MediaPack™ Series

MP-11x & MP-124 Analog VoIP Gateways

Version 6.6

audiocodes

Table of Contents

This page is intentionally left blank.

Notice

Information contained in this document is believed to be accurate and reliable at the time of printing. However, due to ongoing product improvements and revisions, AudioCodes cannot guarantee accuracy of printed material after the Date Published nor can it accept responsibility for errors or omissions. Updates to this document can be downloaded from https://www.audiocodes.com/library/technical-documents.

This document is subject to change without notice.

Date Published: October-22-2023

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 website at https://www.audiocodes.com/services-support/maintenance-and-support.

Stay in the Loop with AudioCodes

Abbreviations and Terminology

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

Throughout this manual, unless otherwise specified, the following naming conventions are used:

- The term *device* refers to the MediaPack series gateways.
- The term MediaPack refers to MP-124, MP-118, MP-114, and MP-112.
- The term *MP-11x* refers to MP-118, MP-114, and MP-112.

Regulatory Information

Click here to download the Regulatory Information from AudioCodes website.

Related Documentation

Manual Name
SIP CPE Release Notes
MP-11x & MP-124 SIP Installation Manual
MP-11x SIP Fast Track Guide
MP-124 AC SIP Fast Track Guide
MP-124 DC SIP Fast Track Guide
CPE Configuration Guide for IP Voice Mail
DConvert User's Guide
SNMP User's Guide

Notes and Warnings

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

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

Note: This device supports the SAS and/or Gateway / IP-to-IP applications; not the SBC application.

Legal Notice:

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

Document Revision Record

LTRT	Description
65423	Warning bulletin regarding indoor installation and cabling; CLI commands in procedure for assigning OAMP IP address; Command Shell commands; VDC for lighting lamp for MWI; irrelevant parameters removed (GWInboundManipulationSet, GWOutboundManipulationSet); command correction for FXS line testing.
65424	TR-069; maximum resolved IP addresses per DNS query; three-way conference example; CPTWizard removed; maximum number of User Info file rules.

LTRT	Description
65425	Automatic update chapter revised.
65425	New parameter added – EnableLLDP.
65427	SAS feature key removed.
65428	FXS Line Testing updated to include MP-118 for Command Shell commands and updated notes.
65429	MP-124 Rev. E added.
65430	Maximum channel capacity updated; Out-of-Band Digit Notifications According to KPML (AdditionalOutOfBandDtmfFormat); Note added for User Information File syntax re spaces; PublicationIPGroupID (new); RTCPXRReportMode (new option [3]); GwSDPConnectionMode (new); description of CutThrough updated.
65431	TLSVersion parameter updated; note added to LifeLineType parameter description.
65432	▪ New sections: Configuring Media Realms; Configuring Quality of Experience per Media Realm; Quality of Experience; Reporting Voice Quality of Experience to OVOC; Configuring the SEM Server; Configuring Clock Synchronization between Device and SEM; Enabling RTCP XR Reporting to SEM; SIP Media Realm Parameters ▪ New parameters: IgnoreAuthorizationStale; CpMediaRealm; QOEServerIP; QOEInterfaceName; QOERules
65433	▪ Updated sections: FXS Voice Menu Guidance; Alternative Routing Based on IP Connectivity (typo) ▪ Updated parameters: PSTNPrefix_SourceAddress; ResetWebPassword; SSHMaxLoginAttempts; SecureCallsFromIP
65434	▪ New sections: Configuring Password Display in ini File ▪ Updated sections: Advanced User Accounts Configuration (ini file parameters); SIP Calling Name Manipulations (max. rows); Selected Technical Specifications (MWI) ▪ Updated parameters: WebUsers_Password (note); FaxBypassPayloadType ▪ New parameters: INIPasswordsDisplayType
65435	▪ Updated sections: Configuring Voice Settings (silence suppression removed); Silence Suppression (removed); Fax / Modem Transparent Mode (silence suppression removed); Configuring Coders (silence suppression removed); Message Waiting Indication (lamp voltage); Viewing Active Alarms (note) ▪ New parameters: ActiveAlarmTableMaxSize; NoAlarmForDisabledPort; EnableLowVoltageMwiGeneration; LedMwiOnDurationTime; LedMwiOffDurationTime; NeonMwiOnDurationTime; NeonMwiOffDurationTime ▪ Updated parameters: IsCiscoSCEMode; EnableSilenceCompression (removed); EnableSilenceDisconnect; UseDisplayNameAsSourceNumber
65436	▪ Updated with patch version 6.60A.340.001 ▪ Updated sections: Manipulating URI user part of Incoming REGISTER ▪ Updated parameters: NoAlarmForDisabledPort (Web); TLSVersion; SASRegistrationManipulation (SASRegistrationManipulation_RuleApplyTo); EnableLowVoltageMwiGeneration; LedMwiOnDurationTime; LedMwiOffDurationTime; NeonMwiOnDurationTime; NeonMwiOffDurationTime

LTRT	Description
65437	▪ Updated with patch version 6.60A.342.003 ▪ Updated sections: Multiple Interface Table Configuration Rules (VLANs); VoIP Interfaces for Combined Application Types (VLANs); Configuring Proxy Sets Table; Configuring Account Table (max.); SAS Configuration (all subsections – path to pages); Manipulating URI user part of Incoming REGISTER and/or INVITE (new SAS Registration Table) ▪ New sections: Provisioning the Device using DHCP Option 160 ▪ New parameters: ProxySet_HomingSuccessDetectionRetries; DhcpOption160Support
65437	▪ Updated with new corporate logo and website URLs ▪ Updated sections: FXS Voice Menu Guidance (note re MP-124 Rev. E); Automatic Update from NFS, FTP and HTTP Servers (AutoUpdatePredefinedTime) ▪ Updated parameters: Test_Call_Play (NOTE REMOVED RE Caller); DisableSNMP (note)
65439	▪ New parameter: BlindTransferOnHook ▪ Updated parameter: MgmtBehaviorOnTimeout (removed)
65440	▪ New parameter: DhcpHasLocalIpAddrDuringInit
65441	Three-way conference call updated with new MP-112 and MP-114 hardware revisions; TrunkGroupSettings_DedicatedConnectionMode added to Hunt Group Settings table.
65442	Silence detection no-support for MP-124 Rev. E.
65443	Note for IPMediaChannels parameter added for new hardware revision in Three-Way Conferencing section
65444	TR-069 removed
65445	Auto-generation of default TLS certificate if expires.
65446	'Dedicated Connection Mode' parameter added to Hunt Group Settings table; note added to EnhancedFXSLineCurrent for MP-124 Rev. E
65447	Note added (By Dest Phone Number not relevant) for Forward on Busy Trunk feature; SASEmergencyModeRouteRegister added; Call Forward table (note re deactivation and 'Time for No Reply Forward' field).
65448	SNMP community string note
65450	IgnoreAuthorizationStale description updated
65451	Typo (BlindTransferOnHook instead of BlindTransferOnOnHook)
65452	AUPDResetURLOnWebConfig added; TLS version and cipher configuration for 802.1x
65453	Prefix2ExtLine
65454	Models (4 FXS/FXO)
65455	802.1x update (new 802.1xMaxFragmentSize);
65456	EnableSIPRemoteReset updated

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 website at https://online.audiocodes.com/documentation-feedback.

1   Overview

The MediaPack series analog Voice-over-IP (VoIP) Session Initiation Protocol (SIP) media gateways (hereafter referred to as *device*) are cost-effective, cutting-edge technology products. These stand-alone analog VoIP devices provide superior voice technology for connecting legacy telephones, fax machines and Private Branch Exchange (PBX) systems to IP-based telephony networks, as well as for integration with new IP-based PBX architectures. These devices are designed and tested to be fully interoperable with leading softswitches and SIP servers.

The device is best suited for small and medium-sized enterprises (SME), branch offices, or residential media gateway solutions. The device enables users to make local or international telephone and / or fax calls over the Internet between distributed company offices, using their existing telephones and fax. These calls are routed over the existing network ensuring that voice traffic uses minimum bandwidth. The device also provides SIP trunking capabilities for Enterprises operating with multiple Internet Telephony Service Providers (ITSP) for VoIP services.

The device supports the SIP protocol, enabling the deployment of VoIP solutions in environments where each enterprise or residential location is provided with a simple media gateway. This provides the enterprise with a telephone connection (i.e., RJ-11 connector) and the capability to transmit voice and telephony signals over a packet network.

The device provides FXO and/or FXS analog ports for direct connection to an enterprise's PBX (FXO), and / or to phones, fax machines, and modems (FXS). Depending on model, the device can support up to 24 simultaneous VoIP calls. The device is also equipped with a 10/100Base-TX Ethernet port for connection to the IP network. The device provides LEDs for indicating operating status of the various interfaces.

The device is a compact unit that can be easily mounted on a desktop, wall, or in a 19-inch rack.

The device provides a variety of management and provisioning tools, including an HTTP-based embedded Web server, Telnet, One Voice Operations Center (OVOC), and Simple Network Management Protocol (SNMP). The user-friendly, Web interface provides remote configuration using any standard Web browser (such as Microsoft™ Internet Explorer™).

The figure below illustrates a typical MediaPack VoIP application.

Figure 1-1: Typical MediaPack VoIP Application
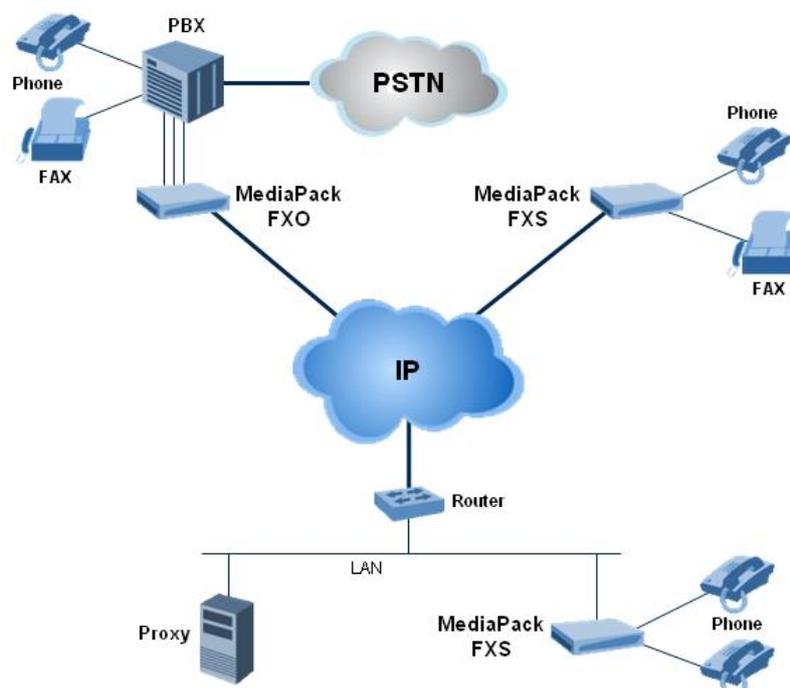
1.1 MediaPack Models

The analog MediaPack 1xx models and their corresponding supported configurations are listed in the table below:

Table 1-1: MediaPack 1xx Models and Configurations

MediaPack Model	FXS	FXO	Combined FXS / FXO	Number of Channels
MP-124	Yes	No	No	24
MP-118	Yes	Yes	4 FXS + 4 FXO	8
MP-114	Yes	Yes	2 FXS + 2 FXO	4
			4 FXS + 4 FXO	8
MP-112*	Yes	No	No	2

* The MP-112 differs from the MP-114 and MP-118 in that its configuration excludes the RS-232 connector and Lifeline option.

1.2 SIP Overview

Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol used on the gateway for creating, modifying, and terminating sessions with one or more participants. These sessions can include Internet telephone calls, media announcements, and conferences.

SIP invitations are used to create sessions and carry session descriptions that enable participants to agree on a set of compatible media types. SIP uses elements called Proxy servers to help route requests to the user's current location, authenticate and authorize users for services, implement provider call-routing policies and provide features to users.

SIP also provides a registration function that enables users to upload their current locations for use by Proxy servers. SIP implemented in the gateway, complies with the Internet Engineering Task Force (IETF) RFC 3261 (refer to http://www.ietf.org).

The SIP call flow, shown in the figure below, describes SIP messages exchanged between two devices during a basic call. In this call flow example, device 10.8.201.108 with phone number 6000, dials device 10.8.201.161 with phone number 2000.

Figure 1-2: SIP Call Flow
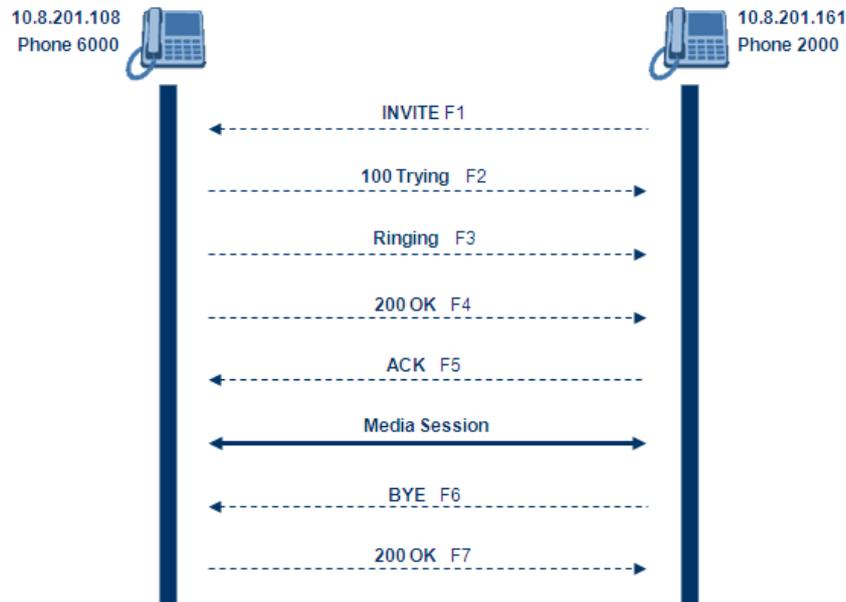
- **F1 INVITE - 10.8.201.108 to 10.8.201.161:**

```
INVITE sip:2000@10.8.201.161;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacsiJkDGd
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
CSeq: 18153 INVITE
Contact: <sip:8000@10.8.201.108;user=phone>
User-Agent: Audiocodes-Sip-Gateway/MediaPack/v.6.60.010.006
Supported: 100rel,em
Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,
NOTIFY,PRACK,REFER,INFO
Content-Type: application/sdp
Content-Length: 208

v=0
o=AudiocodesGW 18132 74003 IN IP4 10.8.201.108
s=Phone-Call
c=IN IP4 10.8.201.108
t=0 0
m=audio 4000 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
```

- **F2 TRYING - 10.8.201.161 to 10.8.201.108:**

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacsiJkDGd
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
Server: Audiocodes-Sip-Gateway/MediaPack/v.6.60.010.006
```

```
CSeq: 18153 INVITE
Content-Length: 0
```

■ **F3 RINGING 180 - 10.8.201.161 to 10.8.201.108:**

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacsiJkDGd
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>;tag=1c7345
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
Server: Audiocodes-Sip-Gateway/MediaPack/v.6.60.010.006
CSeq: 18153 INVITE
Supported: 100rel,em
Content-Length: 0
```

> **Note:** Phone 2000 answers the call and then sends a SIP 200 OK response to device 10.8.201.108.

■ **F4 200 OK - 10.8.201.161 to 10.8.201.108:**

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacsiJkDGd
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>;tag=1c7345
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
CSeq: 18153 INVITE
Contact: <sip:2000@10.8.201.161;user=phone>
Server: Audiocodes-Sip-Gateway/MediaPack/v.6.60.010.006
Supported: 100rel,em
Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,
NOTIFY,PRACK,REFER,INFO
Content-Type: application/sdp
Content-Length: 206

v=0
o=AudiocodesGW 30221 87035 IN IP4 10.8.201.161
s=Phone-Call
c=IN IP4 10.8.201.10
t=0 0
m=audio 7210 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=ptime:20
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
```

■ **F5 ACK - 10.8.201.108 to 10.8.201.10:**

```
ACK sip:2000@10.8.201.161;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacZYpJWxZ
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>;tag=1c7345
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
User-Agent: Audiocodes-Sip-Gateway/MediaPack/v.6.60.010.006
CSeq: 18153 ACK
Supported: 100rel,em
Content-Length: 0
```

> **Note:** Phone 6000 goes on-hook and device 10.8.201.108 sends a BYE to device 10.8.201.161 and a voice path is established.

- **F6 BYE - 10.8.201.108 to 10.8.201.10:**

```
BYE sip:2000@10.8.201.161;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacRKCVBud
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>;tag=1c7345
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
User-Agent: Audiocodes-Sip-Gateway/MediaPack/v.6.60.010.006
CSeq: 18154 BYE
Supported: 100rel,em
Content-Length: 0
```

- **F7 OK 200 - 10.8.201.10 to 10.8.201.108:**

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacRKCVBud
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>;tag=1c7345
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
Server: Audiocodes-Sip-Gateway/MediaPack/v.6.60.010.006
CSeq: 18154 BYE
Supported: 100rel,em
Content-Length: 0
```

This page is intentionally left blank.

Part I

Getting Started with Initial Connectivity

2   Assigning the OAMP IP Address

The device is shipped with a factory default IP address for its operations, administration, maintenance, and provisioning (OAMP) interface, as shown in the table below:

Table 2-1: Default OAMP IP Address

IP Address	Value
IP Address	▪ FXS and FXS / FXO devices: 10.1.10.10 ▪ FXO device: 10.1.10.11 **Note:** FXO interfaces are applicable only to MP-11x series devices.
Subnet Mask	255.255.0.0
Default Gateway IP Address	0.0.0.0

The default IP address can be used for initially accessing the device, using any of its management tools (i.e., embedded Web server, OVOC, or Telnet). Once accessed, you can change this default IP address to correspond with your networking scheme in which the device is deployed. After changing the IP address, you can re-access the device with this new OAMP IP address and start configuring and managing the device as desired.

This section describes the different methods for changing the device's default IP address to suit your networking environment:

- Embedded command line interface (CLI) - see 'CLI' on page 30
- Embedded HTTP/S-based Web server - see 'Web Interface' on page 27
- Bootstrap Protocol (BootP) - see BootP/TFTP Server on page 29
- FXS telephone voice menu - see FXS Voice Menu Guidance on page 32

2.1   Web Interface

The procedure below describes how to assign an OAMP IP address using the Web interface.

➢ **To assign an OAMP IP address using the Web interface:**

1. Disconnect the network cables (if connected) from the device.

2. Connect the Ethernet port located on the rear panel (labeled Ethernet) directly to the network interface of your computer, using a straight-through Ethernet cable.

Figure 2-1: MP-11x Ethernet Connection to PC for Initial Connectivity
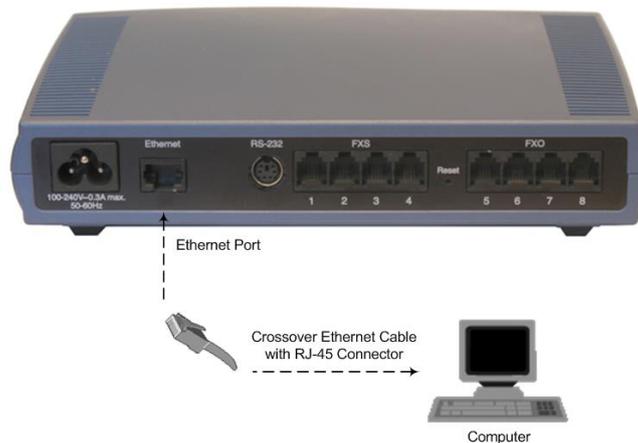
Figure 2-2: MP-124 Ethernet Connection to PC for Initial Connectivity (e.g., MP-124 Rev. E)

3. Change the IP address and subnet mask of your computer to correspond with the default IP address and subnet mask of the device.

4. Access the Web interface:

   a. On your computer, start a Web browser and in the URL address field, enter the default IP address of the device; the Web interface's Login screen appears:

Figure 2-3: Web Login Screen

   b. In the 'Username' and 'Password' fields, enter the default login user name ("Admin" - case-sensitive) and password ("Admin" - case-sensitive), and then click **Login**; the device's Web interface is accessed.

5. Change the default IP address to one that corresponds with your network:

   a. Open the Multiple Interface Table page (**Configuration** tab > **VoIP** menu > **Network** submenu > **IP Settings**).

Figure 2-4: IP Settings Page (Single Network Interface)

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

     c. Change the IP address, subnet mask, and Default Gateway IP address to correspond with your network IP addressing scheme.

     d. Click **Apply**, and then click **Done** to validate your settings.

6. Save your settings to the flash memory with a device reset (see Resetting the Device on page 359).

7. Disconnect the computer from the device and then reconnect the device to your network.

2.2   BootP/TFTP Server

You can assign an IP address to the device using BootP/TFTP protocols. This can be done using the AudioCodes AcBootP utility (supplied) or any standard compatible BootP server.

> ⚠️ **Note:** You can also use the AcBootP utility to load the software file (.cmp) and configuration file (.ini). For a detailed description of the AcBootP utility, refer to *AcBootP Utility User's Guide*.

➢ **To assign an IP address using BootP/TFTP:**

1. Start the AcBootP utility.

2. Select the **Preferences** tab, and then set the 'Timeout' field to "50".

3. Select the **Client Configuration** tab, and then click the **Add New Client** button.

Figure 2-5: BootP Client Configuration Screen
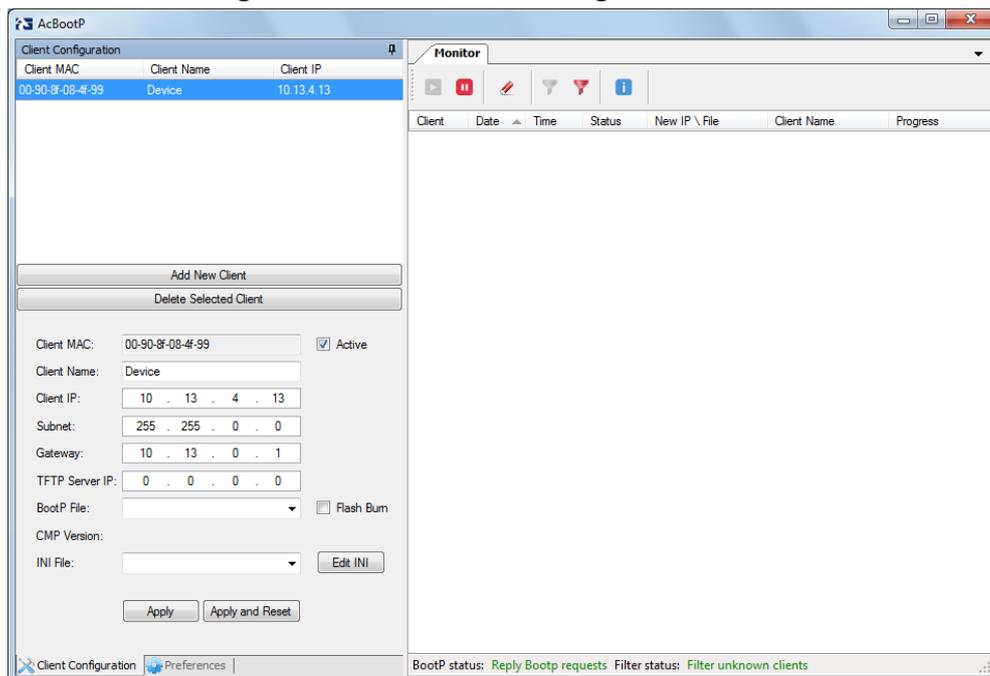
4. Configure the following fields:

- 'Client MAC': Enter the device's MAC address. The MAC address is printed on the label located on the underside of the device. Ensure that the check box to the right of the field is selected in order to enable the client.

- 'Client IP': Enter the new IP address (in dotted-decimal notation) that you want to assign the device.
- 'Subnet': Enter the new subnet mask (in dotted-decimal notation) that you want to assign the device.
- 'Gateway': Enter the IP address of the Default Gateway (if required).

5. Click **Apply** to save the new client.
6. Physically reset the device by powering it down and then up again. This enables the device to receive its new networking parameters through the BootP process.

2.3 CLI

The procedure below describes how to assign an OAMP IP address, using CLI.

Note: Assigning an IP address using CLI is not applicable to MP-112 as this model does not provide RS-232 serial interface.

➤ **To assign an OAMP IP address using CLI:**

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

Figure 2-6: MP-11x Serial Connection with PC for CLI Communication

Figure 2-7: MP-124 Serial Connection with PC for CLI Communication (e.g., MP-124 Rev. E)
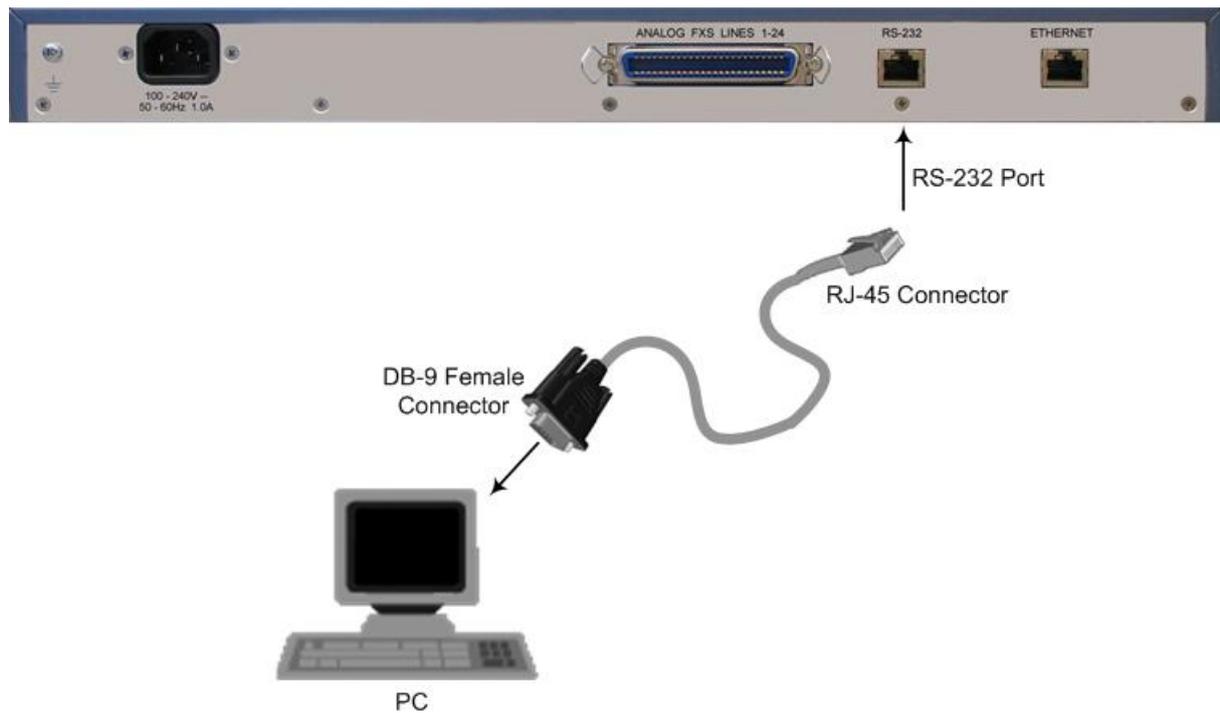
2. Establish serial communication with the device using a terminal emulator program (such as HyperTerminal) with the following communication port settings:

   - Baud Rate: 115,200 bps for MP-124 and 9,600 bps for MP-11x
   - Data Bits: 8
   - Parity: None
   - Stop Bits: 1
   - Flow Control: None

3. At the prompt, type the login username (default is "Admin" - case sensitive):
   login: Admin

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

5. At the prompt, type the following command to access the Configuration folder:
   />CONF

6. View the current network settings, by typing the following command:
   /CONFiguration>GCP IP

7. Change the network settings, by typing the following command:
   /CONFiguration>SCP IP <IP address> <subnet mask> <Default Gateway>
   You must enter all three network parameters, each separated by a space, for example:
   /CONFiguration>SCP IP 10.13.77.7 255.255.0.0 10.13.0.1

8. Save your changes and reset the device, by typing the following command:
   /CONFiguration>SAR

2.4   FXS Voice Menu Guidance

You can assign an IP address that suits your networking scheme using a standard touch-tone telephone connected to one of the FXS ports. The FXS voice menu can also be used to query and modify basic configuration parameters.

Notes:

- The voice menu is not supported by MP-124 Rev. E.
- If you want to disable the FXS voice menu, do one of the following:
  - √   Set the VoiceMenuPassword parameter to 'disable'.
  - √   Change the Web login password for the Admin user from its default value (i.e., "Admin") to any other value, and then reset the device.

➢   **To assign an IP address using the voice menu:**

1.   Connect a telephone to one of the FXS ports.

2.   Lift the handset and dial *****12345** (three stars followed by the digits 1, 2, 3, 4, and 5).

Notes: When dialing *******, a fast-busy tone may be heard.

3.   Wait for the 'configuration menu' voice prompt to be played.

4.   To change the IP address:

    a.   Press **1** followed by the pound key (**#**); the current IP address of the device is played.

    b.   Press the **#** key.

    c.   Dial the new IP address, using the star (*) key instead of periods (.), e.g., 192*168*0*4, and then press **#** to finish.

    d.   Review the new IP address, and then press **1** to save.

5.   To change the subnet mask:

    a.   Press **2** followed by the **#** key; the current subnet mask of the device is played.

    b.   Press the **#** key.

    c.   Dial the new subnet mask (e.g., 255*255*0*0), and then press **#** to finish.

    d.   Review the new subnet mask, and then press **1** to save.

6. To change the Default Gateway IP address:

    a. Press **3** followed by the **#** key; the current Default Gateway address is played.

    b. Press the **#** key.

    c. Dial the new Default Gateway address (e.g., 192*168*0*1), and then press **#** to finish.

    d. Review the new Default Gateway address, and then press **1** to save.

7. Hang up (on-hook) the handset.

Alternatively, initial configuration may be performed using an HTTP server. The Voice Menu may be used to specify the configuration URL.

➢ To set a configuration URL:

1. Obtain the IP address of the configuration HTTP server (e.g., 36.44.0.6).

2. Connect a telephone to one of the FXS ports.

3. Lift the handset and dial *****12345** (three stars followed by the digits 1, 2, 3, 4, and 5).

4. Wait for the "configuration menu" voice prompt to be played.

5. Dial **31** followed by the **#** key; the current IP address is played.

6. To change the IP address:

    a. Press the **#** key.

    b. Dial the configuration server's IP address. Use the star (*) key instead of dots ("."), e.g., 36*44*0*6, and then press **#** to finish.

    c. Review the configuration server's IP address, and then press **1** to save.

7. Dial **32** followed by the **#** key, and then do the following to change the configuration file name pattern:

    a. Press the **#** key.

    b. Select one of the patterns listed in the table below (*aa.bb.cc.dd* denotes the IP address of the configuration server):

#	Configuration File Name Pattern	Description
1	http://aa.bb.cc.dd/config.ini	Standard config.ini.
2	https://aa.bb.cc.dd/config.ini	Secure HTTP.
3	http://aa.bb.cc.dd/audiocodes/<MAC>.ini	The device's MAC address is appended to the file name (e.g., http://36.44.0.6/audiocodes/00908f012300.ini).
4	http://aa.bb.cc.dd:8080/config.ini	HTTP on port 8080.
5	http://aa.bb.cc.dd:1400/config.ini	HTTP on port 1400.
6	http://aa.bb.cc.dd/cgi-bin/acconfig.cgi?mac=<MAC>&ip=<IP>	Generating configuration per IP/MAC address dynamically, using a CGI script. See perl example below.

    a. Press the selected pattern code, and then press **#** to finish.

8. Press **1** to save, and then hang up the handset. The device retrieves the configuration from the HTTP server.

The following is an example perl CGI script, suitable for most Apache-based HTTP servers for generating configuration dynamically per pattern #6 above. Copy this script to /var/www/cgi-bin/acconfig.cgi on your Apache server and edit it as required:

```
#!/usr/bin/perl
use CGI;
```

```
$query = new CGI;
$mac = $query->param('mac');
$ip = $query->param('ip');
print "Content-type: text/plain\n\n";
print "; INI file generator CGI\n";
print "; Request for MAC=$mac IP=$ip\n\n";
print <<"EOF";
SyslogServerIP = 36.44.0.15
EnableSyslog = 1
SSHServerEnable = 1
EOF
```

The table below lists the configuration parameters that can be viewed and modified using the voice menu:

Table 2-2: Voice Menu Configuration Parameters

Item Number at Menu Prompt	Description
1	IP address.
2	Subnet mask.
3	Default Gateway IP address.
4	Primary DNS server IP address.
7	DHCP enable / disable.
31	Configuration server IP address.
32	Configuration file name pattern.
99	Voice menu password (initially 12345). **Note:** The voice menu password can also be changed using the Web interface or *ini* file parameter VoiceMenuPassword.

Part II

Management Tools

3   Introduction

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

The following management tools can be used to configure the device:

■   Embedded HTTP/S-based Web server - see 'Web-based Management' on page 39

■   Command Line Interface (CLI) - see 'CLI-Based Management' on page 77

■   AudioCodes One Voice Operations Center (OVOC) - see OVOC-Based Management on page 93

■   Simple Network Management Protocol (SNMP) browser software - see 'SNMP-Based Management' on page 87

■   Configuration *ini* file - see 'INI File-Based Management' on page 95

Notes:

- Some configuration settings can only be done using a specific management tool. For example, some configuration can only be done using the Configuration *ini* file method.

- Throughout this manual, where a parameter is mentioned, its corresponding Web, CLI, and ini parameter is mentioned. The *ini* file parameters are enclosed in square brackets [...].

- For a list and description of all the configuration parameters, see 'Configuration Parameters Reference' on page 471.

This page is intentionally left blank.

4   Web-Based Management

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

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

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

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

Notes:

- The Web interface allows you to configure most of the device's settings. However, additional configuration parameters may exist that are not available in the Web interface and which can only be configured using other management tools.

- Some Web interface pages and/or parameters are available only for certain hardware configurations or software features. The software features are determined by the installed Software License Key (see 'Software License Key' on page 377).

4.1   Getting Acquainted with the Web Interface

This section provides a description of the Web interface.

4.1.1   Computer Requirements

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

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

---

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

4.1.2 Accessing the Web Interface

The procedure below describes how to access the Web interface.

➢ **To access the Web interface:**

1. Open a standard Web browser (see 'Computer Requirements' on page 39).

2. In the Web browser, specify the IP address of the device (e.g., http://10.1.10.10); the Web interface's Login window appears, as shown below:

Figure 4-1: Web Login Screen

3. In the 'Username' and 'Password' fields, enter the case-sensitive, user name and password respectively.

4. Click **Login**; the Web interface is accessed, displaying the Home page. For a detailed description of the Home page, see 'Viewing the Home Page' on page 63.

Notes:

- The default username and password is "Admin". To change the login user name and password, see 'Configuring the Web User Accounts' on page 66.
- If you want the Web browser to remember your password, select the 'Remember Me' check box and then agree to the browser's prompt (depending on your browser) to save the password for future logins. On your next login attempt, simply press the Tab or Enter keys to auto-fill the 'Username' and 'Password' fields, and then click **Login**.

4.1.3   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 4-2: Main Areas of the Web Interface GUI
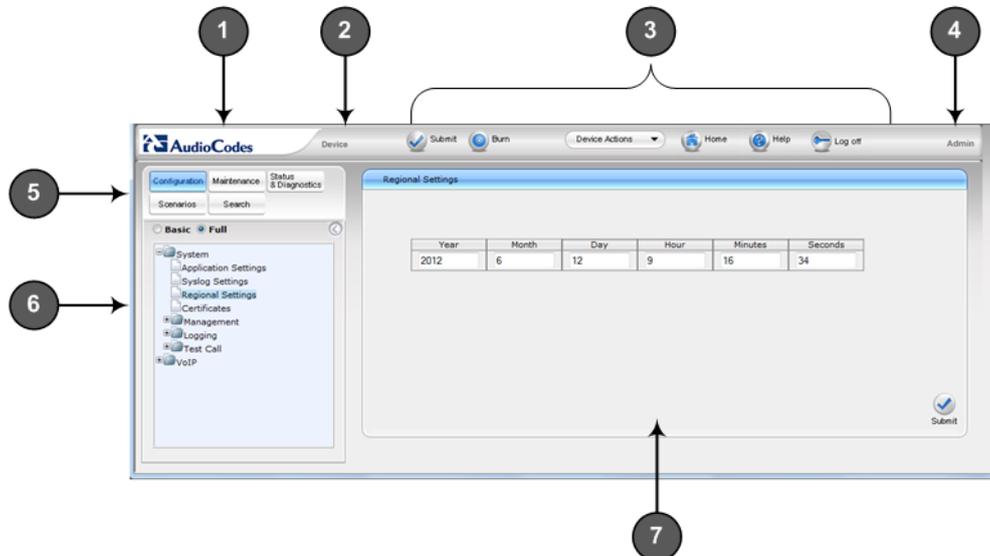
Table 4-1: Description of the Web GUI Areas

Item #	Description
1	Displays AudioCodes (corporate) logo image.
2	Displays the product name.
3	Toolbar, providing frequently required command buttons. For more information, see 'Toolbar Description' on page 42.
4	Displays the username of the Web user that is currently logged in.
5	Navigation bar, providing the following tabs for accessing various functionalities in the Navigation tree: ▪ **Configuration**, **Maintenance**, and **Status & Diagnostics** tabs: Access the configuration menus (see 'Working with Configuration Pages' on page 45) ▪ **Scenarios** tab: Creates configuration scenarios (see Working with Scenarios on page 52) ▪ **Search** tab: Enables a search engine for searching configuration parameters (see 'Searching for Configuration Parameters' on page 51)
6	Navigation tree, displaying a tree-like structure of elements (configuration menus, Scenario steps, or search engine) pertaining to the selected tab on the Navigation bar. For more information, see 'Navigation Tree' on page 42.
7	Work pane, displaying the configuration page of the selected menu in the Navigation tree. This is where configuration is done. For more information, see 'Working with Configuration Pages' on page 45.

4.1.4 Toolbar Description

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

Table 4-2: Description of Toolbar Buttons

Icon	Button Name	Description
	Submit	Applies parameter settings to the device (see 'Saving Configuration' on page 362). **Note:** This icon is grayed out when not applicable to the currently opened page.
	Burn	Saves parameter settings to flash memory (see 'Saving Configuration' on page 362).
Device Actions ▼	**Device Actions**	Opens a drop-down list with frequently needed commands: ▪ **Load Configuration File:** Opens the Configuration File page for loading an *ini* file to the device (see 'Backing Up and Loading Configuration File' on page 384). ▪ **Save Configuration File:** Opens the Configuration File page for saving the *ini* file to a folder on a computer (see 'Backing Up and Loading Configuration File' on page 384). ▪ **Reset:** Opens the Maintenance Actions page for performing various maintenance procedures such as resetting the device (see 'Resetting the Device' on page 359). ▪ **Software Upgrade Wizard:** starts the Software Upgrade wizard for upgrading the device's software (see 'Software Upgrade Wizard' on page 381).
	Home	Opens the Home page (see 'Viewing the Home Page' on page 63).
	Help	Opens the Online Help topic of the currently opened configuration page (see 'Getting Help' on page 61).
	Log off	Logs off a session with the Web interface (see 'Logging Off the Web Interface' on page 62).

> ⚠ **Note:** If you modify a parameter that takes effect only after a device reset, after you click the **Submit** button in the configuration page, the toolbar displays "Reset", as shown in the figure below. This is a reminder that you need to later save your settings to flash memory and reset the device.

Figure 4-3: "Reset" Displayed on Toolbar

4.1.5 Navigation Tree

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

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

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

Figure 4-4: Navigating in Hierarchical Menu Tree (Example)
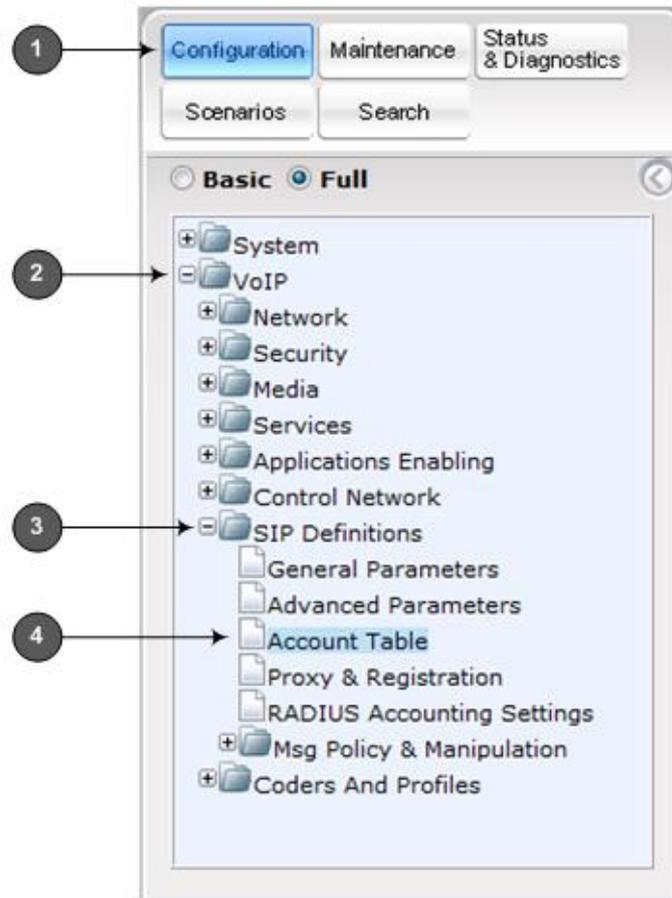
	Note: The figure above is used only as an example. The displayed menus depend on supported features based on the Software License Key installed on your device.

4.1.5.1 Displaying Navigation Tree in Basic and Full View

You can view an expanded or reduced display of the Navigation tree. This affects the number of displayed menus and submenus in the tree. The expanded (*Full*) view displays all the menus pertaining to the selected configuration tab; the reduced (*Basic*) view displays only commonly used menus. This is relevant when using the configuration tabs (i.e., **Configuration**, **Maintenance**, and **Status & Diagnostics**) on the Navigation bar. The advantage of the Basic view is that it prevents "cluttering" of the Navigation tree with menus that may not be required.

➢ **To toggle between Full and Basic view:**

■ To display a reduced menu tree, select the **Basic** option (default).

■ To display all the menus and submenus in the Navigation tree, select the **Full** option.

Figure 4-5: Basic and Full View Options
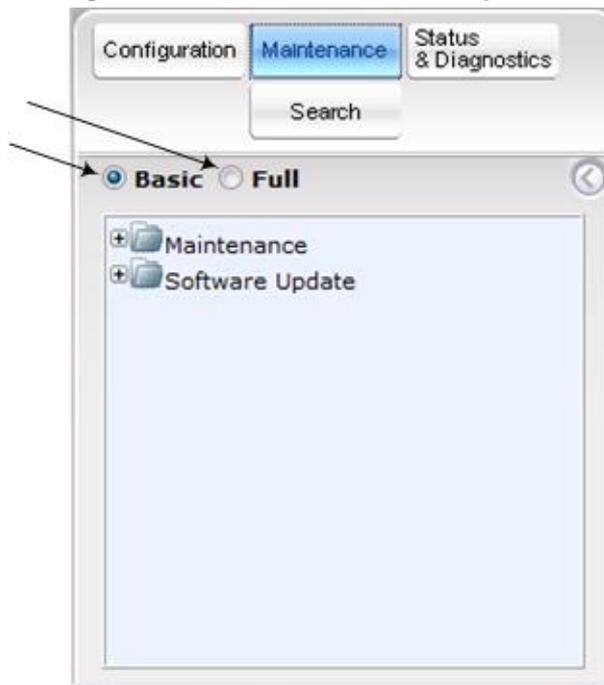
	Notes:
⚠	• After you reset the device, the Web GUI is displayed in Basic view.
	• When in Scenario mode (see Scenarios on page 52), the Navigation tree is displayed in Full view.

4.1.5.2   Showing / Hiding the Navigation Pane

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

➢ **To hide and show the Navigation pane:**

■ **To hide the Navigation pane:** Click the left-pointing arrow 🔘 ; the pane is hidden and the button is replaced by the right-pointing arrow button.

■ **To show the Navigation pane:** Click the right-pointing arrow 🔘 ; the pane is displayed and the button is replaced by the left-pointing arrow button.

Figure 4-6: Show and Hide Button (Navigation Pane in Hide View)
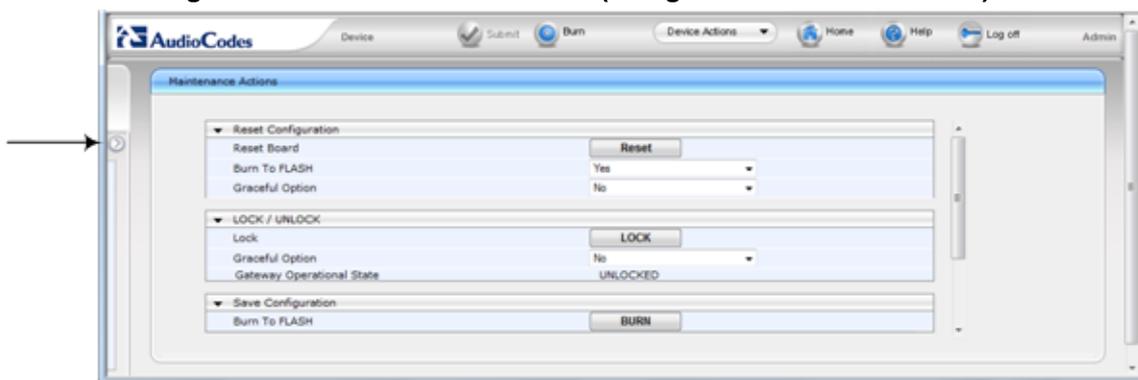
4.1.6   Working with Configuration Pages

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

4.1.6.1   Accessing Pages

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

➢ **To open a configuration page:**

1. On the Navigation bar, click the required tab (**Configuration**, **Maintenance**, or **Status & Diagnostics**); the menus pertaining to the selected tab appear in the Navigation tree.

2. Navigate to the required page item, by performing the following:

 • Drill-down using the **plus** ⊞ sign to expand the menu and submenus.

 • Drill-up using the **minus** ⊟ sign to collapse the menu and submenus.

3. Click the required page item; the page opens in the Work pane.

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

Notes:

- You can also access certain pages from the **Device Actions** button located on the toolbar (see 'Toolbar Description' on page 42).

- To view all the menus in the Navigation tree, ensure that the Navigation tree is in Full view (see 'Displaying Navigation Tree in Basic and Full View' on page 43).

- To get Online Help for the currently displayed page, see 'Getting Help' on page 61.

- Certain pages may not be accessible or may be read-only, depending on the access level of your Web user account (see 'Configuring Web User Accounts' on page 66). If a page is read-only, "Read-Only Mode" is displayed at the bottom of the page.

4.1.6.2 Viewing Parameters

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

■ Displaying "basic" and "advanced" parameters - see 'Displaying Basic and Advanced Parameters' on page 46

■ Displaying parameter groups - see 'Showing / Hiding Parameter Groups' on page 47

4.1.6.2.1 Displaying Basic and Advanced Parameters

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

■ **Advanced Parameter List** button with down-pointing arrow: click this button to display all parameters.

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

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

Figure 4-7: Toggling between Basic and Advanced View
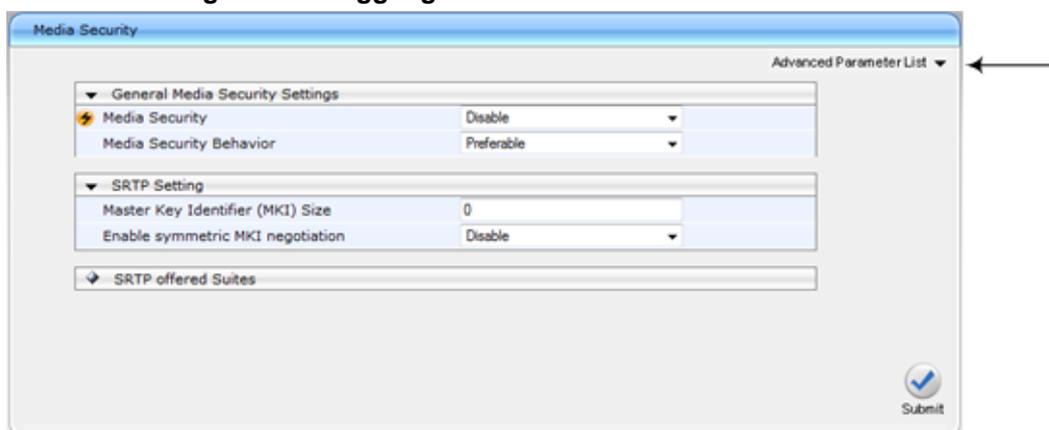
> **Notes:**
>
> - When the Navigation tree is in Full mode (see 'Navigation Tree' on page 42), configuration pages display all their parameters.
> - If a page contains only basic parameters, the **Basic Parameter List** button is not displayed.
> - If you reset the device, the Web pages display only the basic parameters.
> - The basic parameters are displayed in a dark blue background.

4.1.6.2.2 Showing / Hiding Parameter Groups

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

Figure 4-8: Expanding and Collapsing Parameter Groups
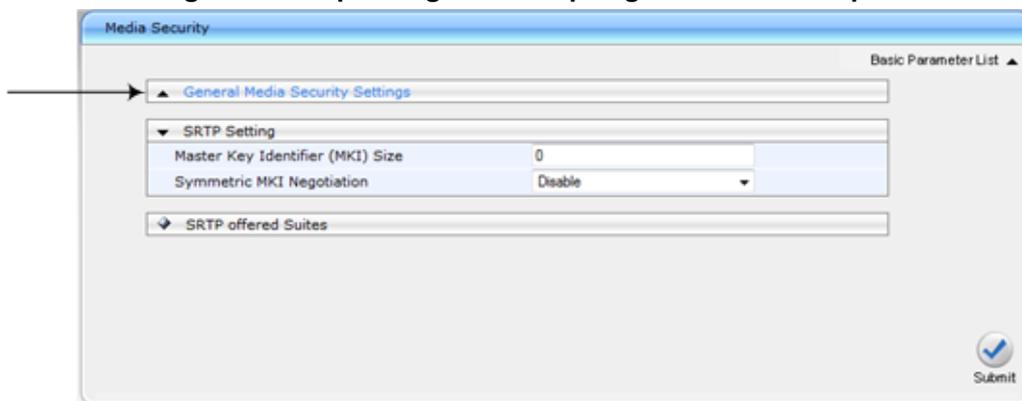
4.1.6.3 Modifying and Saving Parameters

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

Figure 4-9: Edit Symbol after Modifying Parameter Value
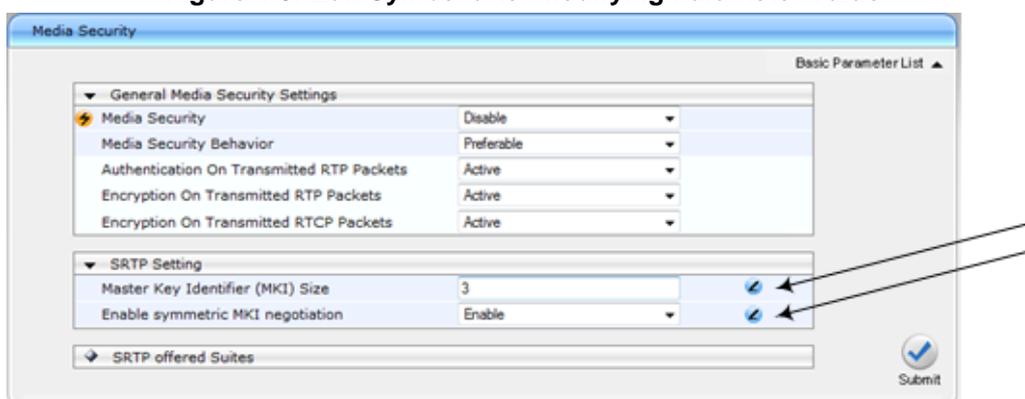
➢ **To save configuration changes on a page to the device's volatile memory (RAM),**

do one of the following:

■ On the toolbar, click the **Submit** [✓] button.

■ At the bottom of the page, click the **Submit** [✓] button.

When you click **Submit**, modifications to parameters with on-the-fly capabilities are immediately applied to the device and take effect. Parameters displayed on the page with the lightning ⚡ symbol take effect only after a device reset. For resetting the device, see 'Resetting the Device' on page 359.

> **Note:** Parameters saved to the volatile memory (by clicking **Submit**), revert to their previous settings after a hardware or software reset, or if the device is powered down. Therefore, to ensure parameter changes (whether on-the-fly or not) are retained, save ('burn') them to the device's non-volatile memory, i.e., flash (see 'Saving Configuration' on page 362).

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

Figure 4-10: Value Reverts to Previous Valid Value
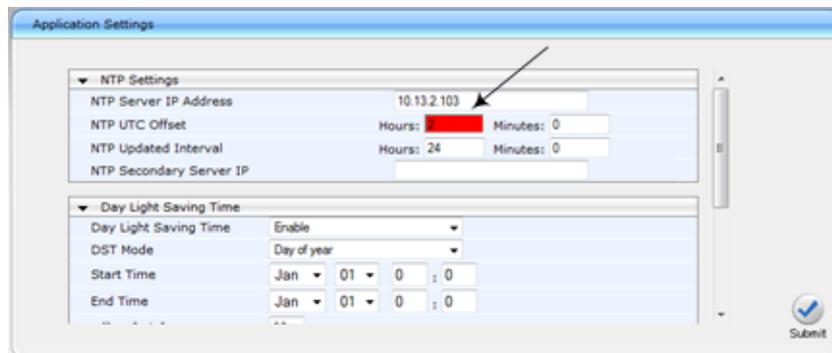
4.1.6.4   Working with Tables

This section describes how to work with configuration tables, which are provided in basic or enhanced design, depending on the configuration page.

4.1.6.4.1 Basic Design Tables

A few of the tables in the Web interface are in basic design format. The figure below displays a typical table in the basic design format and the subsequent table describes its command buttons.

Figure 4-11: Adding an Index Entry to a Table

Table 4-3: Basic Table Design Description

Item #	Button / Field	
1	**Add Index** (or **Add**) button	Adds an index entry row to the table.
2	**Edit**	Edits the selected row.
3	**Delete**	Removes the selected row from the table.
4	'Add Index' field	Defines the index number. When adding a new row, enter the required index number in this field, and then click **Add Index**.
5	**Index** radio button	Selects the row for editing and deleting.
-	**Compact** button	Organizes the index entries in ascending, consecutive order, starting from index 0. For example, assume you have three index entries, 0, 4 and 6. After you click **Compact**, index entry 4 is re-assigned to index 1 and index entry 6 is re-assigned to index 2.
-	**Apply** button	Saves the row configuration. Click this button after you add or edit each index entry.

4.1.6.4.2 Enhanced Design Tables

Most of the tables in the Web interface are designed in the enhanced table format. The figure below displays a typical table in the enhanced design format and the subsequent table describes its command buttons and areas.

Figure 4-12: Displayed Details Pane
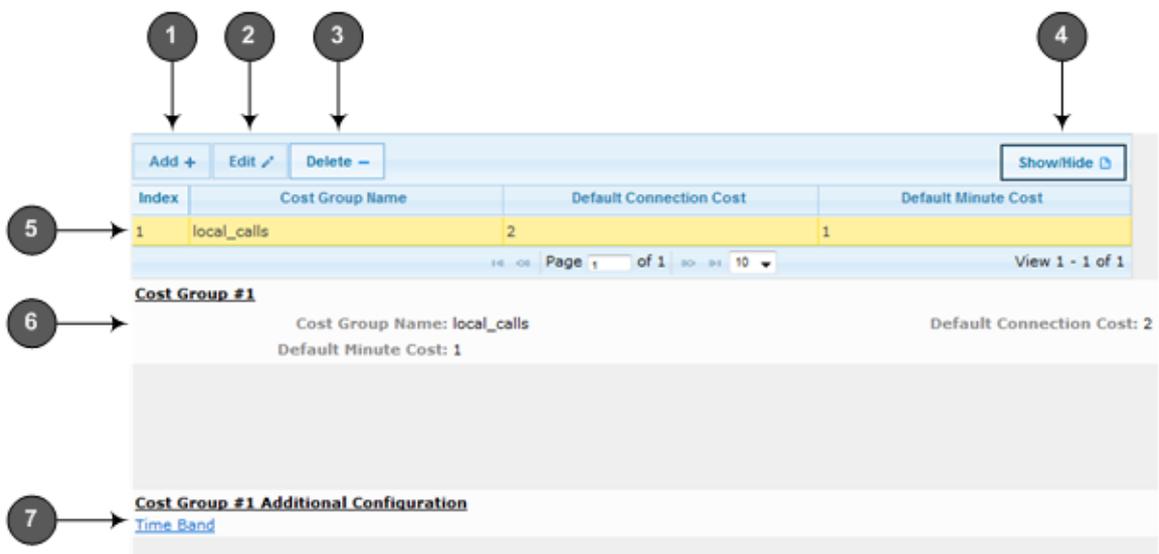
Table 4-4: Enhanced Table Design Description

Item #	Button	
1	**Add**	Adds a new index entry row to the table. When you click this button, a dialog box appears with parameters for configuring the new entry.

Item #	Button	
		When you have completed configuration, click the **Submit** button in the dialog box to add it to the table.
2	**Edit**	Edits the selected row.
3	**Delete**	Removes the selected row from the table. When you click this button, a confirmation box appears requesting you to confirm deletion. Click **Delete** to accept deletion.
4	**Show/Hide**	Toggles between displaying and hiding the full configuration of a selected row. This configuration is displayed below the table (see Item #6) and is useful for large tables that cannot display all its columns in the work pane.
5	-	Selected index row entry for editing, deleting and showing configuration.
6	-	Displays the full configuration of the selected row when you click the **Show/Hide** button.
7	-	Links to access additional configuration tables related to the current configuration.

If the configuration of an entry row is invalid, the index of the row is highlighted in red, as shown below:

Figure 4-13: Invalid Configuration with Index Highlighted in Red
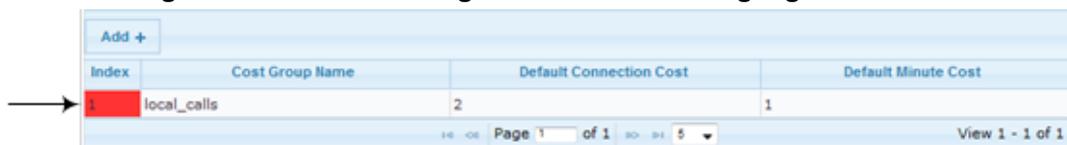
The table also enables you to define the number of rows to display on the page and to navigate between pages displaying multiple rows. This is done using the page navigation area located below the table, as shown in the figure below:

Figure 4-14: Viewing Table Rows per Page
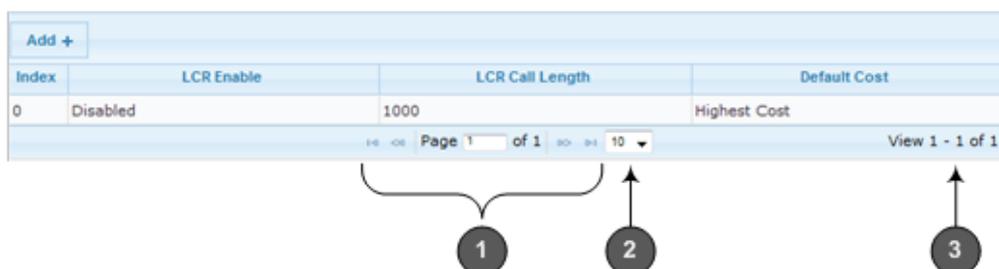
Table 4-5: Row Display and Page Navigation

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

4.1.7 Searching for Configuration Parameters

You can locate the exact Web page on which a specific parameter appears, by using the device's Search feature. The Web parameter's corresponding *ini* file parameter name is used as the search key. The search key can include the full parameter name (e.g., "EnableIPSec") or a substring of it (e.g., "sec"). If you search for a substring, all parameters containing the specified substring in their names are listed in the search result.

> **Note:** If an *ini* file parameter is not configurable in the Web interface, the search fails.

➢ **To search for a parameter:**

1. On the Navigation bar, click the **Search** tab; the Search engine appears in the Navigation pane.
2. In the field alongside the **Search** button, enter the parameter name or a substring of the name for which you want to search. If you have done a previous search for such a parameter, instead of entering the required string, you can use the 'Search History' drop-down list to select the string saved from a previous search.
3. Click **Search**; a list of found parameters based on your search key appears in the Navigation pane. Each searched result displays the following:

   - *ini* file parameter name
   - Link (in green) to the Web page on which the parameter appears
   - Brief description of the parameter
   - Menu navigation path to the Web page on which the parameter appears

4. In the searched list, click the required parameter (green link) to open the page on which the parameter appears; the relevant page opens in the Work pane and the searched

parameter is highlighted in the page for easy identification, as shown in the figure below:

Figure 4-15: Searched Result Screen
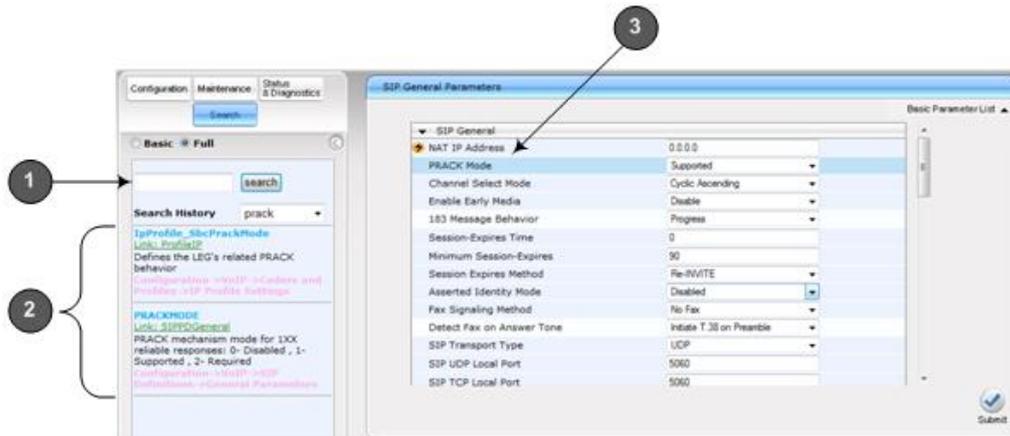
Table 4-6: Search Description

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

4.1.8   Working with Scenarios

The Web interface allows you to create your own menu (*Scenario*) of up to 20 pages, selected from the menus in the Navigation tree (i.e., pertaining to the **Configuration**, **Maintenance**, and **Status & Diagnostics** tabs). Each page in the Scenario is referred to as a *Step*. For each Step, you can select up to 25 parameters on the page to include in the Scenario. Therefore, the Scenario feature is useful in that it allows you quick-and-easy access to commonly used configuration parameters specific to your network environment. When you log in to the Web interface, your Scenario is displayed in the Navigation tree.

Instead of creating a new Scenario, you can load a saved Scenario on a computer to the device (see 'Loading a Scenario to the Device' on page 57).

4.1.8.1 Creating a Scenario

The procedure below describes how to create a Scenario.

➢ **To create a Scenario:**

1. On the Navigation bar, click the **Scenarios** tab; a message box appears, requesting you to confirm creation of a Scenario:

Figure 4-16: Create Scenario Confirmation Message Box
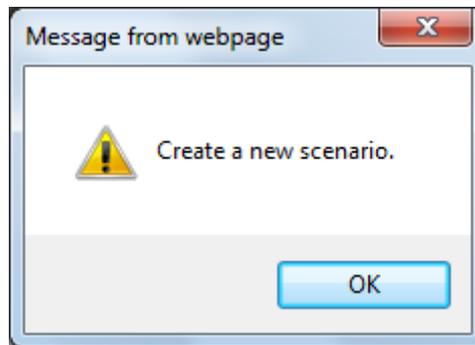
Note: If a Scenario already exists, the Scenario Loading message box appears.

2. Click **OK**; the Scenario mode appears in the Navigation tree as well as the menus of the **Configuration** tab.

3. In the 'Scenario Name' field, enter an arbitrary name for the Scenario.

4. On the Navigation bar, click the **Configuration** or **Maintenance** tab to display their respective menus in the Navigation tree.

5. In the Navigation tree, select the required page item for the Step, and then in the page itself, select the required parameters by selecting the check boxes corresponding to the parameters.

6. In the 'Step Name' field, enter a name for the Step.

7. Click the **Next** button located at the bottom of the page; the Step is added to the Scenario and appears in the Scenario Step list.

8. Repeat steps 5 through 7 to add additional Steps (i.e., pages).

9. When you have added all the required Steps for your Scenario, click the **Save & Finish** button located at the bottom of the Navigation tree; a message box appears informing you that the Scenario has been successfully created.

10. Click **OK**; the Scenario mode is quit and the menu tree of the **Configuration** tab

appears in the Navigation tree.

Figure 4-17: Creating a Scenario
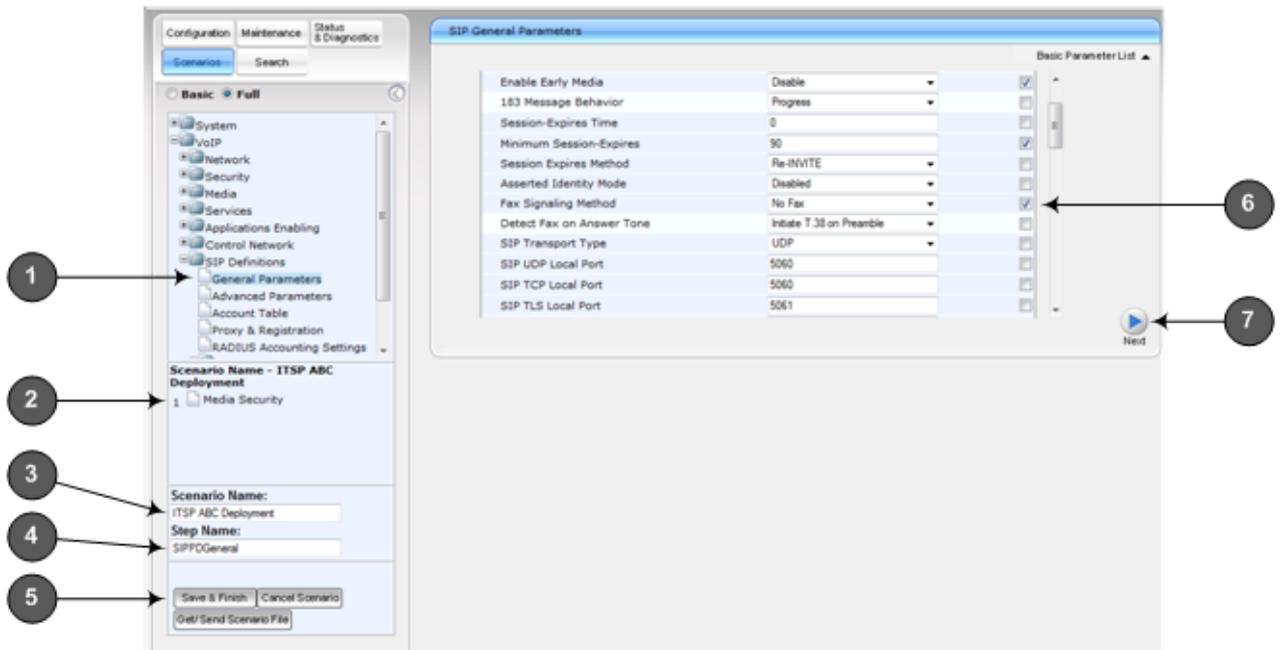
Table 4-7: Scenario Description

	Description
1	Selected page item in the Navigation tree whose page contains the parameter that you want to add to the Scenario Step.
2	Name of a Step that has been added to the Scenario.
3	'Scenario Name' field for defining a name for the Scenario.
4	'Step Name' field for defining a name for a Scenario Step.
5	**Save & Finish** button to save your Scenario.
6	Selected parameter(s) that you want added to a Scenario Step.
7	**Next** button to add the current Step to the Scenario and enables you to add additional Steps.

Notes:

- You can add up to 20 Steps per Scenario, where each Step can contain up to 25 parameters.
- When in Scenario mode, the Navigation tree is in 'Full' display (i.e., all menus are displayed in the Navigation tree) and the configuration pages are in 'Advanced Parameter List' display (i.e., all parameters are shown in the pages). This ensures accessibility to all parameters when creating a Scenario. For a description on the Navigation tree views, see 'Navigation Tree' on page 42.
- If you previously created a Scenario and you click the **Create Scenario** button, the previously created Scenario is deleted and replaced with the one you are creating.
- Only Security Administrator Web users can create Scenarios.

4.1.8.2   Accessing a Scenario

Once you have created the Scenario, you can access it by following the procedure below:

➢   **To access the Scenario:**

1.   On the Navigation bar, select the **Scenario** tab; a message box appears, requesting you to confirm the loading of the Scenario.

2.   Click **OK**; the Scenario and its Steps appear in the Navigation tree, as shown in the example below:

Figure 4-18: Scenario Example
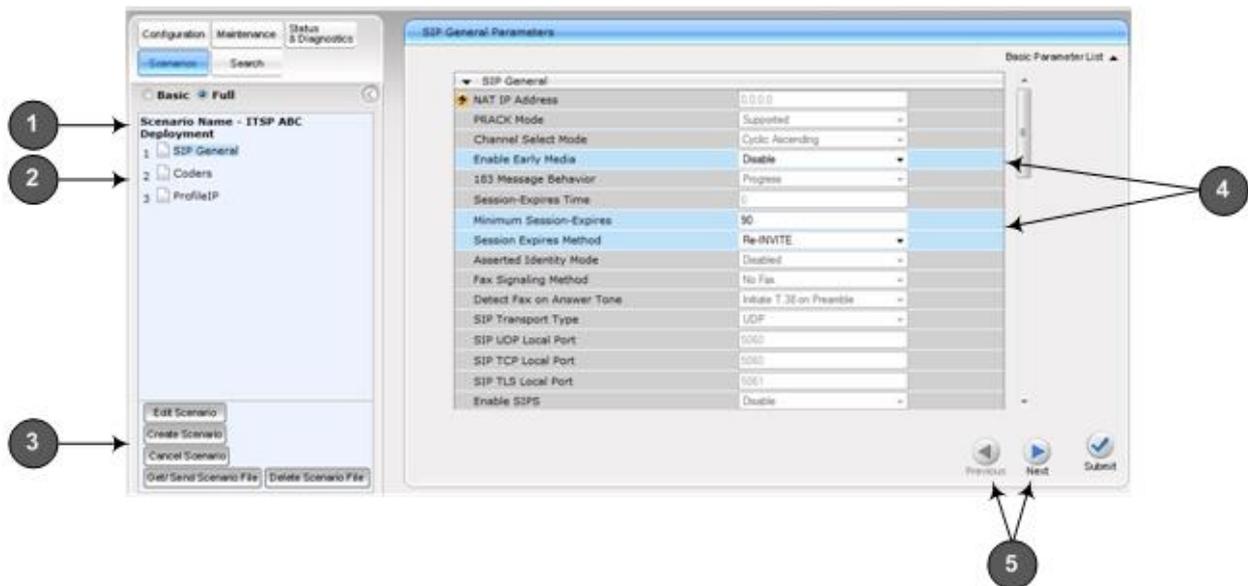
Table 4-8: Loaded Scenario Description

Item	Description
1	Scenario name.
2	Scenario Steps.
3	Scenario configuration command buttons.
4	Parameters available on a page for the selected Scenario Step. These are displayed in a blue background; unavailable parameters are displayed in a gray or light-blue background.
5	Navigation buttons for navigating between Scenario Steps: ▪ **Next** [▶] button to open the next Step listed in the Scenario ▪ **Previous** [◀] button to open the previous Step listed in the Scenario

> ⚠ **Note:**   If you reset the device while in Scenario mode, after the device resets, you are returned once again to the Scenario mode.

4.1.8.3 Editing a Scenario

You can modify a Scenario as described in the procedure below.

> **Note:** Only Security Administrator Web users can edit a Scenario.

➤ **To edit a Scenario:**

1. Open the Scenario.

2. Click the **Edit Scenario** button located at the bottom of the Navigation pane; the 'Scenario Name' and 'Step Name' fields appear.

3. You can perform the following edit operations:

- **Add Steps:**
    - **a.** On the Navigation bar, select the desired tab (i.e., **Configuration** or **Maintenance**); the tab's menu appears in the Navigation tree.
    - **b.** In the Navigation tree, navigate to the desired page item; the corresponding page opens in the Work pane.
    - **c.** On the page, select the required parameters by marking their corresponding check boxes.
    - **d.** Click **Next**.

- **Add or Remove Parameters:**
    - **a.** In the Navigation tree, select the required Step; the corresponding page opens in the Work pane.
    - **b.** To add parameters, select the check boxes corresponding to the desired parameters.
    - **c.** To remove parameters, clear the check boxes corresponding to the desired parameters.
    - **d.** Click **Next**.

- **Edit Step Name:**
    - **a.** In the Navigation tree, select the required Step.
    - **b.** In the 'Step Name' field, modify the Step name.
    - **c.** On the page, click **Next**.

- **Edit Scenario Name:**
    - **a.** In the 'Scenario Name' field, edit the Scenario name.
    - **b.** On the displayed page, click **Next**.

- **Remove a Step:**
    - **a.** In the Navigation tree, select the required Step; the corresponding page opens in the Work pane.
    - **b.** On the page, clear all the check boxes corresponding to the parameters.
    - **c.** Click **Next**.

4. After clicking **Next**, a message box appears notifying you of the change. Click **OK**.

5. Click **Save & Finish**; a message box appears informing you that the Scenario has been successfully modified. The Scenario mode is exited and the menus of the **Configuration** tab appear in the Navigation tree.

4.1.8.4   Saving a Scenario to a PC

You can save a Scenario (as a *dat* file) to a folder on your computer. This is useful when you need multiple Scenarios to represent different deployments. Once you create a Scenario and save it to your computer, you can then keep on saving modifications to it under different Scenario file names. When you require a specific network environment setup, you can load the suitable Scenario file from your computer (see 'Loading a Scenario to the Device' on page 57).

➢   **To save a Scenario to a computer:**

1. On the Navigation bar, click the **Scenarios** tab; the Scenario appears in the Navigation tree.

2. Click the **Get/Send Scenario File** button, located at the bottom of the Navigation tree; the Scenario File page appears, as shown below:

Figure 4-19: Scenario File Page

3. Click the **Get Scenario File** button; the File Download window appears.

4. Click **Save**, and then in the Save As window navigate to the folder to where you want to save the Scenario file. When the file is successfully downloaded to your computer, the Download Complete window appears.

5. Click **Close** to close the window.

4.1.8.5   Loading a Scenario to the Device

The procedure below describes how to load a previously saved Scenario file (*data* file) from your computer to the device. For saving a Scenario, see 'Saving a Scenario to a PC' on page 57.

➢   **To load a Scenario to the device:**

1. On the Navigation bar, click the **Scenarios** tab; the Scenario appears in the Navigation tree.

2. Click the **Get/Send Scenario File** button, located at the bottom of the Navigation tree; the Scenario File page appears.

3. Click the **Browse** button, and then navigate to the Scenario file saved on your computer.

4. Click the **Send File** button.

> **Notes:**
>
> - You can only load a Scenario file to a device that has the same hardware configuration as the device on which it was created.
> - The loaded Scenario replaces any existing Scenario.
> - You can also load a Scenario file using BootP, by loading an ini file that contains the ini file parameter ScenarioFileName (see Web and Telnet Parameters on page 483). The Scenario file must be located in the same folder as the ini file. For information on using AudioCodes AcBootP utility, refer to AcBootP Utility User's Guide.

4.1.8.6 Deleting a Scenario

You can delete the Scenario, as described in the procedure below.

➢ **To delete the Scenario:**

1. On the Navigation bar, click the **Scenarios** tab; a message box appears, requesting you to confirm:

2. Click **OK**; the Scenario mode appears in the Navigation tree.

3. Click the **Delete Scenario File** button; a message box appears requesting confirmation for deletion.

Figure 4-20: Message Box for Confirming Scenario Deletion
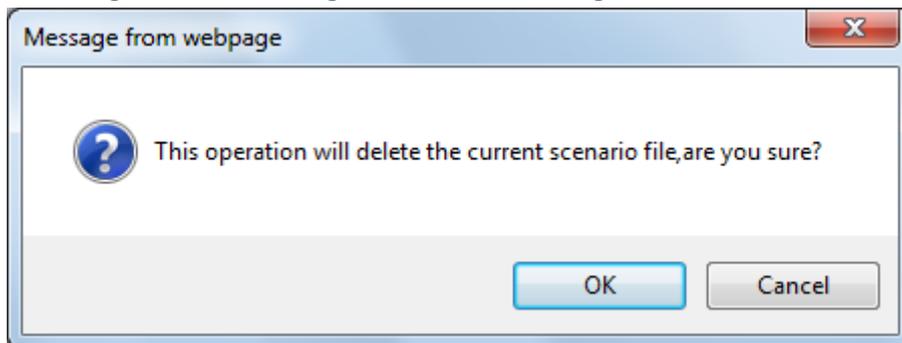
4. Click **OK**; the Scenario is deleted and the Scenario mode closes.

> **Note:** You can also delete a Scenario using the following alternative methods:
> - Loading an empty *dat* file (see 'Loading a Scenario to the Device' on page 57).
> - Loading an *ini* file with the ScenarioFileName parameter set to no value (i.e., ScenarioFileName = "").

4.1.8.7 Quitting Scenario Mode

Follow the procedure below to quit the Scenario mode.

➢ **To quit the Scenario mode:**

1. On the Navigation bar, click any tab except the **Scenarios** tab, or click the **Cancel Scenarios** button located at the bottom of the Navigation tree; a message box appears, requesting you to confirm exiting Scenario mode, as shown below.

Figure 4-21: Confirmation Message Box for Exiting Scenario Mode
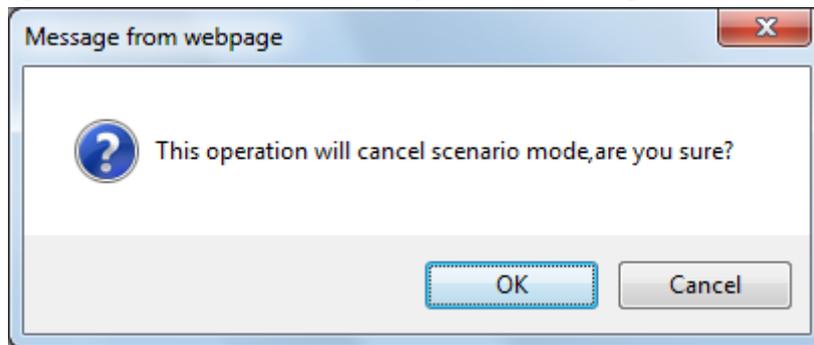
2. Click **OK** to exit.

4.1.9 Creating a Login Welcome Message

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

Figure 4-22: User-Defined Web Welcome Message after Login
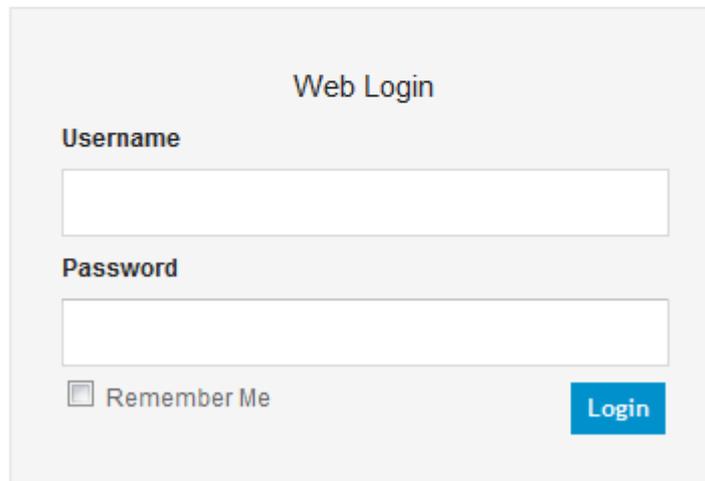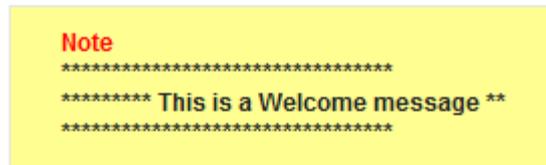
To enable and create a Welcome message, use the WelcomeMessage table ini file parameter. If this parameter is not configured, no Welcome message is displayed.

Table 4-9: ini File Parameter for Welcome Login Message

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

4.1.10 Getting Help

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

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

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

Figure 4-23: Help Topic for Current Page
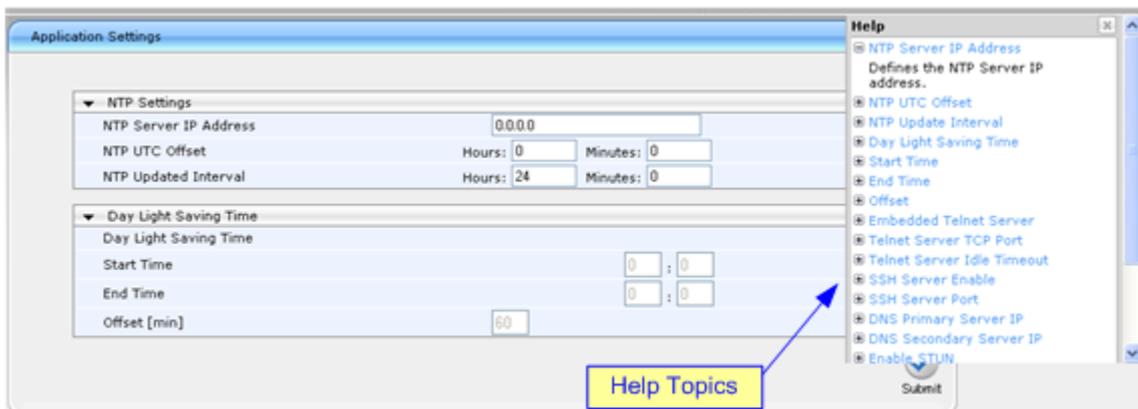
2. To view a description of a parameter, click the **plus** ⊞ sign to expand the parameter. To collapse the description, click the **minus** ⊟ sign.

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

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

4.1.11 Logging Off the Web Interface

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

➢ **To log off the Web interface:**

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

Figure 4-24: Log Off Confirmation Box
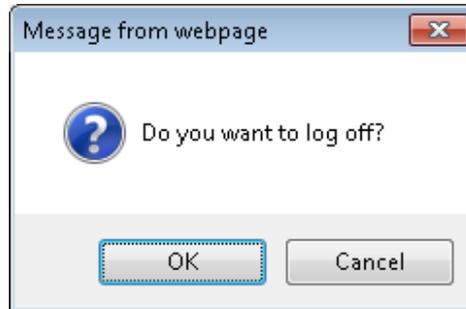
2. Click **OK**; you are logged off the Web session and the Web Login dialog box appears enabling you to re-login, if required.

4.2   Viewing the Home Page

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

➢ **To access the Home page:**

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

Figure 4-25: MP-11x Home Page

Figure 4-26: MP-124 Home Page

> **Note:** The displayed number and type (FXO and/or FXS) of channels depends on the ordered model (e.g., MP-118 or MP-114).

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

■ **IP Address:** IP address of the device

■ **Subnet Mask:** Subnet mask address of the device

■ **Default Gateway Address:** Default gateway used by the device

■ **Firmware Version:** Software version running on the device

■ **Protocol Type:** Signaling protocol currently used by the device (i.e. SIP)

■ **Gateway Operational State:**

• "LOCKED": device is locked (i.e. no new calls are accepted)

• "UNLOCKED": device is not locked

• "SHUTTING DOWN": device is currently shutting down

To perform these operations, see 'Basic Maintenance' on page 359.

The table below describes the areas of the Home page.

Table 4-10: Home Page Description

Label	Description
Alarms	Displays the highest severity of an active alarm raised (if any) by the device: ▪ Green = no alarms ▪ Red = Critical alarm ▪ Orange = Major alarm ▪ Yellow = Minor alarm To view active alarms, click this Alarms area to open the Active Alarms page (see Viewing Active Alarms on page 409).
Channel/Ports	Displays the status of the ports (channels): ▪ 🔴 (red): Line not connected or port out of service due to Serial Peripheral Interface (SPI) failure (applicable only to FXO interfaces) ▪ ⚪ (grey): Channel inactive ▪ 🔵 (blue): Handset is off-hook ▪ 🟢 (green): Active RTP stream If you click a port, a shortcut menu appears with commands allowing you to perform the following: ▪ (Analog ports only) Reset the channel port (see Resetting an Analog Channel on page 363) ▪ View the port settings (see 'Viewing Analog Port Information' on page 411) ▪ Assign a name to the port (see 'Assigning a Port Name' on page 65)
Uplink (MP-11x) **LAN (MP-124**	If clicked, the Ethernet Port Information page opens, displaying Ethernet port configuration settings (see Viewing Ethernet Port Information on page 407).
Fail	Currently not supported.
Ready	Currently not supported.
Power	Always lit green, indicating power received by the device.

4.2.1   Assigning a Port Name

The Home page allows you to assign an arbitrary name or a brief description to each port. This description appears as a tooltip when you move your mouse over the port.

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

> ➢ **To add a port description:**

1. Click the required port icon; a shortcut menu appears, as shown below:

Figure 4-27: Shortcut Menu (Example using MP-11x)
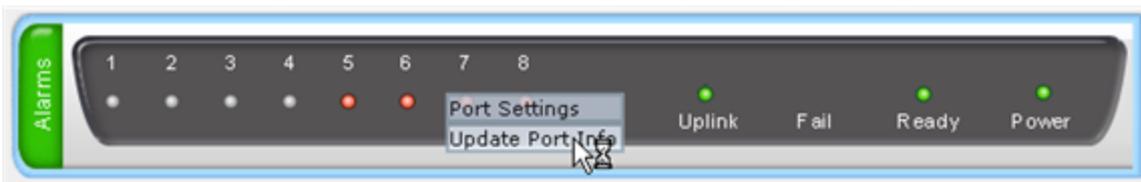
2. From the shortcut menu, choose **Update Port Info**; a text box appears.

Figure 4-28: Typing in Port Name (Example using MP-11x)

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

4.3 Configuring Web User Accounts

You can create up to 5 Web user accounts for the device. Up to five Web users can simultaneously be logged in to the device's Web interface. Web user accounts prevent unauthorized access to the Web interface, enabling login access only to users with correct credentials (i.e., username and password). Each Web user account is composed of the following attributes:

■ **Username and password:** Credentials that enable authorized login access to the Web interface.

■ **Access level (user type):** Access privileges specifying what the user can view in the Web interface and its read/write privileges. The table below describes the different types of Web user account access levels:

Table 4-11: Access Levels of Web User Accounts

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

* The numeric representation of the access level is used only to define accounts in a RADIUS server (the access level ranges from 1 to 255).

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

Table 4-12: Pre-configured Web User Accounts

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

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

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

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

1. From the 'Deny Access On Fail Count' drop-down list, select the number of failed logins after which the user is prevented access to the device for a user-defined time (see next step).

2. In the 'Deny Authentication Timer' field, enter the interval (in seconds) that the user needs to wait before a new login attempt from the same IP address can be done after reaching the number of failed login attempts (defined in the previous step).

Notes:

- For security, it's recommended that you change the default username and password.

- The Security Administrator user can change all attributes of all Web user accounts. Web users with access levels other than Security Administrator can change only their password and username.

- To restore the two Web user accounts to default settings (usernames and passwords), set the *ini* file parameter ResetWebPassword to 1.

- To log in to the Web interface with a different Web user, click the **Log off** button and then login with with a different username and password.

- You can set the entire Web interface to read-only (regardless of Web user access levels), by using the *ini* file parameter DisableWebConfig (see 'Web and Telnet Parameters' on page 483).

- You can define additional Web user accounts using a RADIUS server (see 'Configuring RADIUS Settings' on page 76).

4.3.1   Basic User Accounts Configuration

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

For the Security Administrator, you can change only the username and password; not its access level. For the Monitor user, you can change username and password as well as access level (Administrator, Monitor, or No Access).

Notes:

- The access level of the Security Administrator cannot be modified.

- The access level of the second user account can be modified only by the Security Administrator.

- The username and password can be a string of up to 19 characters. When you log in to the Web interface, the username and password string values are case-sensitive, according to your configuration.

- Up to two users can be logged in to the Web interface at the same time, and they can be of the same user.

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

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

Figure 4-29: WEB User Accounts Page (for Users with 'Security Administrator' Privileges)

2. To change the username of an account:
   a. In the 'User Name' field, enter the new user name.
   b. Click **Change User Name**; if you are currently logged in to the Web interface with this account, the 'Web Login' dialog box appears.
   c. Log in with your new user name.

3. To change the password of an account:
   a. In the 'Current Password' field, enter the current password.
   b. In the 'New Password' and 'Confirm New Password' fields, enter the new password.
   c. Click **Change Password**; if you are currently logged in to the Web interface with this account, the 'Web Login' dialog box appears.
   d. Log in with your new password.

4. To change the access level of the optional, second account:
   a. Under the **Account Data for User: User** group, from the 'Access Level' drop-down list, select a new access level user.
   b. Click **Change Access Level**; the new access level is applied immediately.

4.3.2   Advanced User Accounts Configuration

This section describes advanced Web user account configuration. This is relevant if you need the following management scheme:

■   Enhanced security settings per Web user (e.g., limit session duration)

■   More than two Web user accounts (up to 5 Web user accounts)

■   Master users

This advanced Web user configuration is done in the Web Users table, which is initially accessed from the Web User Accounts page (see procedure below). Once this table is accessed, subsequent access immediately opens the Web Users table instead of the Web User Accounts page.

Notes:

•   Only the Security Administrator user can **initially** access the Web Users table.

•   Only Security Administrator and Master users can add, edit, or delete users.

•   Admin users have read-only privileges in the Web Users table. Monitor users have no access to this page.

•   If you delete a user who is currently in an active Web session, the user is immediately logged off by the device.

•   All users can change their own passwords. This is done in the WEB Security Settings page (see 'Configuring Web Security Settings' on page 73).

•   To remove the Web Users table and revert to the Web User Accounts page with the pre-configured, default Web user accounts, set the ResetWebPassword *ini* file parameter to 1. This also deletes all other Web users.

•   Once the Web Users table is accessed, Monitor users and Admin users can only change their passwords in the Web Security Settings page (see 'Configuring Web Security Settings' on page 73). The new password must have at least four different characters than the previous password. (The Security Administrator users and Master users can change their passwords in the Web Users table and in the Web Security Settings page.)

•   This table can only be configured using the Web interface.

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

1.   Open the Web Users Table page:

   •   Upon initial access:

      a.   Open the Web User Accounts page (**Configuration** tab > **System** menu > **Web User Accounts**).

      b.   Under the **Web Users Table** group, click the **Create Table** button.

   •   Subsequent access: **Configuration** tab > **System** menu > **Web User Accounts**.

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

Figure 4-30: Web Users Table Page

Index	Username	Password	Status	Password Age	Session Limit	Session Timeout	Block Duration	User Level
0	Admin	*	Valid	0	2	60	60	SecAdmin
1	User	*	Valid	0	2	60	60	Monitor
				◁ ◁◁ Page 1 of 1 ▷▷ ▷▷ 10 ▾				View 1 - 2 of 2

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

Figure 4-31: Web Users Table - Add Record Dialog Box

Add Record	✖
Index	0
Username	
Password	
Status	New ▾
Password Age	90
Session Limit	2
Session Timeout	60
Block Duration	60
User Level	Monitor ▾
	🖫 **Submit** ✖ **Cancel**

3. Add a user as required. For a description of the parameters, see the table below.
4. Click **Submit**.

Table 4-13: Web User Parameters Description

Parameter	Description
Web: Username [WebUsers_Username]	Defines the Web user's username. The valid value is a string of up to 40 alphanumeric characters, including the period ".", underscore "_", and hyphen "-" signs.
Web: Password [WebUsers_Password]	Defines the Web user's password. The valid value is a string of 8 to 40 ASCII characters, which must include the following: ▪ At least eight characters ▪ At least two letters that are upper case (e.g., "AA") ▪ At least two letters that are lower case (e.g., "aa") ▪ At least two numbers ▪ At least two signs (e.g., the dollar "$" sign) ▪ No spaces in the string ▪ At least four characters different to the previous password **Note:** For security, password characters are not shown in the Web interface and ini file. In the Web interface, they are displayed as dots when you enter the password and then once applied, the password is displayed as an asterisk (*) in the table. In the ini file, they are displayed as an encrypted string.

Parameter	Description
Web: Status [WebUsers_Status]	Defines the status of the Web user. ▪ New = (Default) User is required to change its password on the next login. When the user logs in to the Web interface, the user is immediately prompted to change the current password. ▪ Valid = User can log in to the Web interface as normal. ▪ Failed Access = This state is automatically set for users that exceed a user-defined number of failed login attempts, set by the 'Deny Access on Fail Count' parameter (see 'Configuring Web Security Settings' on page 73). These users can log in only after a user-defined timeout configured by the 'Block Duration' parameter (see below) or if their status is changed (to New or Valid) by a System Administrator or Master. ▪ Old Account = This state is automatically set for users that have not accessed the Web interface for a user-defined number of days, set by the 'User Inactivity Timer' (see 'Configuring Web Security Settings' on page 73). These users can only log in to the Web interface if their status is changed (to New or Valid) by a System Administrator or Master. **Notes:** ▪ The Old Account status is applicable only to Admin and Monitor users; System Administrator and Master users can be inactive indefinitely. ▪ For security, it is recommended to set the status of a newly added user to New in order to enforce password change.
Web: Password Age [WebUsers_PwAgeInterval]	Defines the duration (in days) of the validity of the password. When this duration elapses, the user is prompted to change the password; otherwise, access to the Web interface is blocked. The valid value is 0 to 10000, where 0 means that the password is always valid. The default is 90.
Web: Session Limit [WebUsers_SessionLimit]	Defines the maximum number of Web interface sessions allowed for the user. In other words, this allows the same user account to log in to the device from different sources (i.e., IP addresses). The valid value is 0 to 5. The default is 2. **Note:** Up to 5 users can be concurrently logged in to the Web interface.
Web: Session Timeout [WebUsers_SessionTimeout]	Defines the duration (in minutes) of Web inactivity of a logged-in user, after which the user is automatically logged off the Web interface. The valid value is 0 to 100000. The default is according to the settings of the 'Session Timeout' global parameter (see 'Configuring Web Security Settings' on page 73).

Parameter	Description
Web: Block Duration [WebUsers_BlockTime]	Defines the duration (in seconds) for which the user is blocked when the user exceeds a user-defined number of failed login attempts. This is configured by the 'Deny Access On Fail Count' parameter (see 'Configuring Web Security Settings' on page 73). The valid value is 0 to 100000, where 0 means that the user can do as many login failures without getting blocked. The default is according to the settings of the 'Deny Authentication Timer' parameter (see 'Configuring Web Security Settings' on page 73). **Note:** The 'Deny Authentication Timer' parameter relates to failed Web logins from specific IP addresses.
Web: User Level [WebUsers_UserLevel]	Defines the user's access level. - Monitor = (Default) Read-only user. This user can only view Web pages and access to security-related pages is denied. - Admin = Read/write privileges for all pages, except security-related pages including the Web Users table where this user has only read-only privileges. - SecAdmin = Read/write privileges for all pages. This user is the Security Administrator. - Master-User = Read/write privileges for all pages. This user also functions as a security administrator. **Notes:** - At least one Security Administrator must exist. The last remaining Security Administrator cannot be deleted. - The first Master user can be added only by a Security Administrator user. - Additional Master users can be added, edited and deleted only by Master users. - If only one Master user exists, it can be deleted only by itself. - Master users can add, edit, and delete Security Administrators (but cannot delete the last Security Administrator). - Only Security Administrator and Master users can add, edit, and delete Admin and Monitor users.

4.4   Displaying Login Information upon Login

The device can display login information immediately upon Web login.

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

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

2. From the 'Display Login Information' drop-down list, select **Yes**.

3. Click **Submit** to apply your changes.

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

Figure 4-32: Login Information Window

4.5   Configuring Web Security Settings

The WEB Security Settings page is used to define a secure Web access communication method. For a description of these parameters, see 'Web and Telnet Parameters' on page 483.

➤ **To define Web access security:**

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

General	
Voice Menu Password	12345
HTTP Authentication Mode	Web Based Authentication ▾
⚡ Secured Web Connection (HTTPS)	HTTP and HTTPS ▾
Requires Client Certificates for HTTPS connection	Disable ▾
⚡ HTTPS Cipher String	RC4:EXP

Session	
Session Timeout (minutes)	60

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

2. Configure the parameters as required.

3. Click **Submit** to apply your changes.

4. To save the changes to flash memory, see 'Saving Configuration' on page 362.

4.6 Web Login Authentication using Smart Cards

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

This feature is enabled using the EnableMgmtTwoFactorAuthentication parameter.

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

➢ **To log in to the Web interface using CAC:**

1. Insert the Common Access Card into the card reader.

2. Access the device using the following URL: https://<host name or IP address>; the device prompts for a username and password.

3. Enter the password only. As some browsers require that the username be provided, it's recommended to enter the username with an arbitrary value.

4.7   Configuring Web and Telnet Access List

The Web & Telnet Access List page is used to define IP addresses (up to ten) that are permitted to access the device's Web, Telnet, and SSH interfaces. Access from an undefined IP address is denied. If no IP addresses are defined, this security feature is inactive and the device can be accessed from any IP address. The Web and Telnet Access List can also be defined using the *ini* file parameter WebAccessList_x (see 'Web and Telnet Parameters' on page 483).

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

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

Figure 4-33: Web & Telnet Access List Page - Add New Entry

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

Figure 4-34: Web & Telnet Access List Table

3. To delete authorized IP addresses, select the Delete Row check boxes corresponding to the IP addresses that you want to delete, and then click **Delete Selected Addresses**; the IP addresses are removed from the table and these IP addresses can no longer access the Web and Telnet interfaces.

4. To save the changes to flash memory, see 'Saving Configuration' on page 362.

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

4.8 Configuring RADIUS Settings

The RADIUS Settings page is used for configuring the Remote Authentication Dial In User Service (RADIUS) accounting parameters. For a description of these parameters, see 'Configuration Parameters Reference' on page 471.

➢ **To configure RADIUS:**

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

Figure 4-35: RADIUS Parameters Page

2. Configure the parameters as required.
3. Click **Submit** to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 362.

5   CLI-Based Management

This section provides an overview of the CLI-based management and configuration relating to CLI management. The device's CLI-based management interface can be accessed using the RS-232 serial port or by using Secure SHell (SSH) or Telnet through the Ethernet interface.

> **Warning:** If you are using the PuTTY terminal emulator for CLI, you must enable the use of the backspace key in the CLI; otherwise, an error will be generated and your settings will not be applied. To enable backspace functionality, start PuTTY and then in the PuTTY Configuration window, expand the Terminal folder, click Keyboard, and then select the Control-H option under the 'The Backspace key' group.

> **Notes:**
>
> - For security, CLI is disabled by default.
> - For information on accessing the CLI interface through the RS-232 port interface, see 'CLI' on page 30.
> - CLI is used only for debugging and mainly allows you to view various information regarding device configuration and performance.

5.1   Enabling CLI using Telnet

The device's CLI can be accessed using Telnet. Secure Telnet using Secure Socket Layer (SSL) can be configured whereby information is not transmitted in the clear. If SSL is used, a special Telnet client is required on your PC to connect to the Telnet interface over a secured connection; examples include C-Kermit for UNIX and Kermit-95 for Windows.

For security, some organizations require the display of a proprietary notice upon starting a Telnet session. You can use the configuration ini file parameter, WelcomeMessage to configure such a message (see Creating a Login Welcome Message on page 60).

➢ **To enable Telnet:**

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

Figure 5-1: Telnet Settings on Telnet/SSH Settings Page

2. Set the 'Embedded Telnet Server' parameter to **Enable Unsecured** or **Enable Secured** (i.e, SSL).
3. Configure the other Tenet parameters as required. For a description of these parameters, see Telnet Parameters on page 486.
4. Click **Submit**.
5. Save the changes to flash memory with a device reset.

5.2 Enabling CLI using SSH and RSA Public Key

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

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

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

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

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

Figure 5-2: Selecting Public RSA Key in PuTTY

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

Figure 5-3: SSH Settings - Pasting Public RSA Key in 'Admin Key' Field

c. For additional security, you can set the 'Require Public Key' to **Enable**. This ensures that SSH access is only possible by using the RSA key and not by using user name and password.

d. Configure the other SSH parameters as required. For a description of these parameters, see SSH Parameters on page 507.

e. Click **Submit**.

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

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

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

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

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

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

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

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

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

4. Click **Submit**.

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

```
ssh –i admin.key xx.xx.xx.xx
```

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

5.3   Establishing a CLI Session

The procedure below describes how to establish a CLI session with the device.

> **Notes:**
> - The default login username and password are both "Admin" (case-sensitive).
> - Only the primary User Account, which has Security Administration access level (200) can access the device using Telnet. For configuring the username and password, see Configuring Web User Accounts on page 66.

➤ **To establish a CLI session with the device:**

1. Establish a Telnet or SSH session with the device using its OAMP IP address.
2. Log in to the session using the username and password assigned to the Admin user of the Web interface.
3. At the login prompt, type the username, and then press Enter:
   ```
   login: Admin
   ```
4. At the password prompt, type the password, and then press Enter:
   ```
   password: Admin
   ```

   After logging in, the current directory (root), available commands, available subdirectories, and a welcome message are displayed at the CLI prompt:
   ```
   login: Admin
   password:
   ready. Type "exit" to close the connection.
   SIP/ SECurity/ DebugRecording/ MGmt/ ControlProtocol/ CONFiguration/
   IPNetworking/ TPApp/ BSP/ PING SHow
   />
   ```

5.4   Command Shell

The Command Shell interface is used mainly to display current configuration and performance.

5.4.1   Getting Familiar with the Command Shell

This section provides a description on how to work with the CLI. This includes basic navigation commands and accessing subdirectories.

5.4.1.1   Basic Command Shell Commands

The table below describes the commands for navigating the Command Shell interface:

Table 5-1: Basic CLI Commands

Purpose	Commands	Description
Help	**h**	Displays the help for a specific command, action, or parameter.
Navigation	**cd**	Enters another directory.
	cd root	Returns you to the root directory "/>".
	..	Goes up one level.
	exit	Terminates the CLI session.

5.4.1.2   Accessing Subdirectories

The Command Shell commands are organized in subdirectories. When you establish a CLI session, you are located in the root directory, indicated by the following prompt:

```
/>
```

To access a subdirectory, type its name, and then press Enter. For example, to enter the **MGmt** subdirectory, type the following:

```
/>MGmt
```

Instead of typing the full name of the subdirectory, you can simply type its abbreviated format. The abbreviated format is shown in upper case (i.e., capital letters). For example, for the **MGmt** subdirectory, you can simply type the following:

```
/>MG
```

If you know the full path to a command inside one of the subdirectories, the short format can be used to run it directly. For example, to run the **PERFormance** command in the **MGmt** subdirectory, type the following:

```
/>MG/PERF
```

> **Note:** The subdirectory names and commands are case-insensitive. For example, it does not matter whether you type "MGmt" or "mgmt".

5.4.2   Status Commands

The following table summarizes the Show commands and their corresponding options.

Show CLI Commands

Command	Short Format	Arguments	Description
SHow	sh	info | dsp | ip | log	Displays operational data. ▪ info: Displays general device information ▪ dsp: Displays DSP resource information ▪ ip: Displays information about IP interfaces
SHow INFO	sh info	-	Displays device hardware information, versions, uptime, temperature reading, and the last reset reason.
SHow DSP	sh dsp	status | perf	Displays status and version for each DSP device, along with overall performance statistics.
SHow IP	sh ip	conf | perf | route	Displays IP interface status and configuration, along with performance statistics. **Note:** The display format may change according to the configuration.
SHow LOG	sh log	[stop]	Displays (or stops displaying) Syslog messages in the CLI session.

Example:

```
/>sh info
Board type: gateway SDH, firmware version 6.60.000.020
Uptime: 0 days, 0 hours, 3 minutes, 54 seconds
Memory usage: 63%
Temperature reading: 39 C
Last reset reason:
Board was restarted due to issuing of a reset from Web interface
Reset Time : 7.1.2012 21.51.13
```

```
/>sh dsp status
DSP firmware: 491096AE8 Version:0660.03 Used=0 Free=480 Total=480
DSP device  0: Active    Used=16   Free= 0   Total=16
DSP device  1: Active    Used=16   Free= 0   Total=16
DSP device  2: Active    Used=16   Free= 0   Total=16
DSP device  3: Active    Used=16   Free= 0   Total=16
DSP device  4: Active    Used=16   Free= 0   Total=16
DSP device  5: Active    Used=16   Free= 0   Total=16
DSP device  6: Inactive
DSP device  7: Inactive
DSP device  8: Inactive
DSP device  9: Inactive
DSP device 10: Inactive
DSP device 11: Inactive
DSP device 12: Active    Used=16   Free= 0   Total=16
DSP device 13: Active    Used=16   Free= 0   Total=16
DSP device 14: Active    Used=16   Free= 0   Total=16
DSP device 15: Active    Used=16   Free= 0   Total=16
DSP device 16: Active    Used=16   Free= 0   Total=16
DSP device 17: Active    Used=16   Free= 0   Total=16
DSP device 18: Inactive
```

```
PSEC - DSP firmware: AC491IPSEC Version: 0660.03
CONFERENCE - DSP firmware: AC491256C Version: 0660.03
/>sh dsp perf
DSP Statistics (statistics for 968 seconds):
Active DSP resources: 480
Total DSP resources: 480
DSP usage %: 100
/>sh ip perf
Networking Statistics (statistics for 979 seconds):
IP KBytes TX: 25
IP KBytes RX: 330
IP KBytes TX per second: 0
IP KBytes RX per second: 1
IP Packets TX: 1171
IP Packets RX: 5273
IP Packets TX per second: 3
IP Packets RX per second: 12
Peak KByte/s TX in this interval: 18
Peak KByte/s RX in this interval: 4
Discarded packets: 186
DHCP requests sent: 0
IPSec Security Associations: 0
/>/mg/perf reset
Done.
/>sh ip perf
Networking Statistics (statistics for 2 seconds):
IP KBytes TX: 2
IP KBytes RX: 4
IP KBytes TX per second: 0
IP KBytes RX per second: 1
IP Packets TX: 24
IP Packets RX: 71
IP Packets TX per second: 3
IP Packets RX per second: 12
Peak KByte/s TX in this interval: 18
Peak KByte/s RX in this interval: 4
Discarded packets: 0
DHCP requests sent: 0
IPSec Security Associations: 0
/>sh ip conf
Interface  IP Address         Subnet Mask     Default Gateway
---------  ------------------ --------------  ----------
 OAM        10.4.64.13     55.255.0.0          10.4.0.1
 Media      10.4.64.13     255.255.0.0         10.4.0.1
 Control    10.4.64.13     255.255.0.0         10.4.0.1
MAC address: 00-90-8f-04-5c-e9
/>sh ip route
Destination      Mask             Gateway          Intf Flags
-------------  ----------------  ------------------------
0.0.0.0          0.0.0.0          10.4.0.1         OAM  A  S
10.4.0.0         255.255.0.0      10.4.64.13       OAM  A  L
127.0.0.0        255.0.0.0        127.0.0.1        AR S
127.0.0.1        255.255.255.255  127.0.0.1        A L    H
Flag legend: A=Active R=Reject L=Local S=Static E=rEdirect
M=Multicast
           B=Broadcast H=Host I=Invalid
End of routing table, 4 entries displayed.
```

5.5   Ping Command

The Ping command is described in the following table:

Ping Command

Command	Short Format	Arguments	Description
PING	**ping**	[-n count] [-l size] [-w timeout] [-p cos] ip-address	Sends ICMP echo request packets to a specified IP address. ▪ count: number of packets to send. ▪ size: payload size in each packet. ▪ timeout: time (in seconds) to wait for a reply to each packet. ▪ cos: Class-of-Service (as per 802.1p) to use.

Example:
```
/>ping 10.31.2.10
Ping process started for address 10.31.2.10. Process ID - 27.
Reply from 10.31.2.10: bytes=0 time<0ms
Reply from 10.31.2.10: bytes=0 time<0ms
Reply from 10.31.2.10: bytes=0 time<0ms
Reply from 10.31.2.10: bytes=0 time<0ms
Ping statistics for 10.31.2.10:
Packets:Sent = 4, Received = 4, Lost 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

5.6   Management Commands

The commands under the **MGmt** directory, described in the table below, display current performance values.

CLI Management Command

Command	Short Format	Arguments	Description
/MGmt/PERFormance	/mg/perf	basic | control | dsp | net | reset	Displays performance statistics. The *reset* argument clears all statistics to zero.

5.7   Configuration Commands

The commands under the **CONFiguration** directory query and modify the current device configuration. The following commands are available:

Configuration CLI Commands

Command	Short Format	Arguments	Description
SetConfigParam IP	/conf/scp ip	ip-addr subnet def-gw	Sets the IP address, subnet mask, and default gateway address of the device (on-the-fly). **Note:** This command may cause disruption of service. The CLI session may disconnect since the device changes its IP address.
RestoreFactorySettings	/conf/rfs		Restores all parameters to factory settings.
SaveAndRestart	/conf/sar		Saves all current configurations to the non-volatile memory and resets the device.
ConfigFile	/conf/cf	view | get | set	Retrieves the full *ini* file from the device and allows loading a new *ini* file directly in the CLI session. **Note:** The argument *view* displays the file, page by page. The argument *get* displays the file without breaks.

This page is intentionally left blank.

6   SNMP-Based Management

The device provides an embedded SNMP Agent to operate with a third-party SNMP Manager (e.g., element management system or EMS) for operation, administration, maintenance, and provisioning (OAMP) of the device. The SNMP Agent supports standard Management Information Base (MIBs) and proprietary MIBs, enabling a deeper probe into the interworking of the device. The SNMP Agent can also send unsolicited events (SNMP traps) towards the SNMP Manager. All supported MIB files are supplied to customers as part of the release.

This section provides configuration relating to SNMP management.

> **Note:** For more information on SNMP support such as SNMP traps, refer to the *SNMP User's Guide*.

6.1   Configuring SNMP Community Strings

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

For detailed descriptions of the SNMP parameters, see 'SNMP Parameters' on page 487.

➢   **To configure the SNMP community strings:**

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

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

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

2.   Configure the SNMP community strings parameters according to the table below.

3.   Click **Submit** to apply your changes.

4.   To save the changes to flash memory, see 'Saving Configuration' on page 362.

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

Table 6-1: SNMP Community String Parameters Description

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

6.2   Configuring SNMP Trap Destinations

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

➢ **To configure SNMP trap destinations:**

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

Figure 6-1: SNMP Trap Destinations Page

		IP Address	Trap Port	Trap User	Trap Enable
☑	SNMP Manager 1	0.0.0.0	162	v2cParams ▼	Enable ▼
☑	SNMP Manager 2	0.0.0.0	162	hq-snmpv3 ▼	Enable ▼
☐	SNMP Manager 3	0.0.0.0	162	v2cParams ▼	Enable ▼
☐	SNMP Manager 4	0.0.0.0	162	v2cParams ▼	Enable ▼
☐	SNMP Manager 5	0.0.0.0	18	v2cParams ▼	Enable ▼

2. Configure the SNMP trap manager parameters according to the table below.

3. Select the check box corresponding to the SNMP Manager that you wish to enable.

4. Click **Submit** to apply your changes.

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

Table 6-2: SNMP Trap Destinations Parameters Description

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

Parameter	Description
Web: Trap User **[SNMPManagerTrapUser]**	Associates a trap user with the trap destination. This determines the trap format, authentication level, and encryption level. ▪ v2cParams (default) = SNMPv2 user community string ▪ SNMPv3 user configured in 'Configuring SNMP V3 Users' on page 91
Trap Enable **[SNMPManagerTrapSendingEnable_x]**	Activates the sending of traps to the SNMP Manager. ▪ **[0]** Disable ▪ **[1]** Enable (Default)

6.3   Configuring SNMP Trusted Managers

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

> **Notes:** The SNMP Trusted Managers table can also be configured using the table ini file parameter, SNMPTrustedMgr_x (see 'SNMP Parameters' on page 487).

➢ **To configure SNMP Trusted Managers:**

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

Figure 6-2: SNMP Trusted Managers

2. Select the check box corresponding to the SNMP Trusted Manager that you want to enable and for whom you want to define an IP address.
3. Define an IP address in dotted-decimal notation.
4. Click **Submit** to apply your changes.
5. To save the changes, see 'Saving Configuration' on page 362.

6.4   Configuring SNMP V3 Users

The SNMP v3 Users page allows you to configure authentication and privacy for up to 10 SNMP v3 users.

➢ **To configure SNMP v3 users:**

1. Open the SNMP v3 Users page (**Configuration** tab > **System** menu > **Management** submenu > **SNMP** submenu > **SNMP V3 Users**).

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

Figure 6-3: SNMP V3 Setting Page - Add Record Dialog Box

3. Configure the SNMP V3 Setting parameters according to the table below.

4. Click **Submit** to apply your settings.

5. To save the changes, see 'Saving Configuration' on page 362.

> **Notes:**
>
> - If you delete a user that is associated with a trap destination (in 'Configuring SNMP Trap Destinations' on page 89), the configured trap destination becomes disabled and the trap user reverts to default (i.e., SNMPv2).
> - The SNMP v3 Users table can also be configured using the table ini file parameter, SNMPUsers (see 'SNMP Parameters' on page 487).

Table 6-3: SNMP V3 Users Parameters

Parameter	Description
Index **[SNMPUsers_Index]**	The table index. The valid range is 0 to 9.
User Name **[SNMPUsers_Username]**	Name of the SNMP v3 user. This name must be unique.
Authentication Protocol **[SNMPUsers_AuthProtocol]**	Authentication protocol of the SNMP v3 user. ▪ **[0]** None (default) ▪ **[1]** MD5 ▪ **[2]** SHA-1

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

7　OVOC-Based Management

AudioCodes One Voice Operations Center (OVOC) is an advanced solution for standards-based management of gateways within VoP networks, covering all areas vital for the efficient monitoring and provisioning of AudioCodes' families of gateways. The OVOC enables Network Equipment Providers (NEPs) and System Integrators (SIs) the ability to offer customers rapid time-to-market and inclusive, cost-effective management of next-generation networks.

> **Note:** For more information on using the OVOC tool, refer to the *OVOC User's Manual* and *OVOC Server IOM Manual*.

This page is intentionally left blank.

8 INI File-Based Management

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

> **Notes:**
> - For a list and description of the *ini* file parameters, see 'Configuration Parameters Reference' on page 471.
> - To restore the device to default settings using the *ini* file, see 'Restoring Factory Defaults' on page 385.

8.1 INI File Format

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

■ Individual parameters - see 'Configuring Individual ini File Parameters' on page 95

■ Table parameters - see 'Configuring Table ini File Parameters' on page 96

8.1.1 Configuring Individual ini File Parameters

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

■ An optional, subsection name (or group name) enclosed in square brackets "[...]". This is used to conveniently group similar parameters by their functionality.

■ Parameter name, followed by an equal "=" sign and then its value.

■ Comments must be preceded by a semicolon ";".

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

For general *ini* file formatting rules, see 'General ini File Formatting Rules' on page 97.

8.1.2 Configuring Table ini File Parameters

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

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

■ **Title of the table:** The name of the table in square brackets, e.g., [MY_TABLE_NAME].

■ **Format line:** Specifies the columns of the table (by their string names) that are to be configured.

• The first word of the Format line must be "FORMAT", followed by the Index field name and then an equal "=" sign. After the equal sign, the names of the columns are listed.

• Columns must be separated by a comma ",".

• The Format line must only include columns that can be modified (i.e., parameters that are not specified as read-only). An exception is Index fields, which are mandatory.

• The Format line must end with a semicolon ";".

■ **Data line(s):** Contain the actual values of the columns (parameters). The values are interpreted according to the Format line.

• The first word of the Data line must be the table's string name followed by the Index field.

• Columns must be separated by a comma ",".

• A Data line must end with a semicolon ";".

■ **End-of-Table Mark:** Indicates the end of the table. The same string used for the table's title, preceded by a backslash "\", e.g., [\MY_TABLE_NAME].

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

```
[Table_Title]
; This is the title of the table.
FORMAT Index = Column_Name1, Column_Name2, Column_Name3;
; This is the Format line.
Index 0 = value1, value2, value3;
Index 1 = value1, $$, value3;
; These are the Data lines.
[\Table_Title]
; This is the end-of-the-table-mark.
```

The table ini file parameter formatting rules are listed below:

■ Indices (in both the Format and the Data lines) must appear in the same order. The Index field must never be omitted.

■ The Format line can include a subset of the configurable fields in a table. In this case, all other fields are assigned with the pre-defined default values for each configured line.

■ The order of the fields in the Format line isn't significant (as opposed to the Index fields). The fields in the Data lines are interpreted according to the order specified in the Format line.

■ The double dollar sign ($$) in a Data line indicates the default value for the parameter.

■ The order of the Data lines is insignificant.

■ Data lines must match the Format line, i.e., it must contain exactly the same number of Indices and Data fields and must be in exactly the same order.

■ A row in a table is identified by its table name and Index field. Each such row may appear only once in the *ini* file.

■ Table dependencies: Certain tables may depend on other tables. For example, one table may include a field that specifies an entry in another table. This method is used to specify additional attributes of an entity, or to specify that a given entity is part of a larger entity. The tables must appear in the order of their dependency (i.e., if Table X is referred to by Table Y, Table X must appear in the *ini* file before Table Y).

For general *ini* file formatting rules, see 'General ini File Formatting Rules' on page 97.

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

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

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

8.1.3   General ini File Formatting Rules

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

■ The *ini* file name must not include hyphens "-" or spaces; if necessary, use an underscore "_" instead.

■ Lines beginning with a semi-colon ";" are ignored. These can be used for adding remarks in the *ini* file.

■ A carriage return (i.e., Enter) must be done at the end of each line.

■ The number of spaces before and after the equals sign "=" is irrelevant.

■ Subsection names for grouping parameters are optional.

■ If there is a syntax error in the parameter name, the value is ignored.

■ Syntax errors in the parameter's value can cause unexpected errors (parameters may be set to the incorrect values).

■ Parameter string values that denote file names (e.g., CallProgressTonesFileName) must be enclosed with inverted commas, e.g., CallProgressTonesFileName = 'cpt_usa.dat'.

■ The parameter name is not case-sensitive.

■ The parameter value is not case-sensitive, except for coder names.

■ The *ini* file must end with at least one carriage return.

8.2 Loading an ini File

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

- Web interface, using any of the following pages:
  - Configuration File - see 'Backing Up and Loading Configuration File' on page 384
  - Load Auxiliary Files - see 'Loading Auxiliary Files' on page 365
- AudioCodes AcBootP utility, which uses Bootstrap Protocol (BootP) and acts as a TFTP server. For information on using the AcBootP utility, refer to AcBootP Utility User's Guide.
- Any standard TFTP server. This is done by storing the ini file on a TFTP server and then having the device download the file from it.

When loaded to the device, the configuration settings of the *ini* file are saved to the device's non-volatile memory. If a parameter is not included in the loaded *ini* file, the following occurs:

- Using the Load Auxiliary Files page: Current settings for parameters that were not included in the loaded ini file are retained.
- All other methods: The default is assigned to the parameters that were not included in the loaded ini file and thereby, overriding values previously configured for these parameters.

> **Notes:**
>
> - For a list and description of the *ini* file parameters, see 'Configuration Parameters Reference' on page 471.
> - Some parameters are configurable only through the *ini* file (and not the Web interface).
> - To restore the device to default settings using the *ini* file, see 'Restoring Factory Defaults' on page 385.

8.3 Modifying an ini File

You can modify an *ini* file currently used by the device. Modifying an *ini* file instead of loading an entirely new *ini* file preserves the device's current configuration.

➢ **To modify an *ini* file:**

1. Save the device's configuration as an *ini* file on your computer, using the Web interface (see 'Loading an ini File' on page 98).

2. Open the *ini* file using a text file editor such as Notepad, and then modify the *ini* file parameters as required.

3. Save the modified *ini* file, and then close the file.

4. Load the modified *ini* file to the device (see 'Loading an ini File' on page 98).

> **Tip:** Before loading the *ini* file to the device, verify that the file extension of the file is *.ini*.

8.4 Secured Encoded ini File

The *ini* file contains sensitive information that is required for the functioning of the device. The file may be loaded to the device using TFTP or 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 *DConvert Utility User's Guide*.

> **Notes:**
>
> • The procedure for loading an encoded *ini* file is identical to the procedure for loading an unencoded *ini* file (see 'Loading an ini File' on page 98).
>
> • If you download from the device (to a folder on your computer) an *ini* file that was loaded encoded to the device, the file is saved as a regular *ini* file (i.e., unencoded).

8.5 Configuring Password Display in ini File

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

■ Obscured: The password characters are concealed and displayed as an encoded string. The password is displayed using the syntax, *1<obscured password>*, for example, 1S3p+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 chosen 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>).

Part III

General System Settings

9   Configuring Certificates

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

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

9.1   Replacing the Device's Certificate

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

➢ **To replace the device's certificate:**

1. Your network administrator should allocate a unique DNS name for the device (e.g., dns_name.corp.customer.com). This DNS name is used to access the device and therefore, must be listed in the server certificate.

2. If the device is operating in HTTPS mode, then set the 'Secured Web Connection (HTTPS)' parameter (HTTPSOnly) to **HTTP and HTTPS** (see 'Configuring Web Security Settings' on page 73). This ensures that you have a method for accessing the device in case the new certificate does not work. Restore the previous setting after testing the configuration.

3. Open the Certificates page (**Configuration** tab > **System** menu > **Certificates**).

4. Under the **Certificate Signing Request** group, do the following:

   a. In the 'Subject Name [CN]' field, enter the DNS name.

   b. Fill in the rest of the request fields according to your security provider's instructions.

   c. Click the **Create CSR** button; a textual certificate signing request is displayed in the area below the button:

Figure 9-1: Certificate Signing Request Group

5. Copy the text and send it to your security provider. The security provider, also known as Certification Authority or CA, signs this request and then sends you a server certificate for the device.

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

```
-----BEGIN CERTIFICATE-----
MIIDkzCCAnugAwIBAgIEAgAAADANBgkqhkiG9w0BAQQFADA/MQswCQYDVQQGEw
JGUjETMBEGA1UEChMKQ2VydGlwb3N0ZTEbMBkGA1UEAxMSQ2VydGlwb3N0ZSBT
ZXJ2ZXVyMB4XDTk4MDYyNDA4MDAwMFoXDTE4MDYyNDA4MDAwMFowPzELMAkGA1
UEBhMCRlIxEzARBgNVBAoTCkNlcnRpcG9zdGUxGzAZBgNVBAMTEkNlcnRpcG9z
dGUgU2VydmV1cjCCASEwDQYJKoZIhvcNAQEBBQADggEOADCCAQkCggEAPqd4Mz
iR4spWldGRx8bQrhZkonWnNm`+Yhb7+4Q67ecf1janH7GcN/SXsfx7jJpreWUL
f7v7Cvpr4R7qIJcmdHIntmf7JPM5n6cDBv17uSW63er7NkVnMFHwK1QaGFLMyb
FkzaeGrvFm4k3lRefiXDmuOe+FhJgHYezYHf44LvPRPwhSrzi9+Aq3o8pWDguJ
uZDIUP1F1jMa+LPwvREXfFcUW+w==
-----END CERTIFICATE-----
```

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

8. After the certificate successfully loads to the device, save the configuration with a device reset (see 'Saving Configuration' on page 362); the Web interface uses the provided certificate.

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

Figure 9-2: Private key "OK" in Certificate Information Group

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

Notes:

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

9.2   Loading a Private Key

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

➢ **To replace the device's private key:**

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

2. If the device is operating in HTTPS mode, then set the 'Secured Web Connection (HTTPS)' field (HTTPSOnly) to **HTTP and HTTPS** (see 'Configuring Web Security Settings' on page 73). This ensures that you have a method for accessing the device in case the new configuration does not work. Restore the previous setting after testing the configuration.

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

Figure 9-3: Upload Certificate Files from your Computer Group
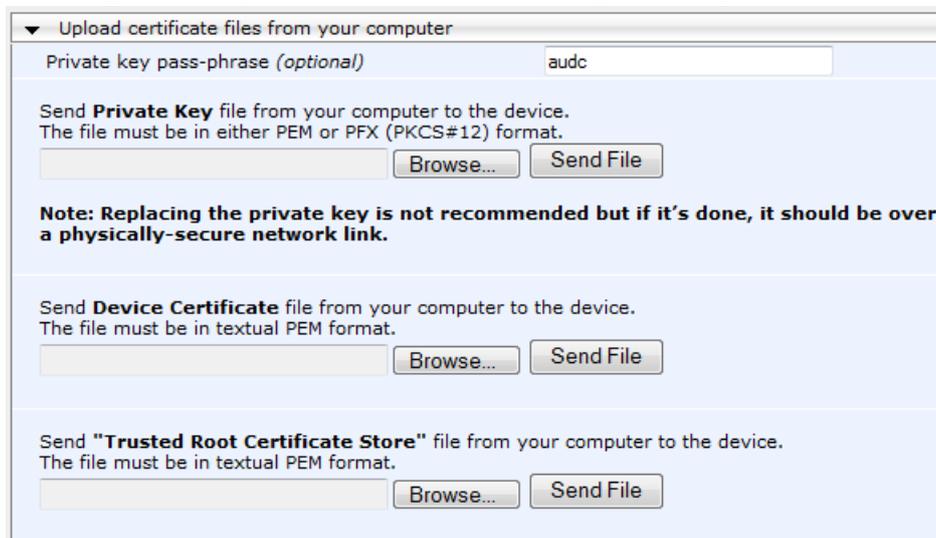
4. Fill in the 'Private key pass-phrase' field, if required.

5. Click the **Browse** button corresponding to the 'Send Private Key' field, navigate to the key file, and then click **Send File**.

6. If the security administrator has provided you with a device certificate file, load it using the 'Send Device Certificate' field.

7. After the files successfully load to the device, save the configuration with a device reset (see 'Saving Configuration' on page 362); the Web interface uses the new configuration.

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

9. If the device was originally operating in HTTPS mode and you disabled it in Step 2, then enable it by setting the 'Secured Web Connection (HTTPS)' field to **HTTPS Only**.

9.3   Mutual TLS Authentication

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

Since X.509 certificates have an expiration date and time, the device must be configured to use NTP (see 'Simple Network Time Protocol Support' on page 111) to obtain the current date and time. Without the correct date and time, client certificates cannot work.

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

1. Set the 'Secured Web Connection (HTTPS)' field to **HTTPS Only** (see 'Configuring Web Security Settings' on page 73) to ensure you have a method for accessing the device in case the client certificate does not work. Restore the previous setting after testing the configuration.

2. Open the Certificates page (see 'Replacing the Device's Certificate' on page 103).

3. In the **Upload certificate files from your computer** group, click the **Browse** button corresponding to the 'Send Trusted Root Certificate Store ...' field, navigate to the file, and then click **Send File**.

4. When the operation is complete, set the 'Requires Client Certificates for HTTPS connection' field to **Enable** (see 'Configuring Web Security Settings' on page 73).

5. Save the configuration with a device reset (see 'Saving Configuration' on page 362).

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

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

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

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

> **Notes:**
>
> • The process of installing a client certificate on your PC is beyond the scope of this document. For more information, refer to your operating system documentation, and/or consult your security administrator.
>
> • The root certificate can also be loaded via the Automatic Update facility, using the HTTPSRootFileName *ini* file parameter.
>
> • You can enable the device to check whether a peer's certificate has been revoked by an Online Certificate Status Protocol (OCSP) server (see Configuring Certificate Revocation Checking (OCSP) on page 108.

9.4   Self-Signed Certificates

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

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

1. Before you begin, ensure the following:

- You have a unique DNS name for the device (e.g., dns_name.corp.customer.com). This name is used to access the device and should therefore, be listed in the server certificate.
- No traffic is running on the device. The certificate generation process is disruptive to traffic and should be executed during maintenance time.

2. Open the Certificates page (see 'Replacing the Device's Certificate' on page 103).

3. In the 'Subject Name [CN]' field, enter the fully-qualified DNS name (FQDN) as the certificate subject, select the desired private key size (in bits), and then click **Generate self-signed**; after a few seconds, a message appears displaying the new subject name.

4. Save the configuration with a device reset (see 'Saving Configuration' on page 362) for the new certificate to take effect.

9.5   TLS Server Certificate Expiry Check

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

> **Note:**   If the device's default self-signed certificate is about to expire (less than a day) or has expired, the device automatically re-generates a new self-signed certificate. The configuration described in this section does not apply to this mechanism, which occurs regardless.

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

1. Open the Certificates page (see 'Replacing the Device's Certificate' on page 103).

2. In the 'TLS Expiry Check Start' field, enter the number of days before the installed TLS server certificate is to expire at which the device must send a trap to notify of this.

Figure 9-4: TLS Expiry Settings Group

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

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

4. Click the **Submit TLS Expiry Settings** button.

9.6 Configuring Certificate Revocation Checking (OCSP)

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

➢ **To configure OCSP:**

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

Figure 9-5: OCSP Parameters

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

2. Configure the OCSP parameters as required. For a description of these parameters, see OCSP Parameters on page 511.

3. Click **Submit**.

Notes:

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

9.7   Loading Certificate Chain for Trusted Root

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

Figure 9-6: Certificate Chain Hierarchy
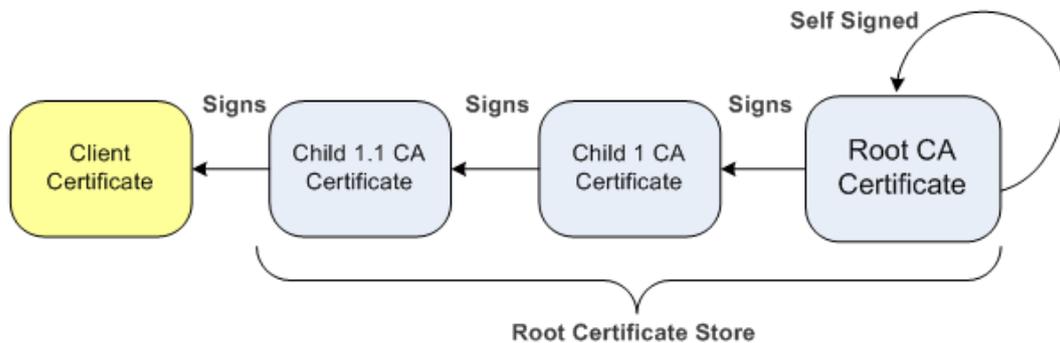
For the device to trust a whole chain of certificates, you need to combine the certificates into one text file (using a text editor). Once done, upload the file using the 'Trusted Root Certificate Store' field in the Certificates page.

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

This page is intentionally left blank.

10   Date and Time

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

10.1   Configuring Date and Time Manually

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

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

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

Figure 10-1: Regional Settings Page

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

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

3.   Click the **Submit** button.

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

10.2   Automatic Date and Time through SNTP Server

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

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

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

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

If required, the clock update is performed by the client as the final step of the update process. The update is performed in such a way as to be transparent to the end users. For instance, the response of the server may indicate that the clock is running too fast on the client. The client slowly robs bits from the clock counter to update the clock to the correct time. If the clock is running too slow, then in an effort to catch the clock up, bits are added to the counter,

causing the clock to update quicker and catch up to the correct time. The advantage of this method is that it does not introduce any disparity in the system time that is noticeable to an end user or that could corrupt call timeouts and timestamps.

The procedure below describes how to configure SNTP.

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

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

Figure 10-2: SNTP Configuration in Application Settings Page

2. Configure the NTP parameters:
   - 'NTP Server DN/IP' (NTPServerIP) - defines the IP address or FQDN of the NTP server.
   - 'NTP UTC Offset' (NTPServerUTCOffset) - defines the time offset in relation to the UTC. For example, if your region is 2 hours ahead of the UTC, enter "2".
   - 'NTP Updated Interval' (NTPUpdateInterval) - defines the period after which the date and time of the device is updated.
   - 'NTP Secondary Server IP' (NTPSecondaryServerIP) - defines the secondary NTP server.

3. Configure daylight saving, if required:
   - 'Day Light Saving Time' (DayLightSavingTimeEnable) - enables daylight saving time.
   - 'DST Mode' - Determines the range type for configuring the start and end date for daylight saving:
     - **Day of Year:** The range is configured by date of month, for example, from January 4 to August 31.
     - **Day of month:** The range is configured by day of month, for example, from the second Sunday of May January to the last Sunday of August.
   - 'Start Time' (DayLightSavingTimeStart) and 'End Time' (DayLightSavingTimeEnd) - defines the period for which daylight saving time is relevant.
   - 'Offset' (DayLightSavingTimeOffset) - defines the offset in minutes to add to the time for daylight saving. For example, if your region has daylight saving of one hour, the time received from the NTP server is 11:00, and the UTC offset for your region is +2 (i.e., 13:00), you need to enter "60" to change the local time to 14:00.

4. Verify that the device is set to the correct date and time. You can do this by viewing the date and time in the Regional Settings page, as described in 'Configuring Date and Time Manually' on page 111.

Part IV

General VoIP Configuration

11 Network

This section describes the network-related configuration.

11.1 Ethernet Interface Configuration

The device's Ethernet connection can be configured, using the *ini* file parameter EthernetPhyConfiguration, to one of the following modes:

■ **Manual:**

- 10Base-T Full-Duplex
- 100Base-TX Half-Duplex or 100Base-TX Full-Duplex

■ **Auto-Negotiation:** chooses common transmission parameters such as speed and duplex mode

The Ethernet connection should be configured according to the following recommended guidelines:

■ When the device's Ethernet port is configured for Auto-Negotiation, the opposite port must also operate in Auto-Negotiation. Auto-Negotiation falls back to Half-Duplex mode when the opposite port is not in Auto-Negotiation mode, but the speed in this mode is always configured correctly. Configuring the device to Auto-Negotiation mode while the opposite port is set manually to Full-Duplex is invalid as it causes the device to fall back to Half-Duplex mode while the opposite port is Full-Duplex. Any mismatch configuration can yield unexpected functioning of the Ethernet connection.

■ When configuring the device's Ethernet port manually, the same mode (i.e., Half Duplex or Full Duplex) and speed must be configured on the remote Ethernet port. In addition, when the device's Ethernet port is configured manually, it is invalid to set the remote port to Auto-Negotiation**.** Any mismatch configuration can yield unexpected functioning of the Ethernet connection.

■ It's recommended to configure the port for best performance and highest bandwidth (i.e., Full Duplex with 100Base-TX), but at the same time adhering to the guidelines listed above.

> **Note:** For remote configuration, the device should be in the correct Ethernet setting prior to the time this parameter takes effect. When, for example, the device is configured using BootP/TFTP, the device performs many Ethernet-based transactions prior to reading the *ini* file containing this device configuration parameter. To resolve this problem, the device always uses the last Ethernet setup mode configured. In this way, if you want to configure the device to operate in a new network environment in which the current Ethernet setting of the device is invalid, you should first modify this parameter in the current network so that the new setting holds next time the device is restarted. After reconfiguration has completed, connect the device to the new network and restart it. As a result, the remote configuration process that occurs in the new network uses a valid Ethernet configuration

11.2 Configuring IP Network Interfaces

You can configure a single VoIP network interface for all applications, which includes OAMP (management traffic), call control (SIP messages), and media (RTP traffic), or you can configure multiple logical, IP network interfaces for these applications. A need often arises to have 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 for the OAMP, call control, and media applications. The device is connected to a VLAN-aware switch for directing traffic from and to the device to the three separated Layer-3 broadcast domains according to VLAN tags (middle pane).

Figure 11-1: Multiple Network Interfaces
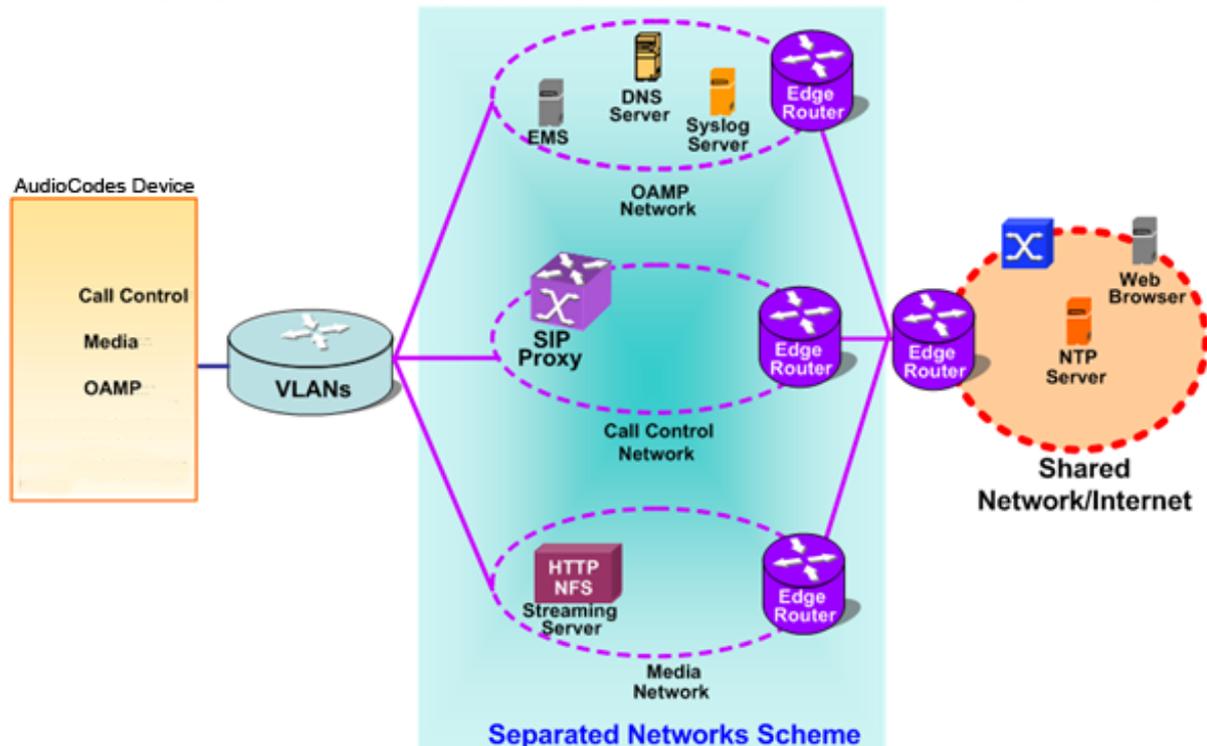
The Multiple Interface Table page allows you to configure these network interfaces. Each row of the table defines a logical IP interface with the following attributes:

■ Application type allowed on the interface:

- Control - call control signaling traffic (i.e., SIP)
- Media - RTP traffic
- Operations, Administration, Maintenance and Provisioning (OAMP) - management (such as Web- and SNMP-based management)

■ IP address and subnet mask represented by prefix length

■ VLAN ID (if VLANs are enabled)

■ Default Gateway - traffic from this interface destined to a subnet that does not meet any of the routing rules, local or static routes, are forwarded to this gateway (as long this application type is allowed on this interface.

■ Primary and secondary DNS IP address (optional)

You can configure up to 16 interfaces, consisting of up to 15 Control and Media interfaces and 1 OAMP interface.

This page also provides VLAN-related parameters for enabling VLANs and defining the Native VLAN ID. This is the VLAN ID to which incoming, untagged packets are assigned. You can also configure Quality of Service (QoS) by assigning VLAN priorities and

Differentiated Services (DiffServ) for the supported Class of Service (CoS). For configuring Quality of Service (QoS), see 'Configuring the QoS Settings' on page 131.

Complementing the Multiple Interface table is the IP Routing table, which allows you to define static routing rules for non-local hosts/subnets. For more information, see 'Configuring the IP Routing Table' on page 127.

Notes:

- Before adding IP network interfaces to the Multiple Interface table, see Multiple Interface Table Configuration Rules on page 121 for the rules on configuring valid IP network interfaces.

- When booting using BootP/DHCP protocols, an IP address is obtained from the server. This address is used as the OAMP address for the initial session, overriding the address configured in the Multiple Interface table. The address configured for OAMP applications in this table becomes available only after you save the configuration to the device's flash with a reset. This enables the device to operate with a temporary address for initial management and configuration while retaining the address configured in this table for deployment.

- The Multiple Interface table can also be configured using the table ini file parameter, InterfaceTable (see 'Networking Parameters' on page 471).

➢ **To configure IP network interfaces:**

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

Figure 11-2: IP Settings Page (Single Network Interface)

Parameter	Description
Web: Application Type OVOC: Application Types **[InterfaceTable_ApplicationTypes]**	Defines the applications allowed on the interface. ▪ **[0]** OAMP = Operations, Administration, Maintenance and Provisioning (OAMP) applications (e.g., Web, Telnet, SSH, and SNMP). ▪ **[1]** Media = Media (i.e., RTP streams of voice). ▪ **[2]** Control = Call Control applications (e.g., SIP). ▪ **[3]** OAMP + Media = OAMP and Media applications. ▪ **[4]** OAMP + Control = OAMP and Call Control applications. ▪ **[5]** Media + Control = Media and Call Control applications. ▪ **[6]** OAMP + Media + Control = All application types are allowed on the interface. **Note:** For valid configuration, see Multiple Interface Table Configuration Rules on page 121.
Web/OVOC: IP Address **[InterfaceTable_IPAddress]**	Defines the IPv4 IP address in dotted-decimal notation. **Note:** For valid configuration, see Multiple Interface Table Configuration Rules on page 121.
Web/OVOC: Prefix Length **[InterfaceTable_PrefixLength]**	Defines the prefix length of the related IP address. This is a Classless Inter-Domain Routing (CIDR)-style representation of a dotted-decimal subnet notation. The CIDR-style representation uses a suffix indicating the number of bits which are set in the dotted-decimal format. For example, 192.168.0.0/16 is synonymous with 192.168.0.0 and subnet 255.255.0.0. This CIDR lists the number of '1' bits in the subnet mask (i.e., replaces the standard dotted-decimal representation of the subnet mask for IPv4 interfaces). For example, a subnet mask of 255.0.0.0 is represented by a prefix length of 8 (i.e., 11111111 00000000 00000000 00000000) and a subnet mask of 255.255.255.252 is represented by a prefix length of 30 (i.e., 11111111 11111111 11111111 11111100). The prefix length is a Classless Inter-Domain Routing (CIDR) style presentation of a dotted-decimal subnet notation. The CIDR-style presentation is the latest method for interpretation of IP addresses. Specifically, instead of using eight-bit address blocks, it uses the variable-length subnet masking technique to allow allocation on arbitrary-length prefixes. The prefix length for IPv4 can range from 0 to 30. **Note:** For valid configuration, see Multiple Interface Table Configuration Rules on page 121.
Web/OVOC: Gateway **[InterfaceTable_Gateway]**	Defines the IP address of the default gateway for the interface. When traffic is sent from this interface to an unknown destination (i.e., not in the same subnet and not defined for any static routing rule), it is forwarded to this default gateway. **Note:** For valid configuration, see Multiple Interface Table Configuration Rules on page 121.

Parameter	Description
Web/OVOC: VLAN ID **[InterfaceTable_VlanID]**	Defines a VLAN ID for the interface. Incoming traffic tagged with this VLAN ID is routed to the corresponding interface. Outgoing traffic from this interface is tagged with this VLAN ID. **Notes:** ▪ To enable VLANs, use the 'VLAN Mode' parameter. ▪ The device can use the discovery protocol, Link Layer Discovery Protocol (LLDP) to obtain (over the Layer-2 data link layer) the VLAN ID for its OAMP interface. For further information, see the EnableLLDP parameter. ▪ For valid configuration, see Multiple Interface Table Configuration Rules on page 121.
Web/OVOC: Interface Name **[InterfaceTable_InterfaceName]**	Defines a name for this interface. It is also displayed in management interfaces (Web, CLI, and SNMP) for clarity where it has no functional use. The valid value is a string of up to 16 characters. **Note:** For valid configuration, see Multiple Interface Table Configuration Rules on page 121.
Web/OVOC: Primary DNS Server IP address **[InterfaceTable_PrimaryDNSServerIPAddress]**	(Optional) Defines the primary DNS server's IP address (in dotted-decimal notation), which is used for translating domain names into IP addresses for the interface. By default, no IP address is defined.
Web/OVOC: Secondary DNS Server IP address **[InterfaceTable_SecondaryDNSServerIPAddress]**	(Optional) Defines the secondary DNS server's IP address (in dotted-decimal notation), which is used for translating domain names into IP addresses for the interface. By default, no IP address is defined.
General Parameters	
Web/OVOC: VLAN Mode **[VLANMode]**	Enables VLANs tagging (IEEE 802.1Q). ▪ [0] Disable (default) ▪ [1] Enable **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ To operate with multiple network interfaces, VLANs must be enabled. ▪ VLANs are available only when booting the device from flash. When booting using BootP/DHCP protocols, VLANs are disabled to allow easier maintenance access. In this scenario, multiple network interface capabilities are unavailable.

Parameter	Description
Web/OVOC: Native VLAN ID **[VLANNativeVLANID]**	Defines the Native VLAN ID. This is the VLAN ID to which untagged incoming traffic is assigned. Outgoing packets sent to this VLAN are sent only with a priority tag (VLAN ID = 0). When the Native VLAN ID is equal to one of the VLAN IDs listed in the Multiple Interface table (and VLANs are enabled), untagged incoming traffic is considered as incoming traffic for that interface. Outgoing traffic sent from this interface is sent with the priority tag (tagged with VLAN ID = 0). When the Native VLAN ID is different to any value in the 'VLAN ID' column in the table, untagged incoming traffic is discarded and all outgoing traffic is tagged. The default Native VLAN ID is 1. **Note:** If this parameter is not configured (i.e., default is 1) and one of the interfaces has a VLAN ID set to 1, this interface is still considered the 'Native' VLAN. If you do not wish to have a 'Native' VLAN ID and want to use VLAN ID 1, set this parameter to a value other than any VLAN ID in the table.

11.2.1 Assigning NTP Services to Application Types

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

11.2.2 Multiple Interface Table Configuration Rules

The Multiple Interface table configuration must adhere to the following rules:

■ Each interface must have its own subnet. Configuring two interfaces with addresses in the same subnet (e.g., 192.168.0.1/16 and 192.168.100.1/16) is invalid.

■ Subnets of different interfaces must not overlap (i.e. 10.0.0.1/8 and 10.50.10.1/24 is invalid); each interface must have its own address space.

■ Each interface must be assigned a unique IP address (i.e., two interfaces may not share the same address space, or even part of it).

■ The prefix length replaces the dotted-decimal subnet mask presentation and must have a value of 0-30 for IPv4 addresses.

■ Only 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 with an IPv4 address.

■ At least one Media interface must be configured with an IPv4 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 (with VLANs)

• Example 3:

♦ One combined OAMP-Media interface with an IPv4 address

♦ One or more combined Media-Control interfaces with IPv4 addresses.

■ 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 IP Routing table.

■ The interface name must be configured (mandatory) and unique for each interface, and can include up to 16 characters.

■ For IPv4 addresses, the 'Interface Mode' column must be set to IPv4 Manual (numeric value 10).

■ A different VLAN ID can be assigned to different network interface types. For example, VLAN ID 100 for OAMP, VLAN ID 200 for Media, and VLAN ID 300 for Control. However, you cannot assign the same Each network interface must be assigned a unique VLAN ID.

■ When configuring more than one IP interface of the same address family, VLANs must be enabled.

■ For network configuration to take effect, you must save the configuration to the device's flash memory (burn) with a device reset.

Notes:

• When configuring the network interfaces and VLANs in the Multiple Interface table using the Web interface, it is recommended to check that your configuration is valid, by clicking the **Done** button in the Multiple Interface Table page.

• Upon device start up, the Multiple Interface table is parsed and passes comprehensive validation tests. If any errors occur during this validation phase, the device sends an error message to the Syslog server and falls back to a "safe mode", using a single interface and no VLANs. Ensure that you view the Syslog messages that the device sends in system startup to see if any errors occurred.

11.2.3 Troubleshooting the Multiple Interface Table

If any of the Multiple Interface table guidelines are violated, the device falls back to a "safe mode" configuration, consisting of a single IPv4 interface without VLANs. For more information on validation failures, consult the Syslog messages.

Validation failures may be caused by one of the following:

■ One of the Application Types (OAMP, Control, or Media) are missing in the IPv4 interfaces.

■ There are too many interfaces for Application Type, OAMP. There is only one interface defined, but the 'Application Types' column is not set to **OAMP + Media + Control** (numeric value 6).

■ An IPv4 interface was defined with 'Interface Type' other than **IPv4 Manual** (10).

■ Two interfaces have the same VLAN ID value while VLANs are enabled.

■ Two interfaces have the same name.

■ At least two interfaces share the same address space or subnet.

Apart from these validation errors, connectivity problems may be caused by one of the following:

■ Trying to access the device with VLAN tags while booting from BootP/DHCP.

■ Trying to access the device with untagged traffic when VLANs are on and Native VLAN is not configured properly.

■ The IP Routing table is not configured properly.

11.2.4 Networking Configuration Examples

This section provides configuration examples of networking interfaces.

11.2.4.1 One VoIP Interface for All Applications

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

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

Table 11-2: Example of Single VoIP Interface in Multiple Interface Table

Index	Application Type	IP Address	Prefix Length	Default	VLAN ID	Interface Name
0	OAMP, Media & Control	192.168.85.14	16	192.168.0.1	1	myInterface

2. VLANS are not required and the Native VLAN ID is irrelevant. Class of Service parameters may have default values.

3. **IP Routing table:** Two routes are configured for directing traffic for subnet 201.201.0.0/16 to 192.168.0.2, and all traffic for subnet 202.202.0.0/16 to 192.168.0.3:

Table 11-3: Example of IP Routing Table

Destination IP Address	Prefix Length	Gateway IP Address	Metric
201.201.0.0	16	192.168.0.2	1
202.202.0.0	16	192.168.0.3	1

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

11.2.4.2 VoIP Interface per Application Type

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

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

Table 11-4: Example of VoIP Interfaces per Application Type in Multiple Interface Table

Index	Application Type	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	OAMP	192.168.85.14	16	0.0.0.0	1	ManagementIF
1	Control	200.200.85.14	24	200.200.85.1	200	myControlIF
2	Media	211.211.85.14	24	211.211.85.1	211	myMediaIF

2. VLANs are required and the Native VLAN ID is the same VLAN ID as the Management interface (configured for Index 0):

    * 'VLAN Mode' is set to **Enable**.
    * 'Native VLAN ID' field is set to "1".

3. **IP Routing table:** A routing rule is required to allow remote management from a host in 176.85.49.0 / 24:

Table 11-5: Example IP Routing Table

Destination IP Address	Prefix Length	Gateway IP Address	Metric	Interface Name
176.85.49.0	24	192.168.0.1	1	-

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

11.2.4.3 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.

1. **Multiple Interface table:**

Table 11-6: Example of VoIP Interfaces of Combined Application Types in Multiple Interface Table

Index	Application Type	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	OAMP	192.168.85.14	16	192.168.0.1	1	Mgmt
1	Media & Control	200.200.85.14	24	200.200.85.1	201	MediaCntrl1
2	Media & Control	200.200.86.14	24	200.200.86.1	201	MediaCntrl2

2. VLANs are required and the Native VLAN ID is the same VLAN ID as the Management interface (index 0):

- 'VLAN Mode' is set to Enable.
- 'Native VLAN ID' field is set to "1".

3. **IP Routing table:** A routing rule is required to allow remote management from a host in 176.85.49.0/24:

Table 11-7: Example of IP Routing Table

Destination IP Address	Prefix Length	Gateway IP Address	Metric	Interface Name
176.85.49.0	24	192.168.0.10	1	-

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

```
EnableNTPasOAM = 1
```

11.2.4.4 VoIP Interfaces with Multiple Default Gateways

Below is a configuration example using default gateways per IP network interface. In this example, the default gateway 200.200.85.1 is available for applications allowed on Interface #1, whereas outgoing management traffic (originating on Interface #0) is never directed to this default gateway.

Table 11-8: Configured Default Gateway Example

Index	Application Type	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	OAMP	192.168.085.214	16	0.0.0.0	100	Mgmt
1	Media & Control	200.200.85.14	24	200.200.85.1	200	CntrlMedia

A separate IP routing table enables you to 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 192.168.0.1.

Table 11-9: Separate Routing Table Example

Destination IP Address	Prefix Length	Gateway IP Address	Metric	Interface Name	Status
17.17.0.0	16	192.168.0.1	1	0	Active

11.3 Configuring the IP Routing Table

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

➢ **To configure static IP routing:**

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

Figure 11-4: IP Routing Table Page

#	Delete Row	Destination IP Address	Prefix Length	Gateway IP Address	Metric	Interface Name	Status
1	☐	169.254.254.252	30	0.0.0.0	0	InternalIF	Active
2	☐	10.9.0.0	16	0.0.0.0	0	Voice	Active
3	☐	0.0.0.0	0	10.9.0.1	1	Voice	Active
4	☐	0.0.0.0	0	169.254.254.253	2	InternalIF	Active

Delete Selected Entries

Add a new table entry

Destination IP Address	Prefix Length	Gateway IP Address	Metric	Interface Name
	16		1	

Add New Entry

2. In the Add a new table entry table, add a new static routing rule according to the parameters described in the table below.

3. Click **Add New Entry**; the new routing rule is added to the IP routing table.

To delete a routing rule from the table, select the 'Delete Row' check box corresponding to the required routing rule, and then click **Delete Selected Entries**.

> **Notes:**
> - You can delete only inactive routing rules.
> - The IP Routing table can also be configured using the table ini file parameter, StaticRouteTable.

Table 11-10: IP Routing Table Description

Parameter	Description
Destination IP Address **[StaticRouteTable_Destination]**	Defines the IP address of the destination host/network. The destination can be a single host or a whole subnet, depending on the Prefix Length configured for this routing rule.
Prefix Length **[StaticRouteTable_PrefixLength]**	Defines the Classless Inter-Domain Routing (CIDR)-style representation of a dotted-decimal subnet notation, of the destination host/network. The CIDR-style representation uses a suffix indicating the number of bits that are set in the dotted-decimal format. For example, 16 is synonymous with subnet 255.255.0.0.
The address of the host/network you want to reach is determined by an AND operation that is applied to the fields 'Destination IP Address' and 'Prefix Length'. For example, to reach the network 10.8.x.x, enter 10.8.0.0 in the 'Destination IP Address' field and 16 in the 'Prefix Length'. As a result of the AND operation, the value of the last two octets in the 'Destination IP Address' field is ignored. To reach a specific host, enter its IP address in the 'Destination IP Address' field and 32 in the 'Prefix Length' field.	
Gateway IP Address **[StaticRouteTable_Gateway]**	Defines the IP address of the router (next hop) used for traffic destined to the subnet/host as defined in the 'Destination IP Address' / 'Prefix Length' field. **Note:** The Gateway address must be in the same subnet as the IP address of the interface over which you configure this static routing rule.
Metric	Defines the number of hops needed to reach the specified destination. **Note:** The recommended value for this parameter is 1. This parameter must be set to a number greater than 0 for the routing rule to be valid. Routing entries with Hop Count equals 0 are local routes set automatically by the device.
Interface Name **[StaticRouteTable_InterfaceName]**	Assigns a network interface through which the 'Gateway IP Address' is reached. This is the string value as configured for the network interface in the 'Interface Name' field of the Multiple Interface table (see 'Configuring IP Network Interfaces' on page 116). **Note:** The IP address of the 'Gateway IP Address' field must be in the same subnet as this interface's IP address.
Status	Read-only field displaying the status of the static IP route: ▪ "Active" - routing rule is used ny the device. ▪ "Inactive" - routing rule is not applied. 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".

11.3.1 Interface Column

This example describes the configuration of static IP routing rules.

1. Configure network interfaces in the Multiple Interface table, as shown below:

Table 11-11: Configured Network Interfaces in Multiple Interface Table

Index	Application Type	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	OAMP	192.168.0.2	16	192.168.0.1	501	Mng
1	Media & Control	10.32.174.50	24	10.32.174.1	2012	MediaCntrl
2	Media	10.33.174.50	24	10.33.174.1	2013	Media1
3	Control	10.34.174.50	24	10.34.174.1	2014	Cntrl1

2. Configure static IP Routing rules in the IP Routing table, as shown below:

Table 11-12: Configured Static IP Routing Rules in IP Routing Table

Destination IP Address	Prefix Length	Gateway IP Address	Metric	Interface Name
10.31.174.0	24	192.168.11.1	1	Mng
174.96.151.15	24	10.32.174.12	1	MediaCntrl
10.35.174.0	24	10.34.174.240	1	Cntrl1

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

11.3.2 Routing Table Configuration Summary and Guidelines

The Routing table configurations must adhere to the following rules:

- Up to 30 different static routing rules can be configured.
- The 'Prefix Length' replaces the dotted-decimal subnet mask presentation. This column must have a value of 0-31 for IPv4 interfaces.
- The 'Gateway IP Address' field must be on the same subnet as the IP address of the associated interface specified in the 'Interface Name' field.
- The 'Metric; field must be set to 1.
- For the configuration settings to take effect, you must reset the device with a "burn" to flash memory.

11.3.3 Troubleshooting the Routing Table

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

Common routing rule configuration errors may include the following:

■ The IP address specified in the 'Gateway IP Address' field is unreachable from the interface specified in the 'Interface Name' field.

■ The same destination is configured in two different routing rules.

■ More than 30 routing rules have been configured.

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

11.4   Configuring Quality of Service

The QoS Settings page is used for configuring the Layer-2 and Layer-3 Quality of Service (QoS) parameters. 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 their priority, thereby, 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 different VLAN priorities (IEEE 802.1p) and DiffServ to the supported Class of Service (CoS):

■ Network Service class – network control traffic (ICMP, ARP)

■ Premium Media service class – used for RTP media traffic

■ Premium Control service class – used for call control (i.e., SIP) traffic

■ Gold service class – used for streaming applications

■ Bronze service class – used for OAMP applications

The Layer-2 QoS parameters define the values for the 3 priority bits in the VLAN tag of frames related to a specific service class (according to the IEEE 802.1p standard). The Layer-3 QoS parameters define the values of the DiffServ field in the IP Header of the frames related to a specific service class.

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

Table 11-13: Traffic/Network Types and Priority

Application	Traffic / Network Types	Class-of-Service (Priority)
Debugging interface	Management	Bronze
Telnet	Management	Bronze
DHCP	Management	Network
Web server (HTTP)	Management	Bronze
SNMP GET/SET	Management	Bronze
Web server (HTTPS)	Management	Bronze
IPSec IKE	Determined by the service	Determined by the service
RTP traffic	Media	Premium media
RTCP traffic	Media	Premium media
T.38 traffic	Media	Premium media
SIP	Control	Premium control
SIP over TLS (SIPS)	Control	Premium control
Syslog	Management	Bronze
ICMP	Management	Determined by the initiator of the request
ARP listener	Determined by the initiator of the request	Network
SNMP Traps	Management	Bronze

Application	Traffic / Network Types	Class-of-Service (Priority)
DNS client	Varies according to DNS settings: ▪ OAMP ▪ Control	Depends on traffic type: ▪ Control: Premium Control ▪ Management: Bronze
NTP	Varies according to the interface type associated with NTP (see 'Assigning NTP Services to Application Types' on page 121): ▪ OAMP ▪ Control	Depends on traffic type: ▪ Control: Premium control ▪ Management: Bronze
NFS	NFSServers_VlanType in the NFSServers table	Gold

➢ **To configure QoS:**

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

Priority Settings
- Network Priority: 7
- Media Premium Priority: 6
- Control Premium Priority: 6
- Gold Priority: 4
- Bronze Priority: 2

Differential Services
- Network QoS: 48
- Media Premium QoS: 46
- Control Premium QoS: 40
- Gold QoS: 26
- Bronze QoS: 10

2. Configure the QoS parameters as required.
3. Click **Submit** to apply your changes.
4. Save the changes to flash memory (see 'Saving Configuration' on page 362).

11.5   Disabling ICMP Redirect Messages

You can configure the device's handling of ICMP Redirect messages. These messages can either be rejected (ignored) or permitted.

⚠️ **Note:** You can also configure this feature using the ini file parameter DisableICMPRedirects (see 'Routing Parameters' on page 473).

➢ **To configure the handling of ICMP Redirect messages:**

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

Figure 11-5: Disabling ICMP Redirect in Network Settings Page

| ▼ | | |
| Disable ICMP Redirects | Enable | ▼ |

2. From the 'Disable ICMP Redirects' drop-down list, select the required option.
3. Click **Submit** to apply your changes.

11.6 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 134

■ Internal SRV table - see 'Configuring the Internal SRV Table' on page 135

11.6.1 Configuring the Internal DNS Table

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

> **Notes:**
>
> • The device initially attempts to resolve a domain name using the Internal DNS table. If the domain name isn't listed in the table, the device performs a DNS resolution using an external DNS server for the related IP network interface, configured in the Multiple Interface table (see 'Configuring IP Network Interfaces' on page 116).
>
> • You can also configure the DNS table using the table ini file parameter, DNS2IP (see 'DNS Parameters' on page 478).

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

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

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

Figure 11-6: Internal DNS Table - Add Record Dialog Box

Add Record	✖
Index	0
Domain Name	dnscompany.com
First IP Address	10.8.2.15
Second IP Address	10.8.4.20
Third IP Address	10.8.16.17
Fourth IP Address	0.0.0.0
	🖫 Submit ✖ Cancel

3. Configure the DNS rule, as required. For a description of the parameters, see the table below.

4. Click **Submit**; the DNS rule is added to the table.

Table 11-14: Internal DNS Table Parameter Description

Parameter	Description
Domain Name **[Dns2Ip_DomainName]**	Defines the host name to be translated. The valid value is a string of up to 31 characters.
First IP Address **[Dns2Ip_FirstIpAddress]**	Defines the first IP address (in dotted-decimal format notation) to which the host name is translated.
Second IP Address **[Dns2Ip_SecondIpAddress]**	Defines the second IP address (in dotted-decimal format notation) to which the host name is translated.
Third IP Address **[Dns2Ip_ThirdIpAddress]**	Defines the third IP address (in dotted-decimal format notation) to which the host name is translated.
Fourth IP Address **[Dns2Ip_FourthIpAddress]**	Defines the fourth IP address (in dotted-decimal format notation) to which the host name is translated.

11.6.2   Configuring the Internal SRV Table

The Internal SRV Table page 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.

> **Notes:**
>
> - If the Internal SRV table is configured, the device initially attempts to resolve a domain name using this table. If the domain name isn't found, the device performs a Service Record (SRV) resolution using an external DNS server configured in the Multiple Interface table (see 'Configuring IP Network Interfaces' on page 116).
> - The Internal SRV table can also be configured using the table ini file parameter, SRV2IP (see 'DNS Parameters' on page 478).

➢ **To configure the Internal SRV table:**

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

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

Figure 11-7: Internal SRV Table Page

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

Table 11-15: Internal SRV Table Parameter Description

Parameter	Description
Domain Name **[Srv2Ip_InternalDomain]**	Defines the host name to be translated. The valid value is a string of up to 31 characters.
Transport Type **[Srv2Ip_TransportType]**	Defines the transport type. ▪ **[0]** UDP (default) ▪ **[1]** TCP ▪ **[2]** TLS
DNS Name (1-3) **[Srv2Ip_Dns1/2/3]**	Defines the first, second or third DNS A-Record to which the host name is translated.
Priority (1-3) **[Srv2Ip_Priority1/2/3]**	Defines the priority of the target host. A lower value means that it is more preferred.
Weight (1-3) **[Srv2Ip_Weight1/2/3]**	Defines a relative weight for records with the same priority.
Port (1-3) **[Srv2Ip_Port1/2/3]**	Defines the TCP or UDP port on which the service is to be found.

11.7   Configuring NFS Settings

Network File System (NFS) enables the device to access a remote server's shared files and directories and to handle them as if they're located locally. The device can use NFS to load *cmp*, *ini*, and auxiliary files through the Automatic Update mechanism (see 'Automatic Update' on page 385).

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

➢   **To add remote NFS file systems:**

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

2.   Under the 'NFS Settings' group, click the **NFS Table** button; the NFS Table page appears.

3.   Click the **Add** button; the Add Record dialog box appears:

Figure 11-8: Add Record Dialog Box for NFS
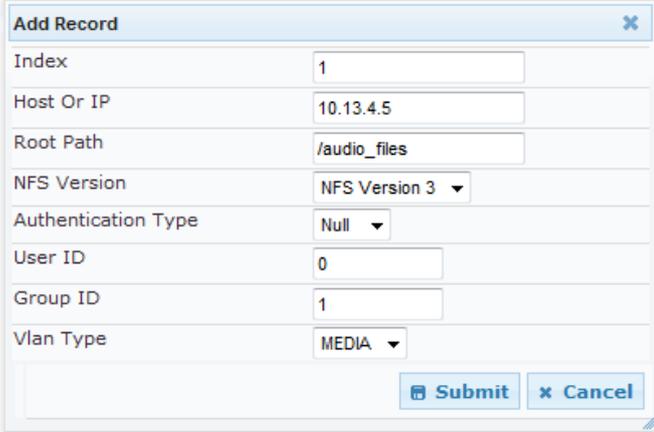
Add Record	
Index	1
Host Or IP	10.13.4.5
Root Path	/audio_files
NFS Version	NFS Version 3
Authentication Type	Null
User ID	0
Group ID	1
Vlan Type	MEDIA
	Submit Cancel

4.   Configure the NFS parameters according to the table below.

5.   Click the **Submit** button; the remote NFS file system is immediately applied, which can be verified by the appearance of the 'NFS mount was successful' message in the Syslog server.

6.   To save the changes to flash memory, see 'Saving Configuration' on page 362.

Notes:

•   To avoid terminating current calls, a row must not be deleted or modified while the device is currently accessing files on that remote NFS file system.

•   The combination of 'Host Or IP' and 'Root Path' must be unique for each row in the table. For example, the table must include only one row with a Host/IP of 192.168.1.1 and Root Path of /audio.

•   The NFS table can also be configured using the table ini file parameter NFSServers (see 'NFS Parameters' on page 477)

Table 11-16: NFS Settings Parameters

Parameter	Description
Index	The row index of the remote file system. The valid range is 1 to 16.
Host Or IP **[NFSServers_HostOrIP]**	The domain name or IP address of the NFS server. If a domain name is provided, a DNS server must be configured.
Root Path **[NFSServers_RootPath]**	Path to the root of the remote file system in the format: /[path]. For example, '/audio'.
NFS Version **[NFSServers_NfsVersion]**	NFS version used to access the remote file system. ▪ **[2]** NFS Version 2 ▪ **[3]** NFS Version 3 (default)
Authentication Type **[NFSServers_AuthType]**	Authentication method used for accessing the remote file system. ▪ **[0]** Null ▪ **[1]** Unix (default)
User ID **[NFSServers_UID]**	User ID used in authentication when using Unix. The valid range is 0 to 65537. The default is 0.
Group ID **[NFSServers_GID]**	Group ID used in authentication when using Unix. The valid range is 0 to 65537. The default is 1.
VLAN Type **[NFSServers_VlanType]**	The VLAN type for accessing the remote file system. ▪ **[0]** OAM ▪ **[1]** MEDIA (default) **Note:** This parameter applies only if VLANs are enabled or if Multiple IPs is configured (see 'Configuring IP Network Interfaces' on page 116).

11.8   Network Address Translation Support

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

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

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

11.8.1   Device Located behind NAT

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

a.   If configured, uses an external STUN server to assign a NAT address to all interfaces - see .Configuring STUN on page 140.

b.   If configured, uses the single Static NAT IP address for all interfaces - see 'Configuring a Static NAT IP Address for All Interfaces' on page 141.

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 Multiple Interface table.

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

Figure 11-9: Device behind NAT and NAT Issues
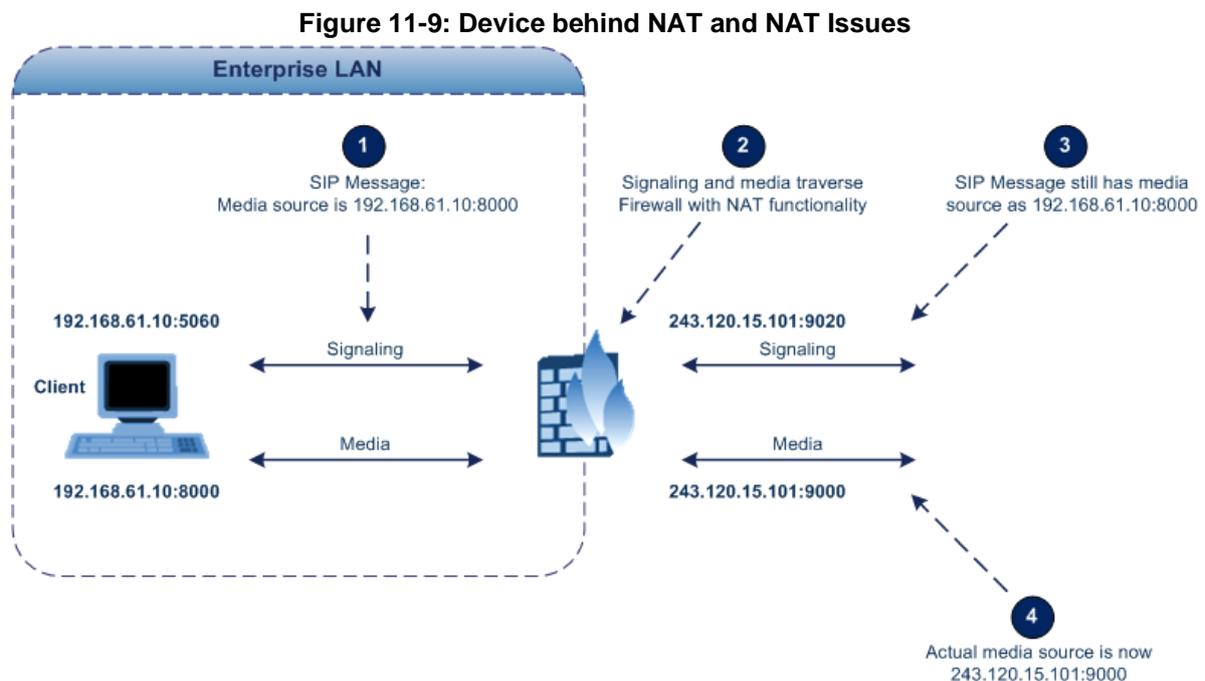
11.8.1.1 Configuring STUN

Simple Traversal of UDP through NATs (STUN), based on RFC 3489 is a client / server protocol that solves most of the NAT traversal problems. The STUN server operates in the public Internet and the STUN clients are embedded in end-devices located behind NAT. STUN is used for signaling and the media streams. STUN works with many existing NAT types and does not require any special behavior.

STUN enables the device to discover the presence (and types) of NATs and firewalls located between it and the public Internet. It provides the device with the capability to determine the public IP address and port allocated to it by the NAT. This information is later embedded in outgoing SIP / SDP messages and enables remote SIP user agents to reach the device. It also discovers the binding lifetime of the NAT - the refresh rate necessary to keep NAT 'pinholes' open.

On startup, the device sends a STUN Binding Request. The information received in the STUN Binding Response (IP address:port) is used for SIP signaling. This information is updated every user-defined period (NATBindingDefaultTimeout).

At the beginning of each call and if STUN is required (i.e., not an internal NAT call), the media ports of the call are mapped. The call is delayed until the STUN Binding Response (that includes a global IP:port) for each media (RTP, RTCP and T.38) is received.

Notes:

- STUN is applicable only to UDP connections (not TCP and TLS).
- STUN can't be used when the device is located behind a symmetric NAT.
- Use either the STUN server IP address (STUNServerPrimaryIP) or domain name (STUNServerDomainName) method, with priority to the first one.

➢ **To enable STUN:**

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

   Figure 11-10: STUN Parameters in Application Settings Page

2. From the 'Enable STUN' (EnableSTUN) drop-down list, select **Enable** to enable the STUN feature.

3. Configure the STUN server address using one of the following methods:

   - Define the IP address of the primary and secondary (optional) STUN servers, using the 'STUN Server Primary IP' field (STUNServerPrimaryIP) and 'STUN Server Secondary IP' field. If the primary STUN server is unavailable, the device attempts to communicate with the second server.

   - Define the domain name of the STUN server using the *ini* file parameter, STUNServerDomainName. The STUN client retrieves all STUN servers with an SRV query to resolve this domain name to an IP address and port, sorts the server list, and uses the servers according to the sorted list.

4. Configure the default NAT binding lifetime (in secondsUse) using the *ini* file parameter, NATBindingDefaultTimeout. STUN refreshes the binding information after this time expires.

11.8.1.2 Configuring a Static NAT IP Address for All Interfaces

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

> ⚠️ **Note:** The NAT IP address can also be configured using the ini file parameter, StaticNATIP.

➢ **To configure a single static NAT IP address for all interfaces:**

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

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

▼ SIP General	
⚡ NAT IP Address	0.0.0.0

2. In the 'NAT IP Address' field, enter the NAT IP address in dotted-decimal notation.

3. Click **Submit**.

4. Save the setting to the device's flash memory with a device reset (see 'Saving Configuration' on page 362).

11.8.2 Remote UA behind NAT

If the remote User Agent with which the device needs to communicate with is located behind NAT, the device can resolve the problem of activating the RTP/RTCP/T.38 streams to an invalid IP address / UDP port.

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

- First Incoming Packet Mechanism - see 'First Incoming Packet Mechanism' on page 142

- RTP No-Op packets according to the avt-rtp-noop draft - see 'No-Op Packets' on page 142

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

Figure 11-12: Remote UA behind NAT
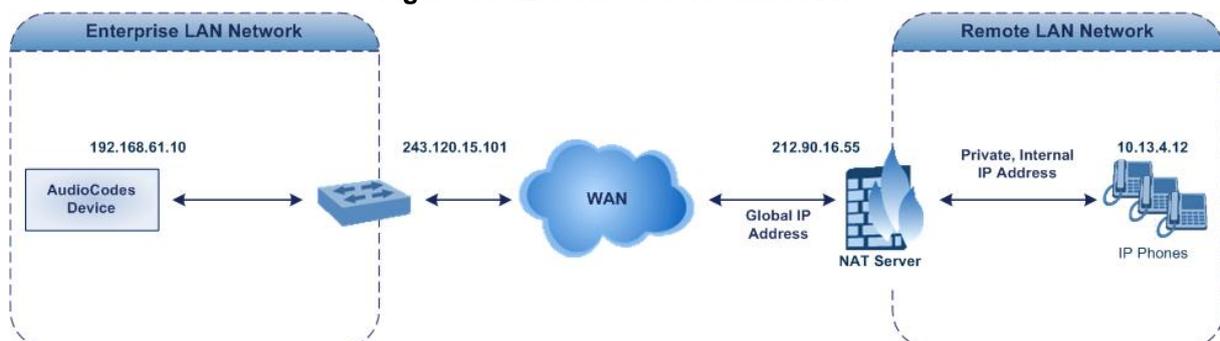
11.8.2.1 First Incoming Packet Mechanism

If the remote device resides behind a NAT device, it's possible that the device can activate the RTP/RTCP/T.38 streams to an invalid IP address / UDP port. To avoid such cases, the device automatically compares the source address of the first received incoming RTP/RTCP/T.38 stream with the IP address and UDP port of the remote device when the session was initially opened. If the two are not identical, then the destination IP address of the outgoing RTP packets is set to the source IP address of the first incoming packet. The RTP, RTCP and T.38 can thus have independent destination IP addresses and UDP ports.

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

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

2. Set the 'NAT Traversal' parameter to **Enable**.

3. Click **Submit**.

The EnableIpAddrTranslation and EnableUdpPortTranslation parameters allow you to specify the type of compare operation that occurs on the first incoming packet. To compare only the IP address, set EnableIpAddrTranslation to 1, and EnableUdpPortTranslation to 0. In this case, if the first incoming packet arrives with only a difference in the UDP port, the sending addresses won't change. If both the IP address and UDP port need to be compared, then both parameters need to be set to 1.

11.8.2.2 No-Op Packets

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

You can control the activation of No-Op packets by using the *ini* file parameter NoOpEnable. If No-Op packet transmission is activated, you can control the time interval in which No-Op packets are sent in the case of silence (i.e., no RTP or T.38 traffic). This is done using the *ini* file parameter NoOpInterval. For a description of the RTP No-Op *ini* file parameters, see 'Networking Parameters' on page 471.

■ **RTP No-Op:** The RTP No-Op support complies with IETF Internet-Draft draft-wing-avt-rtp-noop-03 ("A No-Op Payload Format for RTP"). This IETF document defines a No-Op payload format for RTP. The draft defines the RTP payload type as dynamic. You can control the payload type with which the No-Op packets are sent. This is performed using the RTPNoOpPayloadType *ini* parameter (see 'Networking Parameters' on page 471). The default payload type is 120.

■ **T.38 No-Op:** T.38 No-Op packets are sent only while a T.38 session is activated. Sent packets are a duplication of the previously sent frame (including duplication of the sequence number).

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

11.9   Robust Receipt of Media Streams

The "robust-media" mechanism is an AudioCodes proprietary mechanism to filter out unwanted media (i.e., RTP, RTCP, and T.38) streams that are sent to the same port number on the device. In practice, the media RTP/RTCP ports may receive additional multiple unwanted media streams as result of traces of previous calls, call control errors, or deliberate attacks. When more than one media stream reaches the device on the same port number, the "robust-media" mechanism detects the valid media stream and ignores the rest.

11.10   Multiple Routers Support

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

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

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

11.11   IP Multicasting

The device supports IP Multicasting level 1, according to RFC 2236 (i.e., IGMP version 2) for RTP channels. The device is capable of transmitting and receiving multicast packets.

This page is intentionally left blank.

12 Security

This section describes the VoIP security-related configuration.

12.1 Configuring Firewall Settings

The device provides an internal firewall that enables you to configure network traffic filtering rules (*access list*). You can add up to 50 firewall rules. The access list offers the following firewall possibilities:

■ Block traffic from known malicious sources

■ Allow traffic only from known "friendly" sources, and block all other traffic

■ Mix allowed and blocked network sources

■ Limit traffic to a user-defined rate (blocking the excess)

■ Limit traffic to specific protocols, and specific port ranges on the device

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

Notes:

- This firewall applies to a very low-level network layer and overrides your other security-related configuration. Thus, if you have configured higher-level security features (e.g., on the Application level), you must also configure firewall rules to permit this necessary traffic. For example, if you have configured IP addresses to access the Web and Telnet interfaces in the Web Access List (see 'Configuring Web and Telnet Access List' on page 75), you must configure a firewall rule that permits traffic from these IP addresses.

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

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

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

- You can also configure the firewall settings using the table ini file parameter, AccessList (see 'Security Parameters' on page 499).

> ➢ **To add firewall rules:**

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

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

Figure 12-1: Firewall Settings Page - Add Record

3. Configure the firewall parameters, as required. For a description of the parameters, see the table below.

4. Click **Submit** to add the new firewall rule to the table.

5. Reset the device to activate the rules.

The table below provides an example of configured firewall rules:

Table 12-1: Firewall Rule Examples

Parameter	Value per Rule				
	1	**2**	**3**	**4**	**5**
Source IP	12.194.231.76	12.194.230.7	0.0.0.0	192.0.0.0	0.0.0.0
Prefix Length	16	16	0	8	0
Start Port and End Port	0-65535	0-65535	0-65535	0-65535	0-65535
Protocol	Any	Any	icmp	Any	Any
Use Specific Interface	Enable	Enable	Disable	Enable	Disable
Interface Name	WAN	WAN	None	Voice-Lan	None
Byte Rate	0	0	40000	40000	0

Parameter	Value per Rule				
	1	**2**	**3**	**4**	**5**
Burst Bytes	0	0	50000	50000	0
Action Upon Match	Allow	Allow	Allow	Allow	Block

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

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

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

■ **Rule 4:** Allows traffic from the LAN voice interface and limits bandwidth.

■ **Rule 5:** Blocks all other traffic.

Table 12-2: Internal Firewall Parameters

Parameter	Description
Source IP **[AccessList_Source_IP]**	Defines the IP address (or DNS name) or a specific host name of the source network (i.e., from where the incoming packet is received).
Source Port **[AccessList_Source_Port]**	Defines the source UDP/TCP ports (of the remote host) from where packets are sent to the device. The valid range is 0 to 65535. **Note:** When set to 0, this field is ignored and any source port matches the rule.
Prefix Length **[AccessList_PrefixLen]**	(**Mandatory**) Defines the IP network mask - 32 for a single host or the appropriate value for the source IP addresses. ▪ A value of 8 corresponds to IPv4 subnet class A (network mask of 255.0.0.0). ▪ A value of 16 corresponds to IPv4 subnet class B (network mask of 255.255.0.0). ▪ A value of 24 corresponds to IPv4 subnet class C (network mask of 255.255.255.0). The IP address of the sender of the incoming packet is trimmed in accordance with the prefix length (in bits) and then compared to the parameter 'Source IP'. The default is 0 (i.e., applies to all packets). You **must** change this value to any of the above options. **Note:** A value of 0 applies to **all** packets, regardless of the defined IP address. Therefore, you must set this parameter to a value other than 0.
Start Port **[AccessList_Start_Port]**	Defines the destination UDP/TCP start port (on this device) to where packets are sent. The valid range is 0 to 65535. **Note:** When the protocol type isn't TCP or UDP, the entire range must be provided.

Parameter	Description
End Port **[AccessList_End_Port]**	Defines the destination UDP/TCP end port (on this device) to where packets are sent. The valid range is 0 to 65535. **Note:** When the protocol type isn't TCP or UDP, the entire range must be provided.
Protocol **[AccessList_Protocol]**	Defines the protocol type (e.g., UDP, TCP, ICMP, ESP or 'Any') or the IANA protocol number in the range of 0 (Any) to 255. **Note:** This field also accepts the abbreviated strings 'SIP' and 'HTTP'. Specifying these strings implies selection of the TCP or UDP protocols, and the appropriate port numbers as defined on the device.
Use Specific Interface **[AccessList_Use_Specific_Interface]**	Determines whether you want to apply the rule to a specific network interface defined in the Multiple Interface table (i.e., packets received from that defined in the Source IP field and received on this network interface): ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ If enabled, then in the 'Interface Name' field (described below), select the interface to which the rule is applied. ▪ If disabled, then the rule applies to all interfaces.
Interface Name **[AccessList_Interface_ID]**	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 Multiple Interface table in 'Configuring IP Network Interfaces' on page 116.
Packet Size **[AccessList_Packet_Size]**	Defines the maximum allowed packet size. The valid range is 0 to 65535. **Note:** When filtering fragmented IP packets, this field relates to the overall (re-assembled) packet size, and not to the size of each fragment.
Byte Rate **[AccessList_Byte_Rate]**	Defines the expected traffic rate (bytes per second), i.e., the allowed bandwidth for the specified protocol. In addition to this field, the 'Burst Bytes' field provides additional allowance such that momentary bursts of data may utilize more than the defined byte rate, without being interrupted. For example, if 'Byte Rate' is set to 40000 and 'Burst Bytes' to 50000, then this implies the following: the allowed bandwidth is 40000 bytes/sec with extra allowance of 50000 bytes; if, for example, the actual traffic rate is 45000 bytes/sec, then this allowance would be consumed within 10 seconds, after which all traffic exceeding the allocated 40000 bytes/sec is dropped. If the actual traffic rate then slowed to 30000 bytes/sec, then the allowance would be replenished within 5 seconds.
Burst Bytes **[AccessList_Byte_Burst]**	Defines the tolerance of traffic rate limit (number of bytes). The default is 0.
Action Upon Match **[AccessList_Allow_Type]**	Defines the firewall action to be performed upon rule match. ▪ "Allow" = (Default) Permits these packets ▪ "Block" = Rejects these packets

Parameter	Description
Match Count **[AccessList_MatchCount]**	(Read-only) Displays the number of packets accepted or rejected by the rule.

12.2   Configuring 802.1x Settings

The 802.1x Settings page is used to configure IEEE 802.1X Ethernet security. The device can function as an IEEE 802.1X supplicant. IEEE 802.1X is a standard for port-level security on secure Ethernet switches; when a device is connected to a secure port, no traffic is allowed until the identity of the device is authenticated.

A typical 802.1X deployment consists of an Authenticator (secure LAN switch), an Access Server (e.g., RADIUS), and one or more supplicants. The Authenticator blocks all traffic on the secure port by default and communicates with the supplicant via EAP-over-LAN frames. The supplicant provides credentials which are transmitted to the Access Server. If the Access Server determines that the credentials are valid, it instructs the Authenticator to authorize traffic on the secure port.

The device supports the following Extensible Authentication Protocol (EAP) variants:

- MD5-Challenge (EAP-MD5): Authentication is done with a user-defined 802.1X username and password.

- Protected EAP (PEAPv0 with EAP-MSCHAPv2): Authentication is done with a user-defined 802.1X username and password, however, the protocol is MSCHAPv2 over an encrypted TLS tunnel.

- EAP-TLS: The device's certificate is used to establish a mutually authenticated TLS session with the Access Server. This requires prior configuration of the server certificate and root CA. The user-defined 802.1X username is used to identify the device, however, the 802.1X password is ignored.

➢ **To configure the 802.1x parameters:**

1. Open the 802.1x Settings page (**Configuration** tab > **VoIP** menu > **Security** submenu > **802.1x Settings**).

Figure 12-2: 8021x Settings Page

2. Define the 802.1x parameters (see 802.1X Parameters on page 510 for description).
3. Define the supported TLS version, using the TLSVersion ini file parameter.
4. Define the cipher string for HTTPS, using the HTTPSCipherString ini file parameter. Configure the parameters as required, and then click **Submit**.

12.3   Configuring General Security Settings

The General Security Settings page is used to configure various security features. For a description of the parameters appearing on this page, refer 'Configuration Parameters Reference' on page 471.

➢ **To configure the general security parameters:**

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

IPSec Setting	
Enable IP Security	Disable
IKE Certificate Ext Validate	Disable

TLS Settings	
TLS Version	SSL 2.0-3.0 and TLS 1.0
Strict Certificate Extension Validation	Disable
FIPS140 Mode	Disable
Client Cipher String	ALL:!ADH

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

OCSP Settings	
Enable OCSP Server	Disable
Primary Server IP	0.0.0.0
Secondary Server IP	0.0.0.0
Server Port	2560
Default Response When Server Unreachable	Reject

2. Configure the parameters as required.
3. Click Subm**it to** apply your changes.
4. To save the changes to flash memory, refer to 'Saving Configuration' on page 362.

12.4   IPSec and Internet Key Exchange

IP security (IPSec) and Internet Key Exchange (IKE) protocols are part of the IETF standards for establishing a secured IP connection between two applications (also referred to as peers). Providing security services at the IP layer, IPSec and IKE are transparent to IP applications. IPSec and IKE are used together to provide security for control and management (e.g., SNMP and Web) protocols, but not for media (i.e., RTP, RTCP and T.38).

IKE is used to obtain the Security Associations (SA) between peers (the device and the application it's trying to contact). The SA contains the encryption keys and profile used by IPSec to encrypt the IP stream. IKE negotiation comprises the following two phases:

- **Main Mode** (creates a secured channel for the Quick mode by obtaining a "master" encryption key, without any prior keys, and authenticates the peers to each other):

  - SA negotiation: The peers negotiate their capabilities using up to four proposals. Each proposal includes the Encryption method, Authentication algorithm, and the Diffie-Hellman (DH) group. The master key's lifetime is also negotiated.

  - Key exchange (DH): The DH protocol creates the master key. DH requires both peers to agree on certain mathematical parameters, known as the "group".

  - Authentication: The two peers authenticate one another using a pre-shared key configured in the IP Security Associations Table or by using certificate-based authentication.

- **Quick Mode** (creates the encrypted IPSec tunnel once initial security is set up):

  - SA negotiation: An IPSec SA is created by negotiating encryption and authentication capabilities using the same proposal mechanism as in Main mode.

  - Key exchange: A symmetrical key is created for encrypting IPSec traffic; the peers communicate with each other in encrypted form, secured by the previously negotiated "master" key.

IKE specifications summary:

- Authentication methods: pre-shared key or certificate-based authentication

- Main mode supported for IKE Phase 1

- DH group 1 or group 2

- Encryption algorithms: Data Encryption Standard (DES), Advanced Encryption Standard (AES), and 3DES

- Hash algorithms: SHA1 and MD5

IPSec is responsible for securing the IP traffic. This is accomplished by using the Encapsulation Security Payload (ESP) protocol to encrypt (and decrypt) the IP payload. This is configured in the IPSec Security Association table, which defines the IP peers to which IPSec security is applied.

IPSec specifications summary:

- Transport and Tunneling Mode

- Encapsulation Security Payload (ESP) only

- Encryption algorithms: AES, DES, and 3DES

- Hash types: SHA1 and MD5

12.4.1 Enabling IPSec

To enable IKE and IPSec processing, you must enable the IPSec feature, as described below.

➢ **To enable IPSec:**

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

Figure 12-3: Enabling IPSec

2. Set the 'Enable IP Security' parameter to **Enable**.

3. Click **Submit**, and then reset the device with a flash burn.

12.4.2 Configuring IP Security Proposal Table

The IP Security Proposal Table page is used to configure Internet Key Exchange (IKE) with up to four proposal settings. Each proposal defines an encryption algorithm, an authentication algorithm, and a Diffie-Hellman group identifier. The same set of proposals applies to both Main mode and Quick mode.

> **Note:** You can also configure the IP Security Proposals table using the table ini file parameter IPsecProposalTable (see 'Security Parameters' on page 499).

➢ **To configure IP Security Proposals:**

1. Open the IP Security Proposal Table page (**Configuration** tab > **VoIP** menu > **Security** submenu > **IPSec Proposal Table**).

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

Figure 12-4: IP Security Proposals Table - Add Record Dialog Box

3. Configure the parameters, as required. For a description of the parameters, see the table below.

4. Click **Submit**.

5. To save the changes to flash memory, see 'Saving Configuration' on page 362.

Table 12-3: IP Security Proposals Table Configuration Parameters

Parameter Name	Description
Encryption Algorithm **[IPsecProposalTable_EncryptionAlgorithm]**	Defines the encryption (privacy) algorithm. ▪ **[0]** NONE ▪ **[1]** DES CBC ▪ **[2]** 3DES CBC ▪ **[3]** AES (default)
Authentication Algorithm **[IPsecProposalTable_AuthenticationAlgorithm]**	Defines the message authentication (integrity) algorithm. ▪ **[0]** NONE ▪ **[2]** HMAC SHA1 96 ▪ **[4]** HMAC MD5 96 (default)
Diffie Hellman Group **[IPsecProposalTable_DHGroup]**	Defines the length of the key created by the DH protocol for up to four proposals. For the *ini* file parameter, *X* denotes the proposal number (0 to 3). ▪ **[0]** Group 1 (768 Bits) = DH-786-Bit ▪ **[1]** Group 2 (1024 Bits) (default) = DH-1024-Bit

If no proposals are defined, the default settings (shown in the following table) are applied.

Table 12-4: Default IPSec/IKE Proposals

Proposal	Encryption	Authentication	DH Group
Proposal 0	3DES	SHA1	Group 2 (1024 bit)
Proposal 1	3DES	MD5	Group 2 (1024 bit)
Proposal 2	3DES	SHA1	Group 1 (786 bit)
Proposal 3	3DES	MD5	Group 1 (786 bit)

12.4.3   Configuring IP Security Associations Table

The IP Security Associations Table page allows you to configure up to 20 peers (hosts or networks) for IP security (IPSec)/IKE. Each of the entries in this table controls both Main and Quick mode configuration for a single peer. Each row in the table refers to a different IP destination. IPSec can be applied to all traffic to and from a specific IP address. Alternatively, IPSec can be applied to a specific flow, specified by port (source or destination) and protocol type.

The destination IP address (and optionally, destination port, source port and protocol type) of each outgoing packet is compared to each entry in the table. If a match is found, the device checks if an SA already exists for this entry. If no SA exists, the IKE protocol is invoked and an IPSec SA is established and the packet is encrypted and transmitted. If a match is not found, the packet is transmitted without encryption.

This table can also be used to enable Dead Peer Detection (RFC 3706), whereby the device queries the liveliness of its IKE peer at regular intervals or on-demand. When two peers communicate with IKE and IPSec, the situation may arise in which connectivity between the two goes down unexpectedly. In such cases, there is often no way for IKE and IPSec to

identify the loss of peer connectivity. As such, the Security Associations (SA) remain active until their lifetimes naturally expire, resulting in a "black hole" situation where both peers discard all incoming network traffic. This situation may be resolved by performing periodic message exchanges between the peers. When no reply is received, the sender assumes SA's are no longer valid on the remote peer and attempts to renegotiate.

> **Notes:**
>
> - Incoming packets whose parameters match one of the entries in the IP Security Associations table but is received without encryption, is rejected.
> - If you change the device's IP address on-the-fly, you must then reset the device for IPSec to function properly.
> - The proposal list must be contiguous.
> - For security, once the IKE pre-shared key is configured, it is not displayed in any of the device's management tools.
> - You can also configure the IP Security Associations table using the table ini file parameter IPsecSATable (see 'Security Parameters' on page 499).

> **To configure the IPSec Association table:**

1. Open the IP Security Associations Table page (**Configuration** tab > **VoIP** menu > **Security** submenu > **IPSec Association Table**).
2. Click the **Add** button; the following dialog box appears:

Figure 12-5: IP Security Associations Table Page - Add Record Dialog Box

3. Configure the parameters, as required. In the above figure, a single IPSec/IKE peer (10.3.2.73) is configured. Pre-shared key authentication is selected with the pre-shared key set to 123456789. In addition, a lifetime of 28800 seconds is set for IKE and a lifetime of 3600 seconds is set for IPSec. For a description of the parameters, see the table below.
4. Click **Submit**.

5. To save the changes to flash memory, see 'Saving Configuration' on page 362.

Table 12-5: IP Security Associations Table Configuration Parameters

Parameter Name	Description
Operational Mode **[IPsecSATable_IPsecMode]**	Defines the IPSec mode of operation. ▪ **[0]** Transport (default) ▪ **[1]** Tunnel
Remote Endpoint Addr **[IPsecSATable_RemoteEndpointAddressOrName]**	Defines the IP address or DNS host name of the peer. **Note:** This parameter is applicable only if the Operational Mode is set to Transport.
Authentication Method **[IPsecSATable_AuthenticationMethod]**	Defines the method for peer authentication during IKE main mode. ▪ **[0]** Pre-shared Key (default) ▪ **[1]** RSA Signature = in X.509 certificate **Note:** For RSA-based authentication, both peers must be provisioned with certificates signed by a common CA. For more information on certificates, see 'Replacing the Device's Certificate' on page 103.
Shared Key **[IPsecSATable_SharedKey]**	Defines the pre-shared key (in textual format). Both peers must use the same pre-shared key for the authentication process to succeed. **Notes:** ▪ This parameter is applicable only if the Authentication Method parameter is set to pre-shared key. ▪ The pre-shared key forms the basis of IPSec security and therefore, it should be handled with care (the same as sensitive passwords). It is not recommended to use the same pre-shared key for several connections. ▪ Since the *ini* file is plain text, loading it to the device over a secure network connection is recommended. Use a secure transport such as HTTPS, or a direct crossed-cable connection from a management PC. ▪ After it is configured, the value of the pre-shared key cannot be retrieved.
Source Port **[IPsecSATable_SourcePort]**	Defines the source port to which this configuration applies. The default is 0 (i.e., any port).
Destination Port **[IPsecSATable_DestPort]**	Defines the destination port to which this configuration applies. The default is 0 (i.e., any port).
Protocol **[IPsecSATable_Protocol]**	Defines the protocol type to which this configuration applies. Standard IP protocol numbers, as defined by the Internet Assigned Numbers Authority (IANA) should be used, for example: ▪ 0 = Any protocol (default) ▪ 17 = UDP ▪ 6 = TCP

Parameter Name	Description
IKE SA Lifetime **[IPsecSATable_Phase1SaLifetimeI nSec]**	Defines the duration (in seconds) for which the negotiated IKE SA (Main mode) is valid. After this time expires, the SA is re-negotiated. The default is 0 (i.e., unlimited). **Note:** Main mode negotiation is a processor-intensive operation; for best performance, do not set this parameter to less than 28,800 (i.e., eight hours).
IPSec SA Lifetime (sec) **[IPsecSATable_Phase2SaLifetimeI nSec]**	Defines the duration (in seconds) for which the negotiated IPSec SA (Quick mode) is valid. After this time expires, the SA is re-negotiated. The default is 0 (i.e., unlimited). **Note:** For best performance, a value of 3,600 (i.e., one hour) or more is recommended.
IPSec SA Lifetime (Kbs) **[IPsecSATable_Phase2SaLifetimeI nKB]**	Defines the maximum volume of traffic (in kilobytes) for which the negotiated IPSec SA (Quick mode) is valid. After this specified volume is reached, the SA is re-negotiated. The default is 0 (i.e., the value is ignored).
Dead Peer Detection Mode **[IPsecSATable_DPDmode]**	Defines dead peer detection (DPD), according to RFC 3706. • **[0]** DPD Disabled (default) • **[1]** DPD Periodic = DPD is enabled with message exchanges at regular intervals • **[2]** DPD on demand = DPD is enabled with on-demand checks - message exchanges as needed (i.e., before sending data to the peer). If the liveliness of the peer is questionable, the device sends a DPD message to query the status of the peer. If the device has no traffic to send, it never sends a DPD message.
Remote Tunnel Addr **[IPsecSATable_RemoteTunnelAddr ess]**	Defines the IP address of the peer router. **Note:** This parameter is applicable only if the Operational Mode is set to Tunnel.
Remote Subnet Addr **[IPsecSATable_RemoteSubnetIPA ddress]**	Defines the IP address of the remote subnet. Together with the Prefix Length parameter (below), this parameter defines the network with which the IPSec tunnel allows communication. **Note:** This parameter is applicable only if the Operational Mode is set to Tunnel.
Remote Prefix Length **[IPsecSATable_RemoteSubnetPrefi xLength]**	Defines the prefix length of the Remote Subnet IP Address parameter (in bits). The prefix length defines the subnet class of the remote network. A prefix length of 16 corresponds to a Class B subnet (255.255.0.0); a prefix length of 24 corresponds to a Class C subnet (255.255.255.0). **Note:** This parameter is applicable only if the Operational Mode is set to Tunnel.
Interface Name **[IPsecSATable_InterfaceName]**	Assigns a network interface to this IPSec rule. The network interfaces are defined in the Multiple Interface table ('Interface Name' column) in 'Configuring IP Network Interfaces' on page 116

13 Media

This section describes the media-related configuration.

13.1 Configuring Voice Settings

The Voice Settings page configures various voice parameters such as voice volume and DTMF transport type. For a detailed description of these parameters, see 'Configuration Parameters Reference' on page 471.

➢ **To configure the voice parameters:**

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

Voice Volume (-32 to 31 dB)	0
Input Gain (-32 to 31 dB)	0
Silence Suppression	Disable
DTMF Transport Type	RFC2833 Relay DTMF
DTMF Volume (-31 to 0 dB)	-11
NTE Max Duration	-1
Enable Answer Detector	Disable
Answer Detector Activity Delay	0
Answer Detector Silence Time	10
Answer Detector Redirection	0
Answer Detector Sensitivity	0
⚡ DTMF Generation Twist	0
Echo Canceller	Enable

2. Configure the Voice parameters as required.
3. Click Subm**it to** apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 362.

13.1.1 Configuring Voice Gain (Volume) Control

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

The procedure below describes how to configure gain control using the Web interface:

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

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

Figure 13-1: Voice Volume Parameters in Voice Settings Page

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

2. Configure the following parameters:

- 'Voice Volume' (*VoiceVolume*) - Defines the voice gain control (in decibels) for IP-to-Tel

- 'Input Gain' (*InputGain*) - Defines the PCM input gain control (in decibels) for Tel-to-IP

3. Click **Submit** to apply your settings.

13.1.2 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 procedure below describes how to configure echo cancellation using the Web interface:

➢ **To configure echo cancellation using the Web interface:**

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

Figure 13-2: Enabling Echo Cancellation in Voice Settings Page

Echo Canceller	Enable ▼

2. Set the 'Echo Canceller' field (*EnableEchoCanceller*) to **Enable**.

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

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

---

13.2 Fax and Modem Capabilities

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

> **Notes:**
>
> • Unless otherwise specified, the configuration parameters mentioned in this section are available on this page.
>
> • Some SIP parameters override these fax and modem parameters. For example, the IsFaxUsed parameter and V.152 parameters in Section 'V.152 Support' on page 169).
>
> • For a detailed description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 471.

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

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

Figure 13-3: Fax/Modem/CID Settings Page

2. Configure the parameters, as required.
3. Click **Submit** to apply your changes.

13.2.1   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 161)

■   G.711 Transport: switching to G.711 when fax/modem is detected (see 'G.711 Fax / Modem Transport Mode' on page 162)

■   Fax fallback to G.711 if T.38 is not supported (see 'Fax Fallback' on page 163)

■   Fax and modem bypass: a proprietary method that uses a high bit rate coder (see 'Fax/Modem Bypass Mode' on page 163)

■   NSE Cisco's Pass-through bypass mode for fax and modem (see 'Fax / Modem NSE Mode' on page 165)

■   Transparent with events: passing the fax / modem signal in the current voice coder with adaptations (see 'Fax / Modem Transparent with Events Mode' on page 166)

■   Transparent: passing the fax / modem signal in the current voice coder (see 'Fax / Modem Transparent Mode' on page 166)

■   RFC 2833 ANS Report upon Fax/Modem Detection (see 'RFC 2833 ANS Report upon Fax/Modem Detection' on page 167)

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

13.2.1.1 T.38 Fax Relay Mode

In Fax Relay mode, fax signals are transferred using the T.38 protocol. T.38 is an 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 161)

■   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 162)

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

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

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

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

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

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

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

2. In the Fax/Modem/CID Settings page, configure the following optional parameters:

   - 'Fax Relay Redundancy Depth' (FaxRelayRedundancyDepth)
   - 'Fax Relay Enhanced Redundancy Depth' (FaxRelayEnhancedRedundancyDepth)
   - 'Fax Relay ECM Enable' (FaxRelayECMEnable)
   - 'Fax Relay Max Rate' (FaxRelayMaxRate)

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

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

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

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

1. In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **No Fax** (IsFaxUsed = 0).

2. In the Fax/Modem/CID Settings page, set the 'Fax Transport Mode' parameter to **RelayEnable** (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)

13.2.1.2 G.711 Fax / Modem Transport Mode

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

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

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

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

■ **For G.711 A-law:**

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

■ **For G.711 µ-law:**

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

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

■ 'Fax Transport Mode' (FaxTransportMode)

■ 'Vxx ModemTransportType' (VxxModemTransportType)

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

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

13.2.1.3 Fax Fallback

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

■ Echo Canceller = on

■ Silence Compression = off

■ Echo Canceller Non-Linear Processor Mode = off

■ Dynamic Jitter Buffer Minimum Delay = 40

■ Dynamic Jitter Buffer Optimization Factor = 13

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

■ **For G.711A-law:**

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

■ **For G.711 µ-law:**

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

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

➢ **To configure fax fallback mode:**

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

13.2.1.4 Fax/Modem Bypass Mode

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

■ Disables silence suppression

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

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

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

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

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

1. In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **No Fax** (IsFaxUsed = 0).

2. In the Fax/Modem/CID Settings page, do the following:
   a. Set the 'Fax Transport Mode' parameter to **Bypass** (FaxTransportMode = 2).
   b. Set the 'V.21 Modem Transport Type' parameter to **Enable Bypass** (V21ModemTransportType = 2).
   c. Set the 'V.22 Modem Transport Type' parameter to **Enable Bypass** (V22ModemTransportType = 2).
   d. Set the 'V.23 Modem Transport Type' parameter to **Enable Bypass** (V23ModemTransportType = 2).
   e. Set the 'V.32 Modem Transport Type' parameter to **Enable Bypass** (V32ModemTransportType = 2).
   f. Set the 'V.34 Modem Transport Type' parameter to **Enable Bypass** (V34ModemTransportType = 2).

3. Set the ini file parameter, BellModemTransportType to 2 (Bypass).

4. Configure the following optional parameters:
   - 'Fax/Modem Bypass Coder Type' (FaxModemBypassCoderType).
   - 'Fax Bypass Payload Type' (FaxBypassPayloadType) - in the RTP/RTCP Settings page (**Configuration** tab > **VoIP** menu > **Media** > **RTP/RTCP Settings**).
   - ModemBypassPayloadType (ini file).
   - FaxModemBypassBasicRTPPacketInterval (ini file).
   - FaxModemBypasDJBufMinDelay (ini file).

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

> **Tip:**   When the remote (non-AudioCodes) gateway uses the G.711 coder for voice and doesn't change the coder payload type for fax or modem transmission, it is recommended to use the Bypass mode with the following configuration:
>
> - EnableFaxModemInbandNetworkDetection = 1.
> - 'Fax/Modem Bypass Coder Type' = same coder used for voice.
> - 'Fax/Modem Bypass Packing Factor'(FaxModemBypassM) = same interval as voice.
> - ModemBypassPayloadType = 8 if voice coder is A-Law or 0 if voice coder is Mu-Law.

13.2.1.5 Fax / Modem NSE Mode

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

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

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

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

Where *100* is the NSE payload type.

The Cisco gateway must include the following definition:

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

➢ **To configure NSE mode:**

1. In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **No Fax** (IsFaxUsed = 0).

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

    a. Set the 'Fax Transport Mode' parameter to **Bypass** (FaxTransportMode = 2).
    b. Set the 'V.21 Modem Transport Type' parameter to **Enable Bypass** (V21ModemTransportType = 2).
    c. Set the 'V.22 Modem Transport Type' parameter to **Enable Bypass** (V22ModemTransportType = 2).
    d. Set the 'V.23 Modem Transport Type' parameter to **Enable Bypass** (V23ModemTransportType = 2).
    e. Set the 'V.32 Modem Transport Type' parameter to **Enable Bypass** (V32ModemTransportType = 2).
    f. Set the 'V.34 Modem Transport Type' parameter to **Enable Bypass** (V34ModemTransportType = 2).

3. Set the ini file parameter, BellModemTransportType to 2 (Bypass).

4. Set the ini file parameter, NSEMode parameter to 1 (enables NSE).

5. Set the ini file parameter, NSEPayloadType parameter to 100.

13.2.1.6 Fax / Modem Transparent with Events Mode

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

■ Echo Canceller = on (or off for modems)

■ Echo Canceller Non-Linear Processor Mode = off

■ Jitter buffering optimizations

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

1. In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **No Fax** (IsFaxUsed = 0).

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

   a. Set the 'Fax Transport Mode' parameter to **Events Only** (FaxTransportMode = 3).

   b. Set the 'V.21 Modem Transport Type' parameter to **Events Only** (V21ModemTransportType = 3).

   c. Set the 'V.22 Modem Transport Type' parameter to **Events Only** (V22ModemTransportType = 3).

   d. Set the 'V.23 Modem Transport Type' parameter to **Events Only** (V23ModemTransportType = 3).

   e. Set the 'V.32 Modem Transport Type' parameter to **Events Only** (V32ModemTransportType = 3).

   f. Set the 'V.34 Modem Transport Type' parameter to **Events Only** (V34ModemTransportType = 3).

3. Set the ini file parameter, BellModemTransportType to 3 (transparent with events).

13.2.1.7 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 211) to apply certain adaptations to the channel used for fax / modem. For example, to use the coder G.711, to set the jitter buffer optimization factor to 13, and to enable echo cancellation for fax and disable it for modem.

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

1. In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **No Fax** (IsFaxUsed = 0).

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

   a. Set the 'Fax Transport Mode' parameter to **Disable** (FaxTransportMode = 0).

   b. Set the 'V.21 Modem Transport Type' parameter to **Disable** (V21ModemTransportType = 0).

   c. Set the 'V.22 Modem Transport Type' parameter to **Disable** (V22ModemTransportType = 0).

   d. Set the 'V.23 Modem Transport Type' parameter to **Disable** (V23ModemTransportType = 0).

   e. Set the 'V.32 Modem Transport Type' parameter to **Disable** (V32ModemTransportType = 0).

   f. Set the 'V.34 Modem Transport Type' parameter to **Disable** (V34ModemTransportType = 0).

3. Set the ini file parameter, BellModemTransportType to 0 (transparent mode).

4. Configure the following optional parameters:

   a. Coders table - (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **Coders**).

   b. 'Dynamic Jitter Buffer Optimization Factor' (DJBufOptFactor) - RTP/RTCP Settings page (**Configuration** tab > **VoIP** menu > **Media** > **RTP/RTCP Settings**).

   c. 'Echo Canceller' (EnableEchoCanceller) - Voice Settings page.

> **Note:** This mode can be used for fax, but is not recommended for modem transmission. Instead, use the Bypass (see 'Fax/Modem Bypass Mode' on page 163) or Transparent with Events modes (see 'Fax / Modem Transparent with Events Mode' on page 166) for modem.

13.2.1.8 RFC 2833 ANS Report upon Fax/Modem Detection

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

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

1. In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **No Fax** or **Fax Fallback** (IsFaxUsed = 0 or 3).

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

   a. Set the 'Fax Transport Mode' parameter to **Bypass** (FaxTransportMode = 2).

   b. Set the 'V.xx Modem Transport Type' parameters to **Enable Bypass** (VxxModemTransportType = 2).

3. Set the ini file parameter, FaxModemNTEMode to 1 (enables this feature).

13.2.2 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 168)
- T38 Version 0 relay mode, i.e., fallback to T.38 (see 'Relay Mode for T.30 and V.34 Faxes' on page 168)

> **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**.

13.2.2.1 Bypass Mechanism for V.34 Fax Transmission

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

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

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

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

1. In the Fax/Modem/CID Settings page, do the following:
- **a.** Set the 'Fax Transport Mode' parameter to **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).

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

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

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

1. In the Fax/Modem/CID Settings page, do the following:
- **a.** Set the 'Fax Transport Mode' parameter to **Relay** (FaxTransportMode = 1).
- **b.** Set the 'V.22 Modem Transport Type' parameter to **Disable** (V22ModemTransportType = 0).
- **c.** Set the 'V.23 Modem Transport Type' parameter to **Disable** (V23ModemTransportType = 0).
- **d.** Set the 'V.32 Modem Transport Type' parameter to **Disable** (V32ModemTransportType = 0).
- **e.** Set the 'V.34 Modem Transport Type' parameter to **Disable** (V34ModemTransportType = 0).

13.2.3 V.152 Support

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

For V.152 capability, the device supports T.38 as well as VBD codecs (i.e., G.711 A-law and G.711 μ-law). The selection of capabilities is performed using the coders table (see 'Configuring Coders' on page 211).

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

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

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

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

> **Note:** You can also configure the device to handle G.711 coders received in INVITE SDP offers as VBD coders, using the HandleG711asVBD parameter. For example, if the device is configured with G.729 and G.711 VBD coders and it receives an INVITE with an SDP offer containing G.729 and "regular" G.711 coders, it sends an SDP answer containing G.729 and G.711 VBD coders, allowing subsequent bypass (passthrough) sessions if fax / modem signals are detected during the call.

13.2.4 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.

13.3   Configuring RTP/RTCP Settings

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

13.3.1   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 procedure below describes how to configure the jitter buffer using the Web interface.

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

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

Figure 13-4: Jitter Buffer Parameters in the RTP/RTCP Settings Page

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

2. Set the 'Dynamic Jitter Buffer Minimum Delay' parameter (DJBufMinDelay) to the minimum delay (in msec) for the Dynamic Jitter Buffer.

3. Set the 'Dynamic Jitter Buffer Optimization Factor' parameter (DJBufOptFactor) to the Dynamic Jitter Buffer frame error/delay optimization factor.

4. Click **Submit** to apply your settings.

13.3.2 Comfort Noise Generation

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

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

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

➢ **To configure CNG using the Web interface:**

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

Figure 13-5: Comfort Noise Parameter in RTP/RTCP Settings Page

Comfort Noise Generation Negotiation	Enable

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

3. Click **Submit** to apply your changes.

13.3.3   Dual-Tone Multi-Frequency Signaling

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

13.3.3.1 Configuring DTMF Transport Types

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

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

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

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

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

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

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

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

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

■ **Using NOTIFY messages according to IETF Internet-Draft draft-mahy-sipping-signaled-digits-01**: DTMF digits are sent to the remote side using NOTIFY messages. To enable this mode, define the following:

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

   b. Set the '1st Tx DTMF Option' parameter to **NOTIFY** (TxDTMFOption = 2).

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

■ **Using RFC 2833 relay with Payload type negotiation:** DTMF digits are sent to the remote side as part of the RTP stream according to RFC 2833. To enable this mode, define the following:

   a. Set the 'Declare RFC 2833 in SDP' parameter to **Yes** (RxDTMFOption = 3).

   b. Set the '1st Tx DTMF Option' parameter to **RFC 2833** (TxDTMFOption = 4).

   Note: To set the RFC 2833 payload type with a value other than its default, use the RFC2833PayloadType parameter. The device negotiates the RFC 2833 payload type using local and remote SDP and sends packets using the payload type from the received SDP. The device expects to receive RFC 2833 packets with the same payload type as configured by this parameter. If the remote side doesn't include 'telephony-event' in its SDP, the device sends DTMF digits in transparent mode (as part of the voice stream).

■ **Sending DTMF digits (in RTP packets) as part of the audio stream (DTMF Relay is disabled):** This method is typically used with G.711 coders. With other low-bit rate (LBR) coders, the quality of the DTMF digits is reduced. To enable this mode, define the following:

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

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

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

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

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

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

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

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

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

■ TxDTMFOption, RxDTMFOption, RFC2833TxPayloadType, and RFC2833RxPayloadType

■ MGCPDTMFDetectionPoint, DTMFVolume, DTMFTransportType, DTMFDigitLength, and DTMFInterDigitInterval

13.3.3.2 Configuring RFC 2833 Payload

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

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

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

Figure 13-6: RFC 2833 Payload Parameters in RTP/RTCP Settings Page

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

2. Configure the following parameters:

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

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

• 'RFC 2833 TX Payload Type' (RFC2833TxPayloadType) - defines the Tx RFC 2833 DTMF relay dynamic payload type.

• 'RFC 2833 RX Payload Type' (RFC2833RxPayloadType) - defines the Rx RFC 2833 DTMF relay dynamic payload type.

- 'RFC 2198 Payload Type' (RFC2198PayloadType) - defines the RTP redundancy packet payload type according to RFC 2198.

3. Click **Submit** to apply your settings.

13.3.4 Configuring RTP Base UDP Port

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

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

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

Where the port offsets are as follows:

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

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

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

- MP-112/MP-114: BaseUDPport to (BaseUDPport + 3*10) - for example, if the BaseUDPPort is set to 6,000, then the UDP port range is 6,000 to 6,030

- MP-118: BaseUDPport to (BaseUDPport + 7*10) - for example, if the BaseUDPPort is set to 6,000, then the UDP port range is 6,000 to 6,070

- MP-124: BaseUDPport to (BaseUDPport + 23*10) - for example, if the BaseUDPPort is set to 6,000, then the UDP port range is 6,000 to 6,230

Notes:

- The device allocates the UDP ports randomly to the channels.

- To configure the device to use the same port for both RTP and T.38 packets, set the T38UseRTPPort parameter to 1.

- If you are using Media Realms (see Configuring Media Realms on page 180), the port range configured for the Media Realm must be within this range defined by the BaseUDPPort parameter.

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

➢ **To configure the RTP base UDP port:**

1. Open the RTP/RTCP Settings page (**Configuration** tab > **VoIP** menu > **Media** submenu > **RTP/RTCP Settings**). The relevant parameter is listed under the 'General

Settings' group, as shown below:

Figure 13-7: RTP Based UDP Port in RTP/RTCP Settings Page

RTP Base UDP Port   6000

2. Set the 'RTP Base UDP Port' parameter to the required value.

3. Click **Submit**.

4. Reset the device for the settings to take effect.

13.4   Configuring Analog Settings

The Analog Settings page allows you to configure various analog parameters. For a detailed description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 471.

This page also selects the type (USA or Europe) of FXS and/or FXO coefficient information. The FXS coefficient contains the analog telephony interface characteristics such as DC and AC impedance, feeding current, and ringing voltage.

➢ **To configure the analog parameters:**

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

Figure 13-8: Analog Settings Page

FXS_FXO Settings	
Analog TTX Voltage Level	0.5V
Analog Metering Type	12 kHz sinusoidal bursts
Min. Hook-Flash Detection Period [msec]	300
Max. Hook-Flash Detection Period [msec]	700
FXS Coefficient Type	USA
FXO Coefficient Type	USA

2. Configure the parameters as required.

3. Click **Submit** to apply your changes.

4. To save the changes to flash memory, see 'Saving Configuration' on page 362.

13.5   Configuring DSP Templates

The DSP Template determines the coders that can be used by the device and various other functionalities. For a list of DSP templates and the maximum number of channels supported by each coder, see 'DSP Templates' on page 645. You can select a single DSP Template.

> **Notes:**
>
> - If no entries are defined, the device uses the default DSP template (i.e., Template 0).
> - A single DSP Template can also be configured using the ini file parameter, DSPVersionTemplateNumber.

➢ **To select a DSP Template(s):**

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

Figure 13-9: Defining Single DSP Template in General Settings Page

▼ General Settings	
⚡ DSP Version Template Number	4

2. In the 'DSP Version Template Number' field, enter the required DSP Template number.
3. Click **Submit**.
4. Reset the device with a flash burn for the settings to take effect (see 'Saving Configuration' on page 362).

13.6 Configuring Media Security

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

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

■ AES_CM_128_HMAC_SHA1_32

■ AES_CM_128_HMAC_SHA1_80

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

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

■ UNENCRYPTED_SRTP

■ UNENCRYPTED_SRTCP

■ UNAUTHENTICATED_SRTP

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

Below is an example of crypto attributes usage:

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

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

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

> **Notes:**
>
> • For a detailed description of the SRTP parameters, see SRTP Parameters on page 502.
>
> • When SRTP is used, the channel capacity may be reduced.

➢ **To configure media security:**

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

2. Configure the parameters as required.
3. Click **Submit** to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 362.

13.7 Configuring Media Realms

The Media Realm Table page allows you to define a pool of up to 64 SIP media interfaces, termed *Media Realms*. Media Realms allow you to divide a Media-type interface, which is configured in the Multiple Interface table, into several realms, where each realm is specified by a UDP port range. You can also define the maximum number of sessions per Media Realm. Once configured, Media Realms can be assigned to IP Groups (see 'Configuring IP Groups' on page 197).

Once you have configured a Media Realm, you can configure it with the following:

■ Quality of Experience parameters for reporting to AudioCodes OVOC server used for monitoring the quality of calls (see Configuring Quality of Experience Parameters per Media Realm on page 182)

Notes:

- For this setting to take effect, a device reset is required.
- The Media Realm table can also be configured using the table ini file parameter, CpMediaRealm.

➢ **To define a Media Realm:**

1. Open the Media Realm Table page (**Configuration** tab > **VoIP** menu > **Media** submenu > **Media Realm Configuration**).

2. Click the **Add** button; the following appears:

Figure 13-10: Media Realm Page - Add Record Dialog Box

3. Configure the parameters as required. See the table below for a description of each parameter

4. Click **Submit** to apply your settings.

5. Reset the device to save the changes to flash memory (see 'Saving Configuration' on page 362).

Table 13-1: Media Realm Table Parameter Descriptions

Parameter	Description
Index **[CpMediaRealm_Index]**	Defines the required table index number.

Parameter	Description
Media Realm Name **[CpMediaRealm_MediaRealm Name]**	Defines an arbitrary, identifiable name for the Media Realm. The valid value is a string of up to 40 characters. **Notes:** ▪ This parameter is mandatory. ▪ The name assigned to the Media Realm must be unique. ▪ This Media Realm name is used in the IP Groups table.
IPv4 Interface Name **[CpMediaRealm_IPv4IF]**	Assigns an IPv4 interface to the Media Realm. This is name of the interface as configured for the Interface Name field in the Multiple Interface table.
IPv6 Interface Name **[CpMediaRealm_**IPv6IF**]**	Assigns an IPv6 interface to the Media Realm. This is name of the interface as configured for the Interface Name field in the Multiple Interface table.
Port Range Start **[CpMediaRealm_PortRangeS tart]**	Defines the starting port for the range of Media interface UDP ports. **Notes:** ▪ You must either configure all media realms with port ranges or all without; not some with and some without. ▪ The available UDP port range is calculated using the BaseUDPport parameter: ✓ BaseUDPport to BaseUDPport + 4030*10 ▪ Port ranges over 60,000 must not be used.
Number of Media Session Legs **[CpMediaRealm_MediaSessi onLeg]**	Defines the number of media sessions associated with the range of ports. This is the number of media sessions available in the port range. For example, 100 ports correspond to 10 media sessions, since ports are allocated in chunks of 10.
Port Range End **[CpMediaRealm_PortRangeE nd]**	Read-only field displaying the ending port for the range of Media interface UDP ports. This field is calculated by adding the 'Media Session Leg' field (multiplied by the port chunk size) to the 'Port Range Start' field. A value appears once a row has been successfully added to the table.
Is Default **[CpMediaRealm_IsDefault]**	Defines the Media Realm as the default Media Realm. This default Media Realm is used when no Media Realm is configured for an IP Group for a specific call. ▪ **[0]** No (default) ▪ **[1]** Yes **Notes:** ▪ This parameter can be set to Yes for only **one** defined Media Realm. ▪ If this parameter is not configured, then the first Media Realm in the table is used as the default. ▪ If the table is not configured, then the default Media Realm includes all the configured media interfaces.

13.7.1 Configuring Quality of Experience per Media Realm

You can configure Quality of Experience (QoE) per Media Realm. This enables you to monitor and analyze media and signaling traffic, allowing you to detect problems causing service degradation. The device can save call information and statistics at call start, at call end, or at specific changes in the call. The information is stored as call records on an external server. The device connects, as a client, to the server using TLS over TCP.

You can specify the call parameters to monitor and configure their upper and lower thresholds. If these thresholds are exceeded, the device can be configured to do the following:

- Reports the change in the monitored parameter to the monitoring server (default).
- Sends RFC 2198 RTP redundancy packets on the call leg that crossed the threshold. This enables the device to adapt to the changed network status. In this option, you can also configure the redundancy depth. The channel configuration is unchanged if the change requires channel reopening. Currently, this option is applicable only when the monitored parameter is remote packet loss.

The device can be configured to monitor the following parameters on the local (i.e., at the device) or remote side:

- Packet loss
- Mean Opinion Score (MOS)
- Jitter
- Packet delay
- Residual Echo Return Loss (RERL)

At any given time during a call, each of these parameters can be in one of the following states according to its value in the last RTCP / RTCP XR packet:

- Gray - indicates that the value is unknown
- Green - indicates good call quality
- Yellow - indicates medium call quality
- Red - indicates poor call quality

The mapping between the values of the parameters and the color is according to the configured threshold of these parameters, per Media Realm. The call itself also has a state (color), which is the worst-state color of all the monitored parameters. Each time a color of a parameter changes, the device sends a report to the external server. A report is also sent at the end of each call.

Notes:

- To configure the address of the AudioCodes One Voice Operations Center (OVOC)server to where the device reports the QoE, see 'Configuring OVOC Server for Media Quality of Experience' on page 185.
- You can also configure QoE per Media Realm using the table *ini* file parameter QOERules.

➢ **To configure QoE per Media Realm:**

1. Open the Media Realm Table page (**Configuration** tab > **VoIP** menu > **Media** submenu > **Media Realm Configuration**).
2. Select the Media Realm for which you want to configure Quality of Experience, and then click the **Quality Of Experience** link; the Quality Of Experience page appears.

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

Figure 13-11: Quality of Experience Page - Add Record Dialog Box

The figure above shows value thresholds for the MOS parameter, which are assigned using pre-configured values of the Low Sensitivity profile. In this example setting, if the MOS value changes by 0.1 (hysteresis) to 3.3 or 3.5, the device sends a report to the OVOC indicating this change. If the value changes to 3.3, it sends a yellow state (i.e., medium quality); if the value changes to 3.5, it sends a green state.

4. Configure the parameters as required. See the table below for a description of each parameter.

5. Click **Submit** to apply your settings.

Table 13-2: Quality of Experience Parameter Descriptions

Parameter	Description
Index **[QOERules_RuleIndex]**	Defines the table index entry. Up to four table row entries can be configured per Media Realm.
Monitored Parameter **[QOERules_MonitoredParam]**	Defines the parameter to monitor and report. ▪ **[0]** MOS (default) ▪ **[1]** Delay ▪ **[2]** Packet Loss ▪ **[3]** Jitter ▪ **[4]** RERL
Direction **[QOERules_Direction]**	Defines the monitoring direction. ▪ **[0]** Device Side (default) ▪ **[1]** Remote Side

Parameter	Description
Profile **[QOERules_Profile]**	Defines the pre-configured threshold profile to use. ▪ **[0]** No Profile = No profile is used and you need to define the thresholds in the parameters described below. ▪ **[1]** Low Sensitivity = Automatically sets the thresholds to low sensitivity values. Therefore, reporting is done only if changes in parameters' values is significant. ▪ **[2]** Default Sensitivity = Automatically sets the thresholds to a medium sensitivity. ▪ **[3]** High Sensitivity = Automatically sets the thresholds to high sensitivity values. Therefore, reporting is done for small fluctuations in parameters' values.
Green Yellow Threshold **[QOERules_GreenYellowThreshold]**	Defines the parameter threshold values between green (good quality) and yellow (medium quality) states.
Green Yellow Hysteresis **[QOERules_GreenYellowHystersis]**	Defines the hysteresis (fluctuation) for the green-yellow threshold. When the threshold is exceeded by this hysteresis value, the device sends a report to the OVOC indicating this change.
Yellow Red Threshold **[QOERules_YellowRedThreshold]**	Defines the parameter threshold values between yellow (medium quality) and red (poor quality). When this threshold is exceeded, the device sends a report to the OVOC indicating this change.
Yellow Red Hysteresis **[QOERules_YellowRedHystersis]**	Defines the hysteresis (fluctuation) for the yellow-red threshold. When the threshold is exceeded by this hystersis value, the device sends a report to the OVOC indicating this change.
Green Yellow Operation **[QOERules_GreenYellowOperation]**	Defines the action that is done if the green-yellow threshold is crossed. ▪ **[1]** Notify = (Default) Device sends a report to the OVOC server. ▪ **[2]** Change Redundancy Depth= RTP redundancy packets are sent to the relevant call leg. **Note:** This field is applicable only if the monitored parameter is remote packet loss.
Green Yellow Operation Details **[QOERules_GreenYellowOperationDetails]**	**Note:** This field is currently not supported. Defines the desired RTP redundancy depth. The actual redundancy depth on the relevant call leg is the minimum between the desired depth and the maximum supported depth on that call leg. **Note:** This field is applicable only if the 'Green Yellow Operation' field is set to Change Redundancy Depth.
Yellow Red Operation **[QOERules_YellowRedOperation]**	**Note:** This field is currently not supported. Defines the action that is done if the yellow-red threshold is crossed. ▪ **[1]** Notify = (Default) Device sends a report to the OVOC server. ▪ **[2]** Change Redundancy Depth = RTP redundancy packets are sent to the relevant call leg. ▪ **Note:** This field is applicable only if the monitored parameter is remote packet loss.

Parameter	Description
Yellow Red Operation Details **[QOERules_YellowRedOperationDetails]**	**Note:** This field is currently not supported. Defines the desired RTP redundancy depth. The actual redundancy depth on the relevant call leg is the minimum between the desired depth and the maximum supported depth on that call leg. **Note:** This field is applicable only if the 'Yellow Red Operation' field is set to Change Redundancy Depth.

13.8   Quality of Experience

This chapter describes how to configure the Quality of Experience feature.

13.8.1   Reporting Voice Quality of Experience to OVOC

The device can be configured to report voice (media) Quality of Experience (QoE) to AudioCodes' One Voice Operations Center (OVOC)server, a plug-in for AudioCodes OVOC. The reports include real-time metrics of the quality of the actual call experience, which are then processed by the OVOC.

OVOC is a VoIP-quality monitoring and analysis tool. OVOC 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 OVOC in their VoIP networks to guarantee effective utilization, smooth performance, reliable QoS levels, and SLA fulfillment.

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

13.8.1.1 Configuring the OVOC Server

The device can report QoE voice metrics to AudioCodes OVOC server.

You can also configure at what stage of the call the device must send the report to the OVOC 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.

> **Notes:**
> - To configure the parameters to report and their thresholds per Media Realm, see 'Configuring Quality of Experience per Media Realm' on page 182.
> - For information on the OVOC server, refer to the *OVOC User's Manual*.

For a detailed description of the OVOC parameters, see "SIP Media Realm Parameters" on page 514.

➢ **To configure the OVOC server address and other related features:**

1. Open the Media Quality of Experience page (**Configuration** tab > **VoIP** menu > **Media**

submenu > **Media Quality of Experience**.

Figure 13-12: Configuring OVOC

Quality of Experience	
Server IP Address	0.0.0.0
Port	5000
Interface Name	DEFAULT

2. In the 'Server IP Address' field, enter the OVOC server's IP address.

3. In the 'Port' field, enter the OVOC server's port.

4. In the 'Interface Name' field, enter the device's IP network interface from which the device sends the reports to the OVOC server.

5. Click **Submit**.

13.8.1.2 Configuring Clock Synchronization between Device and OVOC

To ensure accurate call quality statistics and analysis by the OVOC server, you must configure the device and the OVOC 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 OVOC server (also acting as an NTP server)

■ Third-party, external NTP server

Once you have determined the NTP server, all the elements—device and OVOC--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 111.

13.8.1.3 Enabling RTCP XR Reporting to OVOC

In order for the device to be able to send voice metric reports to the OVOC, 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 OVOC server.

For enabling RTCP XR reporting, see "Configuring RTCP XR" on page 419. To configure what to report to the OVOC, see "Configuring Quality of Experience Profiles" on page 182.

14   Services

This section describes configuration for various supported services.

14.1   Least Cost Routing

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

14.1.1   Overview

The LCR feature enables the device to choose the outbound IP destination routing rule based on lowest call cost. This is useful in that it enables service providers to optimize routing costs for customers. For example, you may wish to define different call costs for local and international calls, or different call costs for weekends and weekdays (specifying even the time of call). The device sends the calculated cost of the call to a Syslog server (as Information messages), thereby enabling billing by third-party vendors.

LCR is implemented by defining Cost Groups and assigning them to routing rules in the Outbound IP Routing table. The device searches this routing table for matching routing rules, and then selects the rule with the lowest call cost. If two routing rules have identical costs, then the rule appearing higher up in the table is used (i.e., first-matched rule). If a selected route is unavailable, the device selects the next least-cost routing rule. However, even if a matched rule is not assigned a Cost Group, the device can select it as the preferred route over other matched rules with Cost Groups. This is determined according to the settings of the Default Cost parameter in the Routing Rule Groups table.

The Cost Group defines a fixed connection cost (*connection cost*) and a charge per minute (*minute cost*). Cost Groups can also be configured with time segments (*time bands*), which define connection cost and minute cost based on specific days of the week and time of day (e.g., from Saturday through Sunday, between 6:00 and 18:00). If multiple time bands are configured per Cost Group and a call spans multiple time bands, the call cost is calculated using only the time band in which the call was initially established.

In addition to Cost Groups, the device can calculate the call cost using an optional, user-defined average call duration value. The logic in using this option is that a Cost Group may be cheap if the call duration is short, but due to its high minute cost, may prove very expensive if the duration is lengthy. Thus, together with Cost Groups, the device can use this option to determine least cost routing. The device calculates the Cost Group call cost as follows: Total Call Cost = Connection Cost + (Minute Cost * Average Call Duration).

The below table shows an example of call cost when taking into consideration call duration. This example shows four defined Cost Groups and the total call cost if the average call duration is 10 minutes:

Table 14-1: Call Cost Comparison between Cost Groups for different Call Durations

Cost Group	Connection Cost	Minute Cost	Total Call Cost per Duration	
			1 Minute	**10 Minutes**
A	1	6	7	61
B	0	10	10	100
C	0.3	8	8.3	80.3
D	6	1	7	**16**

If four matching routing rules are located in the routing table and each one is assigned a different Cost Group as listed in the table above, then the rule assigned Cost Group "D" is selected. Note that for one minute, Cost Groups "A" and "D" are identical, but due to the average call duration, Cost Group "D" is cheaper. Therefore, average call duration is an important factor in determining the cheapest routing role.

Below are a few examples of how you can implement LCR:

■ **Example 1:** This example uses two different Cost Groups for routing local calls and international calls:

Two Cost Groups are configured as shown below:

Cost Group	Connection Cost	Minute Cost
1. "Local Calls"	2	1
2. "International Calls"	6	3

The Cost Groups are assigned to routing rules for local and international calls in the Outbound IP Routing table:

Routing Index	Dest Phone Prefix	Destination IP	Cost Group ID
1	2000	x.x.x.x	1 "Local Calls"
2	00	x.x.x.x	2 "International Calls"

■ **Example 2:** This example shows how the device determines the cheapest routing rule in the Outbound IP Routing table:

The Default Cost parameter (global) in the Routing Rule Groups table is set to **Min**, meaning that if the device locates other matching LCR routing rules (with Cost Groups assigned), the routing rule without a Cost Group is considered the lowest cost route.

• The following Cost Groups are configured:

Cost Group	Connection Cost	Minute Cost
1. "A"	2	1
2. "B"	6	3

• The Cost Groups are assigned to routing rules in the Outbound IP Routing table:

Routing Index	Dest Phone Prefix	Destination IP	Cost Group ID
1	201	x.x.x.x	"A'
2	201	x.x.x.x	"B"
3	201	x.x.x.x	0
4	201	x.x.x.x	"B"

The device calculates the optimal route in the following index order: 3, 1, 2, and then 4, due to the following logic:

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

■ **Example 3:** This example shows how the cost of a call is calculated if the call spans over multiple time bands:

Assume a Cost Group, "CG Local" is configured with two time bands, as shown below:

Cost Group	Time Band	Start Time	End Time	Connection Cost	Minute Cost
CG Local	TB1	16:00	17:00	2	1
	TB2	17:00	18:00	7	2

Assume that the call duration is 10 minutes, occurring between 16:55 and 17:05. In other words, the first 5 minutes occurs in time band "TB1" and the next 5 minutes occurs in "TB2", as shown below:

Figure 14-1: LCR using Multiple Time Bands (Example)
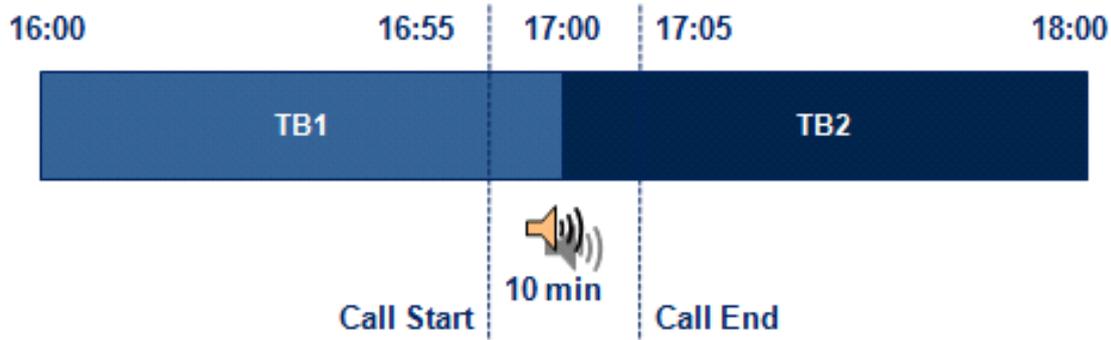
The device calculates the call using the time band in which the call was initially established, regardless of whether the call spans over additional time bands:

Total call cost = "TB1" Connection Cost + ("TB1" Minute Cost x call duration) = 2 + 1 x 10 min = 12

14.1.2 Configuring LCR

The following main steps need to be done to configure LCR:

1. Enable the LCR feature and configure the average call duration and default call connection cost - see 'Enabling LCR and Configuring Default LCR' on page 189.

2. Configure Cost Groups - see 'Configuring Cost Groups' on page 191.

3. Configure Time Bands for a Cost Group - see 'Configuring Time Bands for Cost Groups' on page 192.

4. Assign Cost Groups to outbound IP routing rules - see 'Assigning Cost Groups to Routing Rules' on page 193.

14.1.2.1 Enabling the LCR Feature

The procedure below describes how to enable the LCR feature. This also includes configuring the average call duration and default call cost for routing rules that are not assigned Cost Groups in the Outbound IP Routing table.

> **Note:** The Routing Rule Groups table can also be configured using the table ini file parameter, RoutingRuleGroups.

➢ **To enable LCR:**

1. Open the Routing Rule Groups Table page (**Configuration** tab > **VoIP** menu > **Services** submenu > **Least Cost Routing** > **Routing Rule Groups Table**).

2. Click the **Add** button; the Add Record dialog box appears:

Figure 14-2: Routing Rule Groups Table - Add Record

3. Configure the parameters as required. For a description of the parameters, see the table below.

4. Click **Submit**; the entry is added to the Routing Rule Groups table.

Table 14-2: Routing Rule Groups Table Description

Parameter	Description
Index **[RoutingRuleGroups_Index]**	Defines the table index entry. **Note:** Only one index entry can be configured.
LCR Enable **[RoutingRuleGroups_LCREnable]**	Enables the LCR feature: ▪ **[0]** Disabled (default) ▪ **[1]** Enabled
LCR Call Length **[RoutingRuleGroups_LCRAverageCallLength]**	Defines the average call duration (in minutes) and is used to calculate the variable portion of the call cost. This is useful, for example, when the average call duration spans over multiple time bands. The LCR is calculated as follows: cost = call connect cost + (minute cost * average call duration) The valid value range is 0-65533. The default is 1. For example, assume the following Cost Groups: ▪ "Weekend A": call connection cost is 1 and charge per minute is 6. Therefore, a call of 1 minute cost 7 units. ▪ "Weekend_ B": call connection cost is 6 and charge per minute is 1. Therefore, a call of 1 minute cost 7 units. Therefore, for calls under one minute, "Weekend A" carries the lower cost. However, if the average call duration is more than one minute, then "Weekend B" carries the lower cost.

Parameter	Description
Default Cost **[RoutingRuleGroups_LCRDefaultCost]**	Determines whether routing rules in the Outbound IP Routing table without an assigned Cost Group are considered a higher cost or lower cost route compared to other matched routing rules that are assigned Cost Groups. • **[0]** Lowest Cost = If the device locates other matching LCR routing rules, this routing rule is considered the lowest cost route and therefore, it is selected as the route to use (default.) • **[1]** Highest Cost = If the device locates other matching LCR routing rules, this routing rule is considered as the highest cost route and therefore, is not used or used only if the other cheaper routes are unavailable. **Note:** If more than one valid routing rule without a defined Cost Group exists, the device selects the first-matched rule.

14.1.2.2 Configuring Cost Groups

The procedure below describes how to configure Cost Groups. Cost Groups are defined with a fixed call connection cost and a call rate (charge per minute). Once configured, you can configure Time Bands for each Cost Group. Up to 10 Cost Groups can be configured.

> **Note:** The Cost Group table can also be configured using the table ini file parameter, CostGroupTable.

➢ **To configure Cost Groups:**

1. Open the Cost Group Table page (**Configuration** tab > **VoIP** menu > **Services** submenu > **Least Cost Routing** > **Cost Group Table**).
2. Click the **Add** button; the Add Record dialog box appears:

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

Table 14-3: Cost Group Table Description

Parameter	Description
Index **[CostGroupTable_Index]**	Defines the table index entry.
Cost Group Name **[CostGroupTable_CostGroupName]**	Defines an arbitrary name for the Cost Group. The valid value is a string of up to 30 characters. **Note:** Each Cost Group must have a unique name.
Default Connect Cost **[CostGroupTable_DefaultConnectionCost]**	Defines the call connection cost (added as a fixed charge to the call) for a call outside the time bands. The valid value range is 0-65533. The default is 0. **Note:** When calculating the cost of a call, if the current time of the call is not within a time band configured for the Cost Group, then this default connection cost is used.
Default Time Cost **[CostGroupTable_DefaultMinuteCost]**	Defines the call charge per minute for a call outside the time bands. The valid value range is 0-65533. The default is 0. **Note:** When calculating the cost of a call, if the current time of the call is not within a time band configured for the Cost Group, then this default charge per minute is used.

14.1.2.3 Configuring Time Bands for Cost Groups

The procedure below describes how to configure Time Bands for a Cost Group. The time band defines the day and time range for which the time band is applicable (e.g., from Saturday 05:00 to Sunday 24:00) as well as the fixed call connection charge and call rate per minute for this interval. Up to 70 time bands can be configured, and up to 21 time bands can be assigned to each Cost Group.

> **Notes:**
>
> - You cannot define overlapping time bands.
> - The Time Band table can also be configured using the table ini file parameter, CostGroupTimebands.

➢ **To configure Time Bands for a Cost Group:**

1. Open the Cost Group Table page (**Configuration** tab > **VoIP** menu > **Services** submenu > **Least Cost Routing** > **Cost Group Table**).

2. Select a Cost Group for which you want to assign Time Bands, and then click the **Time Band** link located below the table; the Time Band table for the selected Cost Group appears.

3. Click the **Add** button; the Add Record dialog box appears:

4. Configure the parameters as required. For a description of the parameters, see the table below.
5. Click **Submit**; the entry is added to the Time Band table for the relevant Cost Group.

Table 14-4: Time Band Table Description

Parameter	Description
Index **[CostGroupTimebands_Timeband Index]**	Defines the table index entry.
Start Time **[CostGroupTimebands_StartTime]**	Defines the day and time of day from when this time band is applicable. The format is DDD:hh:mm (e.g., SUN:06:00), where: ▪ *DDD* is 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)
End Time **[CostGroupTimebands_EndTime]**	Defines the day and time of day until when this time band is applicable. For a description of the valid values, see the parameter above.
Connection Cost **[CostGroupTimebands_ConnectionCost]**	Defines the call connection cost during this time band. This is added as a fixed charge to the call. The valid value range is 0-65533. The default is 0. **Note:** The entered value must be a whole number (i.e., not a decimal).
Minute Cost **[CostGroupTimebands_MinuteCost]**	Defines the call cost per minute charge during this timeband. The valid value range is 0-65533. The default is 0. **Note:** The entered value must be a whole number (i.e., not a decimal).

14.1.2.4 Assigning Cost Groups to Routing Rules

Once you have configured your Cost Groups, you need to assign them to routing rules in the Outbound IP Routing table - see Configuring Tel to IP Routing on page

This page is intentionally left blank.

15   Enabling Applications

In addition to the Gateway application (i.e., IP-to-Tel and Tel-to-IP calling), the device supports the following main application:

■   Stand-Alone Survivability (SAS) application

The procedure below describes how to enable these applications. Once an application is enabled, the Web GUI provides menus and parameter fields relevant to the application.

> **Notes:**
>
> • For configuring the SAS application, see 'Stand-Alone Survivability (SAS) Application' on page 325.
> • For enabling an application, a device reset is required.

➢   **To enable an application:**

1.   Open the Applications Enabling page (**Configuration** tab > **VoIP** menu > **Applications Enabling** submenu > **Applications Enabling**).

SAS Application	Enable ▼

2.   From the relevant application drop-down list, select **Enable**.

3.   Save (burn) the changes to the device's flash memory with a device reset (see 'Saving Configuration' on page 362).

This page is intentionally left blank.

16   Control Network

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

16.1   Configuring IP Groups

The IP Group Table page allows you to create up to nine logical IP entities called *IP Groups*. An IP Group is an entity with a set of definitions such as a Proxy Set ID (see 'Configuring Proxy Sets Table' on page 200), which represents the IP address of the IP Group.

IP Groups are used for the following:

■   SIP dialog registration and authentication (digest user/password) of a specific IP Group (Served IP Group, e.g., corporate IP-PBX) with another IP Group (Serving IP Group, e.g., ITSP). This is configured in the Account table (see Configuring Account Table on page 205).

■   Call routing rules:

- Outgoing IP calls (Tel-to-IP): The IP Group identifies the source of the call and is used as the destination of the outgoing IP call (defined in the Tel to IP Routing). For Tel-to-IP calls, the IP Group (Serving IP Group) can be used as the IP destination to where all SIP dialogs that are initiated from a Hunt Group are sent (defined in Configuring Hunt Group Settings on page 229).

- Incoming IP calls (IP-to-Tel): The IP Group identifies the source of the IP call.

- Number Manipulation rules to IP: The IP Group is used to associate the rule with specific calls identified by IP Group.

> **Notes:**
>
> - IP Group ID 0 cannot be used. This IP Group is set to default values and is used by the device when IP Groups are not implemented.
> - When operating with multiple IP Groups, the default Proxy server must not be used (i.e., the parameter IsProxyUsed must be set to 0).
> - You can also configure the IP Groups table using the table ini file parameter, IPGroup (see 'Configuration Parameters Reference' on page 471).

➢ **To configure IP Grops:**

1. Open the IP Group Table page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **IP Group Table**).

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

3. Configure the IP Group parameters according to the table below.

4. Click **Submit**.

5. To save the changes to flash memory, see 'Saving Configuration' **on page 362.**

Table 16-1: IP Group Parameters

Parameter	Description
Common Parameters	
Description **[IPGroup_Description]**	Defines a brief description for the IP Group. The valid value is a string of up to 29 characters. The default is an empty field.
Proxy Set ID **[IPGroup_ProxySetId]**	Assigns a Proxy Set ID to the IP Group. All INVITE messages destined to this IP Group are sent to the IP address configured for the Proxy Set. **Notes:** - Proxy Set ID 0 must **not** be used; this is the device's default Proxy. - To configure Proxy Sets, see 'Configuring Proxy Sets Table' on page 200.
SIP Group Name **[IPGroup_SIPGroupName]**	Defines the SIP Request-URI host name used in INVITE and REGISTER messages sent to this IP Group, or the host name in the From header of INVITE messages received from this IP Group. The valid value is a string of up to 49 characters. The default is an empty field. **Note:** If this parameter is not configured, the value of the global parameter, ProxyName is used instead (see 'Configuring Proxy and Registration Parameters' on page 208).

Parameter	Description
Contact User **[IPGroup_ContactUser]**	Defines the user part of the From, To, and Contact headers of SIP REGISTER messages, and the user part of the Contact header of INVITE messages received from this IP Group and forwarded by the device to another IP Group. **Note:** This parameter is overridden by the 'Contact User' parameter in the 'Account' table (see 'Configuring Account Table' on page 205).
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 to a specific IP Group, and the Contact header in SIP 18x and 200 OK responses for incoming INVITE messages from a specific IP Group. The Inbound IP Routing table can be used to identify the source IP Group from where the INVITE message was received. If this parameter is not configured (default), these headers are populated with the device's dotted-decimal IP address of the network interface on which the message is sent. **Note:** To ensure proper device handling, this parameter should be a valid FQDN.
Media Realm Name **[IPGroup_MediaRealm]**	Assigns a Media Realm to the IP Group. The string value must be identical (including case-sensitive) to the Media Realm name defined in the Media Realm table. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ If the Media Realm is later deleted from the Media Realm table, then this value becomes invalid. ▪ For configuring Media Realms, see Configuring Media Realms on page 180.
IP Profile ID **[IPGroup_ProfileId]**	Assigns an IP Profile to the IP Group. The default is 0. **Note:** To configure IP Profiles, see 'Configuring IP Profiles' on page 217.
Gateway Parameters	
Always Use Route Table **[IPGroup_**AlwaysUseRouteTable**]**	Defines the Request-URI host name in outgoing INVITE messages. ▪ **[0]** No (default). ▪ **[1]** Yes = The device uses the IP address (or domain name) defined in the Tel to IP Routing (see Configuring the Tel to IP Routing on page 250) as the Request-URI host name in outgoing INVITE messages, instead of the value configured in the 'SIP Group Name' field.
SIP Re-Routing Mode **[IPGroup_**SIPReRoutingMode**]**	Defines the routing mode after a call redirection (i.e., a 3xx SIP response is received) or transfer (i.e., a SIP REFER request is received). ▪ **[-1]** Not Configured (Default) ▪ **[0]** Standard = INVITE messages that are generated as a result of Transfer or Redirect are sent directly to the URI,

Parameter	Description
	according to the Refer-To header in the REFER message or Contact header in the 3xx response. ▪ **[1]** Proxy = Sends a new INVITE to the Proxy. This is applicable only if a Proxy server is used and the parameter AlwaysSendtoProxy is set to 0. ▪ **[2]** Routing Table = Uses the Routing table to locate the destination and then sends a new INVITE to this destination. **Notes:** ▪ When this parameter is set to **[1]** and the INVITE sent to the Proxy fails, the device re-routes the call according to the Standard mode **[0].** ▪ When this parameter is set to **[2]** and the INVITE fails, the device re-routes the call according to the Standard mode **[0].** If DNS resolution fails, the device attempts to route the call to the Proxy. If routing to the Proxy also fails, the Redirect / Transfer request is rejected. ▪ When this parameter is set to **[2],** the XferPrefix parameter can be used to define different routing rules for redirected calls. ▪ This parameter is ignored if the parameter AlwaysSendToProxy is set to 1.

16.2 Configuring Proxy Sets Table

The Proxy Sets Table page allows you to define *Proxy Sets*. A Proxy Set is a group of Proxy servers defined by IP address or fully qualified domain name (FQDN). You can define up to 10 Proxy Sets, each with up to five Proxy server addresses. For each Proxy server address you can define the transport type (i.e., UDP, TCP, or TLS). The total number of IP addresses that can be resolved from a DNS query is 15. In addition, Proxy load balancing and redundancy mechanisms can be applied per Proxy Set if it contains more than one Proxy address.

Proxy Sets can later be assigned to Server-type IP Groups (see 'Configuring IP Groups' on page 197). When the device sends an INVITE message to an IP Group, it is sent to the IP address or domain name defined for the Proxy Set that is associated with the IP Group. In other words, the Proxy Set represents the **destination** of the call.

> **Notes:**
>
> • Proxy Sets can be assigned only to Server-type IP Groups.
>
> • The Proxy Set table can also be configured using two complementary tables:
>   - Proxy Set ID with IP addresses: Table ini file parameter, ProxyIP.
>   - Attributes for the Proxy Set: Table ini file parameter, ProxySet.

➢ **To configure Proxy Sets:**

1. Open the Proxy Sets Table page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **Proxy Sets Table**).

Figure 16-1: Proxy Sets Table Page

2. From the 'Proxy Set ID' drop-down list, select an ID for the desired group.
3. Configure the Proxy parameters, as required. For a description of the parameters, see the table below.
4. Click **Submit**.
5. To save the changes to flash memory, see 'Saving Configuration' on page 362.

Table 16-2: Proxy Sets Table Parameters

Parameter	Description
Web: Proxy Set ID OVOC: Index **[ProxySet_Index]**	Defines the Proxy Set identification number. The valid value is 0 to 9. Proxy Set ID 0 is used as the default Proxy Set. **Note:** Although not recommended, you can use both default Proxy Set (ID 0) and IP Groups for call routing. For example, in the Hunt Group Settings page (see Configuring Hunt Group Settings on page 229) you can configure a Serving IP Group to where you want to route specific Hunt Group endpoints, and all other device endpoints then use the default Proxy Set. You can also use IP Groups in the Tel to IP Routing (see Configuring the Tel to IP Routing on page 250) to configure the default Proxy Set if the parameter PreferRouteTable is set to 1. To summarize, if the default Proxy Set is used, the INVITE message is sent according to the following preferences: ▪ To the Hunt Group's Serving IP Group ID, as defined in the Hunt Group Settings table.

Parameter	Description
	• According to the Tel to IP Routing if the parameter PreferRouteTable is set to 1. • To the default Proxy. Typically, when IP Groups are used, there is no need to use the default Proxy and all routing and registration rules can be configured using IP Groups and the Account tables (see 'Configuring Account Table' on page 205).
Proxy Address **[ProxyIp_IpAddress]**	Defines the address (and optionally, port number) of the Proxy server. Up to five 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 as an FQDN. You can also specify the selected port in the format, <IP address>:<port>. If you enable Proxy Redundancy (by setting the parameter EnableProxyKeepAlive to 1 or 2), the device can operate with multiple Proxy servers. If there is no response from the first (*primary*) Proxy defined in the list, the device attempts to communicate with the other (*redundant*) Proxies in the list. When a redundant Proxy is located, the device either continues operating with it until the next failure occurs or reverts to the primary Proxy (refer to the parameter ProxyRedundancyMode). If none of the Proxy servers respond, the device goes over the list again. The device also provides real-time switching (Hot-Swap mode) between the primary and redundant proxies (refer to the parameter IsProxyHotSwap). If the first Proxy doesn't respond to the INVITE message, the same INVITE message is immediately sent to the next Proxy in the list. The same logic applies to REGISTER messages (if RegistrarIP is not defined). **Notes:** • If EnableProxyKeepAlive is set to 1 or 2, the device monitors the connection with the Proxies by using keep-alive messages (OPTIONS or REGISTER). • To use Proxy Redundancy, you must specify one or more redundant Proxies. • When a port number is specified (e.g., domain.com:5080), DNS NAPTR/SRV queries aren't performed, even if ProxyDNSQueryType is set to 1 or 2.
Transport Type **[ProxyIp_TransportType]**	Defines the transport type of the proxy server. • **[0]** UDP • **[1]** TCP • **[2]** TLS • **[-1]** = Undefined **Note:** If no transport type is selected, the value of the global parameter SIPTransportType is used.
Web/OVOC: Enable Proxy Keep Alive **[ProxySet_EnableProxyKeepAlive]**	Enables the Keep-Alive mechanism with the Proxy server(s). • **[0]** Disable (default). • **[1]** Using Options = Enables Keep-Alive with Proxy using SIP OPTIONS messages. • **[2]** Using Register = Enables Keep-Alive with Proxy using SIP REGISTER messages. If set to 'Using Options', the SIP OPTIONS message is sent every user-defined interval (configured by the parameter ProxyKeepAliveTime). If set to 'Using Register', the SIP REGISTER message is sent every user-

Parameter	Description
	defined interval (configured by the RegistrationTime parameter). Any response from the Proxy, either success (200 OK) or failure (4xx response) is considered as if the Proxy is communicating correctly. **Notes:** ▪ This parameter must be set to 'Using Options' when Proxy redundancy is used. ▪ When this parameter is set to 'Using Register', the homing redundancy mode is disabled. ▪ When the active proxy doesn't respond to INVITE messages sent by the device, the proxy is tagged as 'offline'. The behavior is similar to a Keep-Alive (OPTIONS or REGISTER) failure. ▪ If this parameter is enabled and the proxy uses the TCP/TLS transport type, you can enable CRLF Keep-Alive mechanism, using the UsePingPongKeepAlive parameter.
Web: Proxy Keep Alive Time OVOC: Keep Alive Time **[ProxySet_ProxyKeepAliveTime]**	Defines the Proxy keep-alive time interval (in seconds) between Keep-Alive messages. The valid range is 5 to 2,000,000. The default is 60. **Note:** This parameter is applicable only if the parameter EnableProxyKeepAlive is set to 1 (OPTIONS). When the parameter EnableProxyKeepAlive is set to 2 (REGISTER), the time interval between Keep-Alive messages is determined by the RegistrationTime parameter.
Web: Proxy Load Balancing Method OVOC: Load Balancing Method **[ProxySet_ProxyLoadBalancingMethod]**	Enables the Proxy Load Balancing mechanism per Proxy Set ID. ▪ **[0]** Disable = Load Balancing is disabled (default) ▪ **[1]** Round Robin ▪ **[2]** Random Weights When the Round Robin algorithm is used, a list of all possible Proxy IP addresses is compiled. This list includes all IP addresses per Proxy Set, after necessary DNS resolutions (including NAPTR and SRV, if configured). After this list is compiled, the Proxy Keep-Alive mechanism (according to parameters EnableProxyKeepAlive and ProxyKeepAliveTime) tags each entry as 'offline' or 'online'. Load balancing is only performed on Proxy servers that are tagged as 'online'. All outgoing messages are equally distributed across the list of IP addresses. REGISTER messages are also distributed unless a RegistrarIP is configured. The IP addresses list is refreshed according to ProxyIPListRefreshTime. If a change in the order of the entries in the list occurs, all load statistics are erased and balancing starts over again. When the Random Weights algorithm is used, the outgoing requests are not distributed equally among the Proxies. The weights are received from the DNS server by 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. The Random Weights Load Balancing is not used in the following scenarios: ▪ The Proxy Set includes more than one Proxy IP address. ▪ The only Proxy defined is an IP address and not an FQDN. ▪ SRV is not enabled (DNSQueryType). ▪ The SRV response includes several records with a different Priority value.

Parameter	Description
Web/OVOC: Is Proxy Hot-Swap **[ProxySet_IsProxyHotSwap]**	Enables the Proxy Hot-Swap redundancy mode. ▪ **[0]** No (default) ▪ **[1]** Yes If Proxy Hot-Swap is enabled, the SIP INVITE/REGISTER message is initially sent to the first Proxy/Registrar server. If there is no response from the first Proxy/Registrar server after a specific number of retransmissions (configured by the parameter HotSwapRtx), the message is resent to the next redundant Proxy/Registrar server.
Web/OVOC: Proxy Redundancy Mode **[ProxySet_ProxyRedundancyMode]**	Determines whether the device switches back to the primary Proxy after using a redundant Proxy. ▪ **[-1]** Not Configured = (Default) The global parameter, ProxyRedundancyMode applies. ▪ **[0]** Parking = The device continues operating with a redundant (now active) Proxy until the next failure, after which it operates with the next redundant Proxy. ▪ **[1]** Homing = The device always attempts to operate with the primary Proxy server (i.e., switches back to the primary Proxy whenever it's available). **Notes:** ▪ To use the Proxy Redundancy mechanism, you need to enable the keep-alive with Proxy option, by setting the parameter EnableProxyKeepAlive to 1 or 2. ▪ If this parameter is configured, then the global parameter is ignored.
Main Proxy Success Detection Retries **[ProxySet_HomingSuccessDetectionRetries]**	Defines the number of consecutive, successful keep-alive (using OPTIONS method) responses from the primary proxy that are required before the device switches to the proxy after it was offline. This is used when the Proxy Set is configured for homing (i.e., 'Proxy Redundancy Mode' parameter set to **Homing**). The valid value range is 1 to 300 (default 1). **Note:** The parameter is applicable only if 'Proxy Redundancy Mode' is configured to **Homing** and 'Enable Proxy Keep Alive' is configured to **Using Options**.

17   SIP Definitions

This section describes configuration of SIP parameters.

17.1   Configuring SIP Parameters

Many of the stand-alone SIP parameters associated with various features can be configured in the following pages:

■ **SIP General Parameters page:** Provides SIP parameters for configuring general SIP features. To access this page, use the following path: **Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**.

■ **SIP Advanced Parameters page:** Provides SIP parameters for configuring advanced SIP features. To access this page, use the following path: **Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**.

For a description of these parameters, refer to the section corresponding to the feature or see 'Configuration Parameters Reference' on page 471.

17.2   Configuring Account Table

The Account Table page lets you define up to 24 Accounts per ("served") Hunt Group. Accounts are used to register and/or digest authenticate a Hunt Group, using a username and password, to a destination ("serving") IP Group. For example, the device can use the Account table to register a PBX, which is connected to the device, to an ITSP. The device sends the registration requests to the Proxy Set ID (see 'Configuring Proxy Sets Table' on page 200) that is associated with the serving IP Group.

A Hunt Group or served IP Group can register to more than one serving IP Group (e.g., multiple ITSPs). This is done by configuring multiple entries in the Account table for the same Hunt Group or served IP Group, but with different serving IP Groups, user name/password, host name, and contact user values.

When using the Account table to register a Hunt Group, if all channels belonging to the Hunt Group are down, the device un-registers the channels. If any channel belonging to the Hunt Group is returned to service, the device registers them again. This ensures, for example, that the Proxy does not send INVITEs to trunks that are out of service.

> **Notes:**
>
> • For viewing Account registration status, see Viewing Endpoint Registration Status on page 415.
>
> • The Account table can also be configured using the table ini file parameter, Account.

➢ **To configure Accounts:**

1. Open the Account Table page (**Configuration** tab > **VoIP** menu > **SIP Definitions** submenu > **Account Table**).

Index	Served Trunk Group	Serving IP Group	Username	Password	Host Name	Register	ContactUser	Application Type
1 ○	-1	1		*		No		GW\IP2IP
2 ○	-1	1		*		No		GW\IP2IP

2. In the 'Add' field, enter the desired table row index, and then click **Add**. A new row appears.

3. Configure the Account parameters according to the table below.

4. Click the **Apply** button to save your changes.

5. To save the changes, see 'Saving Configuration' on page 362.

6. To perform registration, click the **Register** button; to unregister, click **Unregister**. The registration method for each Trunk Group is according to the setting of the 'Registration Mode' parameter in the Trunk Group Settings page.

Table 17-1: Account Table Parameters Description

Parameter	Description
Served Trunk Group CLI: served-trunk-group [**Account**_ServedTrunkGroup]	Defines the Hunt Group ID that you want to register and/or authenticate to a destination IP Group (i.e., Serving IP Group). ▪ For Tel-to-IP calls, the Served Hunt Group is the source Hunt Group from where the call originated. ▪ For IP-to-Tel calls, the Served Hunt Group is the Hunt Group ID to which the call is sent.
Serving IP Group [**Account_ServingIPGroup**]	Defines the destination IP Group ID to where the SIP REGISTER requests, if enabled, are sent and authentication is done. The actual destination to where the REGISTER requests are sent is the IP address configured for the Proxy Set ID that is associated with the IP Group. Registration occurs only if: ▪ The 'Registration Mode' parameter is set to 'Per Account' in the Hunt Group Settings table (see Configuring Hunt Group Settings on page 229). ▪ The 'Register' parameter in this Account table is set to **Yes**. In addition, for a SIP call that is identified by both the Served Hunt Group and Serving IP Group, the username and password for digest authentication defined in this table is used. For Tel-to-IP calls, the Serving IP Group is the destination IP Group defined in the Hunt Group Settings table or Tel to IP Routing (see Configuring the Tel to IP Routing on page 250). For IP-to-Tel calls, the Serving IP Group is the 'Source IP Group ID' defined in the IP to Hunt Group Routing Table (see Configuring the IP to Hunt Group Routing Table on page 257). **Note:** If no match is found in this table for incoming or outgoing calls, the username and password defined in the Authentication table (see Configuring Authentication on page 302) or by the global parameters, UserName and Password (in the Proxy & Registration page) are used.
Username [**Account_Username**]	Defines the digest MD5 Authentication user name. The valid value is a string of up to 50 characters.
Password [**Account_Password**]	Defines the digest MD5 Authentication password. The valid value is a string of up to 50 characters. **Note:** After you click the **Apply** button, this password is displayed as an asterisk (*).

Parameter	Description
Host Name **[Account_HostName]**	Defines the Address of Record (AOR) host name. It appears in REGISTER From/To headers as ContactUser@HostName. For successful registrations, this host name is also included in the INVITE request's From header URI. This parameter can be up to 49 characters. **Note:** If this parameter is not configured or if registration fails, the 'SIP Group Name' parameter configured in the IP Group table is used instead.

Register
[Account_Register]
	Enables registration.
	
	▪ **[0]** No (Default)
	▪ **[1]** Yes
	
	When enabled, the device sends REGISTER requests to the Serving IP Group. The host name (i.e., host name in SIP From/To headers) and Contact User (user in From/To and Contact headers) are taken from this table upon successful registration. See the example below:
	<pre>REGISTER sip:xyz SIP/2.0
	Via: SIP/2.0/UDP
	10.33.37.78;branch=z9hG4bKac1397582418
	From:
	<sip:ContactUser@HostName>;tag=1c1397576231
	To: <sip: ContactUser@HostName >
	Call-ID: 1397568957261200022256@10.33.37.78
	CSeq: 1 REGISTER
	Contact:
	<sip:ContactUser@10.33.37.78>;expires=3600
	Expires: 3600
	User-Agent: Sip-Gateway/v.6.00A.008.002
	Content-Length: 0</pre>**Notes:**
	▪ To activate registration, you also need to set the parameter 'Registration Mode' to 'Per Account' in the Hunt Group Settings table for the specific Hunt Group.
	▪ The Hunt Group account registration is not affected by the parameter IsRegisterNeeded.

Contact User **[Account_ContactUser]**	Defines the AOR user name. This appears in REGISTER From/To headers as ContactUser@HostName, and in INVITE/200 OK Contact headers as ContactUser@<device's IP address>. **Notes:** ▪ If this parameter is not configured, the 'Contact User' parameter in the IP Group table is used instead. ▪ If registration fails, then the user part in the INVITE Contact header contains the source party number.
Application Type **[Account_ApplicationType]**	Defines the application type: ▪ **[0]** GW/IP2IP = (Default) Gateway application.

17.3   Configuring Proxy and Registration Parameters

The Proxy & Registration page allows you to configure the Proxy server and registration parameters. For a description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 471.

> **Note:**   To view the registration status of endpoints with a SIP Registrar/Proxy server, see Viewing Endpoint Registration Status on page 415.

➢   **To configure the Proxy and registration parameters:**

1.   Open the Proxy & Registration page (**Configuration** tab > **VoIP** menu > **SIP Definitions** submenu > **Proxy & Registration**).

Use Default Proxy	Yes
Proxy Set Table	
Proxy Name	
Redundancy Mode	Parking
Proxy IP List Refresh Time	60
Enable Fallback to Routing Table	Disable
Prefer Routing Table	No
Use Routing Table for Host Names and Profiles	Disable
Always Use Proxy	Disable
Redundant Routing Mode	Routing Table
SIP ReRouting Mode	Standard Mode
Enable Registration	Disable
Gateway Name	
Gateway Registration Name	
DNS Query Type	A-Record
Proxy DNS Query Type	A-Record
Subscription Mode	Per Endpoint
Number of RTX Before Hot-Swap	3
Use Gateway Name for OPTIONS	No
User Name	joe
Password	mikey
Cnonce	Default_Cnonce
Registration Mode	Per Endpoint
Set Out-Of-Service On Registration Failure	Disable
Challenge Caching Mode	None
Mutual Authentication Mode	Optional

2.   Configure the parameters as required.

3.   Click **Submit** to apply your changes.

➢   **To register or un-register the device to a Proxy/Registrar:**

■   Click the **Register** button to register.

■   Click **Un-Register** button to un-register.

Instead of registering the entire device, you can register specific entities as listed below by using the **Register** button located on the page in which these entities are configured:

■   FXS/FXO endpoints - Endpoint Phone Number Table page (see Configuring Endpoint

Phone Numbers on page 227)

■ Accounts - Account table (see 'Configuring Account Table' on page 205)

Click the **Proxy Set Table** button to Open the Proxy Sets Table page to configure groups of proxy addresses. Alternatively, you can open this page from the **Proxy Sets Table** page item (see 'Configuring Proxy Sets Table' on page 200 for a description of this page).

17.3.1 SIP Message Authentication Example

The device supports basic and digest (MD5) authentication types, according to SIP RFC 3261 standard. A proxy server might require authentication before forwarding an INVITE message. A Registrar/Proxy server may also require authentication for client registration. A proxy replies to an unauthenticated INVITE with a 407 Proxy Authorization Required response, containing a Proxy-Authenticate header with the form of the challenge. After sending an ACK for the 407, the user agent can then re-send the INVITE with a Proxy-Authorization header containing the credentials.

User agents, Redirect or Registrar servers typically use the SIP 401 Unauthorized response to challenge authentication containing a WWW-Authenticate header, and expect the re-INVITE to contain an Authorization header.

The following example shows the Digest Authentication procedure, including computation of user agent credentials:

1. The REGISTER request is sent to a Registrar/Proxy server for registration:

```
REGISTER sip:10.2.2.222 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c17940
To: <sip: 122@10.1.1.200>
Call-ID: 634293194@10.1.1.200
User-Agent: Sip-Gateway/MediaPack/v.6.60.010.006
CSeq: 1 REGISTER
Contact: sip:122@10.1.1.200:
Expires:3600
```

2. Upon receipt of this request, the Registrar/Proxy returns a 401 Unauthorized response:

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 10.2.1.200
From: <sip:122@10.2.2.222 >;tag=1c17940
To: <sip:122@10.2.2.222 >
Call-ID: 634293194@10.1.1.200
Cseq: 1 REGISTER
Date: Mon, 30 Jul 2012 15:33:54 GMT
Server: Columbia-SIP-Server/1.17
Content-Length: 0
WWW-Authenticate: Digest realm="audiocodes.com",
nonce="11432d6bce58ddf02e3b5e1c77c010d2",
stale=FALSE,
algorithm=MD5
```

3. According to the sub-header present in the WWW-Authenticate header, the correct REGISTER request is created.

4. Since the algorithm is MD5:

- The username is equal to the endpoint phone number "122".

- The realm return by the proxy is "audiocodes.com".

- The password from the *ini* file is "AudioCodes".

- The equation to be evaluated is "122:audiocodes.com:AudioCodes". According to the RFC, this part is called A1.

- The MD5 algorithm is run on this equation and stored for future usage.

- The result is "a8f17d4b41ab8dab6c95d3c14e34a9e1".

5. The par called A2 needs to be evaluated:
   - The method type is "REGISTER".
   - Using SIP protocol "sip".
   - Proxy IP from *ini* file is "10.2.2.222".
   - The equation to be evaluated is "REGISTER:sip:10.2.2.222".
   - The MD5 algorithm is run on this equation and stored for future usage.
   - The result is "a9a031cfddcb10d91c8e7b4926086f7e".

6. Final stage:
   - A1 result: The nonce from the proxy response is "11432d6bce58ddf02e3b5e1c77c010d2".
   - A2 result: The equation to be evaluated is "A1:11432d6bce58ddf02e3b5e1c77c010d2:A2".
   - The MD5 algorithm is run on this equation. The outcome of the calculation is the response needed by the device to register with the Proxy.
   - The response is "b9c45d0234a5abf5ddf5c704029b38cf".

   At this time, a new REGISTER request is issued with the following response:

```
REGISTER sip:10.2.2.222 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c23940
To: <sip: 122@10.1.1.200>
Call-ID: 654982194@10.1.1.200
Server: Audiocodes-Sip-Gateway/MediaPack/v.6.60.010.006
CSeq: 1 REGISTER
Contact: sip:122@10.1.1.200:
Expires:3600
Authorization: Digest, username: 122,
realm="audiocodes.com",
nonce="11432d6bce58ddf02e3b5e1c77c010d2",
uri="10.2.2.222",
response="b9c45d0234a5abf5ddf5c704029b38cf"
```

7. Upon receiving this request and if accepted by the Proxy, the Proxy returns a 200 OK response, completing the registration transaction:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c23940
To: <sip: 122@10.1.1.200>
Call-ID: 654982194@10.1.1.200
Cseq: 1 REGISTER
Date: Thu, 26 Jul 2012 09:34:42 GMT
Server: Columbia-SIP-Server/1.17
Content-Length: 0
Contact: <sip:122@10.1.1.200>; expires="Thu, 26 Jul 2012
10:34:42 GMT"; action=proxy; q=1.00
Contact: <122@10.1.1.200:>; expires="Tue, 19 Jan 2038 03:14:07
GMT"; action=proxy; q=0.00
Expires: Thu, 26 Jul 2012 10:34:42 GMT
```

18 Coders and Profiles

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

18.1 Configuring Coders

The Coders page allows you to configure up to 10 voice coders for the device. Each coder can be configured with packetization time (ptime), bit rate, payload type, and silence suppression. The first coder configured in the table has the highest priority and is used by the device whenever possible. If the remote side cannot use the first coder, the device attempts to use the next coder in the table, and so on.

Notes:

- A specific coder can only be configured once in the table.
- If packetization time and/or rate are not specified, the default is applied.
- Only the packetization time of the first coder in the coder list is declared in INVITE/200 OK SDP, even if multiple coders are defined.
- The device always uses the packetization time requested by the remote side for sending RTP packets. If not specified, the packetization time is assigned the default value.
- The value of several fields is hard-coded according to common standards (e.g., payload type of G.711 U-law is always 0). Other values can be set dynamically. If no value is specified for a dynamic field, a default is assigned. If a value is specified for a hard-coded field, the value is ignored.
- The G.722 coder provides Packet Loss Concealment (PLC) capabilities, ensuring higher voice quality.
- For G.729, it's also possible to select silence suppression without adaptations.
- If G.729 is selected and silence suppression is disabled, the device includes 'annexb=no' in the SDP of the relevant SIP messages. If silence suppression is enabled or set to 'Enable w/o Adaptations', 'annexb=yes' is included. An exception to this logic is when the remote gateway is a Cisco device (IsCiscoSCEMode).
- The G.727 coder is not supported by MP-124 Rev. E.
- For defining groups of coders, which can be assigned to Tel and IP Profiles, see 'Configuring Coder Groups' on page 214.
- For information on V.152 and implementation of T.38 and VBD coders, see 'Supporting V.152 Implementation' on page 169.
- The Coders table can also be configured using the table *ini* file parameter, CodersGroup.

➢ **To configure the device's coders:**

1. Open the Coders page (**Configuration** tab > **VoIP** menu > **Coders and Profiles**

submenu > **Coders**).

Figure 18-1: Coders Table Page

2. From the 'Coder Name' drop-down list, select the required coder.

3. From the 'Packetization Time' drop-down list, select the packetization time (in msec) for the selected coder. The packetization time determines how many coder payloads are combined into a single RTP packet.

4. From the 'Rate' drop-down list, select the bit rate (in kbps) for the selected coder.

5. In the 'Payload Type' field, if the payload type (i.e., format of the RTP payload) for the selected coder is dynamic, enter a value from 0 to 120 (payload types of 'well-known' coders cannot be modified).

6. From the 'Silence Suppression' drop-down list, enable or disable the silence suppression option for the selected coder.

7. Repeat steps 2 through 6 for the next optional coders.

8. Click **Submit**.

9. To save the changes to flash memory, see 'Saving Configuration' on page 362.

The table below lists the supported coders:

Table 18-1: Supported Coders

Coder Name	Packetization Time (msec)	Rate (kbps)	Payload Type	Silence Suppression
G.711 A-law **[g711Alaw64k]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	64	8	• **[0]** Disable • **[1]** Enable
G.711 U-law **[g711Ulaw64k]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	64	0	• **[0]** Disable • **[1]** Enable
G.711A-law_VBD **[g711AlawVbd]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	64	Dynamic (0-127) Default is 180	N/A
G.711U-law_VBD **[g711UlawVbd]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	64	Dynamic (0-127) Default is 120	N/A
G.722 **[g722]**	20 (default), 40, 60, 80, 100, 120	64 (default)	9	N/A
G.723.1 **[g7231]**	30 (default), 60, 90, 120, 150	• **[0]** 5.3 (default) • **[1]** 6.3	4	• **[0]** Disable • **[1]** Enable

Coder Name	Packetization Time (msec)	Rate (kbps)	Payload Type	Silence Suppression
G.726 **[g726]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	▪ **[0]** 16 ▪ **[1]** 24 ▪ **[2]** 32 (default) ▪ **[3]** 40	Dynamic (0-127) Default is 23	▪ **[0]** Disable ▪ **[1]** Enable
G.727 ADPCM	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	16, 24, 32, 40	Dynamic (0-127)	▪ **[0]** Disable ▪ **[1]** Enable
G.729 **[g729]**	10, 20 (default), 30, 40, 50, 60, 80, 100	8	18	▪ **[0]** Disable ▪ **[1]** Enable ▪ **[2]** Enable w/o Adaptations
T.38 **[t38fax]**	N/A	N/A	N/A	N/A

18.2 Configuring Coder Groups

The Coder Group Settings page allows you to define up to four groups of coders (termed *Coder Groups*). For each Coder Group, you can define up to 10 coders configured with packetization time (ptime), rate, payload type, and silence suppression. The first coder in the Coder Group table has the highest priority and is used by the device whenever possible. If the remote side cannot use the first coder, the device attempts to use the next coder, and so on.

Coder Groups can be used as follows:

■ Assigned to Tel Profiles in the Tel Profiles table (see Configuring Tel Profiles on page 215).

■ Assigned to IP Profiles in the IP Profiles table (see 'Configuring IP Profiles' on page 217).

Notes:

- A specific coder can be selected only once per Coder Group.
- For a list of supported coders, see 'Configuring Coders' on page 211.
- The Coder Group Settings table can also be configured using the table *ini* file parameter, CodersGroup.

➢ **To configure Coder Groups:**

1. Open the Coder Group Settings page (**Configuration** tab > **VoIP** menu > **Coders and Profiles** submenu > **Coders Group Settings**).

Figure 18-2: Coder Group Settings Page

2. From the 'Coder Group ID' drop-down list, select a Coder Group ID.
3. From the 'Coder Name' drop-down list, select the first coder for the Coder Group.
4. From the 'Packetization Time' drop-down list, select the packetization time (in msec) for the coder. The packetization time determines how many coder payloads are combined into a single RTP packet.
5. From the 'Rate' drop-down list, select the bit rate (in kbps) for the coder you selected.
6. In the 'Payload Type' field, if the payload type (i.e., format of the RTP payload) for the coder you selected is dynamic, enter a value from 0 to 120 (payload types of common coders cannot be modified).

7. From the 'Silence Suppression' drop-down list, enable or disable the silence suppression option for the coder you selected.

8. Repeat steps 3 through 7 for the next coders (optional).

9. Repeat steps 2 through 8 for the next coder group (optional).

10. Click **Submit** to apply your changes.

18.3   Configuring Tel Profile

The Tel Profile Settings table allows you to define up to nine configuration profiles for Tel calls. These profiles are termed *Tel Profiles*. The Tel Profile Settings table contains a list of parameters, which can also be configured globally for all calls using their corresponding "global" parameters. The only difference between the Tel Profile parameters and the global parameters regarding description may be their default values.

Tel Profiles provide high-level adaptation when the device interworks between different equipment and protocols (at both the Tel and IP sides), each of which may require different handling by the device. Once configured, Tel Profiles can be assigned to specific channels (endpoints). Therefore, Tel Profiles enable you to assign special configuration settings for device handling of specific calls. For example, if specific channels require the use of the G.711 coder, you can configure a Tel Profile with this coder and assign it to these channels. Tel Profiles are assigned to channels in the <Endpoint Phone Number Table (see Configuring Endpoint Phone Numbers on page 227).

The procedure below describes how to configure Tel Profiles using the Web interface.

> **Note:**   Tel Profiles can also be configured using the table *ini* file parameter, TelProfile (see 'Configuration Parameters Reference' on page 471)

➢   **To configure Tel Profiles:**

1. Open the Tel Profile Settings page (**Configuration** tab > **VoIP** menu > **Coders and Profiles** submenu > **Tel Profile Settings**).

Figure 18-3: Tel Profile Settings

| Profile ID | 1 |
| Profile Name | mike |

Profile Parameters	
Profile Preference	1
Fax Signaling Method	No Fax
Dynamic Jitter Buffer Minimum Delay [msec]	10
Dynamic Jitter Buffer Optimization Factor	10
RTP IP DiffServ	46
Signaling DiffServ	40
Voice Volume (-32 to 31 dB)	0
DTMF Volume (-31 to 0 dB)	-11
Input Gain (-32 to 31 dB)	0
Enable Digit Delivery	Disable
Enable Polarity Reversal	Enable
Enable Current Disconnect	Disable
MWI Analog Lamp	Disable
MWI Display	Disable
Dial Plan Index	-1
Echo Canceler	Enable
Flash Hook Period	700
Enable Early Media	Disable
Progress Indicator to IP	Not Configured
Enable DID Wink	Disable
Dialing Mode	Two Stages
Enable Voice Mail Delay	Enable
Disconnect Call on Detection of Busy Tone	Enable
Time For Reorder Tone [sec]	255
Enable 911 PSAP	Disable
Enable AGC	Disable
EC NLP Mode	Adaptive NLP
Swap Tel To IP Phone Numbers	Disable

| Coder Group | |
| Coder Group | Default Coder Group |

2. From the 'Profile ID' drop-down list, select the Tel Profile index.

3. In the 'Profile Name' field, enter an arbitrary name that enables you to easily identify the Tel Profile.

4. From the 'Profile Preference' drop-down list, select the priority of the Tel Profile, where **1** is the lowest priority and **20** the highest. If both IP and Tel profiles apply to the same call, the coders and other common parameters (noted by an asterisk in the description of the parameter TelProfile) of the preferred Profile are applied to that call. If the Preference of the Tel and IP Profiles is identical, the Tel Profile parameters are applied. **Note:** If the coder lists of both IP and Tel Profiles apply to the same call, only the coders common to both are used. The order of the coders is determined by the preference.

5. Configure the parameters as required. For a description of each parameter, refer to the corresponding "global" parameter.

6. Click **Submit** to apply your changes.

18.4   Configuring IP Profiles

The IP Profile Settings table allows you to define up to nine *IP Profiles*. An IP Profile is a set of special call configuration behaviors relating to signaling and media (e.g., coder used) applied to specific IP calls (inbound and/or outbound). Therefore, IP Profiles provide high-level adaptation when the device interworks between different IP entities (for Tel and IP sides), each of which may require different handling by the device. For example, if a specific IP entity uses the G.711 coder only, you can configure an IP Profile with G.711 for this IP entity.

Many of the parameters in the IP Profile Settings table have a corresponding "global" parameter. If an IP Profile is not associated with specific calls, the settings of the global parameters are applied to these calls.

IP Profiles can be assigned to the following configuration elements:

- IP Groups - see Configuring IP Groups on page 197

- Tel-to-IP routing rules (for Gateway / IP-to-IP application) - see Configuring Tel-to-IP Routing Table on page 250

- IP-to-Tel routing rules (for Gateway / IP-to-IP application) - see Configuring IP-to-Tel Routing Table on page 257

The device selects the IP Profile as follows:

- If different IP Profiles (not default) are assigned to the same specific calls in all these tables, the device uses the IP Profile that has the highest preference level (as set in the 'Profile Preference' field). If they have the same preference level, the device uses the IP Profile assigned in the IP Group table.

- If different IP Profiles are assigned to these tables and one table is set to the default IP Profile, the device uses the IP Profile that is not the default.

Note:

- IP Profiles can also be implemented when using a Proxy server (when the AlwaysUseRouteTable parameter is set to 1).

- RxDTMFOption configures the received DTMF negotiation method: [-1] not configured, use the global parameter; [0] don't declare RFC 2833; [1] declare RFC 2833 payload type is SDP.

- You can also configure IP Profiles using the table ini file parameter, IPProfile (see Configuration Parameters Reference on page 471).

> **To configure IP Profiles:**

1. Open the IP Profile Settings page (**Configuration** tab > **VoIP** menu > **Coders and Profiles** submenu > **IP Profile Settings**).

Figure 18-4: IP Profile Settings

| Profile ID | 1 |
| Profile Name | |

Common Parameters

RTP IP DiffServ	46
Signaling DiffServ	40
Disconnect on Broken Connection	No
Dynamic Jitter Buffer Minimum Delay [msec](*)	10
Dynamic Jitter Buffer Optimization Factor(*)	10
RTP Redundancy Depth(*)	0
Echo Canceler(*)	Enable
Input Gain (-32 to 31 dB)(*)	0
Voice Volume (-32 to 31 dB)(*)	0

Gateway Parameters

Fax Signaling Method	No Fax
Play Ringback Tone to IP	Don't Play
Enable Early Media	Disable
Copy Destination Number to Redirect Number	Disable
Media Security Behavior	Preferable
CNG Detector Mode	Disable
Modems Transport Type	Enable Bypass
NSE Mode	Disable
Number of Calls Limit	-1
Progress Indicator to IP	Not Configured
Profile Preference	1
Coder Group	Default Coder Group
Remote RTP Base UDP Port	0
First Tx DTMF Option	Not Supported
Second Tx DTMF Option	Not Supported
Declare RFC 2833 in SDP	Yes
Enable Hold	Enable

2. From the 'Profile ID' drop-down list, select the IP Profile index.

3. In the 'Profile Name' field, enter an arbitrary name that allows you to easily identify the IP Profile.

4. From the 'Profile Preference' drop-down list, select the priority of the IP Profile, where '1' is the lowest priority and '20' is the highest. If both IP and Tel profiles apply to the same call, the coders and other common parameters (noted by an asterisk) of the preferred Profile are applied to that call. If the Preference of the Tel and IP Profiles is identical, the Tel Profile parameters are applied.
Note: If the coder lists of both IP and Tel Profiles apply to the same call, only the coders common to both are used. The order of the coders is determined by the preference.

5. Configure the parameters as required.

6. Click **Submit** to apply your changes.

Table 18-2: IP Profile Parameters Description

Parameter	Description
Web: Profile ID **[IpProfile_Index]**	Defines a unique index number for the IP Profile.
Web: Profile Name **[IpProfile_ProfileName]**	(Optional) Defines a descriptive name for the IP Profile.
Common Parameters	
Web: RTP IP DiffServ **[IpProfile_IPDiffServ]**	For a description, see the global parameter PremiumServiceClassMediaDiffServ.
Web: Signaling DiffServ **[IpProfile_SigIPDiffServ]**	For a description, see the global parameter PremiumServiceClassControlDiffServ.
Web: Disconnect on Broken Connection **[IpProfile_DisconnectOnBroken Connection]**	For a description, see the global parameter DisconnectOnBrokenConnection.
Web: Media IP Version Preference **[IpProfile_MediaIPVersionPrefer ence]**	For a description, see the global parameter MediaIPVersionPreference.
Web: Dynamic Jitter Buffer Minimum Delay **[IpProfile_JitterBufMinDelay]**	For a description, see the global parameter DJBufMinDelay.
Web: Dynamic Jitter Buffer Optimization Factor **[IpProfile_JitterBufOptFactor]**	For a description, see the global parameter DJBufOptFactor.
Web: RTP Redundancy Depth **[IpProfile_RTPRedundancyDept h]**	For a description, see the global parameter RTPRedundancyDepth.
Web: Echo Canceler **[IpProfile_EnableEchoCanceller]**	For a description, see the global parameter EnableEchoCanceller.
Web: Input Gain **[IpProfile_InputGain]**	For a description, see the global parameter InputGain.
Web: Voice Volume **[IpProfile_VoiceVolume]**	For a description, see the global parameter VoiceVolume.
Web: Symmetric MKI Negotiation **[IpProfile_EnableSymmetricMKI]**	For a description, see the global parameter EnableSymmetricMKI.
Web: MKI Size **[IpProfile_MKISize]**	For a description, see the global parameter SRTPTxPacketMKISize.
Gateway Parameters	
Web: Fax Signaling Method **[IpProfile_IsFaxUsed]**	For a description, see the global parameter IsFaxUsed.
Web: Play Ringback Tone to IP **[IpProfile_PlayRBTone2IP]**	For a description, see the global parameter PlayRBTone2IP.

Parameter	Description
Web: Enable Early Media **[IpProfile_EnableEarlyMedia]**	For a description, see the global parameter EnableEarlyMedia.
Web: Copy Destination Number to Redirect Number **[IpProfile_CopyDest2RedirectN umber]**	For a description, see the global parameter CopyDest2RedirectNumber.
Web: Media Security Behavior **[IpProfile_MediaSecurityBehaviour]**	For a description, see the global parameter MediaSecurityBehaviour.
Web: CNG Detector Mode **[IpProfile_CNGmode]**	For a description, see the global parameter CNGDetectorMode.
Web: Modems Transport Type **[IpProfile_VxxTransportType]**	For a description, see the global parameters V21ModemTransportType, V22ModemTransportType, V23ModemTransportType, V32ModemTransportType, and V34ModemTransportType.
Web: NSE Mode **[IpProfile_NSEMode]**	For a description, see the global parameter NSEMode.
Web: Number of Calls Limit **[IpProfile_CallLimit]**	Defines the maximum number of concurrent calls (incoming and outgoing). If the number of concurrent calls reaches this limit, the device rejects any new incoming and outgoing calls belonging to this IP Profile. This parameter can also be set to the following: **[-1]** = (Default) No limitation on calls. **[0]** = Calls are rejected. **Note:** For IP-to-IP calls, you can configure the device to route calls to an alternative IP Group when this maximum number of concurrent calls is reached. To do so, you need to add an alternative routing rule in the Outbound IP Routing table that reroutes the call to an alternative IP Group. You also need to add a rule to the Reason for Alternative Routing table to initiate an alternative rule for Tel-to-IP calls using cause 805.
Web: Progress Indicator to IP **[IpProfile_ProgressIndicator2IP]**	For a description, see the global parameter ProgressIndicator2IP.
Web: Profile Preference **[IpProfile_IpPreference]**	Defines the priority of the IP Profile, where "1" is the lowest and "20" the highest. If both IP and Tel Profiles apply to the same call, the coders and other common parameters of the preferred profile are applied to the call. If the preference of the Tel and IP Profiles is identical, the Tel Profile parameters are applied. **Note:** If the coder lists of both IP and Tel Profiles apply to the same call, only the coders common to both are used. The order of the coders is determined by the preference.
Web: Coder Group **[IpProfile_CodersGroupID]**	For a description, see the global parameter CodersGroup.
Web: Remote RTP Base UDP Port **[IpProfile_RemoteBaseUDPPort]**	For a description, see the global parameter RemoteBaseUDPPort.
Web: First Tx DTMF Option **[IpProfile_FirstTxDtmfOption]**	For a description, see the global parameter TxDTMFOption.

Parameter	Description
Web: Second Tx DTMF Option **[IpProfile_SecondTxDtmfOption]**	For a description, see the global parameter TxDTMFOption.
Web: Declare RFC 2833 in SDP **[IpProfile_RxDTMFOption]**	For a description, see the global parameter RxDTMFOption.
Web: Enable Hold **[IpProfile_EnableHold]**	For a description, see the global parameter EnableHold.

This page is intentionally left blank.

Part V

Gateway Application

19   Introduction

This section describes configuration of the Gateway applications. The Gateway application refers to IP-to-Tel call routing and vice versa.

> **Notes:**
>
> - In some areas of the Web interface, the term "GW" application refers to the Gateway applications, respectively.
>
> - The terms *IP-to-Tel* and *Tel-to-IP* refer to the direction of the call relative to the device. IP-to-Tel refers to calls received from the IP network and destined to the PBX (i.e., telephone connected directly or indirectly to the device); Tel-to-IP refers to calls received from telephones connected directly to the device's FXS ports or from the PBX, and destined for the IP network.
>
> - FXO (Foreign Exchange Office) is the interface replacing the analog telephone and connects to a Public Switched Telephone Network (PSTN) line from the Central Office (CO) or to a Private Branch Exchange (PBX). The FXO is designed to receive line voltage and ringing current, supplied from the CO or the PBX (just like an analog telephone). An FXO VoIP device interfaces between the CO/PBX line and the Internet.
>
> - FXS (Foreign Exchange Station) is the interface replacing the Exchange (i.e., the CO or the PBX) and connects to analog telephones, dial-up modems, and fax machines. The FXS is designed to supply line voltage and ringing current to these telephone devices. An FXS VoIP device interfaces between the analog telephone devices and the Internet.

This page is intentionally left blank.

20   Hunt Group

This section describes the configuration of the device's channels, which entails assigning them to Hunt Groups.

20.1   Configuring Endpoint Phone Numbers

The Endpoint Phone Number Table page allows you to activate the device's ports (channels or endpoints), by defining telephone numbers for the endpoints and assigning them to Hunt Groups and Tel Profiles.

> **Notes:**
>
> - Each endpoint must be assigned a unique phone number. In other words, no two endpoints can have the same phone number.
> - The number of endpoints depends on the MediaPack model (e.g., MP-118 displays 8 endpoints).
> - You can also configure the endpoint phone numbers using the table ini file parameter TrunkGroup (see 'Number Manipulation Parameters' on page 628).

➢ **To configure the Endpoint Phone Number table:**

1.   Open the Endpoint Phone Number Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Hunt Group** submenu > **Endpoint Phone Number**).

Figure 20-1: Endpoint Phone Number Table Page

	Channel(s)	Phone Number	Hunt Group ID	Tel Profile ID
1	1-4	200	1	0
2	5-8	300	2	1
3				
4				
5				
6				
7				
8				

2.   Configure the endpoint phone numbers according to the table below. You must enter a number in the 'Phone Number' fields for each port that you want to use.

3.   Click **Submit** to apply your changes.

4.   To save the changes to the flash memory, see 'Saving Configuration' on page 362.

To register an endpoint to a Proxy/Registrar server, click the **Register** button; to un-register an endpoint, click **Un-Register**.

Table 20-1: Endpoint Phone Number Table Parameters

Parameter	Description
Channel(s) **[TrunkGroup_FirstBChannel]** **[TrunkGroup_LastBChannel]**	Defines the device's channels (or ports) that you want to activate. Enter the channel numbers as labeled on the device's rear panel. You can enter a range of channels, by using the syntax *n-m*, where *n* represents the lower channel number and *m* the higher channel number. For example, "1-4" specifies channels 1 through 4.
Phone Number **[TrunkGroup_FirstPhoneNumber]**	Defines the telephone number for the channel. For a range of channels, enter only the first telephone number. Subsequent channels are assigned the next consecutive telephone number. For example, if you enter 400 for channels 1 to 4, then channel 1 is assigned phone number 400, channel 2 is assigned phone number 401, and so on. These phone numbers are also used for channel allocation for IP-to-Tel calls if the Hunt Group's 'Channel Select Mode' parameter is set to **By Dest Phone Number**. This value can include up to 50 characters. **Notes:** ▪ If this field includes alphabetical characters and the phone number is defined for a range of channels (e.g., 1-4), then the phone number must end with a number (e.g., 'user1'). ▪ Phone number must be entered 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.
Hunt Group ID **[TrunkGroup_TrunkGroupNum]**	Defines a Hunt Group ID (1-99) to the channels. The same Hunt Group ID can be assigned to more than one group of channels. The Hunt Group ID is used to define a group of common channel behaviors that are used for routing IP-to-Tel calls. If an IP-to-Tel call is assigned to a Hunt Group, the call is routed to the channel(s) pertaining to that Hunt Group ID. **Notes:** ▪ Once you have defined a Hunt Group, you must configure the parameter PSTNPrefix (IP to Hunt Group Routing Table) to assign incoming IP calls to the appropriate Hunt Group. If you do not configure this table, calls cannot be established. ▪ You can define the method for which calls are assigned to channels within the Hunt Groups, using the parameter TrunkGroupSettings.
Tel Profile ID **[TrunkGroup_ProfileId]**	Defines a Tel Profile ID to the channels. **Note:** For configuring Tel Profiles, see 'Configuring Tel Profiles' on page 215.

20.2   Configuring Hunt Group Settings

The Hunt Group Settings allows you to configure the following per Hunt Group:

■   Channel select method by which IP-to-Tel calls are assigned to the Hunt Group's channels.

■   Registration method for registering Hunt Groups to selected Serving IP Group IDs.

> **Notes:**
>
> • For configuring Hunt Groups, see Configuring Endpoint Phone Numbers on page 227.
> • The Hunt Group Settings table can also be configured using the table ini file parameter, TrunkGroupSettings (see 'Number Manipulation Parameters' on page 628).

➢ **To configure the Hunt Group Settings table:**

1.   Open the Hunt Group Settings page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Hunt Group** > **Hunt Group Settings**).

Figure 20-2: Hunt Group Settings Page

2.   From the 'Index' drop-down list, select the range of entries that you want to edit.

3.   Configure the Hunt Group as required. For a description of the parameters, see the table below.

4.   Click **Submit** to apply your changes.

5.   To save the changes to flash memory, see 'Saving Configuration' on page 362.

Table 20-2: Hunt Group Settings Parameters Description

Parameter	Description
Hunt Group ID **[TrunkGroupSettings_TrunkGroupId]**	Defines the Hunt Group ID that you want to configure.

Parameter	Description

Channel Select Mode **[TrunkGroupSettings_ChannelSelectMode]**	Defines the method by which IP-to-Tel calls are assigned to the channels of the Hunt Group.
	▪ **[0]** By Dest Phone Number = (Default) The channel is selected according to the called (destination) number. If the number is not located, the call is released. If the channel is unavailable (e.g., busy), the call is put on call waiting (if call waiting is enabled and no other call is on call waiting); otherwise, the call is released.
	▪ **[1]** Cyclic Ascending = The next available channel in the Hunt Group, in ascending cyclic order is selected. After the device reaches the highest channel number in the Hunt Group, it selects the lowest channel number in the Hunt Group, and then starts ascending again.
	▪ **[2]** Ascending = The lowest available channel in the Hunt Group is selected, and if unavailable, the next higher channel is selected.
	▪ **[3]** Cyclic Descending = The next available channel in descending cyclic order is selected. The next lower channel number in the Hunt Group is always selected. When the device reaches the lowest channel number in the Hunt Group, it selects the highest channel number in the Hunt Group, and then starts descending again.
	▪ **[4]** Descending = The highest available channel in the Hunt Group is selected, and if unavailable, the next lower channel is selected.
	▪ **[5]** Dest Number + Cyclic Ascending = The channel is selected according to the called number. If the called number isn't found, the next available channel in ascending cyclic order is selected.
	Note: If the called number is located, but the port associated with the number is busy, the call is released.
	▪ **[6]** By Source Phone Number = The channel is selected according to the calling number.
	▪ **[9]** Ring to Hunt Group = The device allocates IP-to-Tel calls to all the FXS ports (channels) in the Hunt Group. When a call is received for the Hunt Group, all telephones connected to the FXS ports belonging to the Hunt Group start ringing. The call is eventually received by whichever telephone first answers the call (after which the other phones stop ringing). This option is applicable only to FXS interfaces.
	▪ **[11]** Dest Number + Ascending = The device allocates channels to incoming IP-to-Tel calls as follows:
	  a. The device attempts to route the call to the channel that is associated with the destination (called) number. If located, the call is sent to that channel.
	  b. If the number is not located or the channel is unavailable (e.g., busy), the device searches in ascending order for the next available channel in the Trunk Group. If located, the call is sent to that channel.
	  c. If all the channels are unavailable, the call is released.
	Note: If this parameter is not configured for the Hunt Group, then its channel select method is according to the global parameter, ChannelSelectMode.

Parameter	Description
Registration Mode **[TrunkGroupSettings_Registrati onMode]**	Defines the registration method for the Hunt Group: ▪ **[1]** Per Gateway = (Default) Single registration for the entire device. This is applicable only if a default Proxy or Registrar IP is configured and Registration is enabled (i.e., parameter IsRegisterUsed is set to 1). In this mode, the SIP URI user part in the From, To, and Contact headers is set to the value of the global registration parameter, GWRegistrationName or username if GWRegistrationName is not configured. ▪ **[0]** Per Endpoint = Each channel in the Hunt Group registers individually. The registrations are sent to the 'Serving IP Group ID' if defined in the table, otherwise, it is sent to the default Proxy, and if no default Proxy, then to the Registrar IP. ▪ **[4]** Don't Register = No registrations are sent by endpoints pertaining to the Hunt Group. For example, if the device is configured globally to register all its endpoints (using the parameter ChannelSelectMode), you can exclude some endpoints from being registered by assigning them to a Hunt Group and configuring the Hunt Group registration mode to 'Don't Register'. ▪ **[5]** Per Account = Registrations are sent (or not) to an IP Group, according to the settings in the Account table (see 'Configuring Account Table' on page 205). An example is shown below of a REGISTER message for registering endpoint "101" using the registration Per Endpoint mode: ``` REGISTER sip:SipGroupName SIP/2.0 Via: SIP/2.0/UDP 10.33.37.78;branch=z9hG4bKac862428454 From: <sip:101@GatewayName>;tag=1c862422082 To: <sip:101@GatewayName> Call-ID: 9907977062512000232825@10.33.37.78 CSeq: 3 REGISTER Contact: <sip:101@10.33.37.78>;expires=3600 Expires: 3600 User-Agent: Sip-Gateway/v.6.60A.011.002 Content-Length: 0 ``` The "SipGroupName" in the Request-URI is configured in the IP Group table (see 'Configuring IP Groups' on page 197). **Notes:** ▪ If this parameter is not configured, the registration is performed according to the global registration parameter, ChannelSelectMode. ▪ To enable Hunt Group registration, set the global parameter, IsRegisterNeeded to 1. This is unnecessary for 'Per Account' registration mode. ▪ If the device is configured globally to register Per Endpoint and an endpoint group includes four FXO endpoints to register Per Gateway, the device registers all endpoints except the first four endpoints. The group of these four endpoints sends a single registration request.

Parameter	Description
Serving IP Group ID **[TrunkGroupSettings_ServingIP Group]**	Assigns an IP Group to where INVITE messages received from this Hunt Group are sent. The actual destination to where these INVITE messages are sent is according to the Proxy Set ID associated with the IP Group. The Request-URI host name in the INVITE and REGISTER messages (except for 'Per Account' registration modes) is set to the value of the 'SIP Group Name' parameter configured in the IP Group table (see 'Configuring IP Groups' on page 197). **Notes:** ▪ If this parameter is not configured, the INVITE messages are sent to the default Proxy or according to the Tel to IP Routing (see 'Configuring Tel to IP Routing' on page 250). ▪ If the PreferRouteTable parameter is set to 1 (see 'Configuring Proxy and Registration Parameters' on page 208), the routing rules in the Outbound IP Routing table take precedence over the selected Serving IP Group ID.
'Dedicated Connection Mode' **[TrunkGroupSettings_Dedicated ConnectionMode]**	Enables the use of a dedicated TCP socket for SIP traffic (REGISTER, re-REGISTER, SUBSCRIBE, and INVITE messages) per FXS analog channel (endpoint). ▪ **[0]** Reuse Connection = (Default) The device may use the same TCP socket for all endpoints. ▪ **[1]** Connection per Endpoint = Assigns a dedicated TCP socket to each endpoint. The socket on which the endpoint successfully registers to the server is the dedicated socket. The dedicated socket is used only for SIP requests from the endpoint's Hunt Group whose destination is the same as the destination where the endpoint registered (i.e., same Serving IP Group). If the endpoint is not registered to the Serving IP Group over a TCP connection, calls from the endpoint to the Serving IP Group are rejected (and trigger an immediate registration attempt). **Note:** ▪ If you configure the parameter to Connection per Endpoint, you must configure a Serving IP Group (in the 'Serving IP Group ID' parameter) and you must configure the 'Registration Mode' parameter to Per Endpoint. ▪ If you configure the parameter to Reuse Connection and SIP authentication with the server fails for one of the endpoints, the server may "blacklist" the TCP socket, thereby blocking traffic from all other endpoints using this same socket.
Gateway Name **[TrunkGroupSettings_Gateway Name]**	Defines the host name for the SIP From header in INVITE messages and for the From/To headers in REGISTER requests. **Note:** If this parameter is not configured, the global parameter, SIPGatewayName is used.
Contact User **[TrunkGroupSettings_ContactU ser]**	Defines the user part for the SIP Contact URI in INVITE messages and for the From, To, and Contact headers in REGISTER requests. **Notes:**

Parameter	Description
	▪ This parameter is applicable only if the 'Registration Mode' parameter is set to 'Per Account' and registration through the Account table is successful. ▪ If registration fails, the user part in the INVITE Contact header is set to the source party number. ▪ The 'Contact User' parameter in the Account table overrides this parameter (see 'Configuring Account Table' on page 205).

Trunk Group Name **[TrunkGroupSettings_TrunkGroupName]**	Defines a name for the Trunk Group. This name represents the Trunk Group in the SIP 'tgrp' parameter of the outgoing INVITE messages (according to RFC 4904). For example:
	`sip:+16305550100;tgrp=`**`TG-1`**`;trunk-context=+1-630@isp.example.net;user=phone`
	The valid value can be a string of up to 20 characters. By default, no name is configured.
	Notes:
	▪ If this parameter is not configured, the Trunk Group decimal number is used in the SIP 'tgrp' parameter.
	▪ This feature is enabled by any of the following parameters:
	  ✓ UseSIPtgrp
	  ✓ UseBroadsoftDTG
	▪ Currently, this parameter can only be configured using the ini file.

This page is intentionally left blank.

21   Manipulation

This section describes the configuration of various manipulation processes.

21.1   Configuring General Settings

The General Settings page allows you to configure general manipulation parameters. For a description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 471.

➢   **To configure the general manipulation parameters:**

1.   Open the General Settings page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Manipulations** submenu >**General Settings**).

▼	
Set TEL-to-IP Redirect Reason	Not Configured ⌄

2.   Configure the parameters as required.

3.   Click **Submit** to apply your changes.

21.2   Configuring Source/Destination Number Manipulation Rules

You can configure rules for manipulating destination and/or source telephone numbers for IP-to-Tel and Tel-to-IP calls. The following number manipulation tables are used for this:

■   **Tel-to-IP calls:**

•   Destination Phone Number Manipulation Table for Tel > IP Calls table (up to 120 entries)

•   Source Phone Number Manipulation Table for Tel > IP Calls table (up to 20 entries)

■   **IP-to-Tel calls:**

•   Destination Phone Number Manipulation Table for IP > Tel Calls table (up to 120 entries)

•   Source Phone Number Manipulation Table for IP > Tel Calls table (up to 20 entries)

The number manipulation tables provide two configuration areas:

■   Matching characteristics (*Rule*) of incoming call, for example, prefix of destination number.

■   Manipulation operation (*Action*), for example, remove user-defined number of digits from the left of the number.

If the incoming call matches the characteristics of a rule, then its manipulation action is applied.

The device searches a matching manipulation rule starting from the first entry (i.e., top of the table). In other words, a rule at the top of the table takes precedence over a rule defined lower down in the table. Therefore, define more specific rules above more generic rules. For example, if you enter 551 in Index 1 and 55 in Index 2, the device applies rule 1 to numbers that start with 551 and applies rule 2 to numbers that start with 550, 552, 553, and so on until 559. However, if you enter 55 in Index 1 and 551 in Index 2, the device applies rule 1 to all numbers that start with 55, including numbers that start with 551.

You can perform a second "round" (additional) of destination (NumberMapIP2Tel parameter) and source (SourceNumberMapIP2Tel parameter) number manipulations for IP-to-Tel calls

on an already manipulated number. The initial and additional number manipulation rules are both configured in these tables. The additional manipulation is performed on the initially manipulated number. Therefore, for complex number manipulation schemes, you only need to configure relatively few manipulation rules in these tables (that would otherwise require many rules). This feature is enabled using the following parameters:

■ PerformAdditionalIP2TELSourceManipulation for source number manipulation

■ PerformAdditionalIP2TELDestinationManipulation for destination number manipulation

Telephone number manipulation can be useful, for example, for the following:

■ Stripping or adding dialing plan digits from or to the number, respectively. For example, a user may need to first dial 9 before dialing the phone number to indicate an external line. This number 9 can then be removed by number manipulation before the call is setup.

■ Allowing or blocking Caller ID information according to destination or source prefixes. For more information on Caller ID, see Configuring Caller Display Information on page 305.

Notes:

- Number manipulation can occur before or after a routing decision is made. For example, you can route a call to a specific Hunt Group according to its original number, and then you can remove or add a prefix to that number before it is routed. To determine when number manipulation is performed, configure the 'IP to Tel Routing Mode' parameter (RouteModeIP2Tel) described in 'Configuring IP to Hunt Group Routing Table' on page 257, and 'Tel to IP Routing Mode' parameter (RouteModeTel2IP) described in 'Configuring Tel to IP Routing' on page 250.

- The device manipulates the number in the following order: 1) strips digits from the left of the number, 2) strips digits from the right of the number, 3) retains the defined number of digits, 4) adds the defined prefix, and then 5) adds the defined suffix.

- The source/destination number manipulation tables can also be configured using the ini file:
  1) **Destination Phone Number Manipulation Table for IP > Tel Calls table:** NumberMapIP2Tel (ini)
  2) **Destination Phone Number Manipulation Table for Tel > IP Calls table:** NumberMapTel2IP (ini)
  3) **Source Phone Number Manipulation Table for IP > Tel Calls table:** SourceNumberMapIP2Tel (ini)
  4) **Source Phone Number Manipulation Table for Tel > IP Calls table:** SourceNumberMapTel2IP (ini)

To configure number manipulation rules:

1. Open the required Number Manipulation page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Manipulations** submenu > **Dest Number IP->Tel**, **Dest Number Tel->IP**, **Source Number IP->Tel**, or **Source Number Tel->IP**); the relevant Manipulation table page is displayed.

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

Figure 21-1: Number Manipulation Table - Add Dialog Box

3. Click the **Rule** tab, and then configure the matching characteristics. For a description of the parameters, see the table below.

4. Click the **Action** tab, and then configure the manipulation operation. For a description of the parameters, see the table below.

5. Click **Submit** to apply your changes.

6. To save the changes to flash memory, see 'Saving Configuration' on page 362.

The table below shows configuration examples of Tel-to-IP source phone number manipulation rules, where:

■ **Rule 1:** When the destination number has the prefix 03 (e.g., 035000), source number prefix 201 (e.g., 20155), and from source IP Group ID 2, the source number is changed to, for example, 97120155.

■ **Rule 2:** When the source number has prefix 1001 (e.g., 1001876), it is changed to 587623.

■ **Rule 3:** When the source number has prefix 123451001 (e.g., 1234510012001), it is changed to 20018.

■ **Rule 4:** When the source number has prefix from 30 to 40 and a digit (e.g., 3122), it is changed to 2312.

■ **Rule 5:** When the destination number has the prefix 6, 7, or 8 (e.g., 85262146), source number prefix 2001, it is changed to 3146.

Parameter	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
Source IP Group	2	0	-	-	-
Destination Prefix	03		*	*	**[6**,7,8]

Parameter	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
Source Prefix	201	1001	123451001#	**[30-**40]x	2001
Stripped Digits from Left	-	4	-	-	5
Stripped Digits from Right	-	-	-	1	-
Prefix to Add	971	5	-	2	3
Suffix to Add	-	23	8	-	-
Number of Digits to Leave	-	-	4	-	-
Presentation	Allowed	Restricted	-	-	-

Table 21-1: Number Manipulation Parameters Description

Parameter	Description
Matching Characteristics (Rule)	
Web: Destination Prefix OVOC: Prefix **[DestinationPrefix]**	Defines the destination (called) telephone number prefix and/or suffix. You can use special notations for denoting the prefix. For example, **[100-**199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. You can also use the $ sign to denote calls without a called number. For a description of available notations, see 'Dialing Plan Notation for Routing and Manipulation Tables' on page 469.
Web/OVOC: Source Prefix **[SourcePrefix]**	Defines the source (calling) telephone number prefix and/or suffix. You can use special notations for denoting the prefix. For example, **[100-**199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. You can also use the $ sign to denote calls without a calling number. For a description of available notations, see 'Dialing Plan Notation for Routing and Manipulation Tables' on page 469.
Web/OVOC: Source IP Address **[SourceAddress]**	Defines the source IP address of the caller. This is obtained from the Contact header in the INVITE message. **Notes:** ▪ This parameter is applicable only to the number manipulation tables for IP-to-Tel calls. ▪ The source IP address can include the 'x' wildcard to represent single digits. For example: 10.8.8.xx represents all IP addresses between 10.8.8.10 to 10.8.8.99. ▪ The source IP address can include the asterisk (*) wildcard to represent any number between 0 and 255. For example, 10.8.8.* represents all IP addresses between 10.8.8.0 and 10.8.8.255.

Parameter	Description
Web: Source Host Prefix **[SrcHost]**	Defines the URI host name prefix of the incoming SIP INVITE message in the From header. **Notes:** ▪ This parameter is applicable only to the number manipulation tables for IP-to-Tel calls. ▪ The asterisk (*) wildcard can be used to denote any prefix. ▪ If the P-Asserted-Identity header is present in the incoming INVITE message, then the value of this parameter is compared to the P-Asserted-Identity URI host name (instead of the From header).
Web: Destination Host Prefix **[DestHost]**	Defines the Request-URI host name prefix of the incoming SIP INVITE message. **Notes:** ▪ This parameter is applicable only to the number manipulation tables for IP-to-Tel calls. ▪ The asterisk (*) wildcard can be used to denote any prefix.
Web: Source Trunk Group **[SrcTrunkGroupID]**	Defines the source Hunt Group ID for Tel-to-IP calls. To denote all Hunt Groups, leave this field empty. **Notes:** ▪ The value -1 indicates that this field is ignored in the rule. ▪ This parameter is applicable only to the number manipulation tables for Tel-to-IP calls.
Web: Source IP Group **[SrcIPGroupID]**	Defines the IP Group from where the IP call originated. Typically, the IP Group of an incoming INVITE is determined or classified using the IP to Hunt Group Routing Table. If not used (i.e., any IP Group), leave the field empty. **Notes:** ▪ The value -1 indicates that this field is ignored. ▪ This parameter is applicable only to the number manipulation tables for Tel-to-IP calls.
Web: Destination IP Group **[DestIPGroupID]**	Defines the IP Group to where the call is sent. **Notes:** ▪ The value -1 indicates that this field is ignored. ▪ This parameter is applicable only to the Destination Phone Number Manipulation Table for Tel -> IP Calls.
Operation (Action)	
Web: Stripped Digits From Left OVOC: Number Of Stripped Digits **[RemoveFromLeft]**	Defines the number of digits to remove from the left of the telephone number prefix. For example, if you enter 3 and the phone number is 5551234, the new phone number is 1234.
Web: Stripped Digits From Right OVOC: Number Of Stripped Digits **[RemoveFromRight]**	Defines the number of digits to remove from the right of the telephone number prefix. For example, if you enter 3 and the phone number is 5551234, the new phone number is 5551.
Web: Prefix to Add OVOC: Prefix/Suffix To Add **[Prefix2Add]**	Defines the number or string that you want added to the front of the telephone number. For example, if you enter 9 and the phone number is 1234, the new number is 91234.

Parameter	Description
Web: Suffix to Add OVOC: Prefix/Suffix To Add **[Suffix2Add]**	Defines the number or string that you want added to the end of the telephone number. For example, if you enter 00 and the phone number is 1234, the new number is 123400.
Web/OVOC: Number of Digits to Leave **[LeaveFromRight]**	Defines the number of digits that you want to keep from the right of the phone number. For example, if you enter 4 and the phone number is 00165751234, then the new number is 1234.
Web: Presentation OVOC: Is Presentation Restricted **[IsPresentationRestricted]**	Enables caller ID. - Not Configured = Privacy is determined according to the Caller ID table (see Configuring Caller Display Information on page 305). **[0]** Allowed = Sends Caller ID information when a call is made using these destination/source prefixes. **[1]** Restricted = Restricts Caller ID information for these prefixes. **Notes:** - This field is applicable only to number manipulation tables for source phone number manipulation. - If this field is set to **Restricted** and the 'Asserted Identity Mode' (AssertedIdMode) parameter is set to **Add P-Asserted-Identity**, the From header in the INVITE message includes the following: From: 'anonymous' <sip: anonymous@anonymous.invalid> and 'privacy: id' header.

21.3   Manipulating Number Prefix

The device supports a notation for adding a prefix where part of the prefix is first extracted from a user-defined location in the original destination or source number. This notation is entered in the 'Prefix to Add' field in the Number Manipulation tables (see 'Configuring Source/Destination Number Manipulation' on page 235): x**[n,**l]y...

where,

- x = any number of characters/digits to add at the beginning of the number (i.e. first digits in the prefix).
- **[n,**l] = defines the location in the original destination or source number where the digits y are added:
  - n = location (number of digits counted from the left of the number) of a specific string in the original destination or source number.
  - l = number of digits that this string includes.
- y = prefix to add at the specified location.

For example, assume that you want to manipulate an incoming IP call with destination number +5492028888888 (area code 202 and phone number 8888888) to the number 0202158888888. To perform such a manipulation, the following configuration is required in the Number Manipulation table:

1.   The following notation is used in the 'Prefix to Add' field:

   0**[5,**3]15

   where,

   - 0 is the number to add at the beginning of the original destination number.
   - **[5,**3] denotes a string that is located after (and including) the fifth character (i.e., the first '2' in the example) of the original destination number, and its length being three digits (i.e., the area code 202, in the example).

- 15 is the number to add immediately after the string denoted by **[5,**3] - in other words, 15 is added after (i.e. to the right of) the digits 202.

2. The first seven digits from the left are removed from the original number, by entering "7" in the 'Stripped Digits From Left' field.

Table 21-2: Example of Configured Rule for Manipulating Prefix using Special Notation

Parameter	Rule 1
Destination Prefix	+5492028888888
Source Prefix	*
Source IP Address	*
Stripped Digits from Left	7
Prefix to Add	0**[5,**3]15

In this configuration example, the following manipulation process occurs:

1. The prefix is calculated as 020215.
2. The first seven digits from the left are removed from the original number, thereby changing the number to 8888888.
3. The prefix that was previously calculated is then added.

21.4   SIP Calling Name Manipulations

The Calling Name Manipulations Tel2IP and Calling Name Manipulations IP2Tel tables allow you to configure up to 20 manipulation rules for manipulating the calling name (i.e., caller ID) in SIP messages. This can include modifying or removing the calling name. SIP calling name manipulation is applicable to Tel-to-IP and IP-to-Tel calls.

For example, assume that an incoming SIP INVITE message includes the following header:

```
P-Asserted-Identity: "company:john" sip:6666@78.97.79.104
```

Using the Calling Name Manipulations IP2Tel table, the text "company" can be changed to "worker" in the outgoing INVITE, as shown below:

```
P-Asserted-Identity: "worker:john" sip:996666@10.13.83.10
```

The calling name manipulation tables provide two configuration areas:

■   Matching characteristics (*Rule*) of incoming call, for example, prefix of destination number.

■   Manipulation operation (*Action*), for example, remove user-defined number of digits from the left of the calling name.

If the incoming call matches the characteristics of a rule, then its manipulation action is applied.

> **Notes:**
>
> • The Calling Name Manipulations Tel2IP table can also be configured using the table *ini* file parameter, CallingNameMapTel2Ip.
> • The Calling Name Manipulations IP2Tel table can also be configured using the table *ini* file parameter, CallingNameMapIp2Tel.

➢   **To configure calling name manipulation rules:**

1.   Open the required Calling Name Manipulations page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Manipulations** > **Calling Name IP->Tel** or **Calling Name Tel->IP**).

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

Figure 21-2: Calling Name Manipulation IP2Tel - Rule Tab

3.   Click the **Rule** tab, and then configure the matching characteristics. For a description of the parameters, see the table below.

4.   Click the **Action** tab, and then configure the manipulation operation. For a description of the parameters, see the table below.

5.   Click the **Submit** button to save your changes.

Table 21-3: Calling Name Manipulation Parameters Description

Parameter	Description
Matching Characteristics (Rule)	
Web: Destination Prefix	Defines the destination (called) telephone number prefix and/or suffix. You can use special notations for denoting the prefix. For example, **[100-**199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. You can also use the $ sign to denote calls without a called number. For a description of available notations, see 'Dialing Plan Notation for Routing and Manipulation Tables' on page 469.
Web/OVOC: Source Prefix	Defines the source (calling) telephone number prefix and/or suffix. You can use special notations for denoting the prefix. For example, **[100-**199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. You can also use the $ sign to denote calls without a calling number. For a description of available notations, see 'Dialing Plan Notation for Routing and Manipulation Tables' on page 469.
Web: Calling Name Prefix	Defines the caller name (i.e., caller ID) prefix. You can use special notations for denoting the prefix. For example, to denote any prefix, use the asterisk (*) symbol or to denote calls without a calling name, use the $ sign. For a description of available notations, see 'Dialing Plan Notation for Routing and Manipulation Tables' on page 469.
Web: Source Trunk Group ID	Defines the source Hunt Group ID for Tel-to-IP calls. To denote all Hunt Groups, leave this field empty. **Notes:** ▪ This parameter is applicable only to the Calling Name Manipulations Tel2IP table. ▪ The value -1 indicates that this field is ignored in the rule.
Web: Source IP Group ID	Defines the IP Group from where the IP call originated. **Notes:** ▪ This parameter is applicable only to the Calling Name Manipulations Tel2IP table. ▪ The value -1 indicates that this field is ignored in the rule.
Web/OVOC: Source IP Address	Defines the source IP address of the caller, obtained from the Contact header in the INVITE message. **Notes:** ▪ This parameter is applicable only to the Calling Name Manipulations IP2Tel table. ▪ The source IP address can include the 'x' wildcard to represent single digits. For example: 10.8.8.xx represents all IP addresses between 10.8.8.10 to 10.8.8.99. ▪ The source IP address can include the asterisk (*) wildcard to represent any number between 0 and 255. For example, 10.8.8.* represents all IP addresses between 10.8.8.0 and 10.8.8.255.

Parameter	Description
Web: Source Host Prefix	Defines the URI host name prefix of the incoming SIP INVITE message in the From header. **Notes:** ▪ This parameter is applicable only to the Calling Name Manipulations IP2Tel table. ▪ The asterisk (*) wildcard can be used to denote any prefix. ▪ If the P-Asserted-Identity header is present in the incoming INVITE message, then the value of this parameter is compared to the P-Asserted-Identity URI host name (instead of the From header).
Web: Destination Host Prefix	Defines the Request-URI host name prefix of the incoming SIP INVITE message. **Notes:** ▪ This parameter is applicable only to the Calling Name Manipulations IP2Tel table. ▪ The asterisk (*) wildcard can be used to denote any prefix.
Operation (Action)	
Web: Stripped Digits From Left OVOC: Number Of Stripped Digits	Defines the number of characters to remove from the left of the calling name. For example, if you enter 3 and the calling name is "company:john", the new calling name is "pany:john".
Web: Stripped Digits From Right OVOC: Number Of Stripped Digits	Defines the number of characters to remove from the right of the calling name. For example, if you enter 3 and the calling name is "company:name", the new name is "company:n".
Web/OVOC: Number of Digits to Leave	Defines the number of characters that you want to keep from the right of the calling name. For example, if you enter 4 and the calling name is "company:name", the new name is "name".
Web: Prefix to Add OVOC: Prefix/Suffix To Add	Defines the number or string to add at the front of the calling name. For example, if you enter ITSP and the calling name is "company:name", the new name is ITSPcompany:john".
Web: Suffix to Add OVOC: Prefix/Suffix To Add	Defines the number or string to add at the end of the calling name. For example, if you enter 00 and calling name is "company:name", the new name is "company:name00".

21.5 Configuring Redirect Number IP to Tel

You can configure rules for manipulating the redirect number received in the incoming message:

- Tel-to-IP redirect number manipulation: You can manipulate the prefix of the redirect number, received from the Tel side, in the outgoing SIP Diversion, Resource-Priority, or History-Info headers sent to the IP side. This is configured in the Redirect Number Tel > IP table.

The redirect number manipulation tables provide two configuration areas:

- Matching characteristics (*Rule*) of incoming call, for example, prefix of redirect number.
- Manipulation operation (*Action*), for example, remove user-defined number of digits from the left of the redirect number.

If the incoming call matches the characteristics of a rule, then its manipulation action is applied.

Notes:

- If the device copies the received destination number to the outgoing SIP redirect number (enabled by the CopyDest2RedirectNumber parameter), then no redirect number Tel-to-IP manipulation is done.
- The manipulation rules are done in the following order: Stripped Digits From Left, Stripped Digits From Right, Number of Digits to Leave, Prefix to Add, and then Suffix to Add.
- The Redirect Prefix parameter is used before it is manipulated.
- The redirect number manipulation tables can also be configured using the ini file: Redirect Number Tel to IP table - RedirectNumberMapTel2Ip (ini)

➢ **To configure redirect number manipulation rules:**

1. Open the redirect number manipulation table (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Manipulations** > **Redirect Number Tel > IP**).
2. Click the **Add** button; the following dialog box appears (e.g., Redirect Number Tel > IP table):

Figure 21-3: Redirect Number Manipulation (e.g., Tel to IP)

3. Click the **Rule** tab, and then configure the matching characteristics. For a description of the parameters, see the table below.

4. Click the **Action** tab, and then configure the manipulation operation. For a description of the parameters, see the table below.

5. Click **Submit** to apply your settings.

Table 21-4: Redirect Number Manipulation Parameters Description

Parameter	Description
Matching Characteristics (Rule)	
Web/OVOC: Redirect Prefix **[RedirectPrefix]**	Defines the redirect telephone number prefix. To denote any number, use the wildcard asterisk (*) symbol.
Web/OVOC: Destination Prefix **[DestinationPrefix]**	Defines the destination (called) telephone number prefix. To denote any number, use the wildcard asterisk (*) symbol.
Web: Source Trunk Group ID **[SrcTrunkGroupID]**	Defines the Hunt Group from where the Tel call is received. To denote any Hunt Group, leave this field empty. **Notes:** ▪ This parameter is applicable only to the Redirect Number Tel > IP table. ▪ The value -1 indicates that this field is ignored in the rule.
Operation (Action)	
Web: Stripped Digits From Left OVOC: Remove From Left **[RemoveFromLeft]**	Defines the number of digits to remove from the left of the redirect number prefix. For example, if you enter 3 and the redirect number is 5551234, the new number is 1234.
Web: Stripped Digits From Right OVOC: Remove From Right **[RemoveFromRight]**	Defines the number of digits to remove from the right of the redirect number prefix. For example, if you enter 3 and the redirect number is 5551234, the new number is 5551.
Web/OVOC: Number of Digits to Leave **[LeaveFromRight]**	Defines the number of digits that you want to retain from the right of the redirect number.
Web/OVOC: Prefix to Add **[Prefix2Add]**	Defines the number or string that you want added to the front of the redirect number. For example, if you enter 9 and the redirect number is 1234, the new number is 91234.
Web/OVOC: Suffix to Add **[Suffix2Add]**	Defines the number or string that you want added to the end of the redirect number. For example, if you enter 00 and the redirect number is 1234, the new number is 123400.
Web: Presentation OVOC: Is Presentation Restricted **[IsPresentationRestricted]**	Enables caller ID. ▪ Not Configured = Privacy is determined according to the Caller ID table (see Configuring Caller Display Information on page 305). ▪ **[0]** Allowed = Sends Caller ID information when a call is made using these destination / source prefixes. ▪ **[1]** Restricted = Restricts Caller ID information for these prefixes. **Note:** If 'Presentation' is set to 'Restricted' and the AssertedIdMode parameter is set to **Add P-Asserted-Identity**, the From header in the INVITE message includes the following: From: 'anonymous' <sip: anonymous@anonymous.invalid> and 'privacy: id' header.

21.6 Mapping NPI/TON to SIP Phone-Context

The Phone-Context table page allows you to map Numbering Plan Indication (NPI) and Type of Number (TON) to the SIP 'phone-context' parameter. The 'phone-context' parameter appears in the standard SIP headers where a phone number is used (i.e., Request-URI, To, From, and Diversion). When a call is received from the Tel side, the NPI and TON are compared against the table and the matching 'phone-context' value is used in the outgoing SIP INVITE message. The same mapping occurs when an INVITE with a 'phone-context' parameter is received.

For example, for a Tel-to-IP call with NPI/TON set as E164 National (values 1/2), the device sends the following SIP INVITE URI:

```
sip:12365432;phone-context= na.e.164.nt.com
```

This is configured for entry 3 in the figure below. In the opposite direction (IP-to-Tel call), if the incoming INVITE contains this 'phone-context' (e.g. "phone-context= na.e.164.nt.com"), the NPI/TON of the called number in the outgoing Setup message is changed to E164 National.

➢ **To configure NPI/TON to SIP phone-context rules:**

1. Open the Phone Context Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Manipulations** > **Phone Context**).

Figure 21-4: Phone Context Table Page

	Add Phone Context As Prefix	Enable
	Phone Context Index	1-10

	NPI	TON	Phone Context
1	Unknown	Unknown	unknown.com
2	Private	Level 2 Regional	host.com
3	E.164 Public	National	na.e164.host.com
4			

2. Configure the parameters as required. For a description of the parameters, see the table below.

3. Click **Submit** to apply your changes.

4. To save the changes to flash memory, see 'Saving Configuration' on page 362.

> **Notes:**
>
> - You can configure multiple rows with the same NPI/TON or same SIP 'phone-context'. In such a configuration, a Tel-to-IP call uses the first matching rule in the table.
> - The Phone Context table can also be configured using the table ini file parameter, PhoneContext (see 'Number Manipulation Parameters' on page 628).

Table 21-5: Phone-Context Parameters Description

Parameter	Description
Add Phone Context As Prefix **[AddPhoneContextAsPrefix]**	Determines whether the received SIP 'phone-context' parameter is added as a prefix to the outgoing called and calling numbers. ▪ **[0]** Disable (default) ▪ **[1]** Enable
NPI **[PhoneContext_Npi]**	Defines the Number Plan Indicator (NPI). ▪ **[0]** Unknown (default) ▪ **[1]** E.164 Public ▪ **[9]** Private
TON **[PhoneContext_Ton]**	Defines the Type of Number (TON). ▪ If you selected Unknown as the NPI, you can select Unknown **[0]**. ▪ If you selected Private as the NPI, you can select one of the following: ✓ **[0]** Unknown ✓ **[1]** Level 2 Regional ✓ **[2]** Level 1 Regional ✓ **[3]** PSTN Specific ✓ **[4]** Level 0 Regional (Local) ▪ If you selected E.164 Public as the NPI, you can select one of the following: ✓ **[0]** Unknown ✓ **[1]** International ✓ **[2]** National ✓ **[3]** Network Specific ✓ **[4]** Subscriber ✓ **[6]** Abbreviated
Phone Context **[PhoneContext_Context]**	Defines the SIP 'phone-context' URI parameter.

22   Routing

This section describes the configuration of call routing rules.

22.1   Configuring General Routing Parameters

The Routing General Parameters page allows you to configure general routing parameters. For a description of these parameters, see 'Configuration Parameters Reference' on page 471.

➢   **To configure general routing parameters:**

1.   Open the Routing General Parameters page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Routing** submenu > **General Parameters**).

Figure 22-1: General Parameters

General Parameters	
Add Hunt Group ID as Prefix	No
Add Trunk ID as Prefix	No
Replace Empty Destination with B-channel Phone Number	No
Add NPI and TON to Called Number	No
Add NPI and TON to Calling Number	No
IP to Tel Remove Routing Table Prefix	No
Source IP Address Input	SIP Contact Header
Enable Alt Routing Tel to IP	Disable
Alt Routing Tel to IP Mode	Both
Alt Routing Tel to IP Connectivity Method	ICMP Ping
Alt Routing Tel to IP Keep Alive Time	60
Alternative Routing Tone Duration [ms]	0
Source Manipulation Mode	FROM & PAI (after manipulation)
Max Allowed Packet Loss for Alt Routing [%]	20
Max Allowed Delay for Alt Routing [msec]	250

2.   Configure the parameters as required.
3.   Click **Submit** to apply your changes.

22.2 Configuring Tel to IP Routing

The Tel to IP Routing page allows you to configure up to 50 Tel-to-IP call routing rules. The device uses these rules to route calls from the Tel to a user-defined IP destination.

The Tel to IP Routing table provides two configuration areas:

■ **Matching Characteristics:** Characteristics of the incoming call. If the call characteristics match a table entry, the routing rule is used to route the call to the specified destination. One or more characteristics can be defined for the rule:

- Source and destination Request-URI host name prefix
- Source Hunt Group (from where the call is received)
- Source (calling) and destination (called) telephone number prefix and suffix

■ **Destination:** If the call matches the configured characteristics, the device routes the call to an IP destination. If no characteristics match is found in the table, the call is rejected. The destination can be any of the following:

- IP address in dotted-decimal notation.
- Fully Qualified Domain Name (FQDN).
- E.164 Telephone Number Mapping (ENUM service).
- IP Group, where the call is routed to the IP address configured for the Proxy Set associated with the IP Group (configured in 'Configuring IP Groups' on page 197).

Notes: When using a proxy server, you do not need to configure this table, unless you require one of the following:

- Fallback (alternative) routing if communication is lost with the proxy server.
- IP security, whereby the device routes only received calls whose source IP addresses are defined in this table. IP security is enabled using the SecureCallsFromIP parameter.
- Filter Calls to IP feature: the device checks this table before a call is routed to the proxy server. However, if the number is not allowed, i.e., the number does not exist in the table or a Call Restriction (see below) routing rule is applied, the call is released.
- Obtain different SIP URI host names (per called number).
- Assign IP Profiles to calls.
- For this table to take precedence over a proxy for routing calls, you need to set the parameter PreferRouteTable to 1. The device checks the 'Destination IP Address' field in this table for a match with the outgoing call; a proxy is used only if a match is not found.

In addition to basic outbound IP routing, this table supports the following features:

■ **Least Cost Routing (LCR):** If the LCR feature is enabled, the device searches the routing table for matching routing rules and then selects the one with the lowest call cost. The call cost of the routing rule is done by assigning it a Cost Group. For configuring Cost Groups, see 'Least Cost Routing' on page 187. If two routing rules have identical costs, then the rule appearing higher up in the table (i.e., first-matched rule) is used. If a selected route is unavailable, the device uses the next least-cost routing rule. However, even if a matched rule is not assigned a Cost Group, the device can select it as the preferred route over other matched routing rules with Cost Groups, according to the settings of the LCR parameter, LCRDefaultCost (see 'Enabling LCR and Configuring Default LCR' on page 189).

- ■ **Call Forking:** If the Tel-to-IP Call Forking feature is enabled, the device can send a Tel call to multiple IP destinations. An incoming Tel call with multiple matched routing rules (e.g., all with the same source prefix numbers) can be sent (forked) to multiple IP destinations if the rules are defined with a Forking Group in the table. The call is established with the first IP destination that answers the call.

- ■ **Call Restriction:** Rejects calls whose matching routing rule is configured with the destination IP address of 0.0.0.0.

- ■ **Always Use Routing Table:** Even if a proxy server is used, the SIP Request-URI host name in the outgoing INVITE message is obtained from this table. Using this feature, you can assign a different SIP URI host name for different called and/or calling numbers. This feature is enabled using the AlwaysUseRouteTable parameter.

- ■ **IP Profiles:** IP Profiles can be assigned to destination addresses (also when a proxy is used).

- ■ **Alternative Routing (when a proxy isn't used):** An alternative IP destination can be configured for a specific call. To associate an alternative IP address to a called telephone number prefix, assign it with an additional entry with a different IP address, or use an FQDN that resolves into two IP addresses. For more information on alternative routing, see 'Alternative Routing for Tel-to-IP Calls' on page 262.

> **Notes:**
>
> - The maximum number of alternative routing rules that can be configured for each routing rule in the table is three.
> - Outbound IP routing can be performed before or after number manipulation. This is configured using the RouteModeTel2IP parameter, as described below.
> - The Tel to IP Routing can also be configured using the table *ini* file parameter, Prefix.

➢ **To configure Tel-to-IP routing rules:**

1. Open the Tel to IP Routing page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Routing** > **Tel to IP Routing**).

	Src. Hunt Group ID	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address	Port	Transport Type	Dest. IP Group ID	IP Profile ID	Status
			Routing Index			1-10 ▾				
			Tel To IP Routing Mode			Route calls before manipulation ▾				
1							Not Configured ▾	-1		Not Available
2							Not Configured ▾	-1		
3							Not Configured ▾	-1		
4							Not Configured ▾	-1		

2. From the 'Routing Index' drop-down list, select the range of entries that you want to add.
3. Configure the routing rule as required. For a description of the parameters, see the table below.
4. Click **Submit** to apply your changes.
5. To save the changes to flash memory, see 'Saving Configuration' on page 362.

The table below shows configuration examples of Tel-to-IP routing rules, where:

■ **Rule 1 and 2 (Least Cost Routing rule):** For both rules, the called (destination) phone number prefix is 10, the caller's (source) phone number prefix is 100, and the call is assigned IP Profile ID 1. However, Rule 1 is assigned a cheaper Cost Group than Rule 2, and therefore, the call is sent to the destination IP address (10.33.45.63) associated with Rule 1.

■ **Rule 3 (IP Group destination rule):** For all callers (*), if the called phone number prefix is 20, the call is sent to IP Group 1 (whose destination is the IP address configured for its associated Proxy Set ID).

■ **Rule 4 (domain name destination rule):** If the called phone number prefix is 5, 7, 8, or 9 and the caller belongs to Hunt Group ID 1, the call is sent to domain.com.

■ **Rule 5 (block rule):** For all callers (*), if the called phone number prefix is 00, the call is rejected (discarded).

■ **Rule 6, Rule 7, and Rule 8 (Forking Group rule):** For all callers (*), if the called phone number prefix is 100, the call is sent to Rule 7 and 9 (belonging to Forking Group "1"). If their destinations are unavailable and alternative routing is enabled, the call is sent to Rule 8 (Forking Group "2").

Table 22-1: Example of Tel-to-IP Source Phone Number Manipulation Rules

Parameter	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8
Src. Trunk Group ID	*	0	1	-	-	-	-	-
Src. Trunk Group ID	-	-	*	1	-	*	*	*
Dest. Phone Prefix	10	10	20	**[5,**7-9]	00	100	100	100
Source Phone Prefix	100	100	*	*	*	*	*	*
Dest. IP Address	10.33.45.63	10.33.45.50	-	domain.com	0.0.0.0	10.33.45.68	10.33.45.67	domain.com
Dest IP Group ID	-	-	1	-	-	-	-	-
IP Profile ID	1	1	-	-	-	-	-	-
Cost Group ID	Weekend	Weekend_B	-	-	-	-	-	-
Forking Group			-	-	-	1	2	1

Table 22-2: Tel-to-IP Routing Table Parameters

Parameter	Description
Matching Call Characteristics	
Web/OVOC: Tel to IP Routing Mode **[RouteModeTel2IP]**	Determines whether to route received calls to an IP destination before or after manipulation of the destination number. ▪ **[0]** Route calls before manipulation = Calls are routed before the number manipulation rules are applied (default). ▪ **[1]** Route calls after manipulation = Calls are routed after the number manipulation rules are applied. **Notes:** ▪ This parameter is not applicable if outbound proxy routing is used. ▪ For number manipulation, see 'Configuring Source/Destination Number Manipulation' on page 235.
Web: Src. Trunk Group ID OVOC: Source Trunk Group ID **[PREFIX_SrcTrunkGroupID]**	Defines the Hunt Group from where the call is received. **Note:** To denote any Hunt Group, use the asterisk (*) symbol.
Web: Dest. Phone Prefix OVOC: Destination Phone Prefix **[PREFIX_DestinationPrefix]**	Defines the prefix and/or suffix of the called (destination) telephone number. The suffix is enclosed in parenthesis after the suffix value. You can use special notations for denoting the prefix. For example, **[100-**199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. To denote any prefix, use the asterisk (*) symbol or to denote calls without a called number, use

Parameter	Description
	the $ sign. For a description of available notations, see 'Dialing Plan Notation for Routing and Manipulation Tables' on page 469. The number can include up to 50 digits.
Web/OVOC: Source Phone Prefix [PREFIX_SourcePrefix]	Defines the prefix and/or suffix of the calling (source) telephone number. You can use special notations for denoting the prefix. For example, **[100-**199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. To denote any prefix, use the asterisk (*) symbol or to denote calls without a calling number, use the $ sign. For a description of available notations, see 'Dialing Plan Notation for Routing and Manipulation Tables' on page 469. The number can include up to 50 digits.
Operation (IP Destination)	

Web: Dest. IP Address
OVOC: Address
[PREFIX_DestAddress]
	Defines the IP address (in dotted-decimal notation or FQDN) to where the call is sent. If an FQDN is used (e.g., domain.com), DNS resolution is done according to the DNSQueryType parameter.
	
	For ENUM-based routing, enter the string value "ENUM". The device sends an ENUM query containing the destination phone number to an external DNS server, configured in the Multiple Interface table. The ENUM reply includes a SIP URI which is used as the Request-URI in the subsequent outgoing INVITE and for routing (if a proxy is not used). To configure the type of ENUM service (e.g., e164.arpa), use the EnumService parameter.
	
	Notes:
	▪ This field and any value assigned to it is ignored if you have configured a destination IP Group for this routing rule (in the 'Dest IP Group ID' field).
	▪ To reject calls, enter the IP address 0.0.0.0. For example, if you want to prohibit international calls, then in the 'Dest Phone Prefix' field, enter 00 and in the 'Dest IP Address' field, enter 0.0.0.0.
	▪ For routing calls between phones connected to the device (i.e., local routing), enter the device's IP address.
	▪ When the device's IP address is unknown (e.g., when DHCP is used), enter IP address 127.0.0.1.
	▪ When using domain names, enter the DNS server's IP address or alternatively, configure these names in the Internal DNS table (see 'Configuring the Internal DNS Table' on page 134).
	▪ The IP address can include the following wildcards:
	  ✓ "x": represents single digits. For example, 10.8.8.xx denotes all addresses between 10.8.8.10 and 10.8.8.99.
	  ✓ "*": represents any number between 0 and 255. For example, 10.8.8.* denotes all addresses between 10.8.8.0 and 10.8.8.255.

Web: Port OVOC: Destination Port [PREFIX_DestPort]	Defines the destination port to where you want to route the call.
Web/OVOC: Transport Type [PREFIX_TransportType]	Defines the transport layer type for sending the IP call: ▪ **[-1]** Not Configured ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS

Parameter	Description
	Note: When set to Not Configured (-1), the transport type defined by the SIPTransportType parameter is used.
Web: Dest IP Group ID OVOC: Destination IP Group ID **[PREFIX_DestIPGroupID]**	Defines the IP Group to where you want to route the call. The SIP INVITE message is sent to the IP address defined for the Proxy Set ID associated with the IP Group. **Notes:** ▪ If you select an IP Group, you do not need to configure a destination IP address. However, if both parameters are configured in this table, the INVITE message is sent only to the IP Group (and not the defined IP address). ▪ If the parameter AlwaysUseRouteTable is set to 1 (see 'Configuring IP Groups' on page 197), then the Request-URI host name in the INVITE message is set to the value defined for the parameter 'Dest. IP Address' (above); otherwise, if no IP address is defined, it is set to the value of the parameter 'SIP Group Name' (defined in the IP Group table). ▪ This parameter is used as the 'Serving IP Group' in the Account table for acquiring authentication user/password for this call (see 'Configuring Account Table' on page 205). ▪ For defining Proxy Set ID's, see 'Configuring Proxy Sets Table' on page 200.
IP Profile ID **[PREFIX_ProfileId]**	Assigns an IP Profile ID to this IP destination call. This allows you to assign numerous configuration attributes (e.g., voice codes) per routing rule. To configure IP Profiles, see 'Configuring IP Profiles' on page 217.
Status	Displays the connectivity status of the routing rule's IP destination. If there is connectivity with the destination, this field displays "OK" and the device uses this routing rule if required. The routing rule is not used if any of the following is displayed: ▪ "n/a" = The destination IP Group is unavailable ▪ "No Connectivity" = No connection with the destination (no response to the ping or SIP OPTIONS). ▪ "QoS Low" = Poor Quality of Service (QoS) of the destination. ▪ "DNS Error" = No DNS resolution. This status is applicable only when a domain name is used (instead of an IP address). ▪ "Unavailable" = The destination is unreachable due to networking issues.
Web/OVOC: Charge Code **[PREFIX_MeteringCode]**	Assigns a Charge Code to the routing rule. To configure Charge Codes, see Configuring Charge Codes Table on page 300. **Note:** This parameter is applicable only to FXS interfaces.
Cost Group ID **[PREFIX_CostGroup]**	Assigns a Cost Group with the routing rule for determining the cost of the call. To configure Cost Groups, see 'Configuring Cost Groups' on page 191.

Parameter	Description
Forking Group **[PREFIX_ForkingGroup]**	Defines a forking group ID for the routing rule. This enables forking of incoming Tel calls to two or more IP destinations. The device sends simultaneous INVITE messages and handles multiple SIP dialogs until one of the calls is answered. When a call is answered, the other calls are dropped. If all matched routing rules belong to the same Forking Group number, the device sends an INVITE to all the destinations belonging to this group and according to the following logic: ▪ If matched routing rules belong to different Forking Groups, the device sends the call to the Forking Group of the first matched routing rule. If the call cannot be established with any of the destinations associated with this Forking Group and alternative routing is enabled, the device forks the call to the Forking Group of the next matched routing rules as long as the Forking Group is defined with a **higher** number than the previous Forking Group. For example: ▪ Table index entries 1 and 2 are defined with Forking Group "1", and index entries 3 and 4 with Forking Group "2": The device first sends the call according to index entries 1 and 2, and if unavailable and alternative routing is enabled, sends the call according to index entries 3 and 4. ▪ Table index entry 1 is defined with Forking Group "2", and index entries 2, 3, and 4 with Forking Group "1": The device sends the call according to index entry 1 only and ignores the other index entries even if the destination is unavailable and alternative routing is enabled. This is because the subsequent index entries are defined with a Forking Group number that is lower than that of index entry 1. ▪ Table index entry 1 is defined with Forking Group "1", index entry 2 with Forking Group "2", and index entries 3 and 4 with Forking Group "1": The device first sends the call according to index entries 1, 3, and 4 (all belonging to Forking Group "1"), and if the destination is unavailable and alternative routing is enabled, the device sends the call according to index entry 2. ▪ Table index entry 1 is defined with Forking Group "1", index entry 2 with Forking Group "3", index entry 3 with Forking Group "2", and index entry 4 with Forking Group "1": The device first sends the call according to index entries 1 and 4 (all belonging to Forking Group "1"), and if the destination is unavailable and alternative routing is enabled, the device sends the call according to index entry 2 (Forking Group "3"). Even if index entry 2 is unavailable and alternative routing is enabled, the device ignores index entry 3 because it belongs to a Forking Group that is lower than index entry 2. **Notes:** ▪ To enable Tel-to-IP call forking, set the 'Tel2IP Call Forking Mode' (*Tel2IPCallForkingMode*) parameter to **Enable**. ▪ When the UseDifferentRTPportAfterHold parameter is enabled, every forked call is sent with a different RTP port. Thus, ensure that the device has available RTP ports for these forked calls.

22.3 Configuring IP to Hunt Group Routing Table

The IP to Hunt Group Routing Table page allows you to configure up to 24 inbound call routing rules:

■ For IP-to-Tel routing: This table is used to route incoming IP calls to Hunt Groups. The specific channel pertaining to the Hunt Group to which the call is routed is determined according to the Hunt Group's channel selection mode. The channel selection mode can be defined per Hunt Group (see 'Configuring Hunt Group Settings' on page 229) or for all Hunt Groups using the global parameter ChannelSelectMode.

The IP to Hunt Group Routing Table provides two configuration areas:

■ Matching characteristics of incoming IP call, for example, prefix of destination number.

■ Operation (destination), for example, sends to a specific Hunt Group.

If the incoming call matches the characteristics of a rule, then the call is sent to the destination configured for that rule.

The device also supports alternative routing if the Hunt Group is unavailable:

■ If a call release reason is received for a specific IP-to-Tel call and this reason is configured for alternative IP-to-Tel routing, then the device re-routes the call to an alternative Hunt Group. The alternative route is configured in this table as an additional row (below the main routing rule) with the same call characteristics, but with a destination to a different Hunt Group. For more information on IP-to-Tel alternative routing, see 'Alternative Routing to Trunk upon Q.931 Call Release Cause Code' on page 265.

■ The device can re-route (i.e., call redirection) IP-to-Tel calls to an alternative IP destination using SIP 3xx responses. For more information, see 'Alternative Routing to IP Destinations upon Busy Trunk' on page 267.

The device automatically re-routes an IP-to-Tel call to a different physical FXO port if the initially destined FXO port within the same Hunt Group is detected as out of service (e.g., physically disconnected). When the physical FXO port is disconnected, the device sends the SNMP trap, GWAPP_TRAP_BUSYOUT_LINK notifying of the out-of-service state for the specific FXO line. When the FXO port is physically reconnected, this trap is sent notifying of the back-to-service state.

> **Note:** You can also configure the IP to Hunt Group Routing Table using the table ini file parameter, PSTNPrefix (see 'Number Manipulation Parameters' on page 628).

➢ **To configure IP-to-Tel routing rules:**

1. Open the IP to Hunt Group Routing Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Routing** > **IP to** Hunt **Group Routing**).

Figure 22-2: Inbound IP Routing Table Page

| | Routing Index | 1-12 | |
| | IP To Tel Routing Mode | Route calls before manipulation | |

	Dest. Host Prefix	Source Host Prefix	Dest. Phone Prefix	Source Phone Prefix	Source IP Address	>	Hunt Group ID	IP Profile ID	Source IPGroup ID
1			1x	*			1	2	-1
2			[501-502]	101			2	1	
3		domain.com	*	*			3		
4			*	*	10.13.64.5		3		

The previous figure displays the following configured routing rules:

- **Rule 1:** If the incoming IP call destination phone prefix is between 10 and 19, the call is assigned settings configured for IP Profile ID 2 and routed to Hunt Group ID 1.
- **Rule 2:** If the incoming IP call destination phone prefix is between 501 and 502 and source phone prefix is 101, the call is assigned settings configured for IP Profile ID 1 and routed to Hunt Group ID 2.
- **Rule 3:** If the incoming IP call has a From URI host prefix as domain.com, the call is routed to Hunt Group ID 3.

2. Configure the routing rule, as required. For a description of the parameters, see the table below.

3. Click **Submit** to apply your changes.

Table 22-3: IP-to-Tel Routing Table Description

Parameter	Description
IP to Tel Routing Mode **[RouteModeIP2Tel]**	Determines whether to route the incoming IP call before or after manipulation of destination number, configured in 'Configuring Source/Destination Number Manipulation' on page 235. ▪ **[0]** Route calls before manipulation = (Default) Incoming IP calls are routed before number manipulation. ▪ **[1]** Route calls after manipulation = Incoming IP calls are routed after number manipulation.
Matching Characteristics	
Web: Dest. Host Prefix **[DestPrefix]**	Defines the Request-URI host name prefix of the incoming SIP INVITE message. If this routing rule is not required, leave the field empty. **Note:** The asterisk (*) wildcard can be used to depict any prefix.
Web: Source Host Prefix **[SrcHostPrefix]**	Defines the From URI host name prefix of the incoming SIP INVITE message. If this routing rule is not required, leave the field empty. **Notes:** ▪ The asterisk (*) wildcard can be used to depict any prefix. ▪ If the P-Asserted-Identity header is present in the incoming INVITE message, then the value of this parameter is compared to the P-Asserted-Identity URI host name (and not the From header).
Web: Dest. Phone Prefix **[DestHostPrefix]**	Defines the prefix or suffix of the called (destined) telephone number. You can use special notations for denoting the prefix. For example, **[100-**199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. To denote any prefix, use the asterisk (*) symbol or to denote calls without a called number, use the $ sign. For a description of available notations, see 'Dialing Plan Notation for Routing and Manipulation Tables' on page 469. The prefix can include up to 49 digits.
Web: Source Phone Prefix **[SourcePrefix]**	Defines the prefix or suffix of the calling (source) telephone number. You can use special notations for denoting the prefix. For example, **[100-**199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. To denote any prefix, use the asterisk (*) symbol or to denote calls without a calling number, use the $ sign. For a description of available notations, see 'Dialing Plan Notation for Routing and Manipulation Tables' on page 469. The prefix can include up to 49 digits.

Parameter	Description
Web: Source IP Address **[SourceAddress]**	Defines the source IP address of the incoming IP call that can be used for routing decisions. The IP address must be configured in dotted-decimal notation (e.g., 10.8.8.5); not as an FQDN. **Notes:** ▪ The source IP address is obtained from the Contact header in the INVITE message. ▪ You can configure from where the source IP address is obtained, using the SourceIPAddressInput parameter. ▪ The source IP address can include the following wildcards: ✓ "x": denotes single digits. For example, 10.8.8.xx represents all the addresses between 10.8.8.10 and 10.8.8.99. ✓ "*": denotes any number between 0 and 255. For example, 10.8.8.* represents all addresses between 10.8.8.0 and 10.8.8.255.
Operation (Destination)	
Web: Hunt Group ID **[TrunkGroupId]**	Defines the Hunt Group to where the incoming SIP call is sent.
Web: IP Profile ID **[ProfileId]**	Assigns an IP Profile (configured in 'Configuring IP Profiles' on page 217) to the call.
Web: Source IP Group ID **[SrcIPGroupID]**	Defines the IP Group associated with the incoming IP call. This is the IP Group that sent the INVITE message. This IP Group can later be used as the 'Serving IP Group' in the Account table for obtaining authentication user name/password for this call (see 'Configuring Account Table' on page 205).

22.4   IP Destinations Connectivity Feature

The device can be configured to check the integrity of the connectivity to IP destinations of Tel-to-IP routing rules in the Outbound IP Routing table. The IP Connectivity feature can be used for the Alternative Routing feature, whereby the device attempts to re-route calls from unavailable Tel-to-IP routing destinations to available ones (see 'Alternative Routing Based on IP Connectivity' on page 262).

The device supports the following methods for checking the connectivity of IP destinations:

■ **Network Connectivity:** The device checks the network connectivity of the IP destination using one of the following methods configured by the 'Alt Routing Tel to IP Connectivity Method' parameter:

- Ping: The device periodically (every seven seconds) pings the IP destination.

- **SIP OPTIONS:** The device sends "keep-alive" SIP OPTIONS messages to the IP destination. If the device receives a SIP 200 OK in response, it considers the destination as available. If the destination does not respond to the OPTIONS message, then it is considered unavailable. You can configure the time interval for sending these OPTIONS messages, using the 'Alt Routing Tel to IP Keep Alive Time' parameter.

These parameters are configured in the Routing General Parameters page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Routing** > **General Parameters**), as shown below:

Figure 22-3: IP Connectivity Method in Routing General Parameters Page

Alt Routing Tel to IP Connectivity Method	SIP OPTIONS
Alt Routing Tel to IP Keep Alive Time	60

■ **Quality of Service (QoS):** You can enable the device to check the QoS of IP destinations. The device measures the QoS according to RTCP statistics of previously established calls with the IP destination. The RTCP includes packet delay (in milliseconds) and packet loss (in percentage). If these measured statistics exceed a user-defined threshold, the destination is considered unavailable. Note that if call statistics is not received within two minutes, the QoS data is reset. These thresholds are configured using the following parameters:

- 'Max Allowed Packet Loss for Alt Routing' (IPConnQoSMaxAllowedPL): defines the threshold value for packet loss after which the IP destination is considered unavailable.

- 'Max Allowed Delay for Alt Routing' (IPConnQoSMaxAllowedDelay): defines the threshold value for packet delay after which the IP destination is considered unavailable

These parameters are configured in the Routing General Parameters page, as shown below:

Figure 22-4: IP QoS Thresholds in Routing General Parameters Page

Max Allowed Packet Loss for Alt Routing [%]	20
Max Allowed Delay for Alt Routing [msec]	250

■ **DNS Resolution:** When a host name (FQDN) is used (instead of an IP address) for the IP destination, it is resolved into an IP address by a DNS server. The device checks network connectivity and QoS of the resolved IP address. If the DNS host name is unresolved, the device considers the connectivity of the IP destination as unavailable.

You can view the connectivity status of IP destinations in the following Web interface pages:

■ **Outbound IP Routing Table:** The connectivity status of the IP destination per routing rule is displayed in the 'Status' column. For more information, see 'Configuring Tel to IP Routing' on page 250.

■ **IP Connectivity:** This page displays a more informative connectivity status of the IP destinations used in Tel-to-IP routing rules in the Outbound IP Routing table. For viewing this page, see 'Viewing IP Connectivity' on page 417.

22.5 Alternative Routing for Tel-to-IP Calls

The device supports various alternative Tel-to-IP call routing methods, as described in this section.

22.5.1 Alternative Routing Based on IP Connectivity

You can configure the device to do alternative Tel-to-IP call routing based on IP connectivity. When the connectivity state of an IP destination is unavailable, the device attempts to re-route the Tel-to-IP call to an alternative IP destination. It does this by searching for the next call matching rule (e.g., phone number prefix) in the Outbound IP Routing table.

> **Notes:**
>
> - Alternative routing based on IP connectivity is applicable only when a proxy server is not used.
> - As the device searches the Outbound IP Routing table for a matching rule starting from the top, you must configure the main routing rule above the alternative routing rules.
> - The maximum number of alternative routing rules that can be configured for each routing rule in the table is three.
> - For configuring Tel-to-IP routing rules in the Outbound IP Routing table, see 'Configuring Tel to IP Routing' on page 250.

The device searches for an alternative IP destination when any of the following connectivity states are detected with the IP destination of the initial Tel-to-IP routing rule:

- No response received from a ping or from SIP OPTIONS messages. This depends on the chosen method for checking IP connectivity.
- Poor QoS according to the configured thresholds for packet loss and delay.
- Unresolved DNS, if the configured IP destination is a domain name (or FQDN). If the domain name is resolved into two IP addresses, the timeout for INVITE re-transmissions can be configured using the HotSwapRtx parameter. For example, if you set this parameter to 3, the device attempts up to three times to route the call to the first IP address and if unsuccessful, it attempts up to three times to re-route it to the second resolved IP address.

The connectivity status of the IP destination is displayed in the 'Status' column of the Outbound IP Routing table per routing rule. If it displays a status other than "ok", then the device considers the IP destination as unavailable and attempts to re-route the call to an alternative destination. For more information on the IP connectivity methods and on viewing IP connectivity status, see 'IP Destinations Connectivity Feature' on page 260.

The table below shows an example of alternative routing where the device uses an available alternative routing rule in the Outbound IP Routing table to re-route the initial Tel-to-IP call.

Table 22-4: Alternative Routing based on IP Connectivity Example

	Destination Phone Prefix	IP Destination	IP Connectivity Status	Rule Used?
Main Route	40	10.33.45.68	"No Connectivity"	No
Alternative Route #1	40	10.33.45.70	"QoS Low"	No
Alternative Route #2	40	10.33.45.72	"ok"	Yes

The steps for configuring alternative Tel-to-IP routing based on IP connectivity are summarized below.

➢ **To configure alternative Tel-to-IP routing based on IP connectivity:**

1. In the Outbound IP Routing table, add alternative Tel-to-IP routing rules for specific calls.

2. In the Routing General Parameters page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Routing** > **General Parameters**), do the following:

a. Enable alternative routing based on IP connectivity, by setting the 'Enable Alt Routing Tel to IP' (AltRoutingTel2IPEnable) parameter to **Enable**.

b. Configure the IP connectivity reason for triggering alternative routing, by setting the 'Alt Routing Tel to IP Mode' parameter (AltRoutingTel2IPMode) to one of the following:

♦ Ping or SIP OPTIONS failure

♦ Poor QoS

♦ Ping or SIP OPTIONS failure, poor QoS, or unresolved DNS

c. The device plays a tone to the Tel endpoint (for analog interfaces) whenever an alternative route is used. This tone is played for a user-defined time configured by the 'Alternative Routing Tone Duration' parameter.

22.5.2 Alternative Routing Based on SIP Responses

You can configure the device to do alternative routing based on the received SIP response. If the SIP response code reflects an error (i.e., 4xx, 5xx, or 6xx) and you have configured this specific response code as a trigger for alternative routing, then the device attempts to re-route the call to an alternative destination.

You can configure up to five SIP response codes for triggering alternative routing. This is done in the Reasons for Alternative Routing table, explained in this section.

Typically, the device performs alternative routing when there is no response at all to an INVITE message after a user-defined number of INVITE re-transmissions, configured using the SIPMaxRtx parameter. In such a scenario, the device issues itself the SIP response code 408 "Request Timeout". If this release code is defined in the Reasons for Alternative Routing table, then alternative routing is done.

> **Note:** The device also plays a tone to the endpoint whenever an alternative route is used. This tone is played for a user-defined time, configured by the AltRoutingToneDuration parameter.

Depending on configuration, the alternative routing is done using one of the following configuration entities:

■ **Outbound IP Routing Rules:** You can configure up to two alternative routing rules in the table. If the initial, main routing rule destination is unavailable, the device searches the table (starting from the top) for the next call matching rule (e.g., destination phone number), and if available attempts to re-route the call to the IP destination configured for this alternative routing rule. The table below shows an example of alternative routing where the device uses the first available alternative routing rule to re-route the initial, unsuccessful Tel-to-IP call destination.

Table 22-5: Alternative Routing based on SIP Response Code Example

	Destination Phone Prefix	IP Destination	SIP Response	Rule Used?
Main Route	40	10.33.45.68	408 Request Timeout	No
Alternative Route #1	40	10.33.45.70	486 Busy Here	No
Alternative Route #2	40	10.33.45.72	200 OK	Yes

■ **Proxy Sets:** Proxy Sets are used for Server-type IP Groups (e.g., an IP PBX) and define the actual IP destination (IP address or FQDN) of the server. As you can define up to five IP destinations per Proxy Set, the device supports proxy redundancy, which works together with the alternative routing feature. If the destination of a routing rule in the Outbound IP Routing table is an IP Group, the device routes the call to the IP destination configured for the Proxy Set associated with the IP Group. If the first IP destination of the Proxy Set is unavailable, the device attempts to re-route the call to the next proxy destination, and so on until an available IP destination is located. To enable the Proxy Redundancy feature, set the IsProxyHotSwap parameter to 1 (per Proxy Set) and set the EnableProxyKeepAlive to 1.

When the Proxy Redundancy feature is enabled, the device continually monitors the connection with the proxies by using keep-alive messages (SIP OPTIONS). The device sends these messages every user-defined interval (ProxyKeepAliveTime parameter). Any response from the proxy, either success (200 OK) or failure (4xx response) is considered as if the proxy is communicating. If there is no response from the first (primary) proxy after a user-defined number of re-transmissions (re-INVITEs) configured using the HotSwapRtx parameter, the device attempts to communicate (using the same INVITE) with the next configured (redundant) proxy in the list, and so on until an available redundant proxy is located. The device's behavior can then be one of the following, depending on the ProxyRedundancyMode parameter setting:

• The device continues operating with the redundant proxy (now active) until the next failure occurs, after which it switches to the next redundant proxy. This is referred to as *Parking* mode.

• The device always attempts to operate with the primary proxy. In other words, it switches back to the primary proxy whenever it's available again. This is referred to as *Homing* mode.

If none of the proxy servers respond, the device goes over the list again.

The steps for configuring alternative Tel-to-IP routing based on SIP response codes are summarized below.

➢ **To configure alternative Tel-to-IP routing based on SIP response codes:**

1. Enable alternative routing based on SIP responses, by setting the 'Redundant Routing Mode' parameter to one of the following:

   • **Routing Table** for using the Outbound IP Routing table for alternative routing.

   • **Proxy** for using the Proxy Set redundancy feature for alternative routing.

2. If you are using the Outbound IP Routing table, configure alternative routing rules with identical call matching characteristics, but with different IP destinations. If you are using the Proxy Set, configure redundant proxies.

3. Define SIP response codes (call failure reasons) that invoke alternative Tel-to-IP routing:

a. Open the Reasons for Alternative Routing page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Routing** submenu > **Alternative Routing Reasons**).

Figure 22-5: Tel to IP Reasons - Reasons for Alternative Routing Page

Tel to IP Reasons	
Reason 1	▼
Reason 2	▼
Reason 3	▼
Reason 4	▼
Reason 5	▼

b. Under the 'Tel to IP Reasons' group, select up to five different SIP response codes (call failure reasons) that invoke alternative Tel-to-IP routing.

c. Click **Submit**.

22.6   Alternative Routing for IP-to-Tel Calls

The device supports alternative IP-to-Tel call routing, as described in this section.

22.6.1   Alternative Routing to Trunk upon Q.931 Call Release Cause Code

You can configure the device to do alternative IP-to-Tel call routing based on the received ISDN Q.931 cause code. If an IP-to-Tel call is rejected or disconnected on the Tel side as a result of a specific ISDN Q.931 release cause code that is listed in the Reasons for Alternative Routing table, the device searches for an alternative IP-to-Tel routing rule in the Inbound IP Routing table and sends it to the alternative Hunt Group. For example, you can enable alternative IP-to-Tel routing for scenarios where the initial Tel destination is busy and a Q.931 Cause Code No. 17 is received (or for other call releases that issue the default Cause Code No. 3).

You can also configure a default release cause code that the device issues itself upon the following scenarios:

- The device initiates a call release whose cause is unknown.
- No free channels (i.e., busy) in the Hunt Group.
- No appropriate routing rule located in the Inbound IP Routing table to the Hunt Group.
- Phone number is not found in the Inbound IP Routing table.

By default, it is set to Cause Code No. 3 (No Route to Destination). This default cause code can be changed using the 'Default Release Cause' parameter located in the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**). To enable alternative routing based on Q.931 cause code, you need to define this cause code in the Reasons for Alternative Routing table.

➢ **To configure alternative Hunt Group routing based on Q.931 cause codes:**

1. In the Proxy & Registration page, set the 'Redundant Routing Mode' parameter to **Routing Table** so that the device uses the Inbound IP Routing table for alternative routing.

2. In the Inbound IP Routing table, configure alternative routing rules with the same call matching characteristics, but with different Hunt Group destinations.

3. Configure up to five Q.931 cause codes that invoke alternative IP-to-Tel routing:

   a. Open the Reasons for Alternative Routing page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Routing** > **Alternative Routing Reasons**).

Figure 22-6: IP to Tel Reasons - Reasons for Alternative Routing Page

IP to Tel Reasons	
Reason 1	3
Reason 2	17
Reason 3	
Reason 4	
Reason 5	

   b. Under the 'IP to Tel Reasons' group, select the desired Q.931 cause codes.

   c. Click **Submit** to apply your changes.

Notes:

- You can configure up to two alternative routing rules in the Inbound IP Routing table.

- The default release cause is described in the Q.931 notation and is translated to corresponding SIP 40x or 50x values (e.g., Cause Code No. 3 to SIP 404, and Cause Code No. 34 to SIP 503).

- For information on mapping PSTN release causes to SIP responses, see PSTN Release Cause to SIP Response Mapping.

- For configuring IP-to-Tel routing rules in the Inbound IP Routing table, see 'Configuring IP to Hunt Group Routing Table' on page 257.

- The Reasons for Alternative Routing IP to Tel table can also be configured using the table ini file parameter, AltRouteCauseIP2Tel.

22.6.2 Alternative Routing to an IP Destination upon a Busy Trunk

You can configure the device to forward (i.e., call redirection) IP-to-Tel calls to an alternative IP destination using SIP 3xx responses. This can be done upon the following scenario:

■ Unavailable FXS / FXO Hunt Group. This feature can be used, for example, to forward the call to another FXS / FXO device.

This feature is configured per Hunt Group and is configured in the Forward on Busy Trunk Destination table, as described in this section.

The alternative destination can be defined as a host name or as a SIP Request-URI user name and host part (i.e., user@host). For example, the below configuration forwards IP-to-Tel calls to destination user "112" at host IP address 10.13.4.12, port 5060, using transport protocol TCP, if Trunk Group ID 2 is unavailable:

```
ForwardOnBusyTrunkDest 1 = 2, 112@10.13.4.12:5060;transport=tcp;
```

When configured with user@host, the original destination number is replaced by the user part.

The device forwards calls using this table only if no alternative IP-to-Tel routing rule has been configured in the Inbound IP Routing table or alternative routing fails and the following reason in the SIP Diversion header of 3xx messages exists:

■ "unavailable": All FXS / FXO lines pertaining to a Hunt Group are busy or unavailable

> **Note:**
>
> • This feature is not applicable to Trunk Groups whose 'Channel Select Mode' parameter is configured to **By Dest Phone Number** in the Hunt Group Settings table (see Configuring Hunt Group Settings on page 229).
>
> • You can also configure the Forward on Busy Trunk Destination table using the table ini file parameter, ForwardOnBusyTrunkDest.

➢ **To configure Forward on Busy Trunk Destination rules:**

1. Open the Forward on Busy Trunk Destination page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Routing** > **Forward on Busy Trunk**).

Figure 22-7: Forward on Busy Trunk Destination Page

Index	Trunk Group ID	Forward Destination
0 ○	1	10.13.5.67

The figure above displays a configuration that forwards IP-to-Tel calls destined for Hunt Group ID 1 to destination IP address 10.13.5.67 if the conditions mentioned earlier exist.

2. Configure the table as required, and then click **Submit** to apply your changes.

3. Save the changes to the device's flash memory with a device reset (see 'Saving Configuration' on page 362).

Table 22-6: Forward on Busy Trunk Destination Description Parameters

Parameter	Description
Trunk Group ID **[ForwardOnBusyTrunkDest_TrunkGroupId]**	Defines the Trunk Group ID to which the IP call is destined to.

Parameter	Description
Forward Destination **[ForwardOnBusyTrunkDest_ForwardDestination]**	Defines the alternative IP destination for the call used if the Trunk Group is busy or unavailable. The valid value can be an IP address in dotted-decimal notation, an FQDN, or a SIP Request-URI user name and host part (i.e., user@host). The following syntax can also be used: host:port;transport=xxx (i.e., IP address, port and transport type). **Note:** When configured with a user@host, the original destination number is replaced by the user part.

23   Configuring DTMF and Dialing

The DTMF & Dialing page is used to configure parameters associated with dual-tone multi-frequency (DTMF) and dialing. For a description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 471.

➢ **To configure the DTMF and dialing parameters:**

1. Open the DTMF & Dialing page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **DTMF & Supplementary** submenu > **DTMF & Dialing**).

Figure 23-1: DTMF & Dialing

Max Digits In Phone Num	5
Inter Digit Timeout [sec]	4
Declare RFC 2833 in SDP	Yes
1st Tx DTMF Option	RFC 2833
2nd Tx DTMF Option	
RFC 2833 Payload Type	96
Hook-Flash Option	Not Supported
Digit Mapping Rules	
Dial Plan Index	-1
Dial Tone Duration [sec]	16
Hotline Dial Tone Duration [sec]	16
Enable Special Digits	Disable
Default Destination Number	1000
Special Digit Representation	Special

2. Configure the parameters as required.
3. Click **Submit** to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 362.

23.1   Dialing Plan Features

This section describes various dialing plan features supported by the device.

23.1.1   Digit Mapping

The device collects digits until a match is found in the user-defined digit pattern (e.g., for closed numbering schemes). The device stops collecting digits and starts sending the digits (collected number) when any one of the following scenarios occur:

■ Maximum number of digits is received. You can define (using the MaxDigits parameter) the maximum number of collected destination number digits that can be received (i.e., dialed) from the Tel side by the device. When the number of collected digits reaches the maximum (or a digit map pattern is matched), the device uses these digits for the called destination number.

■ Inter-digit timeout expires (e.g., for open numbering schemes). This is defined using the TimeBetweenDigits parameter. This is the time that the device waits between each received digit. When this inter-digit timeout expires, the device uses the collected digits to dial the called destination number.

■ The phone's pound (#) key is pressed.

■ Digit string (i.e., dialed number) matches one of the patterns defined in the digit map.

Digit map (pattern) rules are defined using the DigitMapping parameter. The digit map pattern can contain up to 52 options (rules), each separated by a vertical bar ("|"). The maximum length of the entire digit pattern is 152 characters. The available notations are described in the table below:

Table 23-1: Digit Map Pattern Notations

Notation	Description
[n-m]	Range of numbers (not letters).
.	(single dot) Repeat digits until next notation (e.g., T).
x	Any single digit. **Note:** This notation does not apply in some scenarios when using the star (*) or hash (#) key. For example, the key sequence of ** must be presented in the dial plan as *x.s (instead of xx).
T	Dial timeout (configured by the TimeBetweenDigits parameter).
S	Short timer (configured by the TimeBetweenDigits parameter; default is two seconds) that can be used when a specific rule is defined after a more general rule. For example, if the digit map is 99|998, then the digit collection is terminated after the first two 9 digits are received. Therefore, the second rule of 998 can never be matched. But when the digit map is 99s|998, then after dialing the first two 9 digits, the device waits another two seconds within which the caller can enter the digit 8.

Below is an example of a digit map pattern containing eight rules:

```
DigitMapping = 11xS|00[1-
7]xxx|8xxxxxxx|#xxxxxxx|*xx|91xxxxxxxxxx|9011x|xx.T
```

In the example, the rule "00**[1-7]**xxx" denotes dialed numbers that begin with 00, and then any digit from 1 through 7, followed by three digits (of any number). Once the device receives these digits, it does not wait for additional digits, but starts sending the collected digits (dialed number) immediately.

> **Notes:**
>
> - If you want the device to accept/dial any number, ensure that the digit map contains the rule "xx.T"; otherwise, dialed numbers not defined in the digit map are rejected.
>
> - If you are using an external Dial Plan file for dialing plans (see 'Dialing Plans for Digit Collection' on page 372), the device first attempts to locate a matching digit pattern in the Dial Plan file, and if not found, then attempts to locate a matching digit pattern in the Digit Map (configured by the DigitMapping parameter).
>
> - It may be useful to configure both Dial Plan file and Digit Maps. For example, the Digit Map can be used for complex digit patterns (which are not supported by the Dial Plan) and the Dial Plan can be used for long lists of relatively simple digit patterns. In addition, as timeout between digits is not supported by the Dial Plan, the Digit Map can be used to define digit patterns (MaxDigits parameter) that are shorter than those defined in the Dial Plan, or left at default. For example, "xx.T" Digit Map instructs the device to use the Dial Plan and if no matching digit pattern, it waits for two more digits and then after a timeout (TimeBetweenDigits parameter), it sends the collected digits. Therefore, this ensures that calls are not rejected as a result of their digit pattern not been completed in the Dial Plan.

23.1.2 External Dial Plan File

The device can be loaded with a Dial Plan file with user-defined dialing plans. For more information, see 'Dial Plan File' on page 372.

This page is intentionally left blank.

24 Configuring Supplementary Services

This section describes SIP supplementary services that can enhance your telephone service.

> **Notes:**
>
> - All call participants must support the specific supplementary service that is used.
> - When working with certain application servers (such as BroadSoft's BroadWorks) in client server mode (the application server controls all supplementary services and keypad features by itself), the device's supplementary services must be disabled.

The Supplementary Services page is used to configure many of the discussed supplementary services parameters. For a description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page .

➤ **To configure supplementary services parameters:**

1. Open the Supplementary Services page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **DTMF & Supplementary** submenu > **Supplementary Services**).

Figure 24-1: Supplementary Services

2. Configure the parameters as required.

3. Click **Submit** to apply your changes, or click the **Subscribe to MWI** or **Unsubscribe to MWI** buttons to save your changes and to subscribe / unsubscribe to the MWI server.

4. To save the changes to flash memory, see 'Saving Configuration' on page 362.

24.1   Call Hold and Retrieve

Initiating Call Hold and Retrieve:

■   Active calls can be put on-hold by pressing the phone's hook-flash button.

■   The party that initiates the hold is called the *holding* party; the other party is called the *held* party.

■   After a successful Hold, the holding party hears a dial tone (HELD_TONE defined in the device's Call Progress Tones file).

■   Call retrieve can be performed only by the holding party while the call is held and active.

■   The holding party performs the retrieve by pressing the telephone's hook-flash button.

■   After a successful retrieve, the voice is connected again.

■   Hold is performed by sending a Re-INVITE message with IP address 0.0.0.0 or a=sendonly in the SDP according to the parameter HoldFormat.

Receiving Hold/Retrieve:

■   When an active call receives a re-INVITE message with either the IP address 0.0.0.0 or the 'inactive' string in SDP, the device stops sending RTP and plays a local held tone.

■   When an active call receives a re-INVITE message with the 'sendonly' string in SDP, the device stops sending RTP and listens to the remote party. In this mode, it is expected that on-hold music (or any other hold tone) is played (over IP) by the remote party.

You can also configure the device to keep a call on-hold for a user-defined time after which the call is disconnected, using the HeldTimeout parameter.

The device also supports "double call hold" for FXS interfaces where the called party, which has been placed on-hold by the calling party, can then place the calling party on hold as well

and make a call to another destination. The flowchart below provides an example of this type of call hold:

Figure 24-2: Double Hold SIP Call Flow
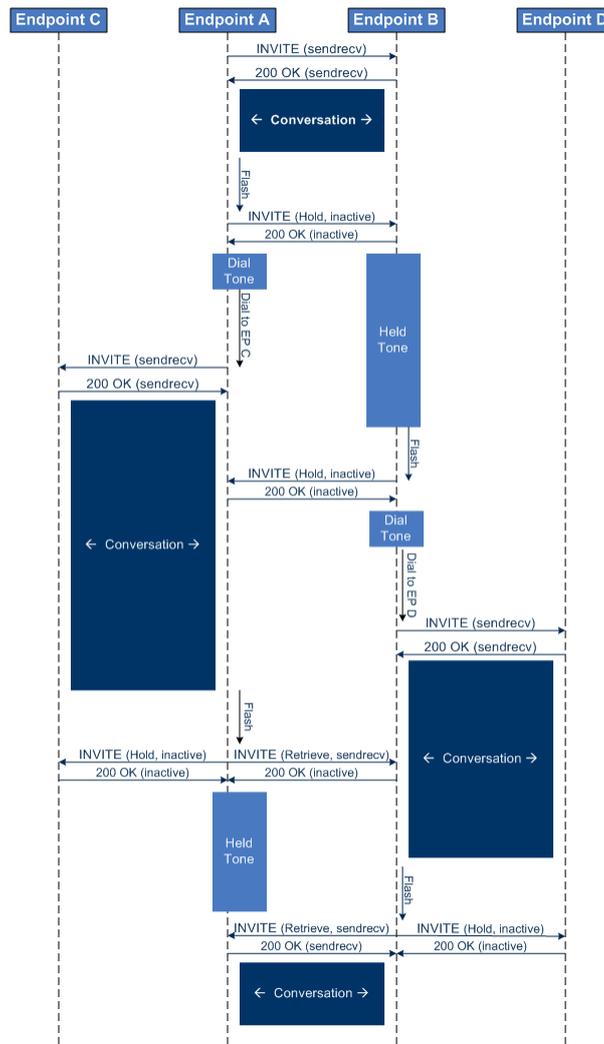
The flowchart above describes the following "double" call-hold scenario:

1. A calls B and establishes a voice path.
2. A places B on hold; A hears a dial tone and B hears a held tone.
3. A calls C and establishes a voice path.
4. B places A on hold; B hears a dial tone.
5. B calls D and establishes a voice path.
6. A ends call with C; A hears a held tone.
7. B ends call with D.
8. B retrieves call with A.

Notes:

- If a party that is placed on hold (e.g., B in the above example) is called by another party (e.g., D), then the on-hold party receives a call waiting tone instead of the held tone.

- While in a Double Hold state, placing the phone on-hook disconnects both calls (i.e. call transfer is not performed).

- You can enable the device to handle incoming re-INVITE messages with "a=sendonly" in the SDP, in the same way as if "a=inactive" is received in the SDP. This is configured using the SIPHoldBehavior parameter. When enabled, the device plays a held tone to the Tel phone and responds with a 200 OK containing "a=recvonly" in the SDP.

24.2   Call Pickup

The device supports the Call Pick-Up feature, whereby the FXS user can answer someone else's telephone call by pressing a user-defined sequence of phone keys. When the user dials the user-defined digits (e.g., #77), the incoming call from the other phone is forwarded to the FXS user's phone. This feature is configured using the parameter KeyCallPickup.

Note: The Call Pick-Up feature is supported only for FXS endpoints pertaining to the same Hunt Group ID.

24.3   Consultation Feature

The device's Consultation feature allows you to place one number on hold and make a second call to another party.

- After holding a call (by pressing hook-flash), the holding party hears a dial tone and can then initiate a new call, which is called a Consultation call.

- While hearing a dial tone, or when dialing to the new destination (before dialing is complete), the user can retrieve the held call by pressing hook-flash.

- The held call can't be retrieved while ringback tone is heard.

- After the Consultation call is connected, the user can toggle between the held and active call by pressing the hook-flash key.

Note: The Consultation feature is applicable only to FXS interfaces.

24.4   Call Transfer

This section describes the device's support for call transfer types.

24.4.1   Consultation Call Transfer

The device supports Consultation Call Transfer using the SIP REFER message and Replaces header. The common method to perform a consultation transfer is described in the following example, which assumes three call parties:

■   Party A = transferring

■   Party B = transferred

■   Party C = transferred to

1.   A Calls B.

2.   B answers.

3.   A presses the hook-flash button and places B on-hold (party B hears a hold tone).

4.   A dials C.

5.   After A completes dialing C, A can perform the transfer by on-hooking the A phone.

6.   After the transfer is complete, B and C parties are engaged in a call.

The transfer can be initiated at any of the following stages of the call between A and C:

■   Just after completing dialing C phone number - transfer from setup

■   While hearing ringback – transfer from alert

■   While speaking to C - transfer from active

24.4.2   Blind Call Transfer

Blind call transfer is done (using SIP REFER messages) after a call is established between call parties A and B, and party A decides to immediately transfer the call to C without first speaking to C. The result of the transfer is a call between B and C (like consultation transfer, but skipping the consultation stage).

The device supports two different blind transfer methods:

■   **Blind Transfer upon On-hook:** When a call is established (e.g., between A and B), A transfers the call to C, by pressing the phone's hook-flash button, dialing the second number (of C) without a "#" at the end, and then on-hooking the phone within a user-defined interval (configured by TimeBetweenDigits) or before a match is found by DigitMapping. The device sends a SIP REFER message to establish the blind transfer call. If the user on-hooks the phone after the timeout expires, a new second call is established (INVITE message sent). The feature is enabled by the BlindTransferOnOnHook parameter.

■   **Blind Transfer using Key Sequence:** When a call is established (e.g., between A and B), A transfers the call to C, by dialing the keypad sequence (configured by the KeyBlindTransfer parameter), followed by the transferee's destination number. Once the key sequence is dialed, the current call is put on hold (using a re-INVITE message), a dial tone is played, and then the phone number collection starts. After the destination phone number is collected by the device, the device sends it to the transferee in a SIP REFER request in the Refer-To header.

24.5   Call Forward

The following methods of call forwarding are supported:

■   Immediate: incoming call is forwarded immediately and unconditionally.

- Busy: incoming call is forwarded if the endpoint is busy.
- No Reply: incoming call is forwarded if it isn't answered for a specified time.
- On Busy or No Reply: incoming call is forwarded if the port is busy or when calls are not answered after a specified time.
- Do Not Disturb: immediately reject incoming calls. Upon receiving a call for a Do Not Disturb, the 603 Decline SIP response code is sent.

Three forms of forwarding parties are available:

- Served party: party configured to forward the call (FXS device).
- Originating party: party that initiates the first call (FXS or FXO device).
- Diverted party: new destination of the forwarded call (FXS or FXO device).

The served party (FXS interface) can be configured through the Web interface (see Configuring Call Forward on page 307) or ini file to activate one of the call forward modes. These modes are configurable per endpoint.

> **Notes:**
>
> - When call forward is initiated, the device sends a SIP 302 response with a contact that contains the phone number from the forward table and its corresponding IP address from the routing table (or when a proxy is used, the proxy's IP address).
> - For receiving call forward, the device handles SIP 3xx responses for redirecting calls with a new contact.

24.5.1 Call Forward Reminder Ring

The device supports the Call Forward Reminder Ring feature for FXS interfaces, whereby the device's FXS endpoint emits a short ring burst, only in **onhook** state, when a third-party Application Server (e.g., softswitch) forwards an incoming call to another destination. This is important in that it notifies (audibly) the FXS endpoint user that a call forwarding service is currently being performed.

Figure 24-3: Call Forward Reminder with Application Server
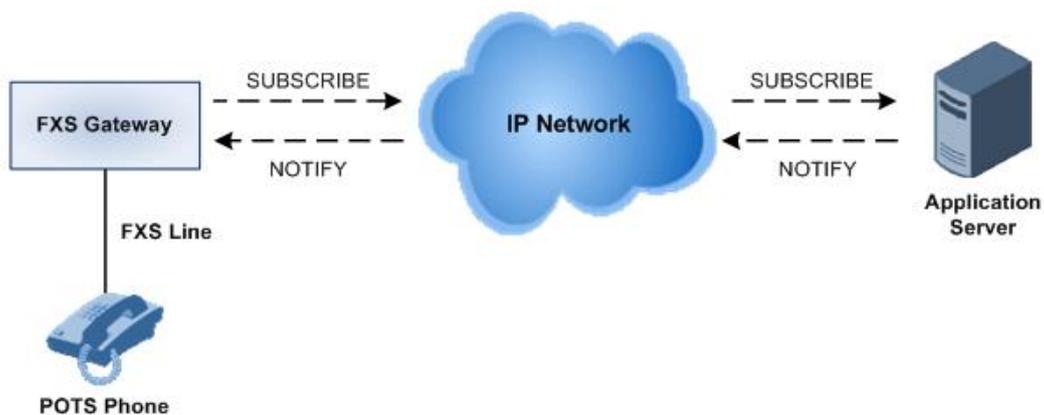
The device generates a Call Forward Reminder ring burst to the FXS endpoint each time it receives a SIP NOTIFY message with a "reminder ring" xml body. The NOTIFY request is sent from the Application Server to the device each time the Application Server forwards an incoming call. The service is cancelled when an UNSUBSCRIBE request is sent from the device, or when the Subscription time expires.

The reminder-ring tone can be defined by using the parameter CallForwardRingToneID, which points to a ring tone defined in the Call Progress Tone file.

The following parameters are used to configure this feature:

- EnableNRTSubscription
- ASSubscribeIPGroupID
- NRTSubscribeRetryTime
- CallForwardRingToneID

24.5.2   Call Forward Reminder (Off-Hook) Special Dial Tone

The device plays a special dial tone (stutter dial tone - Tone Type #15) to a specific FXS endpoint when the phone is off-hooked and when a third-party Application server (AS), e.g., a softswitch is used to forward calls intended for the endpoint, to another destination. This is useful in that it reminds the FXS user of this service. This feature does not involve device subscription (SIP SUBSCRIBE) to the AS.

Activation/deactivation of the service is notified by the server. An unsolicited SIP NOTIFY request is sent from the AS to the device when the Call Forward service is activated or de-activated. Depending on this NOTIFY request, the device plays either the standard dial tone or the special dial tone for Call Forward.

For playing the special dial tone, the received SIP NOTIFY message must contain the following headers:

- **From and To:** contain the same information, indicating the specific endpoint
- **Event:** ua-profile
- **Content-Type:** "application/simservs+xml"
- Message body is the XML body and contains the "dial-tone-pattern" set to "special-condition-tone" (<ss:dial-tone-pattern>special-condition-tone</ss:dial-tone-pattern>), which is the special tone indication.

To cancel the special dial tone and playing the regular dial tone, the received SIP NOTIFY message must contain the following headers:

- **From and To:** contain the same information, indicating the specific endpoint
- **Event:** ua-profile
- **Content-Type:** "application/simservs+xml"
- Message body is the XML body containing the "dial-tone-pattern" set to "standard-condition-tone" (<ss:dial-tone-pattern>standard-condition-tone</ss:dial-tone-pattern>), which is the regular dial tone indication.

Therefore, the special dial tone is valid until another SIP NOTIFY is received that instructs otherwise (as described above).

> **Note:** if the MWI service is active, the MWI dial tone overrides this special Call Forward dial tone.

24.5.3   Call Forward Reminder Dial Tone (Off-Hook) upon Spanish SIP Alert-Info

The device plays a special dial tone to FXS phones in off-hook state that are activated with the call forwarding service. The special dial tone is used as a result of the device receiving a SIP NOTIFY message from a third-party softswitch providing the call forwarding service with the following SIP Alert-Info header:

```
Alert-Info: <http://127.0.0.1/Tono-Espec-Invitacion>;lpi-
aviso=Desvio-Inmediato
```

This special tone is a stutter dial tone (Tone Type = 15), as defined in the CPT file.

The FXS phone user, connected to the device, activates the call forwarding service by dialing a special number (e.g., *21*xxxxx) and as a result, the device sends a regular SIP INVITE message to the softswitch. The softswitch later notifies of the activation of the forwarding service by sending an unsolicited NOTIFY message with the Alert-Info header, as mentioned above.

When the call forwarding service is de-activated, for example, by dialing #21# and sending an INVITE with this number, the softswitch sends another SIP NOTIFY message with the following Alert-Info header:

```
Alert-Info: <http://127.0.0.1/ Tono-Normal-Invitacion>; Aviso =
Desvi¶-Inmediato
```

From this point on, the device plays a normal dial tone to the FXS phone when it goes off-hook.

24.6   Call Waiting

The Call Waiting feature enables FXS devices to accept an additional (second) call on busy endpoints. If an incoming IP call is designated to a busy port, the called party hears a call waiting tone (several configurable short beeps) and (for Bellcore and ETSI Caller IDs) can view the Caller ID string of the incoming call. The calling party hears a call waiting ringback tone. The called party can accept the new call using hook-flash, and can toggle between the two calls.

➢ **To enable call waiting:**

1. Set the parameter EnableCallWaiting to 1.
2. Set the parameter EnableHold to 1.
3. Define the Call Waiting indication and call waiting ringback tones in the Call Progress Tones file. You can define up to four call waiting indication tones (refer to the FirstCallWaitingToneID parameter).
4. To configure the call waiting indication tone cadence, modify the following parameters: NumberOfWaitingIndications, WaitingBeepDuration and TimeBetweenWaitingIndications.
5. To configure a delay interval before a Call Waiting Indication is played to the currently busy port, use the parameter TimeBeforeWaitingIndication. This enables the caller to hang up before disturbing the called party with Call Waiting Indications. Applicable only to FXS modules.

Both the calling and called sides are supported by FXS interfaces; FXO interfaces support only the calling side.

To indicate Call Waiting, the device sends a 182 Call Queued response. The device identifies Call Waiting when a 182 Call Queued response is received.

24.7   Message Waiting Indication

The device supports Message Waiting Indication (MWI) according to IETF RFC 3842, including SUBSCRIBE to an MWI server.

The FXS device can accept an MWI NOTIFY message that indicates waiting messages or that the MWI is cleared. Users are informed of these messages by a stutter dial tone. The stutter and confirmation tones are defined in the CPT file. If the MWI display is configured, the number of waiting messages is also displayed. If the MWI lamp is configured, the phone's lamp (on a phone that is equipped with an MWI lamp) is lit. The device can subscribe to the MWI server per port (usually used on FXS) or per device (used on FXO).

You can also configure the voltage level mode (low or high) that the FXS port generates to the connected phone for lighting the phone's lamp used for indicating a message in waiting. If you require a flashing lamp, you can configure the on-off lamp durations. To configure this feature, see the following parameters:

- EnableLowVoltageMwiGeneration
- LedMwiOnDurationTime
- LedMwiOffDurationTime
- NeonMwiOnDurationTime

■ NeonMwiOffDurationTime

> **Note:**   For more information on IP voice mail configuration, refer to the *IP Voice Mail CPE Configuration Guide*.

To configure MWI, use the following parameters:
- EnableMWI
- MWIServerIP, or MWISubscribeIPGroupID and ProxySet
- MWIAnalogLamp
- MWIDisplay
- StutterToneDuration
- EnableMWISubscription
- MWIExpirationTime
- SubscribeRetryTime
- SubscriptionMode
- CallerIDType (determines the standard for detection of MWI signals)
- ETSIVMWITypeOneStandard
- BellcoreVMWITypeOneStandard
- VoiceMailInterface
- EnableVMURI

24.8   Caller ID

This section describes the device's Caller ID support.

24.8.1   Caller ID Detection / Generation on the Tel Side

By default, generation and detection of Caller ID to the Tel side is disabled. To enable Caller ID, set the parameter EnableCallerID to 1. When the Caller ID service is enabled:
- For FXS: the Caller ID signal is sent to the device's port
- For FXO: the Caller ID signal is detected

The configuration for Caller ID is described below:
- Use the parameter CallerIDType to define the Caller ID standard. Note that the Caller ID standard that is used on the PBX or phone must match the standard defined in the device.
- Select the Bellcore caller ID sub standard using the parameter BellcoreCallerIDTypeOneSubStandard
- Select the ETSI FSK caller ID sub standard using the parameter ETSICallerIDTypeOneSubStandard
- Enable or disable (per port) the caller ID generation (for FXS) and detection (for FXO) using the 'Generate / Detect Caller ID to Tel' table (EnableCallerID). If a port isn't configured, its caller ID generation / detection are determined according to the global parameter EnableCallerID.
- EnableCallerIDTypeTwo: disables / enables the generation of Caller ID type 2 when the phone is off-hooked (used for call waiting).
- RingsBeforeCallerID: sets the number of rings before the device starts detection of

caller ID (FXO only). By default, the device detects the caller ID signal between the first and second rings.

- **AnalogCallerIDTimimgMode:** determines the time period when a caller ID signal is generated (FXS only). By default, the caller ID is generated between the first two rings.
- **PolarityReversalType:** some Caller ID signals use reversal polarity and/or wink signals. In these scenarios, it is recommended to set PolarityReversalType to 1 (Hard) (FXS only).
- The Caller ID interworking can be changed using the parameters UseSourceNumberAsDisplayName and UseDisplayNameAsSourceNumber.

24.8.2   Debugging a Caller ID Detection on FXO

The procedure below describes debugging caller ID detection in FXO interfaces.

➢ **To debug a Caller ID detection on an FXO interface:**

1. Verify that the parameter EnableCallerID is set to 1.
2. Verify that the caller ID standard (and substandard) of the device matches the standard of the PBX (using the parameters CallerIDType, BellcoreCallerIDTypeOneSubStandard, and ETSICallerIDTypeOneSubStandard).
3. Define the number of rings before the device starts the detection of caller ID (using the parameter RingsBeforeCallerID).
4. Verify that the correct FXO coefficient type is selected (using the parameter CountryCoefficients), as the device is unable to recognize caller ID signals that are distorted.
5. Connect a phone to the analog line of the PBX (instead of to the device's FXO interface) and verify that it displays the caller ID.

If the above does not solve the problem, you need to record the caller ID signal (and send it to AudioCodes), as described below.

➢ **To record the caller ID signal using the debug recording mechanism:**

1. Access the FAE page (by appending "FAE" to the device's IP address in the Web browser's URL, for example, http://10.13.4.13/FAE).
2. Press the **Cmd Shell** link.
3. Enter the following commands:

```
dr
ait <IP address of PC to collect the debug traces sent from
the device>
AddChannelIdTrace ALL-WITH-PCM <port number, which starts from
0>
Start
```

4. Make a call to the FXO.
5. To stop the DR recording, at the CLI prompt, type **STOP**.

24.8.3   Caller ID on the IP Side

Caller ID is provided by the SIP From header containing the caller's name and "number", for example:

```
From: "John" <SIP:101@10.33.2.2>;tag=35dfsgasd45dg
```

If Caller ID is restricted (received from Tel or configured in the device), the From header is set to:

```
From: "anonymous" <anonymous@anonymous.invalid>; tag=35dfsgasd45dg
```

The P-Asserted (or P-Preferred) headers are used to present the originating party's caller ID even when the caller ID is restricted. These headers are used together with the Privacy header.

- If Caller ID is restricted:
  - The From header is set to "anonymous" <anonymous@anonymous.invalid>
  - The 'Privacy: id' header is included
  - The P-Asserted-Identity (or P-Preferred-Identity) header shows the caller ID

- If Caller ID is allowed:
  - The From header shows the caller ID
  - The 'Privacy: none' header is included
  - The P-Asserted-Identity (or P-Preferred-Identity) header shows the caller ID

The caller ID (and presentation) can also be displayed in the Calling Remote-Party-ID header.

The 'Caller Display Information' table (CallerDisplayInfo) is used for the following:

- **FXS interfaces** - to define the caller ID (per port) that is sent to IP.

- **FXO interfaces** - to define the caller ID (per port) that is sent to IP if caller ID isn't detected on the Tel side, or when EnableCallerID = 0.

- **FXS and FXO interfaces** - to determine the presentation of the caller ID (allowed or restricted).

- **To maintain backward compatibility** - when the strings 'Private' or 'Anonymous' are set in the Caller ID/Name field, the caller ID is restricted and the value in the Presentation field is ignored.

The value of the 'Presentation' field that is defined in the 'Caller Display Information' table can be overridden by configuring the 'Presentation' parameter in the 'Tel to IP Source Number Manipulation' table. Therefore, this table can be used to set the presentation for specific calls according to Source / Destination prefixes.

The caller ID can be restricted/allowed (per port) using keypad features KeyCLIR and KeyCLIRDeact (FXS only).

AssertedIdMode defines the header that is used (in the generated INVITE request) to deliver the caller ID (P-Asserted-Identity or P-Preferred-Identity). Use the parameter UseTelURIForAssertedID to determine the format of the URI in these headers (sip: or tel:).

The parameter EnableRPIheader enables Remote-Party-ID (RPI) headers for calling and called numbers for Tel-to-IP calls.

24.9   Three-Way Conferencing

The device supports three-way conference calls. These conference calls can also occur simultaneously. The device supports the following conference modes (configured by the parameter 3WayConferenceMode):

■ **Conferencing managed by an external, AudioCodes Conference (media) server:** The Conference-initiating INVITE sent by the device uses the ConferenceID concatenated with a unique identifier as the Request-URI. This same Request-URI is set as the Refer-To header value in the REFER messages that are sent to the two remote parties. For this mode, the 3WayConferenceMode parameter is set to 0 (default.)

■ **Conferencing managed by an external, third-party Conference (media) server:** The Conference-initiating INVITE sent by the device uses only the ConferenceID as the Request-URI. The Conference server sets the Contact header of the 200 OK response to the actual unique identifier (Conference URI) to be used by the participants. This Conference URI is included (by the device) in the Refer-To header value in the REFER messages sent by the device to the remote parties. The remote parties join the conference by sending INVITE messages to the Conference server using this conference URI. For this mode, the 3WayConferenceMode parameter is set to 1.

■ **Local, on-board conferencing:** The conference is established on the device without the need for an external Conference server. This feature includes local mixing and transcoding of the 3-Way Call legs on the device, and even allowing multi-codec conference calls. The number of simultaneous, on-board conferences can be limited using the parameter MaxInBoardConferenceCalls. The device utilizes resources from idle ports to establish the conference call. You can designate ports that can't be used as a resource for conference calls initiated by other ports, using the parameter 3WayConfNoneAllocateablePorts. Ports that are not configured with this parameter (and that are idle) are used by the device as a resource for establishing these types of conference calls. For this mode, the 3WayConferenceMode parameter is set to 2. For maximum on-board, three-way conference calls, see the note below.

> **Notes:**
>
> - Three-way conferencing is applicable only to FXS interfaces.
> - Maximum local on-board, three-way conference call local:
>
>   **New** MP-112 or MP-114 hardware revision (released in December 2018 and compatible with Software Version 6.60A.349.003 and later):
>   √ MP-112 supports one local on-board, three-way conference call
>   √ MP-114 supports up to two local on-board, three-way conference calls
>   For the new hardware revision models, each established conference call disables one FXS port.
>
>   **Old** MP-112 or MP-114 hardware revision:
>   √ MP-112 supports up to two local on-board, three-way conference calls
>   √ MP-114 supports up to three local on-board, three-way conference calls
>   For the old hardware revision models, each conference call doesn't disable an FXS port.
>
>   To identify the hardware revision of your MP-11x model, use the Command Shell Command **GetHwVersions**.
> - Each three-way conference call uses two DSP channels. Therefore, for MP-114, MP-118 and MP-124, each three-way conference call reduces channel capacity by one; for MP-112, no channel reduction occurs.
> - Instead of using the flash-hook button to establish a three-way conference call, you can dial a user-defined hook-flash code (e.g., "*1"), configured by the HookFlashCode parameter.

The following example demonstrates three-way conferencing using the device's local, on-board conferencing feature. In this example, telephone "A" connected to the device establishes a three-way conference call with two remote IP phones, "B" and "C":

1. A establishes a regular call with B.
2. A places B on hold, by pressing the telephone's flash-hook button and the number "1" key.
3. A hears a dial tone and then makes a call to C.
4. C answers the call.
5. A establishes a three-way conference call with B and C, by pressing the flash-hook button and digit 3.

To configure this local, on-board three-way conferencing:

1. Open the Supplementary Services page.
2. Set 'Enable 3-Way Conference' to **Enable** (Enable3WayConference = 1).
3. Set 'Three Way Conference Mode' to **On Board** (3WayConferenceMode = 2).
4. Set 'Flash Keys Sequence Style' to **Sequence 1** or **Sequence 2** (FlashKeysSequenceStyle = 1 or 2).

> **Note:** For local, on-board three-way conferencing on MP-112, in addition to configuring the previously mentioned parameters, the following must be configured:
>
> - New hardware revision:
>
> ```
> EnableIPMediaChannels = 1
>
> [ IPMediaChannels ]
> FORMAT IPMediaChannels_Index = IPMediaChannels_ModuleID,
> IPMediaChannels_DSPChannelsReserved;
> IPMediaChannels 0 = 1, 1;
> [ \IPMediaChannels ]
> ```
>
> - Old hardware revision:
>
> ```
> EnableIPMediaChannels = 1
>
> [ IPMediaChannels ]
> FORMAT IPMediaChannels_Index = IPMediaChannels_ModuleID,
> IPMediaChannels_DSPChannelsReserved;
> IPMediaChannels 0 = 1, 2;
> [ \IPMediaChannels ]
> ```

24.10 Emergency E911 Phone Number Services

This section describes the device's support for emergency phone number services.

24.10.1 Pre-empting Existing Calls for E911 IP-to-Tel Calls

If the device receives an E911 call from the IP network destined to the Tel, and there are unavailable channels (e.g., all busy), the device terminates one of the calls (arbitrary) and then sends the E911 call to that channel. The preemption is done only on a channel pertaining to the same Hunt Group for which the E911 call was initially destined and if the channel select mode (configured by the ChannelSelectMode parameter) is set to a value other than "By Dest Number" (0).

The preemption is done only if the incoming IP-to-Tel call is identified as an emergency call. The device identifies emergency calls by one of the following:

■ The destination number of the IP call matches one of the numbers defined by the EmergencyNumbers parameter. For E911, you must configure this parameter with the value "911".

■ The Priority header of the incoming SIP INVITE message contains the "emergency" value.

Emergency pre-emption of calls can be enabled for all calls, using the global parameter CallPriorityMode, or for specific calls using the Tel Profile parameter CallPriorityMode.

> **Notes:**
>
> - For Hunt Groups configured with call preemption, all must be configured to MLPP **[1]** or all configured to Emergency **[2]**. In other words, you cannot set some trunks to **[1]** and some to **[2]**.
> - The global parameter must be set to the same value as that of the Tel Profile parameter; otherwise, the Tel Profile parameter is not applied.
> - If you configure call preemption using the global parameter and a new Tel Profile is subsequently added, the TelProfile_CallPriorityMode parameter automatically acquires the same setting as well.
> - This feature is applicable to FXO interfaces.
> - For FXO interfaces, the preemption is done only on existing IP-to-Tel calls. In other words, if all the current FXO channels are busy with calls that were answered by the FXO device (i.e., Tel-to-IP calls), new incoming emergency IP-to-Tel calls are rejected.

24.11 Multilevel Precedence and Preemption

The device supports Multilevel Precedence and Preemption (MLPP) service. MLPP is a call priority scheme, which does the following:

■ Assigns a precedence level (priority level) to specific phone calls or messages.

■ Allows higher priority calls (*precedence call*) and messages to preempt lower priority calls and messages (i.e., terminates existing lower priority calls) that are recognized within a user-defined domain (*MLPP domain ID*). The domain specifies the collection of devices and resources that are associated with an MLPP subscriber. When an MLPP subscriber that belongs to a particular domain places a precedence call to another MLPP subscriber that belongs to the same domain, MLPP service can preempt the existing call that the called MLPP subscriber is on for a higher-

precedence call. MLPP service availability does not apply across different domains.

MLPP is typically used in the military where, for example, high-ranking personnel can preempt active calls during network stress scenarios such as a national emergency or degraded network situations.

MLPP can be enabled for all calls, using the global parameter, CallPriorityMode, or for specific calls using the Tel Profile parameter, CallPriorityMode.

Notes:

- For Hunt Groups configured with call preemption, all must be configured to MLPP **[1]** or all configured to Emergency **[2].** In other words, you cannot set some trunks to **[1]** and some to **[2].**
- The global parameter must be set to the same value as that of the Tel Profile parameter; otherwise, the Tel Profile parameter is not applied.
- If you configure call preemption using the global parameter and a new Tel Profile is subsequently added, the TelProfile_CallPriorityMode parameter automatically acquires the same setting as well.

The Resource Priority value in the Resource-Priority SIP header can be any one of those listed in the table below. For each MLPP call priority level, the Multiple Differentiated Services Code Points (DSCP) can be set to a value from 0 to 63.

Table 24-1: MLPP Call Priority Levels (Precedence) and DSCP Configuration Parameters

MLPP Precedence Level	Precedence Level in Resource-Priority SIP Header	DSCP Configuration Parameter
0 (lowest)	routine	MLPPRoutineRTPDSCP
2	priority	MLPPPriorityRTPDSCP
4	immediate	MLPPImmediateRTPDSCP
6	flash	MLPPFlashRTPDSCP
8	flash-override	MLPPFlashOverRTPDSCP
9 (highest)	flash-override-override	MLPPFlashOverOverRTPDSCP

Notes:

- If required, you can exclude the "resource-priority" tag from the SIP Require header in INVITE messages for Tel-to-IP calls when MLPP priority call handling is used. This is configured using the RPRequired parameter.
- For a complete list of the MLPP parameters, see 'MLPP and Emergency Call Parameters' on page 595.

24.11.1 MLPP Preemption Events in SIP Reason Header

The device sends the SIP Reason header (as defined in RFC 4411) to indicate the reason and type of a preemption event. The device sends a SIP BYE or CANCEL request, or SIP 480, 486, 488 response (as appropriate) with a Reason header whose Reason-params can includes one of the following preemption cause classes:

- Reason: preemption ;cause=1 ;text="UA Preemption"
- Reason: preemption ;cause=2 ;text="Reserved Resources Preempted"

■ Reason: preemption ;cause=3 ;text="Generic Preemption"

■ Reason: preemption ;cause=4 ;text="Non-IP Preemption"

This Reason cause code indicates that the session preemption has occurred in a non-IP portion of the infrastructure. The device sends this code in the following scenarios:

- The device performs a network preemption of a busy call (when a high priority call is received), the device sends a SIP BYE or CANCEL request with this Reason cause code.

- The device performs a preemption of a B-channel for a Tel-to-IP outbound call request from the softswitch for which it has not received an answer response (e.g., Connect), and the following sequence of events occurs:

    a. The device sends a Q.931 DISCONNECT over the ISDN MLPP PRI to the partner switch to preempt the remote end instrument.

    b. The device sends a 488 (Not Acceptable Here) response with this Reason cause code.

■ Reason: preemption; cause=5; text="Network Preemption"

This Reason cause code indicates preempted events in the network. Within the Defense Switched Network (DSN) network, the following SIP request messages and response codes for specific call scenarios have been identified for signaling this preemption cause:

- SIP:BYE - If an active call is being preempted by another call

- CANCEL - If an outgoing call is being preempted by another call

- 480 (Temporarily Unavailable), 486 (User Busy), 488 (Not Acceptable Here) - Due to incoming calls being preempted by another call.

The device receives SIP requests with preemption reason cause=5 in the following cases:

- The softswitch performs a network preemption of an active call - the following sequence of events occurs:

    a. The softswitch sends the device a SIP BYE request with this Reason cause code.

    b. The device initiates the release procedures for the B-channel associated with the call request and maps the preemption cause to PRI Cause = #8 'Preemption'. This value indicates that the call is being preempted. For PRI, it also indicates that the B-channel is not reserved for reuse.

    c. The device sends a SIP 200 OK in response to the received BYE, before the SIP end instrument can proceed with the higher precedence call.

- The softswitch performs a network preemption of an outbound call request for the device that has not received a SIP 2xx response - the following sequence of events occur:

    a. The softswitch sends the device a SIP 488 (Not Acceptable Here) response code with this Reason cause code. The device initiates the release procedures for the B-channel associated with the call request and maps the preemption cause to PRI Cause = #8 'Preemption'.

    b. The device deactivates any user signaling (e.g., ringback tone) and when the call is terminated, it sends a SIP ACK message to the softswitch.

24.11.2 Precedence Ring Tone

You can assign a ring tone that is defined in the CPT file to be played when a precedence call is received from the IP side. This is configured by the PrecedenceRingingType parameter.

Emergency Telecommunications Services (ETS) calls (e.g., E911) can be configured with a higher priority than any MLPP call (default), using the E911MLPPBehavior parameter.

24.12 Denial of Collect Calls

You can configure the device to reject (disconnect) incoming Tel-to-IP collect calls and to signal this denial to the PSTN. This capability is required, for example, in the Brazilian telecommunication system to deny collect calls. When this feature is enabled upon rejecting the incoming call, the device sends a sequence of signals to the PSTN. This consists of an off-hook, an on-hook after one second, and then an off-hook after two seconds. In other words, this is in effect, a double-answer sequence.

This feature can be enabled for all calls, using the EnableFXODoubleAnswer "global" parameter, or it can be enabled for specific calls, by enabling this feature in a Tel Profile.

> **Notes:**
>
> - This feature is applicable only to FXO interfaces.
>
> - If automatic dialing is also configured for an FXO port enabled with Denial of Collect Calls, the FXO line does not answer the incoming call (ringing) until a SIP 200 OK is received from the remote destination. When a 200 OK is received, a double answer is sent from the FXO line.
>
> - Ensure that the PSTN side is configured to identify this double-answer signal.

24.13 Configuring Voice Mail

The Voice Mail Settings page allows you to configure the voice mail parameters. For a description of these parameters, see 'Configuration Parameters Reference' on page 471.

> **Notes:**
>
> - The Voice Mail Settings page is available only for FXO interfaces.
> - For more information on configuring voice mail, refer to the *CPE Configuration Guide for Voice Mail User's Manual*.

➢ **To configure the Voice Mail parameters:**

1. Open the Voice Mail Settings page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Advanced Applications** > **Voice Mail Settings**).

Figure 24-4: Voice Mail Settings

Line Transfer Mode	None
Voice Mail Interface	NONE
▼ Digit Patterns	
Forward on Busy Digit Pattern (Internal)	
Forward on No Answer Digit Pattern (Internal)	
Forward on Do Not Disturb Digit Pattern (Internal)	
Forward on No Reason Digit Pattern (Internal)	
Forward on Busy Digit Pattern (External)	
Forward on No Answer Digit Pattern (External)	
Forward on Do Not Disturb Digit Pattern (External)	
Forward on No Reason Digit Pattern (External)	
Internal Call Digit Pattern	
External Call Digit Pattern	
Disconnect Call Digit Pattern	
Digit To Ignore Digit Pattern	
▼ Message Waiting Indication (MWI)	
MWI Off Digit Pattern	
MWI On Digit Pattern	
MWI Suffix Pattern	
MWI Source Number	
▼ SMDI	
Enable SMDI	Disable
SMDI Timeout [msec]	2000

2. Configure the parameters as required.
3. Click **Submit** to apply your changes.
4. To save the changes to flash memory, see 'Saving Configuration' on page 362.

24.14 Out-of-Band Digit Notifications According to KPML

The device supports Out-of-Band digit notifications according to a subset of the KPML specification (RFC 4730). When an INVITE dialog is initiated, even in early dialog, the device may receive a SUBSCRIBE message outside of the dialog. An application that wants to collect digits creates an application/kpml-request+xml document with the digit patterns of interest to the application and places this document in a SUBSCRIBE request. This SUBSCRIBE includes the identifiers of the INVITE dialog. The SUBSCRIBE request consists of DRegex (Digit Regular Expression). Once a subscription is established, the device sends application/kpml-response+xml documents in NOTIFY requests once a match is found.

If the SUBSCRIBE request has no KPML body, any KPML document running on that dialog and addressed by the event id, if present, immediately terminates. However, the SUBSCRIBE-initiated dialog is still active. If the dialog referenced by the KPML subscription does not exist, the device returns a 404 Not Found response to the SUBSCRIBE and terminates the dialog.

It's possible to "re-subscribe". In such a scenario, the device terminates the existing KPML request and replaces it with the new request.

> **Notes:**
>
> - Only one KPML subscription per participant/dialog is supported.
> - Only one regex per pattern is supported.
> - Only single-digit patterns are supported.
> - The following tags are not supported: "pre", "flush", "stream", and "enterkey".

This page is intentionally left blank.

25   Analog Gateway

This section describes configuration of analog settings.

25.1   Configuring Keypad Features

The Keypad Features page enables you to activate and deactivate the following features directly from the connected telephone's keypad:

- Call Forward
- Caller ID Restriction
- Hotline for automatic dialing
- Call Transfer
- Call Waiting
- Rejection of Anonymous Calls

> **Notes:**
>
> - The Keypad Features page is available only for FXS interfaces.
> - The method used by the device to collect dialed numbers is identical to the method used during a regular call (i.e., max digits, interdigit timeout, digit map, etc.).
> - The activation of each feature remains in effect until it is deactivated (i.e., not deactivated after a call).
> - For a description of the keypad parameters, see 'Telephone Keypad Sequence Parameters' on page 612.

➢ **To configure the keypad features**

1.   Open the Keypad Features page (**Configuration** tab > **VoIP** menu > **GW and IP to IP**

> **Analog Gateway** > **Keypad Features**).

Figure 25-1: Keypad Features Page

2. Configure the keypad features as required.
3. Click **Submit** to apply your changes.

25.2   Configuring Metering Tones

The FXS interfaces can generate 12/16 KHz metering pulses toward the Tel side (e.g., for connection to a pay phone or private meter). Tariff pulse rate is determined according to the device's Charge Codes table. This capability enables users to define different tariffs according to the source/destination numbers and the time-of-day. The tariff rate includes the time interval between the generated pulses and the number of pulses generated on answer.

> **Notes:**
>
> - The Metering Tones page is available only for FXS interfaces.
> - Charge Code rules can be assigned to routing rules in the Tel to IP Routing (see 'Configuring Tel to IP Routing' on page 250). When a new call is established, the Tel to IP Routing is searched for the destination IP address. Once a route is located, the Charge Code (configured for that route) is used to associate the route with an entry in the Charge Codes table.

➢ **To configure Metering tones:**

1. Open the Metering Tones page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **Metering Tones**).

Figure 25-2: Metering Tones Page

Generate Metering Tones	Disable
⚡ Metering Tone Type	16 KHz
Charge Codes Table	▣➡

2. Configure the Metering tones parameters as required. For a description of the parameters appearing on this page, see 'Configuration Parameters Reference' on page 471.

3. Click **Submit** to apply your changes.

4. To save the changes to the flash memory, see 'Saving Configuration' on page 362.

If you set the 'Generate Metering Tones' parameter to **Internal Table**, access the Charge Codes Table page by clicking the **Charge Codes Table** 🔷 button. For more information on configuring the Charge Codes table, see 'Configuring Charge Codes' on page 300.

25.3 Configuring Charge Codes

The Charge Codes table is used to configure the metering tones (and their time interval) that the FXS interfaces generate to the Tel side. To associate a charge code to an outgoing Tel-to-IP call, use the Tel to IP Routing.

You can configure up to 25 different charge codes, where each table row represents a charge code. Each charge code can include up to four different time periods in a day (24 hours). The device selects the time period by comparing the device 's current time to the end time of each time period of the selected Charge Code. The device generates the Number of Pulses on Answer once the call is connected and from that point on, it generates a pulse each Pulse Interval. If a call starts at a certain time period and crosses to the next, the information of the next time period is used.

> **Notes:**
> * The Charge Codes Table page is available only for FXS interfaces.
> * The Charge Codes table can also be configured using the table ini file parameter, ChargeCode.

> **To configure the Charge Codes:**

1. Open the Charge Codes Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Analog Gateway** submenu > **Charge Codes**). Alternatively, you can access this page from the Metering Tones page (see 'Configuring Metering Tones' on page 299).

Figure 25-3: Charge Codes Table Page

Table Index	0-4	

Index	Time Period 1			Time Period 2			Time Period 3			Time Period 4		
	End Time	Pulse Interval	Pulses On Answer	End Time	Pulse Interval	Pulses On Answer	End Time	Pulse Interval	Pulses On Answer	End Time	Pulse Interval	Pulses On Answer
0	07	30	1	14	20	2	20	15	1	00	60	1
1	05	60	1	14	20	1	00	60	1			
2	00	60	1									
3												
4												

2. Configured the charge codes, as required. For a description of the parameters, see the table below.
3. Click **Submit** to apply your changes.
4. To save the changes to the flash memory, see 'Saving Configuration' on page 362.

Table 25-1: Charge Codes Table Parameter Description

Parameter	Description
End Time **[ChargeCode_EndTime<1-4>]**	Defines the end of the time period in a 24 hour format, *hh*. For example, "04" denotes 4 A.M. **Notes:** ▪ The first time period always starts at midnight (00). ▪ It is mandatory that the last time period of each rule end at midnight (00). This prevents undefined time frames in a day.
Pulse Interval **[ChargeCode_PulseInterval<1-4>]**	Defines the time interval between pulses (in tenths of a second).
Pulses On Answer **[ChargeCode_PulsesOnAnswer<1-4>]**	Defines the number of pulses sent on answer.

25.4 Configuring FXO Settings

The FXO Settings page allows you to configure the device's specific FXO parameters. For a description of these parameters, see 'Configuration Parameters Reference' on page 471.

> **Note:** The FXO Settings page is available only for FXO interfaces.

➢ **To configure the FXO parameters:**

1. Open the FXO Settings page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **FXO Settings**).

Figure 25-4: FXO Settings Page

Dialing Mode	Two Stages
Waiting for Dial Tone	No
Time to Wait before Dialing [msec]	1000
Ring Detection Timeout [sec]	8
Reorder Tone Duration [sec]	255
Answer Supervision	No
Rings before Detecting Caller ID	1
Send Metering Message to IP	No
Disconnect Call on Busy Tone Detection (CAS)	Enable
Disconnect On Dial Tone	Disable
Guard Time Between Calls	1
FXO Double Answer	Disable
FXO AutoDial Play BusyTone	Disable

2. Configure the parameters as required.

3. Click **Submit** to apply your changes.

4. To save the changes to flash memory, see 'Saving Configuration' on page 362.

25.5 Configuring Authentication

The Authentication page defines a user name and password for authenticating each device port. Authentication is typically used for FXS interfaces, but can also be used for FXO interfaces.

> **Notes:**
>
> - For configuring whether authentication is done per port or for the entire device, use the parameter AuthenticationMode.
> - If authentication is configured for the entire device, the configuration in this table is ignored.
> - If the user name or password is not configured in this table, the port's phone number (configured in the Endpoint Phone Number tableand global password (configured by the global parameter, Password) are used instead for authentication of the port.
> - After you click **Submit**, the password is displayed as an asterisk (*).
> - The Authentication table can also be configured using the table ini file parameter, Authentication (see 'Configuration Parameters Reference' on page 471).

➢ **To configure authentication credentials per port:**

1. Set the parameter 'Registration Mode' (AuthenticationMode) to **Per Endpoint.** This can be configured in any of the following pages:

   - Proxy & Registration page (see 'Configuring Proxy and Registration Parameters' on page 208).
   - Trunk Group Settings page (see 'Configuring Hunt Group Settings' on page 229), where registration method is configured per Trunk Group.

2. Open the Authentication page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **Authentication**).

Gateway Port	User Name	Password
Port 1 FXS		
Port 2 FXS		
Port 3 FXS		
Port 4 FXS		
Port 5 FXO		
Port 6 FXO		
Port 7 FXO		
Port 8 FXO		

3. Configure port authentication credentials as required. For a description of the parameters, see the table below.

4. Click **Submit** to apply your changes.

Table 25-2: Authentication Table Parameter Description

Parameter	Description
User Name **[Authentication_UserId]**	Defines the user name used for authenticating the port.
Password **[Authentication_UserPassword]**	Defines the password used for authenticating the port.

25.6   Configuring Automatic Dialing

The Automatic Dialing page allows you to define a telephone number that is automatically dialed when an FXS or FXO port goes off-hook. The dialing can be done immediately upon off-hook, or after a user-defined interval after off-hook referred to as *Hotline* dialing.

> **Note:**   The Automatic Dialing can also be configured using the table ini file parameter, TargetOfChannel.

➢   **To configure automatic dialing per port:**

1.   Open the Automatic Dialing page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** submenu > **Analog Gateway** > **Automatic Dialing**).

Figure 25-5: Automatic Dialing Page

Gateway Port	Destination Phone Number	Auto Dial Status	Hotline Dial Tone Duration [sec]
Port 1 FXS	911	Enable	15
Port 2 FXS	400	Enable	0
Port 3 FXS		Enable	0
Port 4 FXS		Enable	0
Port 5 FXO		Enable	0
Port 6 FXO		Enable	0
Port 7 FXO		Enable	0
Port 8 FXO		Enable	0

The first table entry in the figure above enables Hotline automatic dialing for an FXS port, whereby if the port is off-hooked for over 15 seconds, the device automatically dials 911.

2.   Configure automatic dialing per port, as required. See the table below for parameter descriptions.

3.   Click **Submit** to apply your changes.

4.   To save the changes to flash memory, see 'Saving Configuration' on page 362.

Table 25-3: Automatic Dialing Table Parameter Description

Parameter	Description
Gateway Port	Lists the FXS or FXO port for which you want to configure automatic dialing.
Destination Phone Number **[TargetOfChannel_Destination]**	Defines the destination telephone number to automatically dial.
Auto Dial Status **[TargetOfChannel_Type]**	Enables automatic dialing. ▪ **[0]** Disable = Automatic dialing for the specific port is disabled. ▪ **[1]** Enable = (Default) Automatic dialing is enabled and the phone number configured in the 'Destination Phone Number' field is automatically dialed if the following occurs: ✓ FXS interfaces: The phone is off-hooked ✓ FXO interfaces: A ring signal (from a PBX/PSTN switch) is detected on the FXO line. The device initiates a call to the destination without seizing the FXO line. The line is seized only after the SIP call is answered. ▪ **[2]** Hotline = Automatic dialing is done after an interval configured by the 'Hotline Dial Tone Duration' parameter: ✓ FXS interfaces: When the phone is off-hooked and no digit is dialed within a user-defined time, the configured destination number is automatically dialed. ✓ FXO interfaces: If a ring signal is detected, the device seizes the FXO line, plays a dial tone, and then waits for DTMF digits. If no digits are detected within a user-defined time, the configured destination number is automatically dialed by sending a SIP INVITE message with this number.
Hotline Dial Tone Duration **[TargetOfChannel_HotLineToneDuration]**	Defines the duration (in seconds) after which the destination phone number is automatically dialed. This is applicable only if the port has been configured for Hotline (i.e., 'Auto Dial Status' is set to **Hotline**). The valid value is 0 to 60. The default is 16. **Note:** You can configure this Hotline interval for all ports, using the global parameter, HotLineToneDuration.

25.7   Configuring Caller Display Information

The Caller Display Information table allows you to define a caller identification string (Caller ID) for FXS and FXO ports and enable the device to send the Caller ID to the IP when a call is made. The called party can use this information for caller identification.

The device sends the configured caller ID in the outgoing INVITE message's From header. For information on Caller ID restriction according to destination/source prefixes, see 'Configuring Source/Destination Number Manipulation' on page 235.

Notes:

- If an FXS port receives 'Private' or 'Anonymous' strings in the SIP From header, the calling name or number is not sent to the Caller ID display.

- If Caller ID is detected on an FXO line (EnableCallerID = 1), it is used instead of the Caller ID configured in this table.

- If you set the 'Caller ID/Name' parameter to the strings "Private" or "Anonymous", Caller ID is restricted and the settings of the 'Presentation' parameter is ignored.

- The Caller Display Information table can also be configured using the table ini file parameter, CallerDisplayInfo.

➢ **To configure Caller Display:**

1. Open the Caller Display Information page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **Caller Display Information**).

Figure 25-6: Caller Display Information Page

Gateway Port	Caller ID/Name	Presentation
Port 1 FXS	Private	Restricted
Port 2 FXS	Susan C.	Restricted
Port 3 FXS	Lee P.	Allowed
Port 4 FXS	Ronaldino E.	Restricted
Port 5 FXO	Hung L.	Allowed
Port 6 FXO		Allowed
Port 7 FXO		Allowed
Port 8 FXO		Allowed

2. Configure the table as required. For a description of the parameters, see the table below.

3. Click **Submit** to apply your changes.

Table 25-4: Caller Display Parameter Description

Parameter	Description
Gateway Port **[CallerDisplayInfo_Port]**	Displays the port.
Caller ID/Name **[CallerDisplayInfo_DisplayString]**	Defines the Caller ID string. The valid value is a string of up to 18 characters.
Presentation **[CallerDisplayInfo_IsCidRestricted]**	Enables the sending of the caller ID string. ▪ **[0]** Allowed = The caller ID string is sent when a Tel-to-IP call is made. ▪ **[1]** Restricted = The caller ID string is not sent. **Notes:** ▪ This parameter is overridden by the 'Presentation' parameter in the Source Number Manipulation table (see 'Configuring Source/Destination Number Manipulation' on page 235). ▪ If this parameter is set to **Restricted**, the Caller ID is sent to the remote side using only the SIP P-Asserted-Identity and P-Preferred-Identity headers (AssertedIdMode).

25.8   Configuring Call Forward

The Call Forwarding table allows you to configure call forwarding per port for IP-to-Tel calls. This redirects the call (using SIP 302 response) initially destined to a specific device Tel port, to a different device port or to an IP destination.

> **Notes:**
>
> - To enable call forwarding, set the 'Enable Call Forward' parameter to **Enable**. This is done in the Supplementary Services page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **DTMF and Supplementary** > **Supplementary Services**).
> - The Call Forward table can also be configured using the table ini file parameter, FwdInfo.

➢ **To configure Call Forward per port:**

1. Open the Call Forward Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **Call Forward**).

Figure 25-7: Call Forward Table Page

Gateway Port	Forward Type	Forward to Phone Number	Time for No Reply Forward
Port 1 FXS	On busy	201	30
Port 2 FXS	On busy	201	30
Port 3 FXS	No Answer	203	30
Port 4 FXS	Unconditional	202@10.2.1.1	30
Port 5 FXO	Deactivate		30

2. Configure the table as required. For descriptions of the parameters, see the table below.
3. Click **Submit** to apply your changes.

Table 25-5: Call Forward Table Parameter Description

Parameter	Description
Forward Type **[FwdInfo_Type]**	Defines the condition upon which the call is forwarded. ▪ **[0]** Deactivate = (Default) Don't forward incoming calls. ▪ **[1]** On Busy = Forward incoming calls when the port is busy. ▪ **[2]** Unconditional = Always forward incoming calls. ▪ **[3]** No Answer = Forward incoming calls that are not answered within the time specified in the 'Time for No Reply Forward' field. ▪ **[4]** On Busy or No Answer = Forward incoming calls when the port is busy or when calls are not answered within the time specified in the 'Time for No Reply Forward' field. ▪ **[5]** Do Not Disturb = Immediately reject incoming calls.

Parameter	Description
Forward to Phone Number **[FwdInfo_Destination]**	Defines the telephone number or URI (<number>@<IP address>) to where the call is forwarded. **Note:** If this parameter is configured with only a telephone number and a Proxy isn't used, this forwarded-to phone number must be specified in the Tel to IP Routing (see 'Configuring Tel to IP Routing' on page 250).
Time for No Reply Forward **[FwdInfo_NoReplyTime]**	If you have set the 'Forward Type' for this port to **No Answer**, then configure the number of seconds the device waits before forwarding the call to the specified phone number. **Note:** If you deactivate an active call forwarding rule (i.e., 'Forward Type' field changed to **Deactivate**), the configured value in the 'Time for No Reply Forward' field is maintained and the value in the 'Forward to Phone Number' field is deleted.

25.9 Configuring Caller ID Permissions

The Caller ID Permissions table allows you to enable per port, Caller ID generation for FXS interfaces and Caller ID detection for FXO interfaces.

> **Notes:**
>
> - If Caller ID permissions is not configured for a port in this table, its Caller ID generation / detection is determined according to the global parameter, 'Enable Call ID' in the Supplementary Services page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **DTMF and Supplementary** > **Supplementary Services**).
> - The Caller ID Permissions table can also be configured using the table ini file parameter, EnableCallerID.

➢ **To configure Caller ID permissions per port:**

1. Open the Caller ID Permissions page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **Caller ID Permissions**).

Figure 25-8: Caller ID Permissions Page

Gateway Port	Caller ID
Port 1 FXS	Enable
Port 2 FXS	Enable
Port 3 FXS	
Port 4 FXS	
Port 5 FXO	
Port 6 FXO	
Port 7 FXO	
Port 8 FXO	

2. Configure the table as required. For a description of the parameter, see the table below.
3. Click **Submit** to apply your changes.

Table 25-6: Caller ID Permissions Table Parameter Description

Parameter	Description
Caller ID **[EnableCallerId_IsEnabled]**	Enables Caller ID generation (FXS) or detection (FXO) per port. ▪ **[0]** Disable ▪ **[1]** Enable

25.10 Configuring Call Waiting

The Call Waiting table allows you to enable or disable call waiting per FXS port.

> **Notes:**
>
> - This page is applicable only to FXS interfaces.
> - You can enable or disable call waiting for all the device's ports using the global parameter, 'Enable Call Waiting' in the Supplementary Services page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **DTMF and Supplementary** > **Supplementary Services**).
> - The CPT file installed on the device must include a 'call waiting Ringback' tone (caller side) and a 'call waiting' tone (called side, FXS interfaces only).
> - The EnableHold parameter must be enabled on both the calling and the called sides.
> - For additional call waiting configuration, see the following parameters: FirstCallWaitingToneID (in the CPT file), TimeBeforeWaitingIndication, WaitingBeepDuration, TimeBetweenWaitingIndications, and NumberOfWaitingIndications.
> - The Call Waiting table can also be configured using the table ini file parameter, CallWaitingPerPort.

➢ **To enable call waiting per port:**

1. Open the Call Waiting page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **Call Waiting**).

Figure 25-9: Call Waiting Page

2. Configure the table as required. For a description of the parameter, see the table below.
3. Click **Submit** to apply your changes.

Table 25-7: Call Waiting Table Parameter Description

Parameter	Description
Call Waiting Configuration **[CallWaitingPerPort_IsEnabled]**	Enables call waiting for the port. ▪ **[0]** Disable

Parameter	Description
	▪ **[1]** Enable = Enables call waiting for the port. When the device receives a call on a busy port, it responds with a SIP 182 response (not with a 486 busy). The device plays a call waiting indication signal. When the device detects a hook-flash from the FXS port, the device switches to the waiting call. The device that initiated the waiting call plays a call waiting ringback tone to the calling party after a 182 response is received.

25.11 Rejecting Anonymous Calls

You can configure the device to reject anonymous calls received from the IP and destined for FXS interfaces. This can be configured using the ini file parameter, RejectAnonymousCallPerPort. If configured, when an FXS interface receives an anonymous call, the device rejects the call and responds with a SIP 433 (Anonymity Disallowed) response. For a description of the parameter see 'Caller ID Parameters' on page 578.

25.12 Configuring FXS Distinctive Ringing and Call Waiting Tones per Source/Destination Number

You can configure a distinctive ringing tone and call waiting tone per calling (source) and/or called (destination) number (or prefix) for IP-to-Tel calls. This feature can be configured per FXS endpoint or for a range of FXS endpoints. Therefore, different tones can be played per FXS endpoint depending on the source and/or destination number of the received call. You can also configure multiple entries with different source and/or destination prefixes and tones for the same FXS port.

Typically, the played ring and/or call waiting tone is indicated in the SIP Alert-info header field of the received INVITE message. If this header is not present in the received INVITE, then this feature is used and the tone played is according to the settings in this table.

> **Notes:**
>
> • This page is applicable only to FXS interfaces.
> • The Tone Index table can also be configured using the table ini file parameter, ToneIndex.

➢ **To configure distinctive ringing and call waiting per FXS port:**

1. Open the Tone Index Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **Tone Index**).

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

Figure 25-10: Tone Index Table Page

The figure above shows a configuration example for using distinctive ringing and call waiting tones of Index #9 ('Priority Index' 1) in the CPT file for FXS endpoints 1 to 4 when a call is received from a source number with prefix 2.

3. Configure the table as required. For a description of the parameters, see the table below.

4. Click **Submit** to apply your changes.

Table 25-8: Tone index Table Parameter Description

Parameter	Description
Index	Defines the table index entry. Up to 50 entries can be defined.
FXS Port First **[ToneIndex_FXSPort_First]**	Defines the first port in the FXS port range.
FXS Port Last **[ToneIndex_FXSPort_Last]**	Defines the last port in the FXS port range.
Source Prefix **[ToneIndex_SourcePrefix]**	Defines the prefix of the calling number.
Destination Prefix **[ToneIndex_DestinationPrefix]**	Defines the prefix of the called number.
Priority Index **[ToneIndex_PriorityIndex]**	Defines the index of the distinctive ringing and call waiting tones. The call waiting tone index equals to the Priority Index plus the value of the FirstCallWaitingToneID parameter. For example, if you want to use the call waiting tone in the CPT file at Index #9, you need to enter "1" as the Priority Index value and set the FirstCallWaitingToneID parameter to "8". The summation of these values is 9, i.e., index #9. The default is 0.

25.13 FXS/FXO Coefficient Types

The FXS Coefficient and FXO Coefficient types used by the device can be one of the following:

■ US line type of 600 ohm AC impedance and 40 V RMS ringing voltage for REN = 2

■ European standard (TBR21)

These Coefficient types are used to increase return loss and trans-hybrid loss performance for two telephony line type interfaces (US or European). This adaptation is performed by modifying the telephony interface characteristics. This means, for example, that changing impedance matching or hybrid balance doesn't require hardware modifications, so that a single device is able to meet requirements for different markets. The digital design of the filters and gain stages also ensures high reliability, no drifts (over temperature or time) and simple variations between different line types.

The FXS Coefficient types provide best termination and transmission quality adaptation for two FXS line type interfaces. This parameter affects the following AC and DC interface parameters:

■ DC (battery) feed characteristics

■ AC impedance matching

■ Transmit gain

■ Receive gain

■ Hybrid balance

■ Frequency response in transmit and receive direction

■ Hook thresholds

■ Ringing generation and detection parameters

➢ **To select the FXO and FXS Coefficient types:**

1. Open the Analog Settings page (**Configuration** tab > **VoIP** menu > **Media** > **Analog Settings**). This page includes the Coefficient type parameters, as shown below:

Figure 25-11: FXS/FXO Coefficient Parameters in Analog Settings Page

2. From the 'FXS Coefficient Type' drop-down list (FXSCountryCoefficients), select the required FXS Coefficient type.

3. From the 'FXO Coefficient Type' drop-down list (CountryCoefficients), select the required FXO Coefficient type.

4. Click **Submit**.

5. Save your settings to the flash memory ("burn") with a device reset.

25.14 FXO Operating Modes

This section provides a description of the device's FXO operating modes:

25.14.1 FXO Operations for IP-to-Tel Calls

The FXO device provides the following operating modes for IP-to-Tel calls:

- One-stage dialing (see 'One-Stage Dialing' on page 313)
  - Waiting for dial tone (see 'Two-Stage Dialing' on page 314)
  - Time to wait before dialing
  - Answer supervision
- Two-stage dialing (see 'Two-Stage Dialing' on page 314)
- Dialing time: DID wink (see 'DID Wink' on page 315)

25.14.1.1   One-Stage Dialing

One-stage dialing is when the FXO device receives an IP-to-Tel call, off-hooks the PBX line connected to the telephone, and then immediately dials the destination telephone number. In other words, the IP caller doesn't dial the PSTN number upon hearing a dial tone.

Figure 25-12: Call Flow for One-Stage Dialing
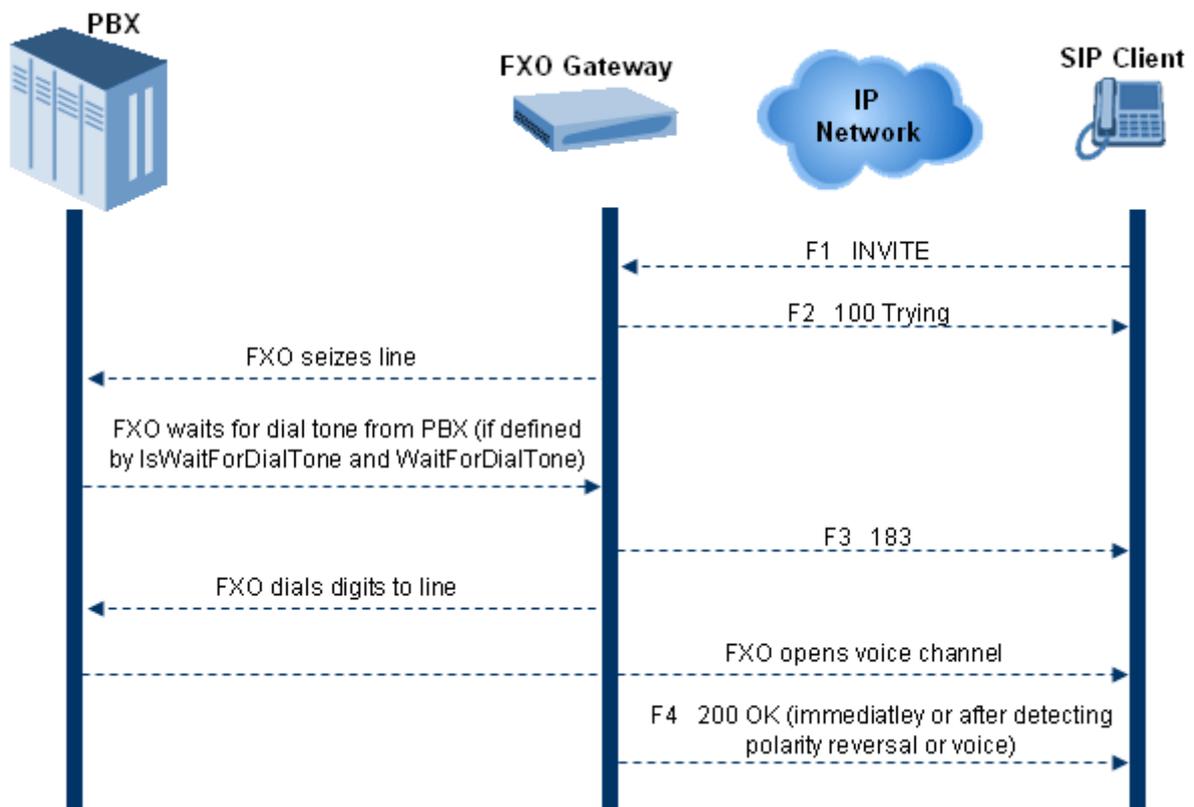
One-stage dialing incorporates the following FXO functionality:

■ **Waiting for Dial Tone:** Enables the device to dial the digits to the Tel side only after detecting a dial tone from the PBX line. The *ini* file parameter IsWaitForDialTone is used to configure this operation.

■ **Time to Wait Before Dialing:** Defines the time (in msec) between seizing the FXO line and starting to dial the digits. The *ini* file parameter WaitForDialTime is used to configure this operation.

> **Note:** The *ini* file parameter IsWaitForDialTone must be disabled for this mode.

■ **Answer Supervision:** The Answer Supervision feature enables the FXO device to determine when a call is connected, by using one of the following methods:

- **Polarity Reversal:** the device sends a 200 OK in response to an INVITE only when it detects a polarity reversal.
- **Voice Detection:** the device sends a 200 OK in response to an INVITE only when it detects the start of speech (fax or modem answer tone) from the Tel side. Note that the IPM detectors must be enabled.

25.14.1.2   Two-Stage Dialing

Two-stage dialing is when the IP caller is required to dial twice. The caller initially dials to the FXO device and only after receiving a dial tone from the PBX (via the FXO device), dials the destination telephone number.
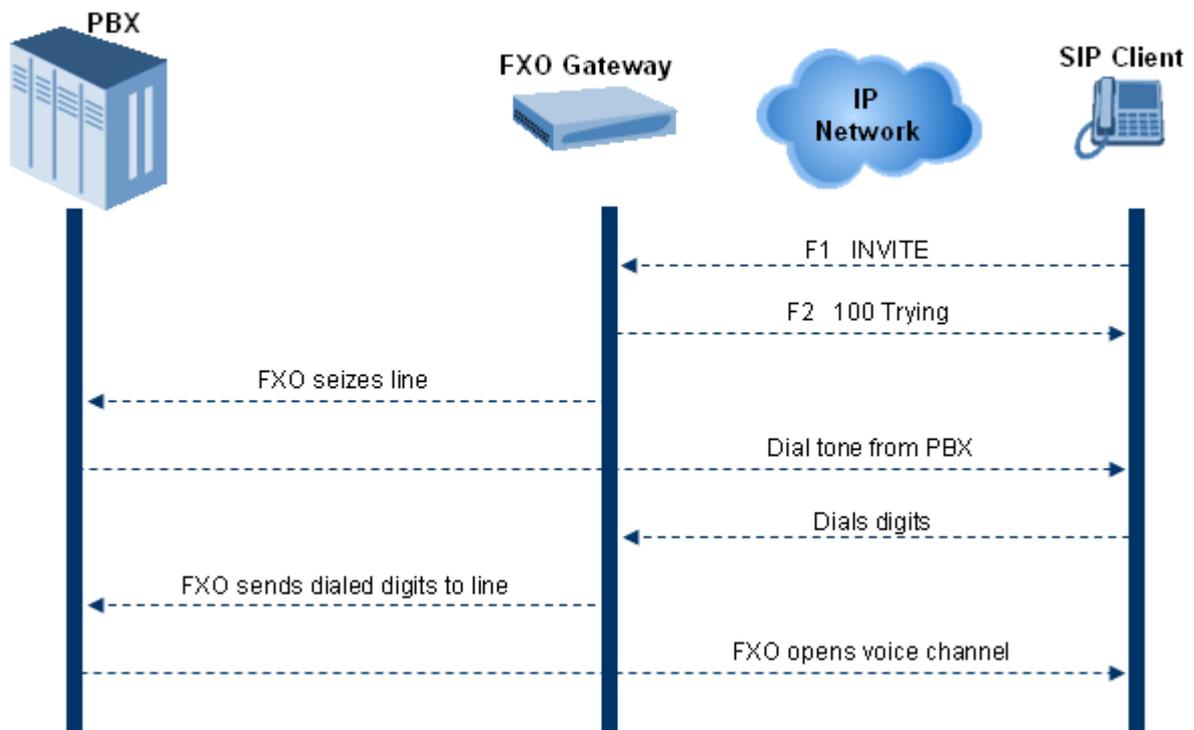
Two-stage dialing implements the Dialing Time feature. Dialing Time allows you to define the time that each digit can be separately dialed. By default, the overall dialing time per digit is 200 msec. The longer the telephone number, the greater the dialing time.

The relevant parameters for configuring Dialing Time include the following:

■ DTMFDigitLength (100 msec): time for generating DTMF tones to the PSTN (PBX) side

■ DTMFInterDigitInterval (100 msec): time between generated DTMF digits to PSTN (PBX) side

25.14.1.3   DID Wink

The device's FXO ports support Direct Inward Dialing (DID). DID is a service offered by telephone companies that enables callers to dial directly to an extension on a PBX without the assistance of an operator or automated call attendant. This service makes use of DID trunks, which forward only the last three to five digits of a phone number to the PBX. If, for example, a company has a PBX with extensions 555-1000 to 555-1999, and a caller dials 555-1234, the local central office (CO) would forward, for example, only 234 to the PBX. The PBX would then ring extension 234.

DID wink enables the originating end to seize the line by going off-hook. It waits for acknowledgement from the other end before sending digits. This serves as an integrity check that identifies a malfunctioning trunk and allows the network to send a re-order tone to the calling party.

The "start dial" signal is a wink from the PBX to the FXO device. The FXO then sends the last four to five DTMF digits of the called number. The PBX uses these digits to complete the routing directly to an internal station (telephone or equivalent).

■ DID Wink can be used for connection to EIA/TIA-464B DID Loop Start lines

■ Both FXO (detection) and FXS (generation) are supported

25.14.2 FXO Operations for Tel-to-IP Calls

The FXO device provides the following FXO operating modes for Tel-to-IP calls:

■ Automatic Dialing (see 'Automatic Dialing' on page 315)

■ Collecting Digits Mode (see 'Collecting Digits Mode' on page 317)

■ FXO Supplementary Services (see 'FXO Supplementary Services' on page 317)

     • Hold/Transfer Toward the Tel side

     • Hold/Transfer Toward the IP side

     • Blind Transfer to the Tel side

25.14.2.1   Automatic Dialing

Automatic dialing is defined using the Web interface's Automatic Dialing (TargetOfChannel ini file parameter) page, described in see 'Configuring Automatic Dialing' on page 303.

The SIP call flow diagram below illustrates Automatic Dialing.

Figure 25-13: Call Flow for Automatic Dialing
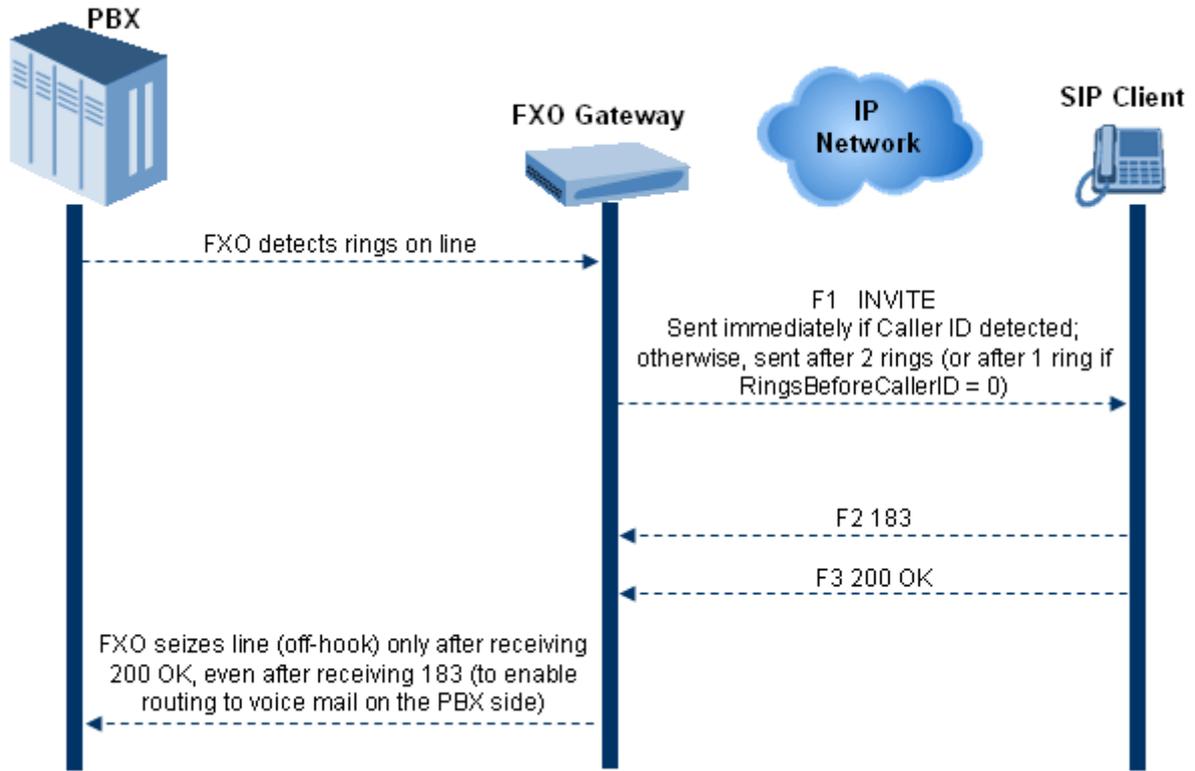
25.14.2.2   Collecting Digits Mode

When automatic dialing is not defined, the device collects the digits.

The SIP call flow diagram below illustrates the Collecting Digits Mode.
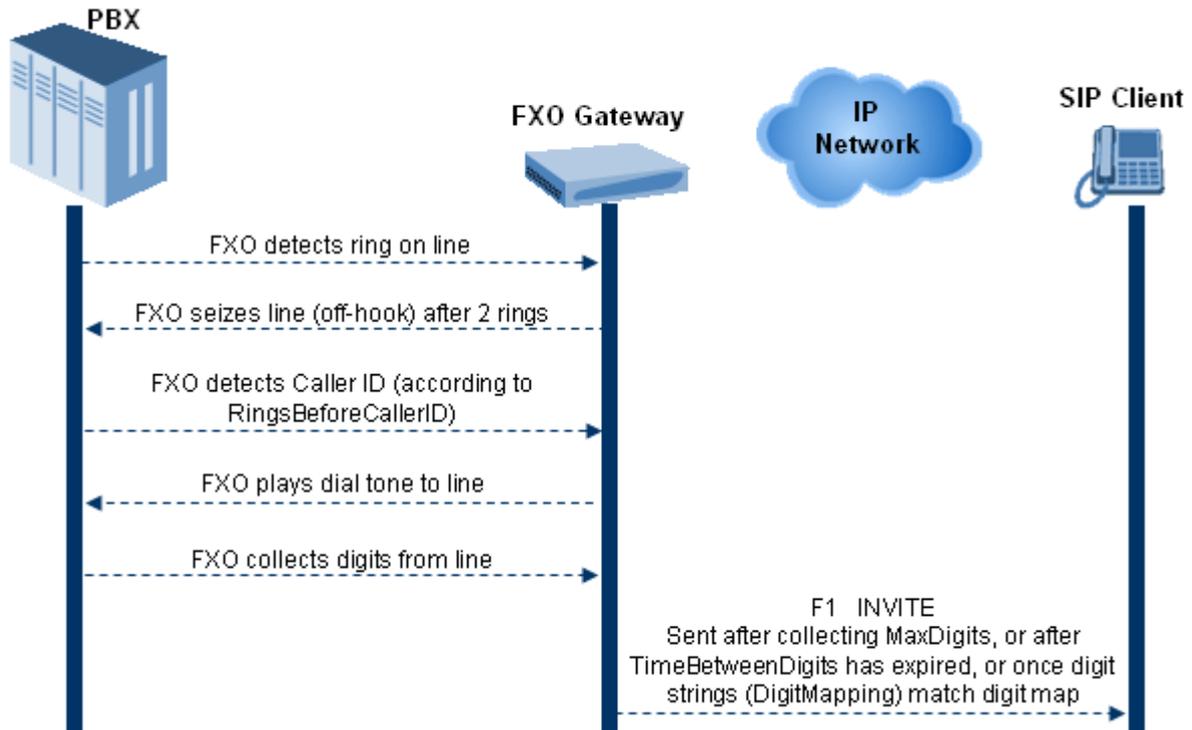
25.14.2.3   FXO Supplementary Services

The FXO supplementary services include the following:

■ **Hold / Transfer toward the Tel side:** The *ini* file parameter LineTransferMode must be set to 0 (default). If the FXO receives a hook-flash from the IP side (using out-of-band or RFC 2833), the device sends the hook-flash to the Tel side by performing one of the following:

- Performing a hook flash (i.e., on-hook and off-hook)
- Sending a hook-flash code (defined by the *ini* file parameter HookFlashCode)

The PBX may generate a dial tone that is sent to the IP, and the IP side may dial digits of a new destination.

■ **Blind Transfer to the Tel side:** A blind transfer is one in which the transferring phone connects the caller to a destination line before ringback begins. The *ini* file parameter LineTransferMode must be set to 1.

The blind transfer call process is as follows:

- FXO receives a REFER request from the IP side
- FXO sends a hook-flash to the PBX, dials the digits (that are received in the Refer-To header), and then drops the line (on-hook). Note that the time between flash to dial is according to the WaitForDialTime parameter.
- PBX performs the transfer internally

■ **Hold / Transfer toward the IP side:** The FXO device doesn't initiate hold / transfer as a response to input from the Tel side. If the FXO receives a REFER request (with or without replaces), it generates a new INVITE according to the Refer-To header.

25.14.3 Call Termination on FXO Devices

This section describes the device's call termination capabilities for its FXO interfaces:

■ Calls terminated by a PBX (see 'Call Termination by PBX' on page 318)

■ Calls terminated before call establishment (see 'Call Termination before Call Establishment' on page 319)

■ Ring detection timeout (see 'Ring Detection Timeout' on page 319)

25.14.3.1 Calls Termination by PBX

The FXO device supports various methods for identifying when a call has been terminated by the PBX.

The PBX doesn't disconnect calls, but instead signals to the device that the call has been disconnected using one of the following methods:

■ **Detection of polarity reversal/current disconnect:** The call is immediately disconnected after polarity reversal or current disconnect is detected on the Tel side (assuming the PBX/CO generates this signal). This is the recommended method.

Relevant parameters: EnableReversalPolarity, EnableCurrentDisconnect, CurrentDisconnectDuration, CurrentDisconnectDefaultThreshold, and TimeToSampleAnalogLineVoltage.

■ **Detection of Reorder, Busy, Dial, and Special Information Tone (SIT) tones:** The call is immediately disconnected after a Reorder, Busy, Dial, or SIT tone is detected on the Tel side (assuming the PBX / CO generates this tone). This method requires the correct tone frequencies and cadence to be defined in the Call Progress Tones file. If these frequencies are unknown, define them in the CPT file. The tone produced by the PBX / CO must be recorded and its frequencies analyzed.

Relevant parameters: DisconnectOnBusyTone and DisconnectOnDialTone.

■ **Detection of silence:** The call is disconnected after silence is detected on both call directions for a specific (configurable) amount of time. The call isn't disconnected immediately; therefore, this method should only be used as a backup option.

Relevant parameters: EnableSilenceDisconnect and FarEndDisconnectSilencePeriod.

> **Note:** This method is not supported by MP-124 Rev. E.

■ **Special DTMF code:** A digit pattern that when received from the Tel side, indicates to the device to disconnect the call.

Relevant *ini* file parameter: TelDisconnectCode.

■ **Interruption of RTP stream:** Relevant parameters: BrokenConnectionEventTimeout and DisconnectOnBrokenConnection.

> **Note:** This method operates correctly only if silence suppression is not used.

■ **Protocol-based termination of the call from the IP side**

⚠️

> **Note:** The implemented disconnect method must be supported by the CO or PBX.

25.14.3.2   Call Termination before Call Establishment

The device supports the following call termination methods before a call is established:

■ **Call termination upon receipt of SIP error response (in Automatic Dialing mode):** By default, when the FXO device operates in Automatic Dialing mode, there is no method to inform the PBX if a Tel-to-IP call has failed (SIP error response - 4xx, 5xx or 6xx - is received). The reason is that the FXO device does not seize the line until a SIP 200 OK response is received. Use the FXOAutoDialPlayBusyTone parameter to allow the device to play a busy / reorder tone to the PSTN line if a SIP error response is received. The FXO device seizes the line (off-hook) for the duration defined by the TimeForReorderTone parameter. After playing the tone, the line is released (on-hook).

■ **Call termination after caller (PBX) on-hooks phone (Ring Detection Timeout feature):** This method operates in one of the following manners:

• **Automatic Dialing is enabled:** if the remote IP party doesn't answer the call and the ringing signal (from the PBX) stops for a user-defined time (configured by the parameter FXOBetweenRingTime), the FXO device releases the IP call.

• **No automatic dialing and Caller ID is enabled:** the device seizes the line after detection of the second ring signal (allowing detection of caller ID sent between the first and the second rings). If the second ring signal is not received within this timeout, the device doesn't initiate a call to IP.

25.14.3.3   Ring Detection Timeout

The operation of Ring Detection Timeout depends on the following:

■ **Automatic dialing is disabled and Caller ID is enabled:** if the second ring signal is not received for a user-defined time (using the parameter FXOBetweenRingTime), the FXO device doesn't initiate a call to the IP.

■ **Automatic dialing is enabled:** if the remote party doesn't answer the call and the ringing signal stops for a user-defined time (using the parameter FXOBetweenRingTime), the FXO device releases the IP call.

Ring Detection Timeout supports full ring cycle of ring on and ring off (from ring start to ring start).

25.15 Remote PBX Extension between FXO and FXS Devices

Remote PBX extension offers a company the capability of extending the "power" of its local PBX by allowing remote phones (remote offices) to connect to the company's PBX over the IP network (instead of via PSTN). This is as if the remote office is located in the head office (where the PBX is installed). PBX extensions are connected through FXO ports to the IP network, instead of being connected to individual telephone stations. At the remote office, FXS units connect analog phones to the same IP network. To produce full transparency, each FXO port is mapped to an FXS port (i.e., one-to-one mapping). This allows individual extensions to be extended to remote locations. To call a remote office worker, a PBX user or a PSTN caller simply dials the PBX extension that is mapped to the remote FXS port.

This section provides an example on how to implement a remote telephone extension through the IP network, using 8-port FXO and 8-port FXS interfaces. In this configuration,

the FXO device routes calls received from the PBX to the 'Remote PBX Extension' connected to the FXS device. The routing is transparent as if the telephone connected to the FXS device is directly connected to the PBX.

The following is required:

- One FXO interfaces with ports connected directly to the PBX lines (shown in the figure below)
- One FXS interfaces for the 'remote PBX extension'
- Analog phones (POTS)
- PBX (one or more PBX loop start lines)
- LAN network

Figure 25-14: FXO-FXS Remote PBX Extension (Example)
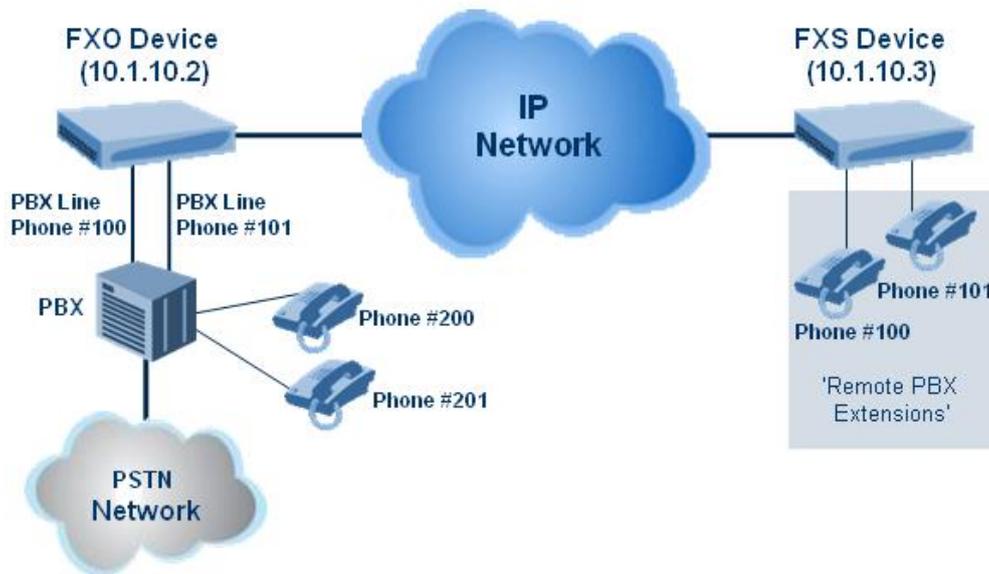
25.15.1 Dialing from Remote Extension (Phone at FXS)

The procedure below describes how to dial from the 'remote PBX extension' (i.e., phone connected to the FXS interface).

➢ **To make a call from the FXS interface:**

1. Off-hook the phone and wait for the dial tone from the PBX. This is as if the phone is connected directly to the PBX. The FXS and FXO interfaces establish a voice path connection from the phone to the PBX immediately after the phone is off-hooked.

2. Dial the destination number (e.g., phone number 201). The DTMF digits are sent over IP directly to the PBX. All the audible tones are generated from the PBX (such as ringback, busy, or fast busy tones). One-to-one mapping occurs between the FXS ports and PBX lines.

3. The call disconnects when the phone connected to the FXS goes on-hook.

25.15.2 Dialing from PBX Line or PSTN

The procedure below describes how to dial from a PBX line (i.e., from a telephone directly connected to the PBX) or from the PSTN to the 'remote PBX extension' (i.e., telephone connected to the FXS interface).

➢ **To dial from a telephone directly connected to the PBX or from the PSTN:**

■ Dial the PBX subscriber number (e.g., phone number 101) in the same way as if the user's phone was connected directly to the PBX. As soon as the PBX rings the FXO device, the ring signal is 'sent' to the phone connected to the FXS device. Once the phone connected to the FXS device is off-hooked, the FXO device seizes the PBX line and the voice path is established between the phone and PBX.

There is one-to-one mapping between PBX lines and FXS device ports. Each PBX line is routed to the same phone (connected to the FXS device). The call disconnects when the phone connected to the FXS device is on-hooked.

25.15.3 Message Waiting Indication for Remote Extensions

The device supports the relaying of Message Waiting Indications (MWI) for remote extensions (and voice mail applications). Instead of subscribing to an MWI server to receive notifications of pending messages, the FXO device receives subscriptions from the remote FXS device and notifies the appropriate extension when messages (and the number of messages) are pending.

The FXO device detects an MWI message from the Tel (PBX) side using any one of the following methods:

■ 100 VDC (sent by the PBX to activate the phone's lamp)

■ Stutter dial tone from the PBX

■ MWI display signal (according to the parameter CallerIDType)

Upon detection of an MWI message, the FXO device sends a SIP NOTIFY message to the IP side. When receiving this NOTIFY message, the remote FXS device generates an MWI signal toward its Tel side.

Figure 25-15: MWI for Remote Extensions
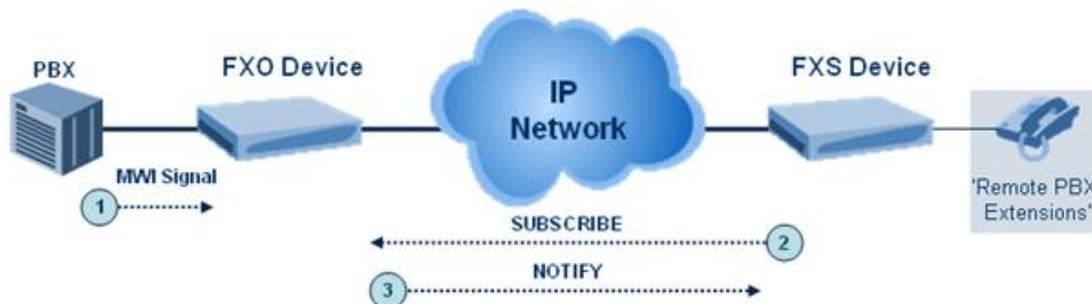
25.15.4 Call Waiting for Remote Extensions

When the FXO device detects a Call Waiting indication (FSK data of the Caller Id - CallerIDType2) from the PBX, it sends a proprietary INFO message, which includes the caller identification to the FXS device. Once the FXS device receives this INFO message, it plays a call waiting tone and sends the caller ID to the relevant port for display. The remote extension connected to the FXS device can toggle between calls using the Hook Flash button.

Figure 25-16: Call Waiting for Remote Extensions

25.15.5 FXS Gateway Configuration

The procedure below describes how to configure the FXS interface (at the 'remote PBX extension').

➢ **To configure the FXS interface:**

1. In the Endpoint Phone Numbers page (see Configuring Endpoint Phone Numbers on page 227, assign the phone numbers 100 to 107 to the device's endpoints.

	Channel(s)	Phone Number	Hunt Group ID
1	1-8	100	
2			

2. In the Automatic Dialing page (see 'Configuring Automatic Dialing' on page 303), enter the phone numbers of the FXO device in the 'Destination Phone Number' fields. When a phone connected to Port #1 off-hooks, the FXS device automatically dials the number '200'.

Gateway Port	Destination Phone Number	Auto Dial Status
Port 1 FXS	200	Enable
Port 2 FXS	201	Enable
Port 3 FXS	202	Enable
Port 4 FXS	203	Enable
Port 5 FXS	204	Enable
Port 6 FXS	205	Enable
Port 7 FXS	206	Enable
Port 8 FXS	206	Enable

3. In the Tel to IP Routing page (see 'Configuring Tel to IP Routing' on page 250), enter 20 for the destination phone prefix, and 10.1.10.2 for the IP address of the FXO device.

	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address
1	20	*		10.1.10.2

> **Note:** For the transfer to function in remote PBX extensions, Hold must be disabled at the FXS device (i.e., Enable Hold = 0) and hook-flash must be transferred from the FXS to the FXO (HookFlashOption = 4).

25.15.6 FXO Gateway Configuration

The procedure below describes how to configure the FXO interface (to which the PBX is directly connected).

> **To configure the FXO interface:**

1. In the Endpoint Phone Numbers page (see Configuring Endpoint Phone Numbers on page 227, assign the phone numbers 200 to 207 to the device's FXO endpoints.

	Channel(s)	Phone Number	Hunt Group ID
1	1-8	200	

2. In the Automatic Dialing page, enter the phone numbers of the FXS device in the 'Destination Phone Number' fields. When a ringing signal is detected at Port #1, the FXO device automatically dials the number '100'.

Gateway Port	Destination Phone Number	Auto Dial Status
Port 1 FXO	100	Enable
Port 2 FXO	101	Enable
Port 3 FXO	102	Enable
Port 4 FXO	103	Enable
Port 5 FXO	104	Enable
Port 6 FXO	105	Enable
Port 7 FXO	106	Enable
Port 8 FXO	107	Enable

3. In the Tel to IP Routing page, enter 10 in the 'Destination Phone Prefix' field, and the IP address of the FXS device (10.1.10.3) in the field 'IP Address'.

Figure 25-17: FXO Tel-to-IP Routing Configuration

	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address
1	10	*		10.1.10.3

4. In the FXO Settings page (see 'Configuring FXO Parameters' on page 301), set the parameter 'Dialing Mode' to **Two Stages** (IsTwoStageDial = 1).

Part VI

Stand-Alone Survivability Application

26   SAS Overview

The device's Stand-Alone Survivability (SAS) feature ensures telephony communication continuity (survivability) for enterprises using hosted IP services (such as IP Centrex) or IP-PBX in cases of failure of these entities. In case of failure of the IP Centrex, IP-PBX servers (or even WAN connection and access Internet modem), the enterprise typically loses its internal telephony service at any branch, between its offices, and with the external environment. Typically, these failures also lead to the inability to make emergency calls (e.g., 911 in North America). Despite these possible points of failure, the device's SAS feature ensures that the enterprise's telephony services (e.g., SIP IP phones or soft phones) are maintained, by routing calls to the PSTN (i.e., providing PSTN fallback).

> **Notes:**
>
> - Throughput this section, the term *user agent* (UA) refers to the enterprise's LAN phone user (i.e., SIP telephony entities such as IP phones).
> - Throughout this section, the term *proxy* or *proxy server* refers to the enterprise's centralized IP Centrex or IP-PBX.
> - Throughout this section, the term *SAS* refers to the SAS application running on the device.

26.1   SAS Operating Modes

The device's SAS application can be implemented in one of the following main modes:

- **Outbound Proxy:** In this mode, SAS receives SIP REGISTER requests from the enterprise's UAs and forwards these requests to the external proxy (i.e., outbound proxy). When a connection with the external proxy fails, SAS enters SAS emergency state and serves as a proxy, by handling internal call routing for the enterprise's UAs - routing calls between UAs and if setup, routing calls between UAs and the PSTN. For more information, see 'SAS Outbound Mode' on page 327.
- **Redundant Proxy:** In this mode, the enterprise's UAs register with the external proxy and establish calls directly through the external proxy, without traversing SAS (or the device per se'). Only when connection with the proxy fails, do the UAs register with SAS, serving now as the UAs redundant proxy. SAS then handles the calls between UAs, and between the UAs and the PSTN (if setup). This mode is operational only during SAS in emergency state. This mode can be implemented, for example, for proxies that accept only SIP messages that are sent directly from the UAs. For more information, see 'SAS Redundant Mode' on page 329.

> **Note:** It is recommended to implement the SAS outbound mode.

26.1.1   SAS Outbound Mode

This section describes the SAS outbound mode, which includes the following states:

- Normal state (see 'Normal State' on page 328)
- Emergency state (see 'Emergency State' on page 328)

26.1.1.1 Normal State

In normal state, SAS receives REGISTER requests from the enterprise's UAs and forwards them to the external proxy (i.e., outbound proxy). Once the proxy replies with a SIP 200 OK, the device records the Contact and address of record (AOR) of the UAs in its internal SAS registration database. Therefore, in this mode, SAS maintains a database of all the registered UAs in the network. SAS also continuously maintains a keep-alive mechanism toward the external proxy, using SIP OPTIONS messages. The figure below illustrates the operation of SAS outbound mode in normal state:

Figure 26-1: SAS Outbound Mode in Normal State (Example)
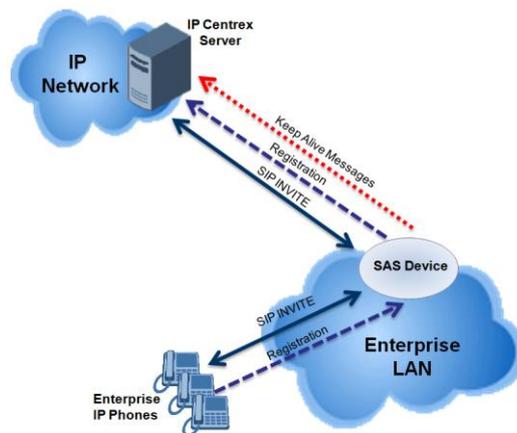
26.1.1.2 Emergency State

When a connection with the external proxy fails (detected by the device's keep-alive messages), the device enters SAS emergency state. The device serves as a proxy for the UAs, by handling internal call routing of the UAs (within the LAN enterprise).

> **Note:** SAS can also enter Emergency state if no response is received from the proxy for sent OPTIONS, INVITE, or REGISTER messages. To configure this, set the SASEnteringEmergencyMode parameter to 1.

When the device receives calls (INVITE), it searches its SAS registration database to locate the destination address (according to AOR or Contact). If the destination address is not found, SAS forwards the call to the default gateway. Typically, the default gateway is defined as the device itself (on which SAS is running), and if the device has PSTN interfaces, the enterprise preserves its capability for outgoing calls (from UAs to the PSTN network).

For incoming REGISTER messages, the device by default only adds the UA to its internal registration database. Alternativey, you can enable the device to route REGISTER messages using the SAS IP-to-IP Routing table during Emergency state. To configure this, see Section Routing REGISTERs in Emergency by IP-to-IP Routing Table on page 351.

The routing logic of SAS in emergency state is described in detail in 'SAS Routing in Emergency State' on page 333.

The figure below illustrates the operation of SAS outbound mode in emergency state:

Figure 26-2: SAS Outbound Mode in Emergency State (Example)
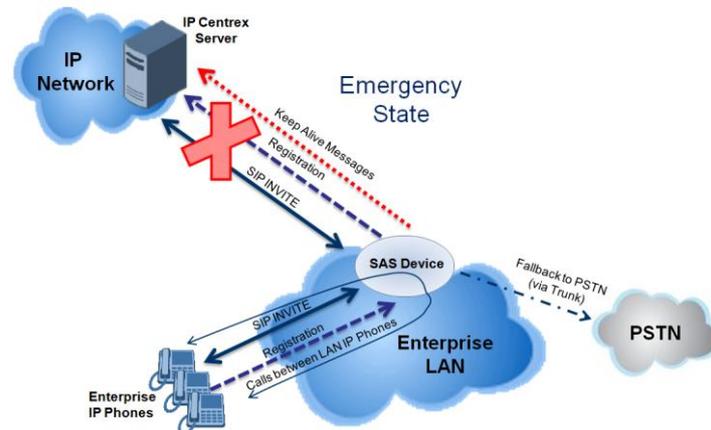
When emergency state is active, SAS continuously attempts to communicate with the external proxy, using keep-alive SIP OPTIONS. Once connection to the proxy returns, the device exits SAS emergency state and returns to SAS normal state, as explained in 'Exiting Emergency and Returning to Normal State' on page 330.

26.1.2   SAS Redundant Mode

In SAS redundant mode, the enterprise's UAs register with the external proxy and establish calls directly through it, without traversing SAS (or the device per se'). Only when connection with the proxy fails, do the UAs register with SAS, serving now as the UAs redundant proxy. SAS then handles the calls between UAs, and between the UAs and the PSTN (if setup).

This mode is operational only during SAS in emergency state.

> **Note:**   In this SAS deployment, the UAs (e.g., IP phones) must support configuration for primary and secondary proxy servers (i.e., proxy redundancy), as well as homing. Homing allows the UAs to switch back to the primary server from the secondary proxy once the connection to the primary server returns (UAs check this using keep-alive messages to the primary server). If homing is not supported by the UAs, you can configure SAS to ignore messages received from UAs in normal state (the 'SAS Survivability Mode' parameter must be set to 'Always Emergency' / 2) and thereby, "force" the UAs to switch back to their primary proxy.

26.1.2.1 Normal State

In normal state, the UAs register and operate directly with the external proxy.

Figure 26-3: SAS Redundant Mode in Normal State (Example)
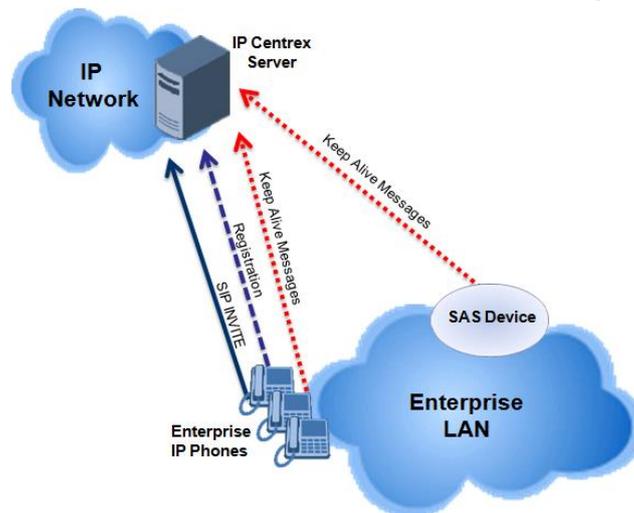
26.1.2.2 Emergency State

If the UAs detect that their primary (external) proxy does not respond, they immediately register to SAS and start routing calls to it.

Figure 26-4: SAS Redundant Mode in Emergency State (Example)
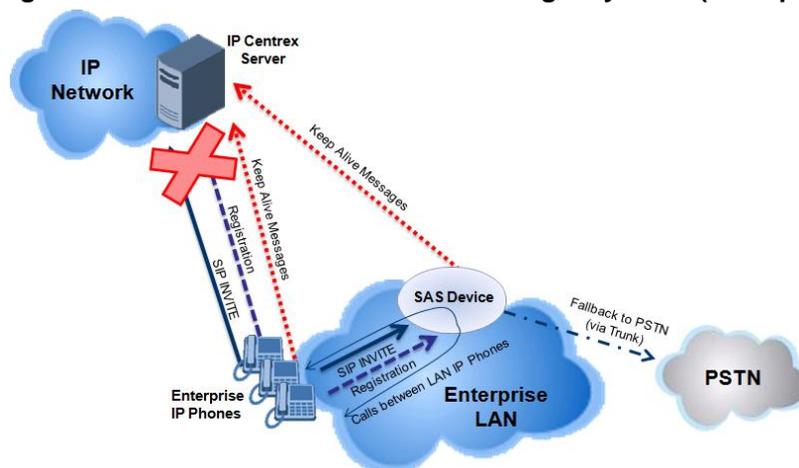
26.1.2.3 Exiting Emergency and Returning to Normal State

Once the connection with the primary proxy is re-established, the following occurs:

■ **UAs:** Switch back to operate with the primary proxy.

■ **SAS:** Ignores REGISTER requests from the UAs, forcing the UAs to switch back to the primary proxy.

Note: This is applicable only if the 'SAS Survivability Mode' parameter is set to 'Always Emergency' (2).

26.2   SAS Routing

This section provides flowcharts describing the routing logic for SAS in normal and emergency states.

26.2.1   SAS Routing in Normal State

The flowchart below displays the routing logic for SAS in normal state for INVITE messages received from UAs:

Figure 26-5: Flowchart of INVITE from UA's in SAS Normal State
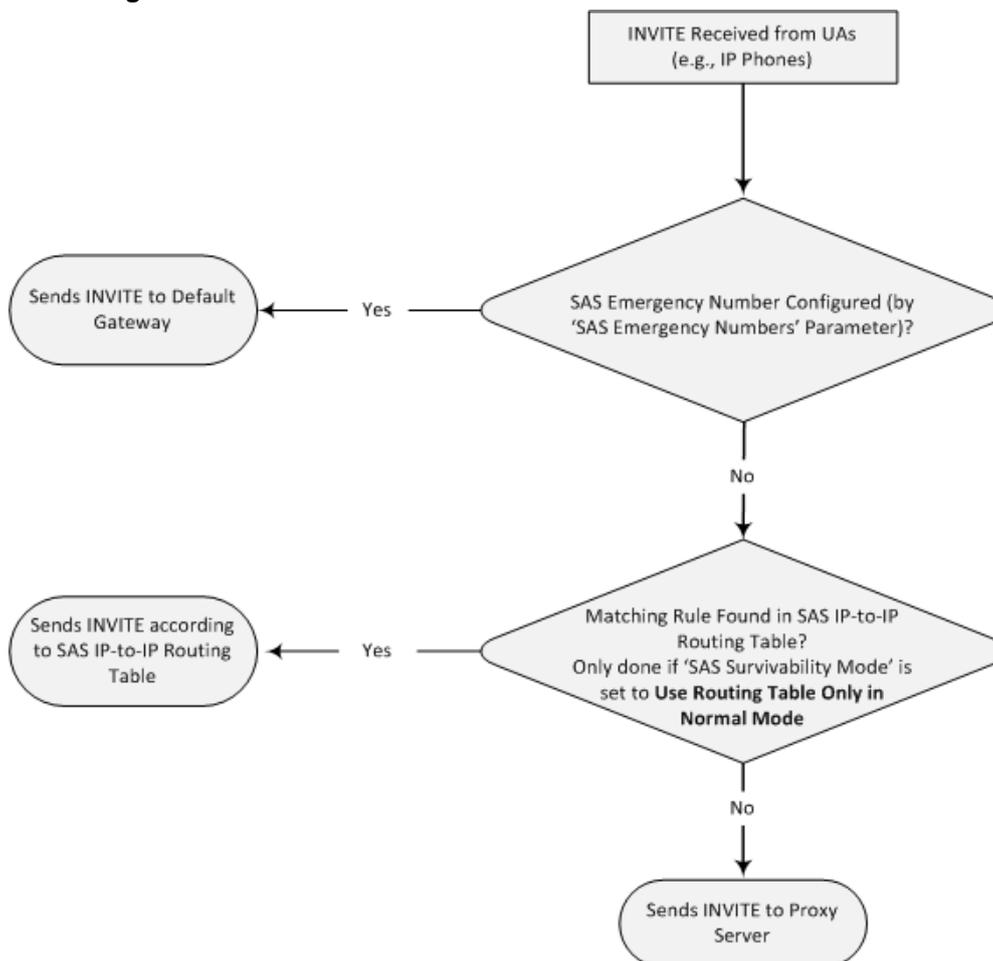
The flowchart below displays the routing logic for SAS in normal state for INVITE messages received from the external proxy:

Figure 26-6: Flowchart of INVITE from Primary Proxy in SAS Normal State
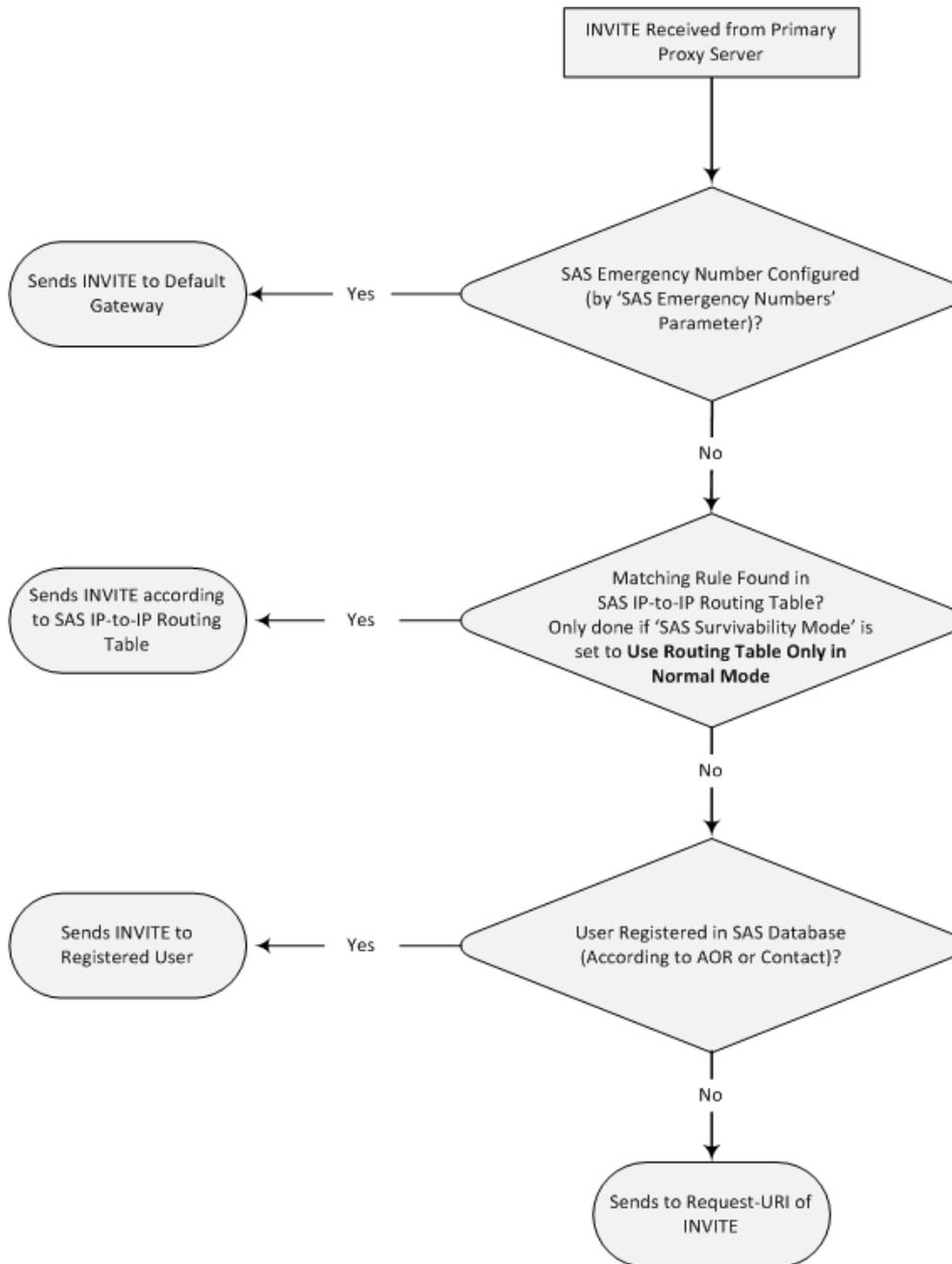
26.2.2   SAS Routing in Emergency State

The flowchart below shows the routing logic of INVITE messages when SAS is in emergency state:

Figure 26-7: Flowchart for Routing INVITEs During SAS Emergency State
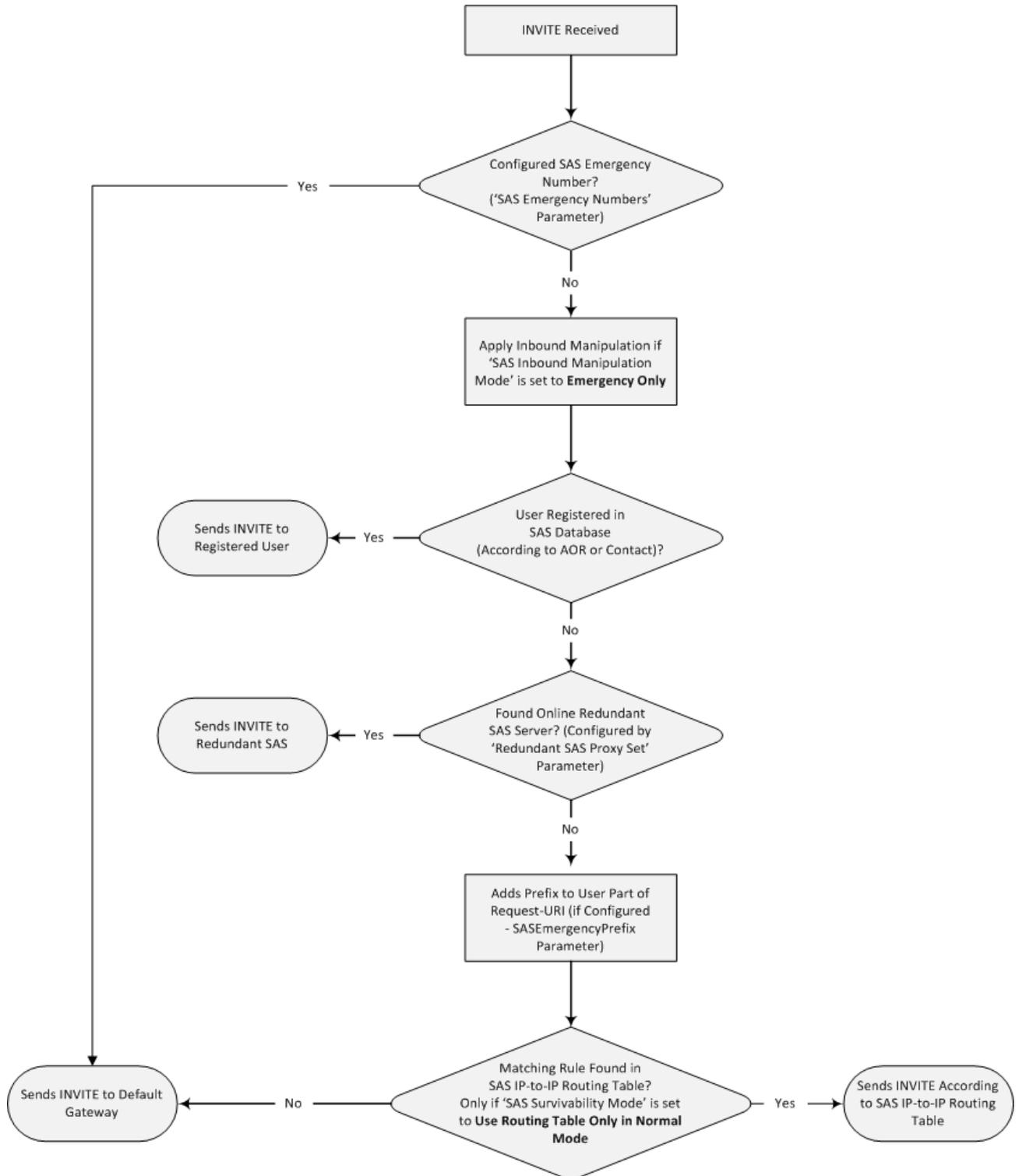
For incoming REGISTER messages, the device by default only adds the UA to its internal registration database. Alternativey, you can enable the device to route REGISTER

messages using the SAS IP-to-IP Routing table during Emergency state. To configure this, see Section Routing REGISTERs in Emergency by IP-to-IP Routing Table on page .

27   SAS Configuration

SAS supports various configuration possibilities, depending on how the device is deployed in the network and the network architecture requirements. This section provides step-by-step procedures on configuring the SAS application, using the device's Web interface.

27.1   General SAS Configuration

This section describes the general configuration required for the SAS application. This configuration is applicable to all SAS modes.

27.1.1   Enabling the SAS Application

Before you can configure SAS, you need to enable the SAS application on the device. Once enabled, the **SAS** menu and related pages appear in the device's Web interface.

➢   **To enable the SAS application:**

1.   Open the Applications Enabling page (**Configuration** tab > **VoIP** menu > **Applications Enabling** > **Applications Enabling**).

2.   From the 'SAS Application' drop-down list, select **Enable**.
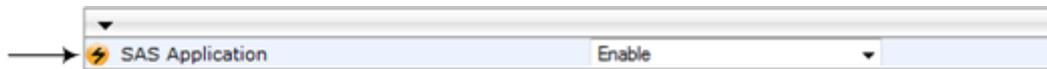
3.   Click **Submit**.

4.   Save the changes to the flash memory with a device reset.

27.1.2   Configuring Common SAS Parameters

The procedure below describes how to configure SAS settings that are common to all SAS modes. This includes various SAS parameters as well as configuring the Proxy Set for the SAS proxy (if required). The SAS Proxy Set ID defines the address of the UAs' external proxy.

➢   **To configure common SAS settings:**

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

2.   Define the port used for sending and receiving SAS messages. This can be any of the following port types:

- UDP port - defined in the 'SAS Local SIP UDP Port' field
- TCP port - defined in the 'SAS Local SIP TCP Port' field
- TLS port - defined in the 'SAS Local SIP TLS Port' field

> **Note:**   This SAS port must be different than the device's local gateway port (i.e., that defined for the 'SIP UDP/TCP/TLS Local Port' parameter in the SIP General Parameters page - **Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**).

3.   In the 'SAS Default Gateway IP' field, define the IP address and port (in the format *x.x.x.x:port)* of the device (i.e., Gateway application). Note that the port of the device is defined by the parameter 'SIP UDP Local Port' (refer to the note in Step 2 above).

4. In the 'SAS Registration Time' field, define the value for the SIP Expires header, which is sent in the 200 OK response to an incoming REGISTER message when SAS is in emergency state.

5. From the 'SAS Binding Mode' drop-down list, select the database binding mode:

   • **0-URI:** If the incoming AOR in the REGISTER request uses a 'tel:' URI or 'user=phone', the binding is done according to the Request-URI user part only. Otherwise, the binding is done according to the entire Request-URI (i.e., user and host parts - user@host).

   • **1-User Part Only:** Binding is done according to the user part only.

   You must select **1-User Part Only** in cases where the UA sends REGISTER messages as SIP URI, but the INVITE messages sent to this UA include a Tel URI. For example, when the AOR of an incoming REGISTER is sip:3200@domain.com, SAS adds the entire SIP URI (e.g., sip:3200@domain.com) to its database (when the parameter is set to '0-URI'). However, if a subsequent Request-URI of an INVITE message for this UA arrives with sip:3200@10.1.2.3 user=phone, SAS searches its database for "3200", which it does not find. Alternatively, when this parameter is set to '1-User Part Only', then upon receiving a REGISTER message with sip:3200@domain.com, SAS adds only the user part (i.e., "3200") to its database. Therefore, if a Request-URI of an INVITE message for this UA arrives with sip:3200@10.1.2.3 user=phone, SAS can successfully locate the UA in its database.

Figure 27-1: Configuring Common Settings

6. In the 'SAS Proxy Set' field, enter the Proxy Set used for SAS. The SAS Proxy Set must be defined only for the following SAS modes:

   • **Outbound mode:** In SAS normal state, SAS forwards REGISTER and INVITE messages received from the UAs to the proxy servers defined in this Proxy Set.

   • **Redundant mode and only if UAs don't support homing:** SAS sends keep-alive messages to this proxy and if it detects that the proxy connection has resumed, it ignores the REGISTER messages received from the UAs, forcing them to send their messages directly to the proxy.

   If you define a SAS Proxy Set ID, you must configure the Proxy Set as described in Step 8 below.

7. Click **Submit** to apply your settings.

8. If you defined a SAS Proxy Set ID in Step 6 above, then you must configure the SAS Proxy Set ID:

   a. Open the Proxy Sets Table page (**Configuration** tab > **VoIP** menu > **Control Networks** > **Proxy Set Table**).

   b. From the 'Proxy Set ID' drop-down list, select the required Proxy Set ID.

> **Notes:**
>
> - The selected Proxy Set ID number must be the same as that specified in the 'SAS Proxy Set' field in the 'SAS Configuration page (see Step 6).
> - Do not use Proxy Set ID 0.

   c. In the 'Proxy Address' field, enter the IP address of the external proxy server.

   d. From the 'Enable Proxy Keep Alive' drop-down list, select **Using Options**. This instructs the device to send SIP OPTIONS messages to the proxy for the keep-alive mechanism.

Figure 27-2: Defining SAS Proxy Server

   e. Click **Submit** to apply your settings.

27.2 Configuring SAS Outbound Mode

This section describes how to configure the SAS outbound mode. These settings are in addition to the ones described in 'Configuring Common SAS Parameters' on page 335.

> **Note:** The VoIP CPEs (such as IP phones or residential gateways) need to be defined so that their proxy and registrar destination addresses and ports are the same as that configured for the device's SAS IP address and SAS local SIP port. In some cases, on the UAs, it is also required to define SAS as their outbound proxy, meaning that messages sent by the UAs include the host part of the external proxy, but are sent (on Layer 3/4) to the IP address / UDP port of SAS.

> ➢ **To configure SAS outbound mode:**

1. Open the SAS Configuration page (**Configuration** tab > **VoIP** menu > **SAS** > **Configuration**).
2. From the 'SAS Survivability Mode' drop-down list, select **Standard**.
3. Click **Submit**.

27.3 Configuring SAS Redundant Mode

This section describes how to configure the SAS redundant mode. These settings are in addition to the ones described in 'Configuring Common SAS Parameters' on page 335.

> **Note:** The VoIP CPEs (such as IP phones or residential gateways) need to be defined so that their primary proxy is the external proxy, and their redundant proxy destination addresses and port is the same as that configured for the device's SAS IP address and SAS SIP port.

> ➢ **To configure SAS redundant mode:**

1. Open the SAS Configuration page (**Configuration** tab > **VoIP** menu > **SAS** > **Configuration**).
2. From the 'SAS Survivability Mode' drop-down list, select one of the following, depending on whether the UAs support homing (i.e., they always attempt to operate with the primary proxy, and if using the redundant proxy, they switch back to the primary proxy whenever it's available):
   - **UAs support homing:** Select **Always Emergency**. This is because SAS does not need to communicate with the primary proxy of the UAs; SAS serves only as the redundant proxy of the UAs. When the UAs detect that their primary proxy is available, they automatically resume communication with it instead of with SAS.
   - **UAs do not support homing:** Select **Ignore REGISTER**. SAS uses the keep-alive mechanism to detect availability of the primary proxy (defined by the SAS Proxy Set). If the connection with the primary proxy resumes, SAS ignores the messages received from the UAs, forcing them to send their messages directly to the primary proxy.
3. Click **Submit**.

27.4 Configuring Gateway Application with SAS

If you want to run both the Gateway and SAS applications on the device, the configuration described in this section is required. The configuration steps depend on whether the Gateway application is operating with SAS in outbound mode or SAS in redundant mode.

> **Note:** The Gateway application must use the same SAS operation mode as the SIP UAs. For example, if the UAs use the SAS application as a redundant proxy (i.e., SAS redundancy mode), then the Gateway application must do the same.

27.4.1 Gateway with SAS Outbound Mode

The procedure below describes how to configure the Gateway application with SAS outbound mode.

➢ **To configure Gateway application with SAS outbound mode:**

1. Define the proxy server address for the Gateway application:

   a. Open the Proxy & Registration page (**Configuration** tab > **VoIP** menu > **SIP Definitions** submenu > **Proxy & Registration**).

   b. From the 'Use Default Proxy' drop-down list, select **Yes**.

   Figure 27-3: Enabling Proxy Server for Gateway Application
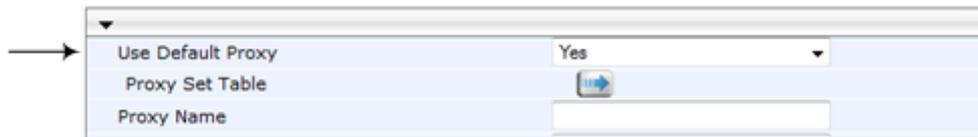
   c. Click **Submit**.

   d. Open the Proxy Sets Table page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **Proxy Sets** Table).

   e. From the 'Proxy Set ID' drop-down list, select **0**.

   f. In the first 'Proxy Address' field, enter the IP address and port of the device (in the format *x.x.x.x:port*). This is the port as defined in the 'SAS Local UDP/TCP/TLS Port' field (see 'Configuring Common SAS Parameters' on page 335).

   Figure 27-4: Defining Proxy Server for Gateway Application
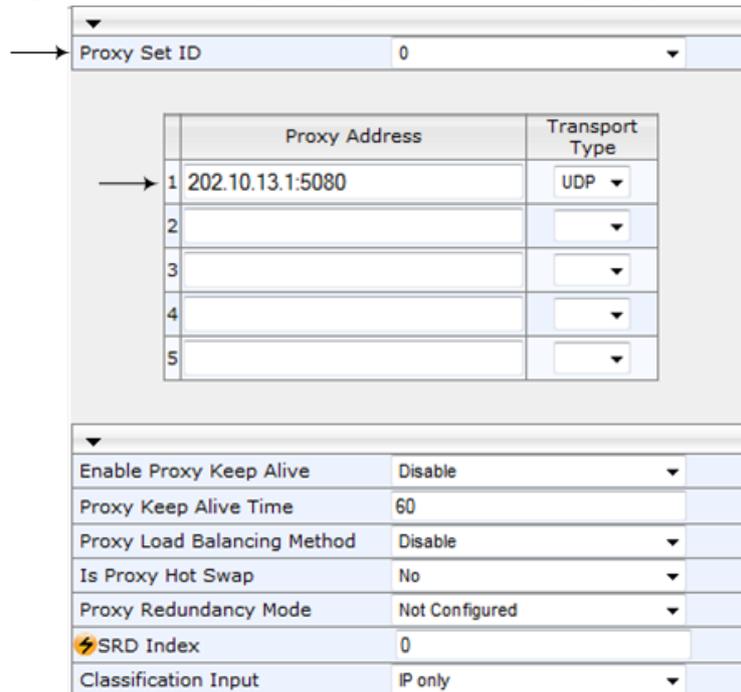
   g. Click **Submit**.

2. Disable use of user=phone in SIP URL:

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

b. From the 'Use user=phone in SIP URL' drop-down list, select **No**. This instructs the Gateway application not to use *user=phone* in the SIP URL and therefore, REGISTER and INVITE messages use SIP URI. (By default, REGISTER messages are sent with *sip uri* and INVITE messages with *tel uri*.)

Figure 27-5: Disabling user=phone in SIP URL

SIP General	
⚡ NAT IP Address	0.0.0.0
PRACK Mode	Supported
Channel Select Mode	Cyclic Ascending
Enable Early Media	Disable
183 Message Behavior	Progress
Session-Expires Time	0
Minimum Session-Expires	90
Session Expires Method	Re-INVITE
Asserted Identity Mode	Disabled
Fax Signaling Method	No Fax
Detect Fax on Answer Tone	Initiate T.38 on Preamble
SIP Transport Type	UDP
SIP UDP Local Port	5060
SIP TCP Local Port	5060
SIP TLS Local Port	5061
Enable SIPS	Disable
Enable TCP Connection Reuse	Enable
TCP Timeout	0
SIP Destination Port	5060
→ Use user=phone in SIP URL	No

c. Click **Submit**.

27.4.2 Gateway with SAS Redundant Mode

The procedure below describes how to configure the Gateway application with SAS redundant mode.

➤ **To configure Gateway application with SAS redundant mode:**

1. Define the proxy servers for the Gateway application:

a. Open the Proxy & Registration page (**Configuration** tab > **VoIP** menu > **SIP Definitions** submenu > **Proxy & Registration**).

b. From the 'Use Default Proxy' drop-down list, select **Yes**.

Figure 27-6: Enabling Proxy Server for Gateway Application

→ Use Default Proxy	Yes
Proxy Set Table	⟶
Proxy Name	

c. Click **Submit**.

d. Open the Proxy Sets Table page (**Configuration** tab > **VoIP** menu > **Control Network** submenu > **Proxy Sets Table**).

e. From the 'Proxy Set ID' drop-down list, select **0**.

f. In the first 'Proxy Address' field, enter the IP address of the external proxy server.

g. In the second 'Proxy Address' field, enter the IP address and port of the device (in the format *x.x.x.x:port*). This is the same port as defined in the 'SAS Local UDP/TCP/TLS Port' field (see 'Configuring Common SAS Parameters' on page 335).

h. From the 'Proxy Redundancy Mode' drop-down list, select **Homing**.

Figure 27-7: Defining Proxy Servers for Gateway Application

i. Click **Submit**.

2. Disable the use of *user=phone* in the SIP URL:

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

b. From the 'Use user=phone in SIP URL' drop-down list, select **No**. This instructs the Gateway application not to use *user=phone* in SIP URL and therefore, REGISTER and INVITE messages use SIP URI. (By default, REGISTER messages are sent with *sip uri* and INVITE messages with *tel uri*.)

c. Click **Submit**.

27.5 Advanced SAS Configuration

This section describes the configuration of advanced SAS features that can optionally be implemented in your SAS deployment.

27.5.1 Manipulating URI user part of Incoming REGISTER and/or INVITE

There are scenarios in which the UAs register to the proxy server with their full phone number (for example, "976653434"), but can receive two types of INVITE messages (calls):

■ INVITEs whose destination is the UAs' full number (when the call arrives from outside the enterprise)

■ INVITES whose destination is the last four digits of the UAs' phone number ("3434" in our example) when it is an internal call within the enterprise

Therefore, it is important that the device registers the UAs in the SAS registered database with their extension numbers (for example, "3434") in addition to their full numbers. To do this, you can define a manipulation rule to manipulate the SIP Request-URI user part of the AOR (in the To header) in incoming REGISTER requests. Once manipulated, it is saved in this manipulated format in the SAS registered users database in addition to the original (un-manipulated) AOR. Alternatively, you can register the user with the original contact details of the REGISTER message, but then configure a manipulation rule to manipulate the INVITE messages so that the device can locate the correct user in its registration database.

For example: Assume the following incoming REGISTER message is received and that you want to register in the SAS database the UA's full number as well as the last four digits from the right of the SIP URI user part:

```
REGISTER sip:10.33.38.2 SIP/2.0
Via: SIP/2.0/UDP 10.33.4.226:5050;branch=z9hG4bKac10827
Max-Forwards: 70
From: <sip: 976653434@10.33.4.226>;tag=1c30219
To: <sip: 976653434@10.33.4.226>
Call-ID: 16844@10.33.4.226
CSeq: 1 REGISTER
Contact: <sip: 976653434@10.10.10.10:5050>;expires=180
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUB
SCRIBE,UPDATE
Expires: 180
User-Agent: Audiocodes-Sip-Gateway-/v.
Content-Length: 0
```

After manipulation, SAS registers the user in its database as follows:

■ **AOR:** 976653434@10.33.4.226

■ **Associated AOR:** 3434@10.33.4.226 (after manipulation, in which only the four digits from the right of the URI user part are retained)

■ **Contact:** 976653434@10.10.10.10

➢ **To manipulate incoming Request-URI user part of REGISTER/INVITE message:**

1. Open the SAS Registration Manipulation table (**Configuration** tab > **VoIP** menu > **SAS** > **Registration Manipulation Table**).

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

Add Record	✕
Index	0
Remove From Right	0
Leave From Right	0
Rule Apply To	Register and Invite ▼
	🖫 Submit ✕ Cancel

3. Configure the rule as required. For a description of the parameters, see the table below.

4. Click **Submit** to apply your changes.

Table 27-1: SAS Registration Manipulation Table Parameter Description

Parameter	Description
Index [SASRegistrationManipulation_Index]	Defines the table index entry. **Note:** You can configure up to three SAS Registration Manipulation rules.
Remove From Right [SASRegistrationManipulation_RemoveFromRight]	Defines the number of digits (e.g., "4") to remove from the right side of the user part; all other digits in the user part are kept.
Leave From Right [SASRegistrationManipulation_LeaveFromRight]	Defines the number of digits (e.g., "4") to leave from the right side of the user part; all other digits in the user part are removed.
Rule Apply To [SASRegistrationManipulation_RuleApplyTo]	Defines the type of SIP message you want to manipulate. ▪ [0] Register and Invite (default) ▪ [1] Register Only ▪ [2] Invite Only

⚠️ **Note:** The device first does manipulation according to the 'Remove From Right' parameter and only then according to the 'Leave From Right' parameter.

27.5.2 Manipulating Destination Number of Incoming INVITE

You can define a manipulation rule to manipulate the destination number in the Request-URI of incoming INVITE messages when SAS is in emergency state. This is required, for example, if the call is destined to a registered user but the destination number in the received INVITE is not the number assigned to the registered user in the SAS registration database. To overcome this and successfully route the call, you can define manipulation rules to change the INVITE's destination number so that it matches that of the registered user in the database. This is done using the IP to IP Inbound Manipulation table.

For example, in SAS emergency state, assume an incoming INVITE has a destination number "7001234" which is destined to a user registered in the SAS database as

"552155551234". In this scenario, the received destination number needs to be manipulated to the number "552155551234". The outgoing INVITE sent by the device then also contains this number in the Request-URI user part.

In normal state, the numbers are not manipulated. In this state, SAS searches the number 552155551234 in its database and if found, it sends the INVITE containing this number to the UA.

➢ **To manipulate the destination number in SAS emergency state:**

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

2. From the 'SAS Inbound Manipulation Mode' (*SASInboundManipulationMode*) drop-down list, select **Emergency Only**.

3. Click **Submit**; the **SAS Inbound Manipulation Mode Table** button appears on the page.

4. Click this button to open the IP to IP Inbound Manipulation page.

5. Add your SAS manipulation rule as required. See the table below for descriptions of the parameters.

6. Click **Submit** to save your changes.

Notes:

- The following fields in the IP to IP Inbound Manipulation table are not applicable to SAS and must be left at their default values:
  - 'Additional Manipulation' - default is **0**
  - 'Manipulation Purpose' - default is **Normal**
  - 'Source IP Group' - default is **-1**
- The IP to IP Inbound Manipulation table can also be configured using the table ini file parameter, IPInboundManipulation.

Table 27-2: SAS IP to IP Inbound Manipulation Parameters

Parameter	Description
Matching Characteristics (Rule)	
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 **[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.

Parameter	Description
Source IP Group ID **[IPInboundManipulation_SrcIpGroup]**	Defines the IP Group from where the incoming INVITE is received. For any IP Group, enter the value "-1".
Source Username Prefix **[IPInboundManipulation_SrcUsernamePrefix]**	Defines the prefix of the source SIP URI user name (usually in the From header). For any prefix, enter the asterisk "*" symbol (default). **Note:** The prefix can be a single digit or a range of digits. For available notations, see 'Dialing Plan Notation for Routing and Manipulation' on page 469.
Source Host **[IPInboundManipulation_SrcHost]**	Defines the source SIP URI host name - full name (usually in the From header). For any host name, enter the asterisk "*" symbol (default).
Destination Username Prefix **[IPInboundManipulation_DestUsernamePrefix]**	Defines the prefix of the destination SIP URI user name (usually in the Request-URI). For any prefix, enter the asterisk "*" symbol (default). **Note:** The prefix can be a single digit or a range of digits. For available notations, see 'Dialing Plan Notation for Routing and Manipulation' on page 469.
Destination Host **[IPInboundManipulation_DestHost]**	Defines the destination SIP URI host name - full name (usually in the Request URI). For any host name, enter the asterisk "*" symbol (default).
Request Type **[IPInboundManipulation_RequestType]**	Defines the SIP request type to which the manipulation rule is applied. ▪ **[0]** All = (Default) All SIP messages. ▪ **[1]** INVITE = All SIP messages except REGISTER and SUBSCRIBE. ▪ **[2]** REGISTER = Only REGISTER messages. ▪ **[3]** SUBSCRIBE = Only SUBSCRIBE messages. ▪ **[4]** INVITE and REGISTER = All SIP messages except SUBSCRIBE. ▪ **[5]** INVITE and SUBSCRIBE = All SIP messages except REGISTER.
Manipulated URI **[IPInboundManipulation_ManipulatedURI]**	Determines whether the source or destination SIP URI user part is manipulated. ▪ **[0]** Source = (Default) Manipulation is done on the source SIP URI user part. ▪ **[1]** Destination = Manipulation is done on the destination SIP URI user part.
Operation Rule (Action)	
Remove From Left **[IPInboundManipulation_RemoveFromLeft]**	Defines the number of digits to remove from the left of the user name prefix. For example, if you enter 3 and the user name is "john", the new user name is "n".

Parameter	Description
Remove From Right **[IPInboundManipulation_RemoveFromRight]**	Defines the number of digits to remove from the right of the user name prefix. For example, if you enter 3 and the user name is "john", the new user name is "j". **Note:** If both 'Remove From Right' and 'Leave From Right' parameters are configured, the 'Remove From Right' setting is applied first.
Leave From Right **[IPInboundManipulation_LeaveFromRight]**	Defines the number of characters that you want retained from the right of the user name. **Note:** If both 'Remove From Right' and 'Leave From Right' parameters are configured, the 'Remove From Right' setting is applied first.
Prefix to Add **[IPInboundManipulation_Prefix2Add]**	Defines the number or string that you want added to the front of the user name. For example, if you enter 'user' and the user name is "john", the new user name is "userjohn".
Suffix to Add **[IPInboundManipulation_Suffix2Add]**	Defines the number or string that you want added to the end of the user name. For example, if you enter '01' and the user name is "john", the new user name is "john01".

27.5.3 Routing INVITEs by SAS IP-to-IP Routing Table

SAS routing that is based on SAS IP-to-IP Routing table rules is applicable for the following SAS states:

■ Normal, if the 'SAS Survivability Mode' parameter is set to **Use Routing Table only in Normal mode**.

■ Emergency, if the 'SAS Survivability Mode' parameter is **not** set to **Use Routing Table only in Normal mode**.

The SAS routing rule destination can be an IP Group, IP address, Request-URI, or ENUM query.

The IP-to-IP Routing Table page allows you to configure up to 120 SAS routing rules (for Normal and Emergency modes). The device routes the SAS call (received SIP INVITE message) once a rule in this table is matched. If the characteristics of an incoming call do not match the first rule, the call characteristics is then compared to the settings of the second rule, and so on until a matching rule is located. If no rule is matched, the call is rejected.

When SAS receives a SIP INVITE request from a proxy server, the following routing logic is performed:

a. Sends the request according to rules configured in the IP-to-IP Routing table.

b. If no matching routing rule exists, the device sends the request according to its SAS registration database.

c. If no routing rule is located in the database, the device sends the request according to the Request-URI header.

> **Note:** The IP-to-IP Routing table can also be configured using the table *ini* file parameter, IP2IPRouting (see 'Configuration Parameters Reference' on page 471).

➢ **To configure the IP-to-IP Routing table for SAS:**

1. Open the IP-to-IP Routing Table (**Configuration** tab > **VoIP** menu > **SAS** > **IP-to-IP Routing Table**).

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

Figure 27-8: Add Record Dialog Box of SAS IP2IP Routing Page

Add Record	✖
Index	
Source IP Group ID	-1
Source Username Prefix	*
Source Host	*
Destination Username Prefix	*
Destination Host	*
Request Type	All ▾
Message Condition	None ▾
Destination Type	IP Group ▾
Destination IP Group ID	-1
Destination SRD ID	None ▾
Destination Address	
Destination Port	0
Destination Transport Type	▾
Alternative Route Options	Route Row ▾
Cost Group	None ▾
	🖫 Submit ✖ Cancel

3. Configure the rule according to the table below.

4. Click **Submit** to apply your changes.

5. To save the changes to flash memory, see 'Saving Configuration' on page 362.

⚠ **Note:** The following parameters are not applicable to SAS and must be ignored:

- 'Source IP Group ID'
- 'Destination IP Group ID'
- 'Destination SRD ID'
- 'Alternative Route Options'

Table 27-3: SAS IP-to-IP Routing Table Parameters

Parameter	Description
Matching Characteristics	
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. For example, to denote any prefix, use the asterisk (*) symbol; 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 469. The default is * (i.e., any prefix).

Parameter	Description
Source Host **[IP2IPRouting_SrcHost]**	Defines the host part of the incoming SIP dialog's source URI (usually the From URI). If this rule is not required, leave the field empty. To denote any host name, use the asterisk (*) symbol (default).
Destination Username Prefix **[IP2IPRouting_DestUsername Prefix]**	Defines the prefix of the incoming SIP dialog's destination URI (usually the Request URI) user part. You can use special notations for denoting the prefix. For example, to denote any prefix, use the asterisk (*) symbol; 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 469. The default is * (i.e., any prefix).
Destination Host **[IP2IPRouting_DestHost]**	Defines the host part of the incoming SIP dialog's destination URI (usually the Request-URI). If this rule is not required, leave the field empty. The asterisk (*) symbol (default) can be used to denote any destination host.
Message Condition **[IP2IPRouting_MessageCondit ion]**	Selects a Message Condition rule. To configure Message Condition rules, see Configuring Condition Rules.
ReRoute IP Group ID **[IP2IPRouting_ReRouteIPGrou pID]**	Defines the IP Group that initiated (sent) the SIP redirect response (e.g., 3xx) or REFER message. This field is typically used for re-routing requests (e.g., INVITEs) when interworking is required for SIP 3xx redirect responses or REFER messages (for more information, see Interworking SIP 3xx Redirect Responses and Interworking SIP REFER Messages, respectively). This parameter functions together with the 'Call Trigger' field (see below). The default is -1 (i.e., not configured).
Call Trigger **[IP2IPRouting_Trigger]**	Defines the reason (i.e, trigger) for re-routing the SIP request: ▪ **[0]** Any = (Default) This routing rule is used for all scenarios (re-routes and non-re-routes). ▪ **[1]** 3xx = Re-routes the request if it was triggered as a result of a SIP 3xx response. ▪ **[2]** REFER = Re-routes the INVITE if it was triggered as a result of a REFER request. ▪ **[3]** 3xx or REFER = Applies to options **[1]** and **[2]**. ▪ **[4]** Initial only = This routing rule is used for regular requests that the device forwards to the destination. This rule is not used for re-routing of requests triggered by the receipt of REFER or 3xx.
Operation Routing Rule	

Parameter	Description

Destination 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's Proxy Set (SERVER-type IP Group) or registered contact from the database (if USER-type IP Group).
	▪ **[1]** Dest Address = The SIP dialog is sent to the address configured in the following fields: 'Destination SRD ID', 'Destination Address', 'Destination Port', and 'Destination Transport Type'.
	▪ **[2]** Request URI = The SIP dialog is sent to the address indicated in the incoming Request-URI. If the fields 'Destination Port' and 'Destination Transport Type' are configured, the incoming Request-URI parameters are overridden and these fields take precedence.
	▪ **[3]** ENUM = An ENUM query is sent to include the destination address. If the fields 'Destination Port' and 'Destination Transport Type' are configured, the incoming Request-URI parameters are overridden and these fields take precedence.
	▪ **[4]** Hunt Group = Used for call center survivability. For more information, see Call Survivability for Call Centers.
	▪ **[5]** Dial Plan = The IP destination is determined by a Dial Plan index of the loaded Dial Plan file. The syntax of the Dial Plan index in the Dial Plan file is as follows: <destination / called prefix number>,0,<IP destination>
	Note that the second parameter "0" is ignored. An example of a configured Dial Plan (# 6) in the Dial Plan file is shown below:
	<pre>[PLAN6]
	200,0,10.33.8.52 ; called prefix 200 is
	routed to destination 10.33.8.52
	201,0,10.33.8.52
	300,0,itsp.com ; called prefix 300 is
	routed to destination itsp.com</pre>
	Once the Dial Plan is defined, you need to assign it (0 to 7) to the routing rule as the destination in the 'Destination Address' parameter, where "0" denotes **[PLAN1],** "1" denotes **[PLAN2],** and so on.

Destination IP Group ID **[IP2IPRouting_DestIPGroupID]**	Defines the IP Group ID to where you want to route the call. The SIP dialog messages are sent to the IP address defined for the Proxy Set associated with this IP Group. If you select an IP Group, it is unnecessary to configure a destination IP address (in the 'Destination Address' field). However, if both parameters are configured, then the IP Group takes precedence. If the destination IP Group is of USER type, the device searches for a match between the Request-URI (of the received SIP dialog) to an AOR registration record in the device's database. The SIP dialog is then sent to the IP address of the registered contact. The default is -1. **Notes:** ▪ This parameter is only relevant if the parameter 'Destination Type' is set to 'IP Group'. However, regardless of the settings

Parameter	Description
	of the parameter 'Destination Type', the IP Group is still used - only for determining the IP Profile or outgoing SRD. If neither IP Group nor SRD are defined in this table, the destination SRD is determined according to the source SRD associated with the Source IP Group (configured in the IP Group table, see 'Configuring IP Groups' on page 197). If this table does not define an IP Group but only an SRD, then the first IP Group associated with this SRD (in the IP Group table) is used. ▪ If the selected destination IP Group ID is type SERVER, the request is routed according to the IP Group addresses. ▪ If the selected destination IP Group ID is type USER, the request is routed according to the IP Group specific database (i.e., only to registered users of the selected database). ▪ If the selected destination IP Group ID is ANY USER (**[-2]**), the request is routed according to the general database (i.e., any matching registered user).
Destination Address **[IP2IPRouting_DestAddress]**	Defines the destination IP address (or domain name, e.g., domain.com) to where the call is sent. If ENUM-based routing is used (i.e., the 'Destination Type' parameter is set to ENUM) this parameter defines the IP address or domain name (FQDN) of the ENUM service, for example, e164.arpa, e164.customer.net, or NRENum.net. The device sends the ENUM query containing the destination phone number to an external DNS server, configured in the Multiple Interface table. The ENUM reply includes a SIP URI (user@host) which is used as the destination Request-URI in this routing table. **Notes:** ▪ This parameter is applicable only if the parameter 'Destination Type' is set to 'Dest Address' **[1]** or ENUM **[3]**. ▪ When using domain names, enter a DNS server IP address or alternatively, define these names in the 'Internal DNS Table' (see 'Configuring the Internal SRV Table' on page 135).
Destination Port **[IP2IPRouting_DestPort]**	Defines the destination port to where the call is sent.
Destination Transport Type **[IP2IPRouting_DestTransportType]**	Defines the transport layer type for sending the call: ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When this parameter is set to -1, the transport type is determined by the parameter SIPTransportType.
Cost Group **[IP2IPRouting_CostGroup]**	Assigns a Cost Group to the routing rule for determining the cost of the call. To configure Cost Groups, see 'Configuring Cost Groups' on page 191. By default, no Cost Group is assigned to the rule.

27.5.4   Routing REGISTERs in Emergency by IP-to-IP Routing Table

You can enable the device to route REGISTER messages based on the SAS IP-to-IP Routing table during Emergency state. For example, you can configure a routing rule to send these REGISTER messages to a different proxy.

➢ **To configure routing of REGISTER messages during Emergency state:**

1. Enable routing of REGISTER messages during Emergency state, by configuring the ini file parameter SASEmergencyModeRouteRegister to 1.

2. Open the SAS Configuration page (**Configuration** tab > **VoIP** menu > **SAS** > **Configuration**), and then configure the 'SAS Survivability Mode' parameter to any value other than 'Use Routing Table only in Normal mode'.

3. Open the IP-to-IP Routing Table (**Configuration** tab > **VoIP** menu > **SAS** > **IP-to-IP Routing Table**), and then configure a routing rule for the REGISTER messages.

27.5.5   Blocking Calls from Unregistered SAS Users

To prevent malicious calls, for example, service theft, it is recommended to configure the feature for blocking SIP INVITE messages received from SAS users that are not registered in the SAS database. This applies to SAS in normal and emergency states.

➢ **To block calls from unregistered SAS users:**

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

2. From the 'SAS Block Unregistered Users' drop-down list, select **Block**.

3. Click **Submit** to apply your changes.

27.5.6   Configuring SAS Emergency Calls

You can configure SAS to route emergency calls (such as 911 in North America) directly to the PSTN through its FXO interface. Thus, even during a communication failure with the external proxy, enterprise UAs can still make emergency calls.

You can define up to four emergency numbers, where each number can include up to four digits. When SAS receives a SIP INVITE (from a UA) that includes one of the user-defined emergency numbers in the SIP user part, it forwards the INVITE directly to the default gateway (see 'SAS Routing in Emergency State' on page 333). The default gateway is defined in the 'SAS Default Gateway IP' field, and this is the device itself. The device then sends the call directly to the PSTN.

This feature is applicable to SAS in normal and emergency states.

➢ **To configure SAS emergency numbers:**

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

2. In the 'SAS Default Gateway IP' field, define the IP address and port (in the format *x.x.x.x:port*) of the device (Gateway application).

> **Note:**   The port of the device is defined in the 'SIP UDP/TCP/TLS Local Port' field in the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**).

3. In the 'SAS Emergency Numbers' field, enter an emergency number in each field box.

Figure 27-9: Configuring SAS Emergency Numbers

4. Click **Submit** to apply your changes.

27.5.7 Adding SIP Record-Route Header to SIP INVITE

You can configure SAS to add the SIP Record-Route header to SIP requests (e.g. INVITE) received from enterprise UAs. SAS then sends the request with this header to the proxy. The Record-Route header includes the IP address of the SAS application. This ensures that future requests in the SIP dialog session from the proxy to the UAs are routed through the SAS application. If not configured, future request within the dialog from the proxy are sent directly to the UAs (and do not traverse SAS). When this feature is enabled, the SIP Record-Route header includes the URI "lr" parameter, indicating loose routing, as shown in the following example:

```
Record-Route: <sip:server10.biloxi.com;lr>
```

> **Note:** This feature is applicable only to the SAS Outbound mode.

➢ **To enable the Record-Route header:**

1. Open the SAS Configuration page (**Configuration** tab > **VoIP** menu > **SAS** > **Configuration**).
2. From the 'Enable Record-Route' drop-down list, select **Enable**.
3. Click **Submit** to apply your changes.

27.5.8   Re-using TCP Connections

You can enable the SAS application to re-use the same TCP connection for sessions (multiple SIP requests / responses) with the same SIP UA. The benefits of this feature include less CPU and memory usage because fewer TCP connections are open and reduced network congestion. For example, assume User A sends a REGISTER message to SAS with transport=TCP, and User B sends an INVITE message to A using SAS. In this scenario, the SAS application forwards the INVITE request using the same TCP connection that User A initially opened with the REGISTER message.

➢ **To re-use TCP connection sessions in SAS**

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

2. From the 'SAS Connection Reuse' drop-down list, select **Enable**.

3. Click **Submit** to apply your changes.

27.5.9   Replacing Contact Header for SIP Messages

You can configure SAS to change the SIP Contact header so that it points to the SAS host. This ensures that in the message, the top-most SIP Via header and the Contact header point to the same host.

> **Notes:**
>
> - This feature is applicable only to the SAS Outbound mode.
> - The device may become overloaded if this feature is enabled, as all incoming SIP dialog requests traverse the SAS application.

Currently, this feature can be configured only by the *ini* file parameter, SASEnableContactReplace:

- **[0]** (Default): Disable - when relaying requests, SAS adds a new Via header (with the IP address of the SAS application) as the top-most Via header and retains the original Contact header. Thus, the top-most Via header and the Contact header point to different hosts.

- **[1]**: Enable - SAS changes the Contact header so that it points to the SAS host and therefore, the top-most Via header and the Contact header point to the same host.

27.6   Viewing Registered SAS Users

You can view all the users that are registered in the SAS registration database. This is displayed in the 'SAS/SBC Registered Users page, as described in 'Viewing Registered Users' on page .

> **Note:** You can increase the maximum number of registered SAS users, by implementing the SAS Cascading feature, as described in 'SAS Cascading' on page .

28 SAS Cascading

The SAS Cascading feature allows you to increase the number of SAS users above the maximum supported by the SAS gateway. This is achieved by deploying multiple SAS gateways in the network. For example, if the SAS gateway supports up to 600 users, but your enterprise has 1,500 users, you can deploy three SAS gateways to accommodate all users: the first SAS gateway can service 600 registered users, the second SAS gateway the next 600 registered users, and the third SAS gateway the rest (i.e., 300 registered users).

In SAS Cascading, the SAS gateway first attempts to locate the called user in its SAS registration database. Only if the user is not located, does the SAS gateway send it on to the next SAS gateway according to the SAS Cascading configuration.

There are two methods for configuring SAS Cascading. This depends on whether the users can be identified according to their phone extension numbers:

■ **SAS Routing Table:** If users can be identified with unique phone extension numbers, then the SAS Routing table is used to configure SAS Cascading. This SAS Cascading method routes calls directly to the SAS Gateway (defined by IP address) to which the called SAS user is registered.

The following is an example of a SAS Cascading deployment of users with unique phone extension numbers:

- users registered to the first SAS gateway start with extension number "40"
- users registered to the second SAS gateway start with extension number "20"
- users registered to the third SAS gateway start with extension number "30"

The SAS Routing table rules for SAS Cascading are created using the destination (called) extension number prefix (e.g., "30") and the destination IP address of the SAS gateway to which the called user is registered. Such SAS routing rules must be configured at each SAS gateway to allow routing between the SAS users. The routing logic for SAS Cascading is similar to SAS routing in Emergency state (see the flowchart in 'SAS Routing in Emergency State' on page 333). For a description on the SAS Routing table, see 'SAS Routing Based on IP-to-IP Routing Table' on page 346.

The figure below illustrates an example of a SAS Cascading call flow configured using the SAS Routing table. In this example, a call is routed from SAS Gateway (A) user to a user on SAS Gateway (B).

Figure 28-1: SAS Cascading Using SAS Routing Table - Example
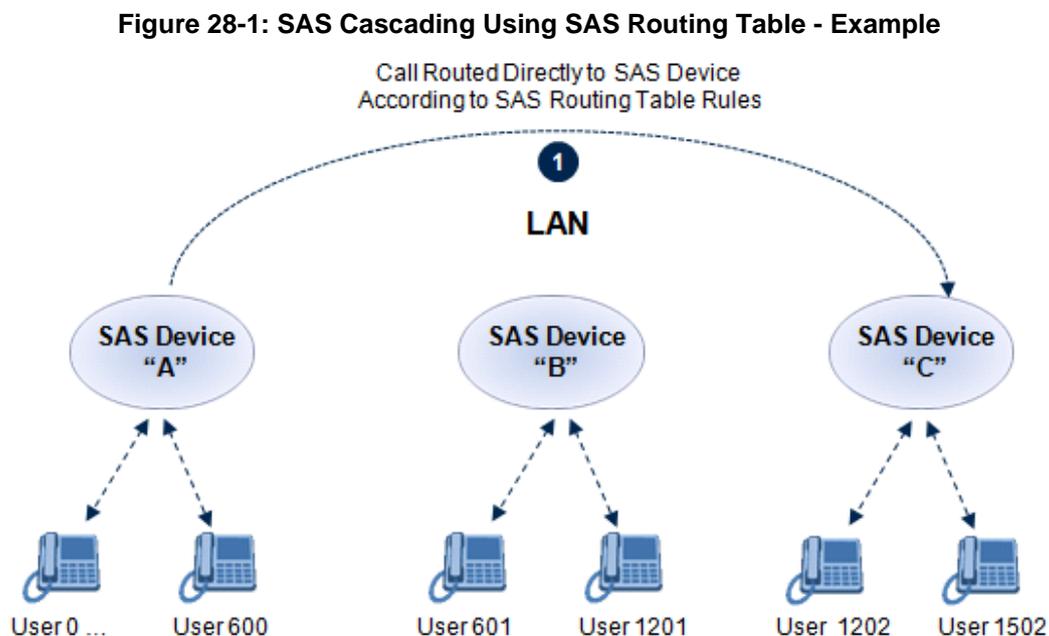
■ **SAS Redundancy mode:** If users cannot be distinguished (i.e., associated to a

specific SAS gateway), then the SAS Redundancy feature is used to configure SAS Cascading. This mode routes the call in a loop fashion, from one SAS gateway to the next, until the user is located. Each SAS gateway serves as the redundant SAS gateway ("redundant SAS proxy server") for the previous SAS gateway (in a one-way direction). For example, if a user calls a user that is not registered on the same SAS gateway, the call is routed to the second SAS gateway, and if not located, it is sent to the third SAS gateway. If the called user is not located on the third (or last) SAS gateway, it is then routed back to the initial SAS gateway, which then routes the call to the default gateway (i.e., to the PSTN).

Each SAS gateway adds its IP address to the SIP via header in the INVITE message before sending it to the next ("redundant") SAS gateway. If the SAS gateway receives an INVITE and its IP address appears in the SIP via header, it sends it to the default gateway (and not to the next SAS gateway), as defined by the SASDefaultGatewayIP parameter. Therefore, this mode of operation prevents looping between SAS gateways when a user is not located on any of the SAS gateways.

The figure below illustrates an example of a SAS Cascading call flow when configured using the SAS Redundancy feature. In this example, a call is initiated from a SAS Gateway (A) user to a user that is not located on any SAS gateway. The call is subsequently routed to the PSTN.

Figure 28-2: SAS Cascading Using SAS Redundancy Mode - Example
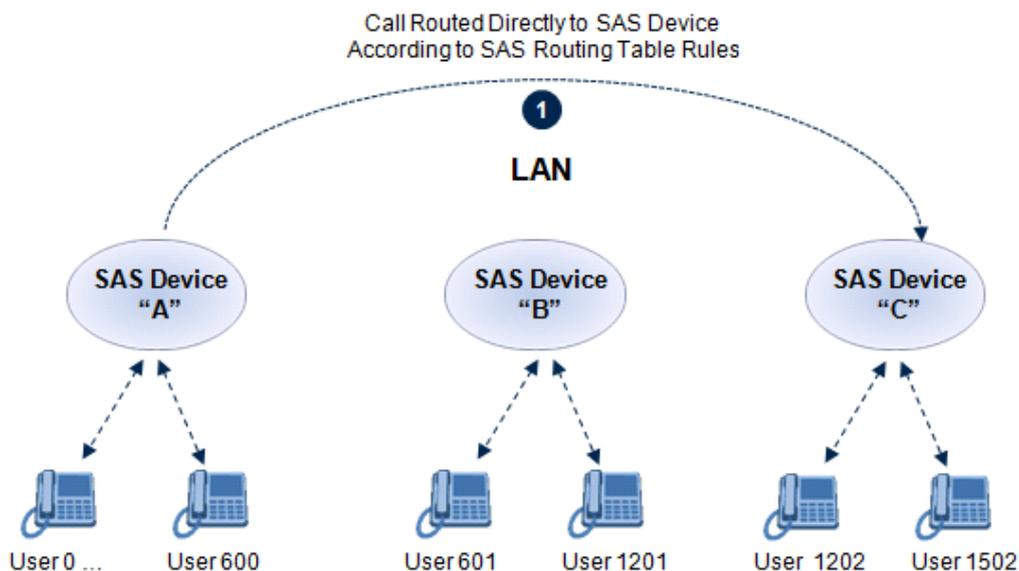
Part VII

Maintenance

29   Basic Maintenance

The Maintenance Actions page allows you to perform the following:

■   Reset the device - see 'Resetting the Device' on page 359

■   Lock and unlock the device - see 'Locking and Unlocking the Device' on page 361

■   Save configuration to the device's flash memory - see 'Saving Configuration' on page 362

➢   **To access the Maintenance Actions page, do one of the following:**

■   On the toolbar, click the **Device Actions** button, and then from the drop-down menu, choose **Reset**.

■   On the Navigation bar, click the **Maintenance** tab, and then in the Navigation tree, select the **Maintenance** menu and choose **Maintenance Actions**.

Figure 29-1: Maintenance Actions Page

29.1   Resetting the Device

The Maintenance Actions page allows you to remotely reset the device. In addition, before resetting the device, you can choose the following options:

■   Save the device's current configuration to the device's flash memory (non-volatile).

■   Perform a graceful shutdown, whereby device reset starts only after a user-defined time (i.e., timeout) or after no more active traffic exists (the earliest thereof).

Notes:

•   Throughout the Web interface, parameters displayed with a lightning ⚡ symbol are not applied on-the-fly and require that you reset the device for them to take effect.

•   When you modify parameters that require a device reset, once you click the **Submit** button in the relevant page, the toolbar displays "Reset" (see 'Toolbar Description' on page 42) to indicate that a device reset is required.

•   After you reset the device, the Web GUI is displayed in Basic view (see 'Displaying Navigation Tree in Basic and Full View' on page 43).

➢ **To reset the device:**

1. Open the Maintenance Actions page (see 'Basic Maintenance' on page 359).

2. Under the 'Reset Configuration' group, from the 'Burn To FLASH' drop-down list, select one of the following options:

- **Yes:** The device's current configuration is saved (*burned*) to the flash memory prior to reset (default).

- **No:** Resets the device without saving the current configuration to flash (discards all unsaved modifications).

3. Under the 'Reset Configuration' group, from the 'Graceful Option' drop-down list, select one of the following options:

- **Yes:** Reset starts only after the user-defined time in the 'Shutdown Timeout' field (see Step 4) expires or after no more active traffic exists (the earliest thereof). In addition, no new traffic is accepted.

- **No:** Reset starts regardless of traffic, and any existing traffic is terminated at once.

4. In the 'Shutdown Timeout' field (relevant only if the 'Graceful Option' in the previous step is set to **Yes**), enter the time after which the device resets. Note that if no traffic exists and the time has not yet expired, the device resets.

5. Click the **Reset** button; a confirmation message box appears, requesting you to confirm.

Figure 29-2: Reset Confirmation Message Box
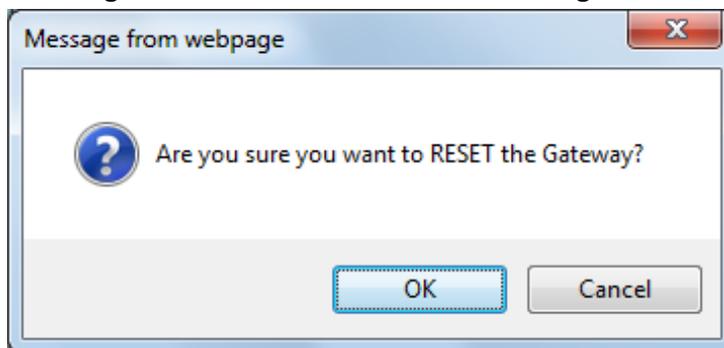
6. Click **OK** to confirm device reset; if the parameter 'Graceful Option' is set to **Yes** (in Step 3), the reset is delayed and a screen displaying the number of remaining calls and time is displayed. When the device begins to reset, a message appears notifying you of this.

29.2 Remotely Resetting Device using SIP NOTIFY

The device can be remotely reset upon the receipt of a SIP NOTIFY that includes an Event header set to 'check-sync;reboot=true', as shown in the example below:

```
NOTIFY sip:<user>@<dsthost> SIP/2.0
To: sip:<user>@<dsthost>
From: sip:sipsak@<srchost>
CSeq: 10 NOTIFY
Call-ID: 1234@<srchost>
Event: check-sync;reboot=true
```

> ➢ **To enable remote reset upon receipt of SIP NOTIFY:**

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

2. Under the Misc Parameters group, set the 'SIP Remote Rest' parameter to **Enable**.

3. Click **Submit**.

> **Note:** This SIP Event header value is proprietary to AudioCodes.

29.3 Locking and Unlocking the Device

The Lock and Unlock option allows you to lock the device so that it doesn't accept any new calls and maintains only the current calls. This is useful when, for example, you are uploading new software files to the device and you don't want any traffic to interfere with the process.

> ➢ **To lock the device:**

1. Open the Maintenance Actions page (see 'Basic Maintenance' on page 359).

2. Under the 'LOCK / UNLOCK' group, from the 'Graceful Option' drop-down list, select one of the following options:

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

- **No:** The device is 'locked' regardless of traffic. Any existing traffic is terminated immediately.

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

3. In the 'Lock Timeout' field (relevant only if the parameter 'Graceful Option' in the previous step is set to **Yes**), enter the time (in seconds) after which the device locks. Note that if no traffic exists and the time has not yet expired, the device locks.

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

Figure 29-3: Device Lock Confirmation Message Box
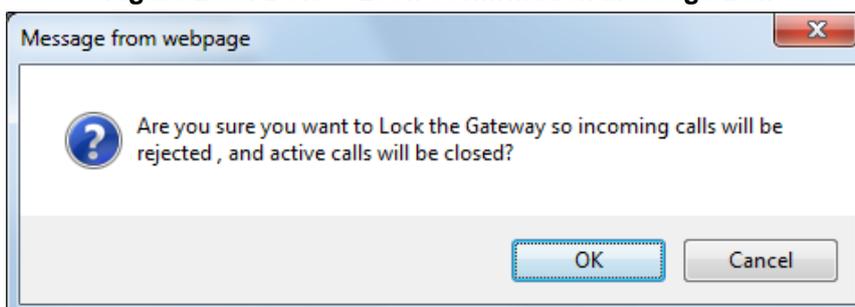
5. Click **OK** to confirm device Lock; if 'Graceful Option' is set to **Yes**, the lock is delayed and a screen displaying the number of remaining calls and time is displayed. Otherwise, the lock process begins immediately. The Current Admin State' field displays the current state - "LOCKED" or "UNLOCKED".

> ➢ **To unlock the device:**

1. Open the Maintenance Actions page (see 'Maintenance Actions' on page 359).

2. Under the 'LOCK / UNLOCK' group, click the **UNLOCK** button. Unlock starts immediately and the device accepts new incoming calls.

> **Note:** The Home page's General Information pane displays whether the device is locked or unlocked (see 'Viewing the Home Page' on page 63).

29.4 Saving Configuration

The Maintenance Actions page allows you to save (*burn*) the current parameter configuration (including loaded auxiliary files) to the device's *non-volatile* memory (i.e., flash). The parameter modifications that you make throughout the Web interface's pages are temporarily saved (to the *volatile* memory - RAM) when you click the **Submit** button on these pages. Parameter settings that are saved only to the device's RAM revert to their previous settings after a hardware/software reset (or power failure). Therefore, to ensure that your configuration changes are retained, you must save them to the device's flash memory using the burn option described below.

> ➢ **To save the changes to the non-volatile flash memory :**

1. Open the Maintenance Actions page (see 'Basic Maintenance' on page 359).

2. Under the 'Save Configuration' group, click the **BURN** button; a confirmation message appears when the configuration successfully saves.

> **Notes:**
>
> - Saving configuration to the *non-volatile* memory may disrupt current traffic on the device. To avoid this, disable all new traffic before saving, by performing a graceful lock (see 'Locking and Unlocking the Device' on page 361).
> - Throughout the Web interface, parameters displayed with the lightning ⚡ symbol are not applied on-the-fly and require that you reset the device for them to take effect (see 'Resetting the Device' on page 359).
> - The Home page's General Information pane displays whether the device is currently "burning" the configuration (see 'Viewing the Home Page' on page 63).

30   Resetting an Analog Channel

You can inactivate (*reset*) an FXO or FXS analog channel. This is sometimes useful, for example, when the device (FXO) is connected to a PBX and the communication between the two can't be disconnected (e.g., when using reverse polarity). This is done in the Web interface's Home page.

➢   **To reset an analog channel:**

1.   Open the Home page.

2.   Click the required **FXS** or **FXO** port icon; a shortcut menu appears.

3.   From the shortcut menu, choose **Reset Channel**; the channel is changed to inactive and the port icon is displayed in gray.

This page is intentionally left blank.

31 Software Upgrade

The **Software Update** menu allows you do the following:

- Load Auxiliary Files (see 'Loading Auxiliary Files' on page 365)
- Load Software License Key (see 'Software License Key' on page 377)
- Upgrade device using Software Upgrade Wizard (see 'Software Upgrade Wizard' on page 381)
- Load / save Configuration File (see 'Backing Up and Loading Configuration File' on page 384)

31.1 Loading Auxiliary Files

Various Auxiliary files can be installed on the device. These Auxiliary files provide the device with additional configuration settings. The table below lists the different types of Auxiliary files:

Table 31-1: Auxiliary Files

File	Description
INI	Configures the device. The Web interface enables practically full device provisioning. However, some features may only be configured by ini file or you may wish to configure your device using the ini file. For more information on using the ini file to configure the device, see 'INI File-Based Management' on page 95.
Call Progress Tones	Region-specific, telephone exchange-dependent file that contains the Call Progress Tones (CPT) levels and frequencies for the device. The default CPT file is U.S.A. For more information, see 'Call Progress Tones File' on page 367.
Prerecorded Tones	The Prerecorded Tones (PRT) file enhances the device's capabilities of playing a wide range of telephone exchange tones that cannot be defined in the CPT file. For more information, see Prerecorded Tones File on page 371. **Note:** PRT is not supported by MP-124 Rev. E.
Dial Plan	Provides dialing plans, for example, to know when to stop collecting dialed digits and start forwarding them or for obtaining the destination IP address for outbound IP routing. For more information, see 'Dial Plan File' on page 372.
User Info	The User Information file maps PBX extensions to IP numbers. This file can be used to represent PBX extensions as IP phones in the global 'IP world'. For more information, see 'User Information File' on page 375.

The Auxiliary files can be loaded to the device using one of the following methods:

- Web interface.
- TFTP: This is done by specifying the name of the Auxiliary file in an *ini* file (see Auxiliary and Configuration Files Parameters) and then loading the *ini* file to the device. The Auxiliary files listed in the *ini* file are then automatically loaded through TFTP during device startup. If the *ini* file does not contain a specific auxiliary file type, the device uses the last auxiliary file of that type that was stored on its non-volatile memory.

Notes:

- You can schedule automatic loading of updated auxiliary files using HTTP/HTTPS, FTP, or NFS. For more information on automatic updates, ee 'Automatic Update' on page 385.

- When loading an *ini* file using this Web page, parameters that are excluded from the loaded *ini* file retain their current settings (*incremental*).

- Saving an auxiliary file to flash memory may disrupt traffic on the device. To avoid this, disable all traffic on the device by performing a graceful lock as described in 'Locking and Unlocking the Device' on page 361.

- For deleting auxiliary files, see 'Viewing Device Information' on page 407.

The procedure below describes how to load Auxiliary files using the Web interface.

➢ **To load auxiliary files to the device using the Web interface:**

1. Open the Load Auxiliary Files page (**Maintenance** tab > **Software Update** menu > **Load Auxiliary Files**).

Note: The appearance of certain file load fields depends on the installed Software License Key.

2. Click the **Browse** button corresponding to the file type that you want to load, navigate to the folder in which the file is located, and then click **Open**; the name and path of the file appear in the field next to the **Browse** button.

3. Click the **Load File** button corresponding to the file you want to load.

4. Repeat steps 2 through 3 for each file you want to load.

5. Save the loaded auxiliary files to flash memory, see 'Saving Configuration' on page 362 and reset the device (if you have loaded a Call Progress Tones file), see 'Resetting the Device' on page 359.

You can also load auxiliary files using an ini file that is loaded to the device with BootP. Each auxiliary file has a specific ini file parameter that specifies the name of the auxiliary file that you want to load to the device with the ini file. For a description of these ini file parameters, see Auxiliary and Configuration Files Parameters.

> ➢ **To load auxiliary files using an ini file:**

1. In the ini file, define the auxiliary files to be loaded to the device. You can also define in the ini file whether the loaded files must be stored in the non-volatile memory so that the TFTP process is not required every time the device boots up.

2. Save the auxiliary files and the ini file in the same directory on your local PC.

3. Invoke a BootP/TFTP session; the ini and associated auxiliary files are loaded to the device.

31.1.1 Call Progress Tones File

The Call Progress Tones (CPT) and Distinctive Ringing auxiliary file is comprised of two sections:

- The first section contains the definitions of the Call Progress Tones (levels and frequencies) that are detected / generated by the device.

- The second section contains the characteristics of the Distinctive Ringing signals that are generated by the device (see Distinctive Ringing on page 369).

You can use one of the supplied auxiliary files (.dat file format) or create your own file. To create your own file, it's recommended to modify the supplied *usa_tone.ini* file (in any standard text editor) to suit your specific requirements and then convert the modified *ini* file into binary format, using AudioCodes DConvert utility. For a description on converting a CPT *ini* file into a binary *dat* file, refer to the *DConvert Utility User's Guide*.

> **Note:** Only the *dat* file format can be loaded to the device.

You can create up to 32 different Call Progress Tones, each with frequency and format attributes. The frequency attribute can be single or dual-frequency (in the range of 300 to 1980 Hz) or an Amplitude Modulated (AM). Up to 64 different frequencies are supported. Only eight AM tones, in the range of 1 to 128 kHz, can be configured (the detection range is limited to 1 to 50 kHz). Note that when a tone is composed of a single frequency, the second frequency field must be set to zero.

The format attribute can be one of the following:

- **Continuous:** A steady non-interrupted sound (e.g., a dial tone). Only the 'First Signal On time' should be specified. All other on and off periods must be set to zero. In this case, the parameter specifies the detection period. For example, if it equals 300, the tone is detected after 3 seconds (300 x 10 msec). The minimum detection time is 100 msec.

- **Cadence:** A repeating sequence of on and off sounds. Up to four different sets of on/off periods can be specified.

- **Burst:** A single sound followed by silence. Only the 'First Signal On time' and 'First Signal Off time' should be specified. All other on and off periods must be set to zero. The burst tone is detected after the off time is completed.

You can specify several tones of the same type. These additional tones are used only for tone detection. Generation of a specific tone conforms to the first definition of the specific tone. For example, you can define an additional dial tone by appending the second dial tone's definition lines to the first tone definition in the *ini* file. The device reports dial tone detection if either of the two tones is detected.

The Call Progress Tones section of the *ini* file comprises the following segments:

■ **[NUMBER OF CALL PROGRESS TONES]:** Contains the following key: 'Number of Call Progress Tones' defining the number of Call Progress Tones that are defined in the file.

■ **[CALL PROGRESS TONE #X]:** containing the Xth tone definition, starting from 0 and not exceeding the number of Call Progress Tones less 1 defined in the first section (e.g., if 10 tones, then it is 0 to 9), using the following keys:

- **Tone Type:** Call Progress Tone types:
  - ♦ **[1]** Dial Tone
  - ♦ **[2]** Ringback Tone
  - ♦ **[3]** Busy Tone
  - ♦ **[4]** Congestion Tone
  - ♦ **[6]** Warning Tone
  - ♦ **[7]** Reorder Tone
  - ♦ **[8]** Confirmation Tone
  - ♦ **[9]** Call Waiting Tone - heard by the called party
  - ♦ **[15]** Stutter Dial Tone
  - ♦ **[16]** Off Hook Warning Tone
  - ♦ **[17]** Call Waiting Ringback Tone - heard by the calling party
  - ♦ **[18]** Comfort Tone
  - ♦ **[23]** Hold Tone
  - ♦ **[46]** Beep Tone

- **Tone Modulation Type:** Amplitude Modulated (1) or regular (0)

- **Tone Form:** The tone's format can be one of the following:
  - ♦ Continuous (1)
  - ♦ Cadence (2)
  - ♦ Burst (3)

- **Low Freq [Hz]:** Frequency (in Hz) of the lower tone component in case of dual frequency tone, or the frequency of the tone in case of single tone. This is not relevant to AM tones.

- **High Freq [Hz:** Frequency (in Hz) of the higher tone component in case of dual frequency tone, or zero (0) in case of single tone (not relevant to AM tones).

- **Low Freq Level [-dBm]:** Generation level 0 dBm to -31 dBm in dBm (not relevant to AM tones).

- **High Freq Level:** Generation level of 0 to -31 dBm. The value should be set to 32 in the case of a single tone (not relevant to AM tones).

- **First Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the first cadence on-off cycle. For continuous tones, this parameter defines the detection period. For burst tones, it defines the tone's duration.

- **First Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the first cadence on-off cycle (for cadence tones). For burst tones, this parameter defines the off time required after the burst tone ends and the tone detection is reported. For continuous tones, this parameter is ignored.

- **Second Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.

- **Second Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.

- **Third Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the third cadence on-off cycle. Can be omitted if there isn't a third cadence.

- **Third Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the third cadence on-off cycle. Can be omitted if there isn't a third cadence.
- **Fourth Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the fourth cadence on-off cycle. Can be omitted if there isn't a fourth cadence.
- **Fourth Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the fourth cadence on-off cycle. Can be omitted if there isn't a fourth cadence.
- **Carrier Freq [Hz]:** Frequency of the carrier signal for AM tones.
- **Modulation Freq [Hz]:** Frequency of the modulated signal for AM tones (valid range from 1 to 128 Hz).
- **Signal Level [-dBm]:** Level of the tone for AM tones.
- **AM Factor [steps of 0.02]:** Amplitude modulation factor (valid range from 1 to 50). Recommended values from 10 to 25.

> **Notes:**
>
> - When the same frequency is used for a continuous tone and a cadence tone, the 'Signal On Time' parameter of the continuous tone must have a value that is greater than the 'Signal On Time' parameter of the cadence tone. Otherwise, the continuous tone is detected instead of the cadence tone.
> - The tones frequency must differ by at least 40 Hz between defined tones.

For example, to configure the dial tone to 440 Hz only, enter the following text:

```
[NUMBER OF CALL PROGRESS TONES]
Number of Call Progress Tones=1
#Dial Tone
[CALL PROGRESS TONE #0]
Tone Type=1
Tone Form =1 (continuous)
Low Freq [Hz]=440
High Freq [Hz]=0
Low Freq Level [-dBm]=10 (-10 dBm)
High Freq Level [-dBm]=32 (use 32 only if a single tone is
required)
First Signal On Time [10msec]=300; the dial tone is detected after
3 sec
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=0
Second Signal Off Time [10msec]=0
```

31.1.1.1 Distinctive Ringing

Distinctive Ringing is applicable only to FXS interfaces. Using the Distinctive Ringing section of the Call Progress Tones auxiliary file, you can create up to 16 Distinctive Ringing patterns. Each ringing pattern configures the ringing tone frequency and up to four ringing cadences. The same ringing frequency is used for all the ringing pattern cadences. The ringing frequency can be configured in the range of 10 to 200 Hz with a 5 Hz resolution.

Each of the ringing pattern cadences is specified by the following parameters:

- **Burst Ring On Time:** Configures the cadence to be a burst cadence in the entire ringing pattern. The burst relates to On time and the Off time of the same cadence. It must appear between 'First/Second/Third/Fourth' string and the 'Ring On/Off Time' This cadence rings once during the ringing pattern. Otherwise, the cadence is interpreted as cyclic: it repeats for every ringing cycle.

■ **Ring On Time:** Specifies the duration of the ringing signal.

■ **Ring Off Time:** Specifies the silence period of the cadence.

The Distinctive Ringing section of the *ini* file format contains the following strings:

■ **[NUMBER OF DISTINCTIVE RINGING PATTERNS]:** Contains the following key:

- 'Number of Distinctive Ringing Patterns' defining the number of Distinctive Ringing signals that are defined in the file.

■ **[Ringing Pattern #X]:** Contains the Xth ringing pattern definition (starting from 0 and not exceeding the number of Distinctive Ringing patterns defined in the first section minus 1) using the following keys:

- **Ring Type:** Must be equal to the Ringing Pattern number.

- **Freq [Hz]:** Frequency in hertz of the ringing tone.

- **First (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the first cadence on-off cycle.

- **First (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the first cadence on-off cycle.

- **Second (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the second cadence on-off cycle.

- **Second (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the second cadence on-off cycle.

- **Third (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the third cadence on-off cycle.

- **Third (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the third cadence on-off cycle.

- **Fourth (Burst) Ring On Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the fourth cadence on-off cycle.

- **Fourth (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the fourth cadence on-off cycle.

> **Note:** In SIP, the Distinctive Ringing pattern is selected according to the Alert-Info header in the INVITE message. For example:
> Alert-Info:<Bellcore-dr2>, or Alert-Info:<http://…/Bellcore-dr2>
> 'dr2' defines ringing pattern #2. If the Alert-Info header is missing, the default ringing tone (0) is played.

An example of a **ringing burst** definition is shown below:

```
#Three ringing bursts followed by repeated ringing of 1 sec on and
3 sec off.
[NUMBER OF DISTINCTIVE RINGING PATTERNS]
Number of Ringing Patterns=1
[Ringing Pattern #0]
Ring Type=0
Freq [Hz]=25
First Burst Ring On Time [10msec]=30
First Burst Ring Off Time [10msec]=30
Second Burst Ring On Time [10msec]=30
Second Burst Ring Off Time [10msec]=30
Third Burst Ring On Time [10msec]=30
Third Burst Ring Off Time [10msec]=30
Fourth Ring On Time [10msec]=100
Fourth Ring Off Time [10msec]=300
```

An example of **various ringing signals** definition is shown below:

```
[NUMBER OF DISTINCTIVE RINGING PATTERNS]
Number of Ringing Patterns=3
#Regular North American Ringing Pattern
[Ringing Pattern #0]
Ring Type=0
Freq [Hz]=20
First Ring On Time [10msec]=200
First Ring Off Time [10msec]=400
#GR-506-CORE Ringing Pattern 1
[Ringing Pattern #1]
Ring Type=1
Freq [Hz]=20
First Ring On Time [10msec]=200
First Ring Off Time [10msec]=400
#GR-506-CORE Ringing Pattern 2
[Ringing Pattern #2]
Ring Type=2
Freq [Hz]=20
First Ring On Time [10msec]=80
First Ring Off Time [10msec]=40
Second Ring On Time [10msec]=80
Second Ring Off Time [10msec]=400
```

31.1.2 Prerecorded Tones File

The CPT file mechanism has several limitations such as a limited number of predefined tones and a limited number of frequency integrations in one tone. To overcome these limitations and provide tone generation capability that is more flexible, the Prerecorded Tones (PRT) file can be used. If a specific prerecorded tone exists in the PRT file, it takes precedence over the same tone that exists in the CPT file and is played instead of it.

> **Notes**
> - The PRT are used only for generation of tones. Detection of tones is performed according to the CPT file.
> - PRT is not supported by MP-124 Rev. E.

The PRT is a *.dat* file containing a set of prerecorded tones that can be played by the device. Up to 40 tones (totaling approximately 10 minutes) can be stored in a single PRT file on the device's flash memory. The prerecorded tones are prepared offline using standard recording utilities (such as CoolEdit™) and combined into a single file, using AudioCodes DConvert utility (refer to *DConvert Utility User's Guide* for more information).

The raw data files must be recorded with the following characteristics:

- **Coders:** G.711 A-law or G.711 μ-law
- **Rate:** 8 kHz
- **Resolution:** 8-bit
- **Channels:** mono

Once created, the PRT file can then be loaded to the device using AudioCodes' AcBootP utility or the Web interface (see 'Loading Auxiliary Files' on page 365).

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 PRT

module repeatedly plays this cadence for the configured duration. Similarly, a continuous tone can be played by repeating only part of it.

31.1.3 Dial Plan File

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

31.1.3.1 Creating a Dial Plan File

Creating a Dial Plan file is similar between 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 use for the specific feature.

The Dial Plan file is a text-based file that can contain up to eight 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.

➢ **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. Install the converted file on the device, as described in 'Loading Auxiliary Files' on page 365.

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

31.1.3.2 Dialing Plans for Digit Collection

The device enables you to configure multiple dialing plans in an external Dial Plan file, which can be installed on the device. If a Dial Plan file is implemented, the device first attempts to locate a matching digit pattern in a specified Dial Plan index listed in the file and if not found, attempts to locate a matching digit pattern in the Digit Map. The Digit Map is configured by the 'Digit Mapping Rules' parameter, located in the DTMF & Dialing page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **DTMF and Supplementary** > **DTMF & Dialing**).

The Dial Plan is used for the following:

■ FXS, and FXO collecting digit mode (Tel-to-IP calls): The file allows the device to know when digit collection ends, after which it starts sending all the collected (or dialed) digits in the outgoing INVITE message. This also provides enhanced digit mapping.

The Dial Plan file can contain up to eight Dial Plans (Dial Plan indices), with a total of up to 8,000 dialing rules (lines) of distinct prefixes (e.g. area codes, international telephone number patterns) for the PSTN to which the device is connected.

The Dial Plan file is created in a textual *ini* file with the following syntax:

`<called number prefix>,<total digits to wait before sending>`

■ Each new Dial Plan index begins with a Dial Plan name enclosed in square brackets "**[...]" on** a new line.

- Each line under the Dial Plan index defines a dialing prefix and the number of digits expected to follow that prefix. The prefix is separated by a comma "," from the number of additional digits.
- The prefix can include numerical ranges in the format **[x-**y], as well as multiple numerical ranges **[n-**m]**[x-**y] (no comma between them).
- The prefix can include the asterisk "*" and number "#" signs.
- The number of additional digits can include a numerical range in the format x-y.
- Empty lines are ignored.
- Lines beginning with a semicolon ";" are ignored. The semicolon can be used for comments.

Below shows an example of a Dial Plan file (in *ini*-file format), containing two dial plans:

```
; Example of dial-plan configuration.
; This file contains two dial plans:
[ PLAN1 ]
; Destination cellular area codes 052, 054, and 050 with 8 digits.

052,8
054,8
050,8
; Defines International prefixes 00, 012, 014.
; The number following these prefixes may
; be 7 to 14 digits in length.
00,7-14
012,7-14
014,7-14
; Defines emergency number 911. No additional digits are expected.
911,0
[ PLAN2 ]
; Defines area codes 02, 03, 04.
; In these area codes, phone numbers have 7 digits.
0[2-4],7
; Operator services starting with a star: *41, *42, *43.
; No additional digits are expected.
*4[1-3],0
```

The procedure below provides a summary on how to create a Dial Plan file and select the required Dial Plan index.

➢ **To create a Dial Plan file:**

1. Create a new file using a text-based editor (such as Notepad) and configure your Dial Plans, as required.
2. Save the file with the *ini* file extension name (e.g., mydialplans.ini).
3. Convert the *ini* file to a *dat* binary file, using AudioCodes DConvert utility. For more information, refer to *DConvert Utility User's Guide*.
4. Install the converted file on the device, as described in 'Loading Auxiliary Files' on page 365.
5. The required Dial Plan is selected using the 'Dial Plan Index' parameter. This parameter can be set to **0** through **7**, where **0** denotes PLAN1, **1** denotes PLAN2, and so on.

Notes:

- The Dial Plan file must not contain overlapping prefixes. Attempting to process an overlapping configuration by the DConvert utility results in an error message specifying the problematic line.

- The Dial Plan index can be selected globally for all calls (as described in the previous procedure), or per specific calls using Tel Profiles.

- It may be useful to configure both Dial Plan file and Digit Maps. For example, the Digit Map can be used for complex digit patterns (which are not supported by the Dial Plan) and the Dial Plan can be used for long lists of relatively simple digit patterns. In addition, as timeout between digits is not supported by the Dial Plan, the Digit Map can be used to configure digit patterns that are shorter than those defined in the Dial Plan or left at default (MaxDigits parameter). For example, the "xx.T" digit map instructs the device to use the Dial Plan and if no matching digit pattern is found, it waits for two more digits and then after a timeout (TimeBetweenDigits parameter), it sends the collected digits. Therefore, this ensures that calls are not rejected as a result of their digit pattern not been completed in the Dial Plan.

- By default, if no matching digit pattern is found in both the Dial Plan and Digit Map, the device rejects the call. However, if you set the DisableStrictDialPlan parameter to 1, the device attempts to complete the call using the MaxDigits and TimeBetweenDigits parameters. In such a setup, it collects the number of digits configured by the MaxDigits parameters. If more digits are received, it ignores the settings of this parameter and collects the digits until the inter-digit timeout configured by the TimeBetweenDigits parameter is exceeded.

31.1.3.3 Obtaining IP Destination from Dial Plan File

You can use a Dial Plan index listed in a loaded Dial Plan file for determining the IP destination of Tel-to-IP /IP-to-IP calls. This enables the mapping of called numbers to IP addresses (in dotted-decimal notation) or FQDNs (up to 15 characters).

➢ **To configure routing to an IP destination based on Dial Plan:**

1. Create the Dial Plan file. The syntax of the Dial Plan index for this feature is as follows:

```
<destination / called prefix number>,0,<IP destination>
```

Note that 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.

3. Assign the Dial Plan index to the required routing rule:

   a. Open the Outbound IP Routing table.

   b. In the 'Destination Address' field, enter the required Dial Plan index using the following syntax:

   DialPlan<index>

   Where "DialPlan0" denotes [PLAN1] in the Dial Plan file, "DialPlan1" denotes [PLAN2], and so on.

> **Note:**   The "DialPlan" string is case-sensitive.

31.1.4   User Information File

This section describes the various uses of the User Info file.

You can load the User Info file using any of the following methods:

- Web interface (see 'Loading Auxiliary Files' on page 365)
- *ini* file - using the UserInfoFileName parameter, e.g., UserInfoFileName = 'UserInformationFile.txt' (see 'Auxiliary and Configuration File Name Parameters' on page 640)
- Automatic update mechanism - using the UserInfoFileURL parameter, e.g., UserInfoFileUrl = 'http://192.168.0.250/Audiocodes/ UserInformationFile.txt' (see 'Automatic Update Mechanism' on page 385)

31.1.4.1 User Information File for PBX Extensions and "Global" Numbers

The User Info file contains a User Info table that can be used for the following Gateway-related:

- **Mapping (Manipulating) PBX Extension Numbers with Global Phone Numbers:** maps PBX extension number, connected to the device, with any "global" phone number (alphanumerical) for the IP side. In this context, the "global" phone number serves as a routing identifier for calls in the "IP world" and the PBX extension uses this mapping to emulate the behavior of an IP phone. This feature is especially useful in scenarios where unique or non-consecutive number translation per PBX is needed. This number manipulation feature supports the following call directions:

    - IP-to-Tel Calls: Maps the called "global" number (in the Request-URI user part) to the PBX extension number. For example, if the device receives an IP call destined for "global" number 638002, it changes this called number to the PBX extension number 402, and then sends the call to the PBX extension on the Tel side.

> **Note:**   If you have configured regular IP-to-Tel manipulation rules (see 'Configuring Source/Destination Number Manipulation' on page 235), the device applies these rules before applying the mapping rules of the User Info table.

    - Tel-to-IP Calls: Maps the calling (source) PBX extension to the "global" number. For example, if the device receives a Tel call from PBX extension 402, it changes this calling number to 638002, and then sends call to the IP side with this calling number. In addition to the "global" phone number, the display name (caller ID) configured for the PBX user in the User Info table is used in the SIP From header.

> **Note:**   If you have configured regular Tel-to-IP manipulation rules (see 'Configuring Source/Destination Number Manipulation' on page 235), the device applies these rules before applying the mapping rules of the User Info table.

- IP-to-IP Calls: Maps SIP From (calling number) and To (called number) of IP PBX extension numbers with "global" numbers. For example, if the device receives a call from IP PBX extension number 402 (calling / SIP From) that is destined to IP PBX extension number 403 (called / SIP To), the device changes both these numbers into their "global" numbers 638002 and 638003, respectively.

■ **Registering Users:** The device can register each PBX user configured in the User Info table. For each user, the device sends a SIP REGISTER to an external IP-based Registrar server, using the "global" number in the From/To headers. If authentication is necessary for registration, the device sends the user's username and password, configured in the User Info table, in the SIP MD5 Authorization header.

> **Notes:**
>
> - To enable the User Info table, see 'Enabling the User Info Table' on page 377.
> - To modify the Use Info table, you need to load a new User Info table containing your modifications.
> - To enable user registration, set the following parameters on the Proxy & Registration page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Proxy & Registration**) as shown:
>   - √ 'Enable Registration' parameter set to **Enable** (IsRegisterNeeded is set to 1).
>   - √ 'Registration Mode' parameter set to **Per Endpoint** (AuthenticationMode is set to 0).
> - For FXS ports, when the device needs to send a new SIP request with the Authorization header (e.g., after receiving a SIP 401 response), it uses the username and password configured in the Authentication table (see 'Configuring Authentication per Port' on page 302). To use the username and password configured in the User Info file, set the 'Password' parameter to any value other than its default value.

The User Info file is a text-based file that you can create using any text-based program such as Notepad. To add mapping rules to this file, use the following syntax:

```
[ GW ]
FORMAT
PBXExtensionNum,GlobalPhoneNum,DisplayName,UserName,Password
```

Where:

■ *PBXExtensionNum* is the PBX extension number (up to 10 characters)

■ *GlobalPhoneNum* is the "global" phone number (up to 20 characters) for the IP side

■ *DisplayName* is the Caller ID (string of up to 30 characters) of the PBX extension

■ *UserName* is the username (string of up to 40 characters) for registering the user when authentication is necessary

■ *Password* is the password (string of up to 20 characters) for registering the user when authentication is necessary

Each line in the file represents a mapping rule of a single PBX extension user.

You can add up to 25 mapping rules. The maximum size of the User Info file is 10,800 bytes.

> **Note:**
>
> - Make sure that there are no spaces between the values.
> - Make sure that the last line in the User Info file ends with a carriage return (i.e., by pressing the <Enter> key).

An example of a configured User Info file is shown below:

```
[ GW ]
FORMAT
PBXExtensionNum,GlobalPhoneNum,DisplayName,UserName,Password
401,638001,Mike,miked,1234
402,638002,Lee,leep,4321
403,638003,Sue,suer,8790
404,638004,John,johnd,7694
405,638005,Pam,pame,3928
406,638006,Steve,steveg,1119
407,638007,Fred,frede,8142
408,638008,Maggie,maggiea,9807
```

31.1.4.2 Enabling the User Info Table

The procedure below describes how to load a User Info file to the device and enable the use of the User Info table:

➢   **To enable the User Info table:**

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

2.   Set the 'Enable User-Information Usage' parameter to **Enable**.

31.2   Software License Key

The device is shipped with a pre-installed Software License Key, which determines the device's supported features, capabilities, and available resources. You can upgrade or change your device's supported features by purchasing and installing a new Software License Key to match your requirements.

> **Note:** The availability of certain Web pages depends on the installed Software License Key.

31.2.1   Obtaining the Software License Key File

Before you can install a new Software License Key, you need to obtain a Software License Key file for your device with the required features from your AudioCodes representative. The Software License Key is an encrypted key in string format that is associated with the device's serial number ("S/N") and supplied in a text-based file.

If you need a Software License Key for more than one device, the Software License Key file can include multiple Software License Keys (see figure below). In such cases, each Software License Key in the file is associated with a unique serial number identifying the specific

device. When loading such a Software License Key file, the device installs only the Software License Key that is associated with its serial number.

Figure 4: Software License Key File with Multiple S/N Lines

➢ **To obtain a Software License Key:**

1. Make a note of the MAC address and/or serial number of the device:

   a. Open the Device Information page (**Status & Diagnostics** tab > **System Status** menu > **Device Information**).

   b. The MAC address is displayed in the "MAC Address" field and the serial number in the "Serial Number" field.

2. If you need a Software License Key for more than one device, repeat Step 1 for each device.

3. Request the required Software License Key from your AudioCodes representative and provide them with the MAC address and/or serial number of the device(s).

4. When you receive the new Software License Key file, check the file as follows:

   a. Open the file with any text-based program such as Notepad.

   b. Verify that the first line displays "[LicenseKeys]".

   c. Verify that the file contains one or more lines in the following format:
   "S/N<serial number> = <Software License Key string>".
   For example: "S/N370604 = jCx6r5tovCIKaBBbhPtT53Yj..."

   d. Verify that the "S/N" value reflects the serial number of your device. If you have multiple Software License Keys, ensure that each "S/N" value corresponds to a device.

Warning: Do not modify the contents of the Software License Key file.

---

5. Install the Software License Key on the device as described in 'Installing the Software License Key' on page 378.

31.2.2 Installing the Software License Key

Once you have received your Software License Key file from your AudioCodes representative, you can install it on the device using one of the following management tools:

■ Web interface - see 'Installing Software License Key using Web Interface' on page 379

■ AudioCodes AcBootP utility - see Installing Software License Key using AcBootP on page 380

■ AudioCodes OVOC - refer to the OVOC User's Manual or OVOC Product Description

⚠

> **Note:** When you install a new Software License Key, it is loaded to the device's non-volatile flash memory and overwrites the previously installed Software License Key.

31.2.2.1 Installing Software License Key using Web Interface

The procedure below describes how to install the Software License Key using the Web interface.

➢ **To install the Software License Key using the Web interface:**

1. Open the Software Upgrade Key Status page (**Maintenance** tab > **Software Update** menu > **Software Upgrade Key**).

```
Current Key  tmMmr5to0i4naRp4qxdVu1wQ83lZpT4ob3lr8hovLFVGIB12p1?CFO0c8z4FcOIcfOx3ejcYgC8IalNerN
```

```
Key features:
Board Type: MP-118
Coders: G723 G729 G728 NETCODER GSM-FR GSM-EFR AMR EVRC-QCELP G727 ILBC EVRC-B
AMR-WB G722 EG711 MS_RTA_NB
DSP Voice features:
Channel Type: RTP ATM PCI DspCh=30 IPMediaDspCh=30
E1Trunks=84
T1Trunks=84
FXSPorts=24
FXOPorts=24
Control Protocols: MGCP MEGACO H323 SIP
Default features:
Coders: G711 G726
```

Add a Software Upgrade Key

[Add Key]

Send "Upgrade Key" file from your computer to the device
[Browse...] [Send File]
Reset with flash burn is required after file is loaded.

2. As a precaution, backup the Software License Key currently installed on the device. If the new Software License Key does not comply with your requirements, you can re-load this backup to restore the device's original capabilities.

a. In the 'Current Key' field, select the entire text string and copy it to any standard text file (e.g., Notepad).

b. Save the text file with any file name and file extension (e.g., key.txt) to a folder on your computer.

3. Depending on whether you are loading a Software License Key file with a single Software License Key (i.e., one "S/N") or with multiple Software License Keys (i.e., more than one "S/N"), do one of the following:

- **Loading a File with a Single Software License Key:**
  a. Open the Software License Key file using a text-based program such as Notepad.
  b. Copy-and-paste the string from the file to the 'Add a Software Upgrade Key' field.
  c. Click the **Add Key** button.

- **Loading a File with Multiple Software License Keys:**
  a. In the 'Load Upgrade Key file ...' field, click the **Browse** button and navigate to the folder in which the Software License Key file is located on your computer.
  b. Click **Load File**; the new key is installed on the device.

If the Software License Key is valid, it is burned to the device's flash memory and displayed in the 'Current Key' field.

4. Verify that the Software License Key was successfully installed, by doing one of the following:

- In the Software Upgrade Key Status page, check that the listed features and capabilities activated by the installed Software License Key match those that were ordered.

- Access the Syslog server and ensure that the following message appears in the Syslog server:

  "S/N___ Key Was Updated. The Board Needs to be Reloaded with ini file\n"

5. Reset the device; the new capabilities and resources enabled by the Software License Key are active.

> **Note:** If the Syslog server indicates that the Software License Key was unsuccessfully loaded (i.e., the "SN_" line is blank), do the following preliminary troubleshooting procedures:
>
> 1. Open the Software License Key file and check that the "S/N" line appears. If it does not appear, contact AudioCodes.
> 2. Verify that you have loaded the correct file. Open the file and ensure that the first line displays "[LicenseKeys]".
> 3. Verify that the content of the file has not been altered.

31.2.2.2 Installing Software License Key using BootP/TFTP

The procedure below describes how to install a Software License Key using AudioCodes AcBootP utility.

> **Notes:**
>
> - When loading the Software License Key file, a cmp file must also be loaded during this BootP process.
> - For more information on using the AcBootP utility, refer to the document *AcBootP Utility User's Guide*.

➢ **To install a Software License Key using the AcBootP utility:**

1. Change the file extension name of the Software License Key file from .txt to .ini.

2. Place the Software License Key file in the same folder in which the device's *cmp* file is located.

3. Start the AcBootP utility.

4. Click the **Client Configuration** tab, and then from the 'INI File' drop-down list, select the Software License Key file.

5. From the 'BootP File' drop-down list, select the device's *cmp* file.

6. Configure the initial BootP/TFTP parameters as required, and then click **Apply**.

7. Reset the device; the *cmp* and Software License Key files are loaded to the device.

31.3   Software Upgrade Wizard

The Software Upgrade Wizard allows you to upgrade the device's firmware. The firmware file has the .cmp file extension name. The wizard also enables you to load an *ini* file and/or auxiliary files (typically loaded using the Load Auxiliary File page described in 'Loading Auxiliary Files' on page 365). However, it is mandatory when using the wizard to first load a .cmp file to the device. You can then choose to also load an *ini* file and/or auxiliary files, but this cannot be done without first loading a .cmp file. For the *ini* and each auxiliary file type, you can choose to load a new file or not load a file but use the existing file (i.e., maintain existing configuration) running on the device.

> **Warning:** The Software Upgrade Wizard requires the device to be reset at the end of the process, which may disrupt traffic. To avoid this, disable all traffic on the device before initiating the wizard by performing a graceful lock (see 'Basic Maintenance' on page 359).

> **Notes:**
>
> • You can get the latest software files from AudioCodes Services Portal (registered users only) at https://services.audiocodes.com.
>
> • Before upgrading the device, it is recommended that you save a copy of the device's configuration settings (i.e., *ini* file) to your computer. If an upgrade failure occurs, you can then restore your configuration settings by uploading the backup file to the device. For saving and restoring configuration, see 'Backing Up and Loading Configuration File' on page 384.
>
> • If you wish to also load an *ini* or auxiliary file, it is mandatory to first load a .cmp file.
>
> • When you activate the wizard, the rest of the Web interface is unavailable. After the files are successfully loaded, access to the full Web interface is restored.
>
> • If you upgraded your .cmp and the "SW version mismatch" message appears in the Syslog or Web interface, then your Software License Key does not support the new .cmp file version. If this occurs, contact AudioCodes support for assistance.
>
> • If you use the wizard to load an *ini* file, parameters excluded from the *ini* file are assigned default values (according to the .cmp file running on the device) thereby, overriding values previously defined for these parameters.
>
> • You can schedule automatic loading of these files using HTTP/HTTPS, FTP, or NFS (see 'Automatic Update' on page 385).

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

1. Stop all traffic on the device using the Graceful Lock feature (refer to the warning bulletin above).

2. Open the Software Upgrade wizard, by performing one of the following:

- Select the **Maintenance** tab, click the **Software Update** menu, and then click **Software Upgrade Wizard**.

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

Figure 31-5: Start Software Upgrade Wizard Screen
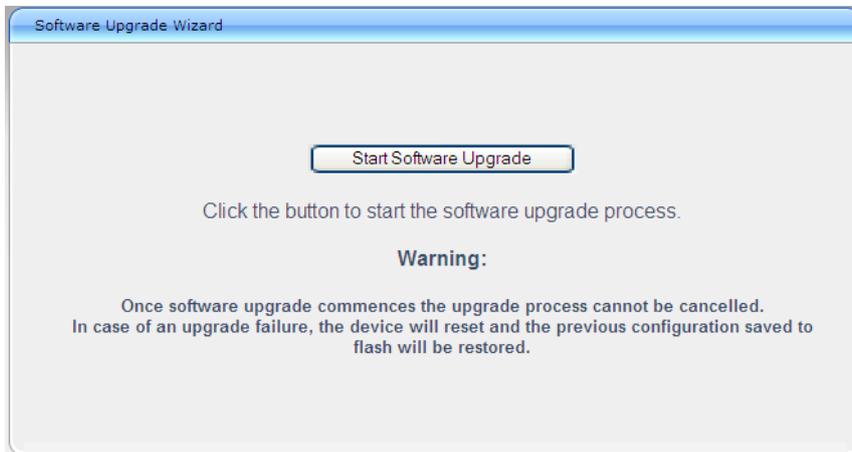
3. Click the **Start Software Upgrade** button; the wizard starts, requesting you to browses to a .cmp file for uploading.

> **Note:** At this stage, you can quit the Software Update Wizard, by clicking **Cancel** , without requiring a device reset. However, once you start uploading a cmp file, the process must be completed with a device reset. If you choose to quit the process in any of the subsequent pages, the device resets.

4. Click the **Browse** button, navigate to the .cmp file, and then click **Load File**; a progress bar appears displaying the status of the loading process. When the .cmp file is successfully loaded to the device, a message appears notifying you of this.

5. If you want to load **only** a .cmp file, then click the **Reset** button to reset the device with the newly loaded .cmp file, utilizing the existing configuration (*ini*) and auxiliary files. To load additional files, skip to the next Step.

> **Note:** Device reset may take a few minutes depending on cmp file version (this may even take up to 10 minutes).

6. Click the **Next** button; the wizard page for loading an *ini* file appears. You can now perform one of the following:

- Load a new *ini* file: Click **Browse**, navigate to the *ini* file, and then click **Send File**; the *ini* file is loaded to the device and you're notified as to a successful loading.

- Retain the existing configuration (*ini* file): Do not select an *ini* file, and ensure that the 'Use existing configuration' check box is selected (default).

- Return the device's configuration settings to factory defaults: Do not select an *ini* file, and clear the 'Use existing configuration' check box.

7. Click the **Next** [▶] button to progress to the relevant wizard pages for loading the desired auxiliary files. To return to the previous wizard page, click the **Back** [◀] button. As you navigate between wizard pages, the relevant file type corresponding to the Wizard page is highlighted in the left pane.

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

9. Click the **Reset** [⟳] button to complete the upgrade process; the device 'burns' the newly loaded files to flash memory and then resets the device.

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

After the device resets, the End of Process wizard page appears displaying the new .cmp and auxiliary files loaded to the device.

Figure 31-6: Software Upgrade Process Completed Successfully
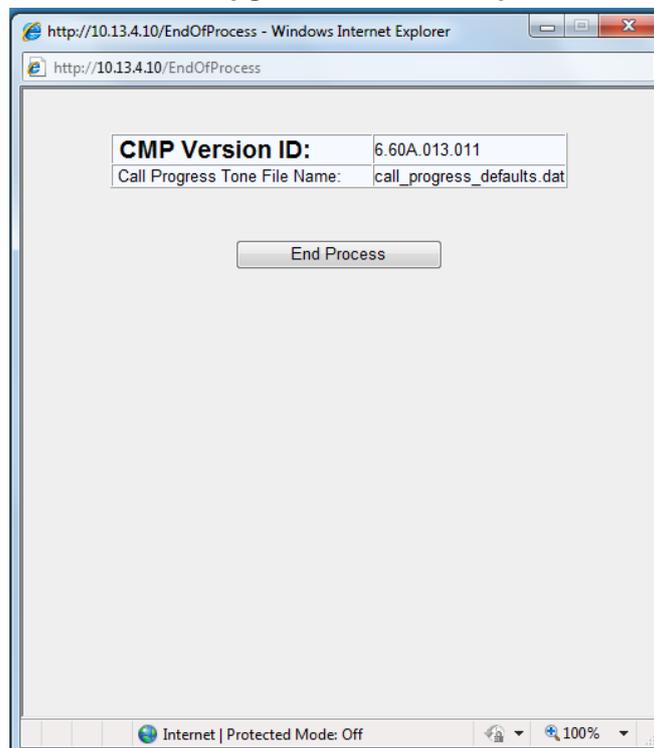
10. Click **End Process** to close the wizard; the Web Login dialog box appears.

11. Enter your login user name and password, and then click **OK**; a message box appears informing you of the new .cmp file.

12. Click **OK**; the Web interface becomes active, reflecting the upgraded device.

31.4 Backing Up and Loading Configuration File

You can save a copy/backup of the device's current configuration settings as an *ini* file to a folder on your computer, using the Configuration File page. The saved *ini* file includes only parameters that were modified and parameters with other than default values. The Configuration File page also allows you to load an *ini* file to the device. If the device has "lost" its configuration, you can restore the device's configuration by loading the previously saved *ini* file or by simply loading a newly created *ini* file.

> **Notes:**
>
> - When loading an *ini* file using this Web page, parameters not included in the *ini* file are reset to default settings.
>
> -

➢ **To save the ini file:**

1. Open the Configuration File page by doing one of the following:
   - From the Navigation tree, click the **Maintenance** tab, click the **Software Update** menu, and then click **Configuration File**.
   - On the toolbar, click **Device Actions**, and then from the drop-down menu, choose **Load Configuration File** or **Save Configuration File**.

2. To save the *ini* file to a folder on your computer, do the following:
   a. Click the **Save INI File** button; the File Download dialog box appears.
   b. Click the **Save** button, navigate to the folder where you want to save the *ini* file, and then click **Save**.
3. To load the *ini* file to the device, do the following:
   a. Click the **Browse** button, navigate to the folder where the *ini* file is located, select the file, and then click **Open**; the name and path of the file appear in the field beside the **Browse** button.
   b. Click the **Load INI File** button, and then at the prompt, click **OK**; the device uploads the *ini* file and then resets (from the *cmp* version stored on the flash memory). Once complete, the Web Login screen appears, requesting you to enter your user name and password.

32   Automatic Update

This chapter describes the device's automatic provisioning mechanisms.

32.1   Automatic Configuration Methods

The device supports the following automatic provisioning methods:

- DHCP (Option 66, Option 67, Option 160)
- HTTP/S
- TFTP
- FTP
- NFS
- SNMP (OVOC)
- BootP / TFTP

32.1.1   BootP Request and DHCP Discovery upon Device Initialization

After the device powers up or is physically reset, it broadcasts a BootP request to the network. If it receives a reply from a BootP server, it changes its network parameters (IP address, subnet mask and default gateway address) according to the values provided by the BootP server. If there is no reply from a BootP server and if DHCP is enabled, the device initiates a standard DHCP procedure to configure its network parameters.

After changing the network parameters, the device attempts to load the firmware file (cmp) and various configuration files from the TFTP server's IP address received from the BootP/DHCP server. If a TFTP server's IP address is not obtained, the device attempts to load the cmp file and/or configuration files from a preconfigured TFTP server. Thus, the device can obtain its network parameters from BootP or DHCP servers, and its software and configuration files from a different TFTP server (preconfigured in the ini configuration file).

If BootP/DHCP servers are not found or when the device is reset using the Web interface or SNMP, it retains its network parameters and attempts to load the cmp file and/or configuration files from a preconfigured TFTP server. If a preconfigured TFTP server does

not exist, the device operates using the existing software and configuration files on its flash memory.

Figure 32-1: BootP Request and DHCP Discovery upon Startup
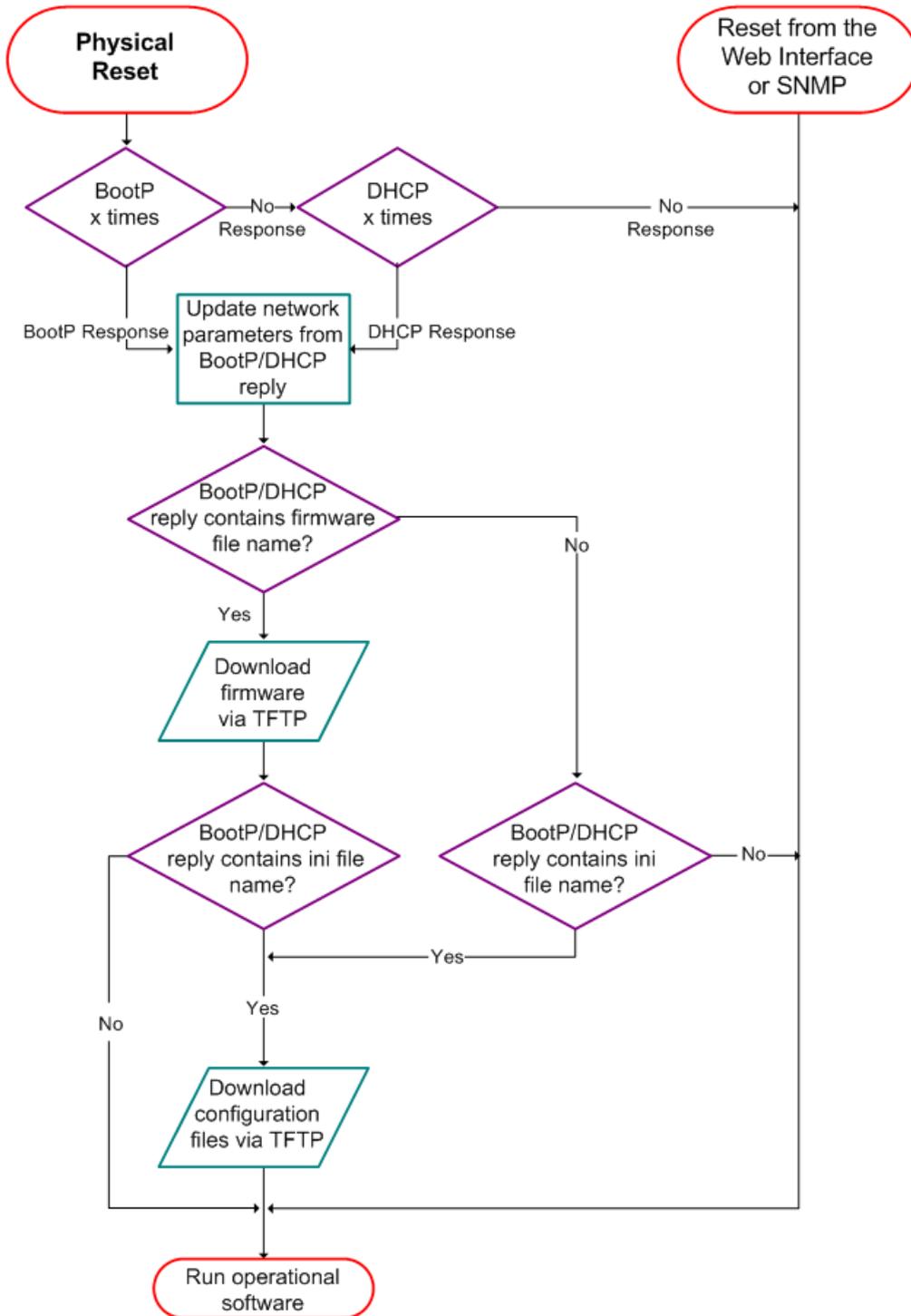
> ⚠️ **Note:** By default, the duration between BootP/DHCP requests sent by the device is one second (configured by the BootPDelay ini file parameter). By default, the number of requests is three (configured by the BootPRetries ini file parameter).

32.1.2   VLAN ID Discovery using LLDP

You can enable the device, through the EnableLLDP ini file parameter, to use the Link Layer Discovery Protocol (LLDP) discovery protocol to obtain over the Layer-2 data link layer, a VLAN ID for its OAMP interface (per IEEE 802.1, IEEE 802.3 and TR-41). The device supports LLDP attributes referred to as TLVs (type, length, and value descriptions), which are sent and received in LLDP messages. The device is defined as a Media Endpoint Device (LLDP-MED, Class II), which is an extension of LLDP. The device supports all TLV's per specification:

■ IEEE 802.1 TLV's: Chassis ID; Port ID; Time To Live; End Of LLDPDU.

■ Basic management TLV's: Port Description; System Name; System Description; System Capabilities.

■ IEEE 802.3 Organizationally Specific TLV's: MAC/PHY Configuration/Status (indicates the auto-negotiation capability and the duplex/speed status of IEEE 802.3 MAC/PHYs); Maximum Frame Size (indicates the maximum supported IEEE 802.3 frame size).

■ LLDP-MED TLVs: LLDP-MED Capabilities; Network Policy.

Upon startup (reset or powered up), the device sends an LLDP broadcast message containing its identity to request a VLAN ID from a server in the network. If it receives an LLDP reply with a VLAN ID from a server within 30 seconds, the device overwrites the current VLAN ID of its OAMP interface (in the Interface table) with this new VLAN ID. If no LLDP reply is received within 30 seconds, the device continues with its normal startup process.

32.1.3   Local Configuration Server with BootP/TFTP

Local configuration server with BootP/TFTP provides an easy and efficient method for automatic provisioning, where configuration occurs at a staging warehouse, as follows:

1. Install AudioCodes AcBootP/TFTP utility program on a computer located in a staging warehouse.

2. Prepare a standard configuration *ini* file and place it in the TFTP directory.

3. Enter the MAC address of each device in the AcBootP utility.

4. For each device added in the BootP utility, select the cmp and *ini* file in the 'BootP File' field.

5. Connect each device to the network and then power up the device.

6. The BootP reply contains the cmp and *ini* file names entered in the 'BootP File' field. Each device retrieves these files using BootP and stores them in its flash memory. If Auxiliary files are required (e.g., call progress tones), they may also be specified in the *ini* file and downloaded from the same TFTP server.

7. When the devices' LEDs turn green indicating that the files were successfully loaded, disconnect the devices and ship to the customer.

> **Notes:**
>
> • Typically, IP addressing at the customer site is done by DHCP.
>
> • For more information on the AcBootP utility, refer to the *AcBootP Utility User's Guide*.

32.1.4 DHCP-based Provisioning

This method is similar to the setup described in Local Configuration Server with BootP/TFTP on page 387, except that DHCP is used instead of BootP. A third-party DHCP server can be configured to automatically provide each device, acting as a DHCP client, with a temporary IP address so that individual MAC addresses are not required. The DHCP server can provide additional networking parameters such as subnet mask, default gateway, primary and secondary DNS server, and two SIP server addresses. These network parameters have a time limit, after which the device must 'renew' its lease from the DHCP server.

The device can use a host name in the DHCP request. The host name is set to acl_nnnnn, where *nnnnn* denotes the device's serial number. The serial number is the last six digits of the MAC address converted to decimal representation. In networks that support this feature and if the DHCP server registers this host name to a DNS server, you can access the device (through a Web browser) using the URL, http://acl_<serial number> (instead of using the device's IP address). For example, if the device's MAC address is 00908f010280, the DNS name is acl_66176.

Notes:

- Throughout the DHCP procedure, make sure that the BootP/TFTP program (AcBootP utility) is deactivated; otherwise the device receives a response from the BootP server instead of the DHCP server. Typically, after the device powers up, it attempts to communicate with a BootP server. If a BootP server does not respond and DHCP is enabled, the device attempts to obtain its networking parameters from the DHCP server.

- When using DHCP to acquire an IP address, the Interface table, VLANs and other advanced configuration options are disabled.

- For more information on DHCP, see BootP Request and DHCP Discovery upon Device Initialization on page 385.

- For additional DHCP parameters, see "DHCP Parameters" on page 479.

➢ **To enable the device as a DHCP client:**

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

Figure 32-2: Enabling DHCP - Application Settings Page
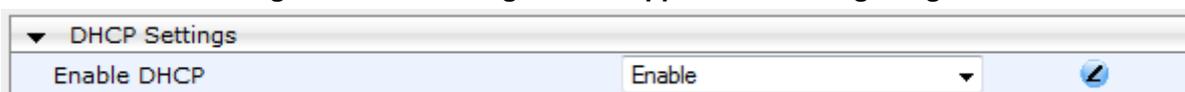
2. From the 'Enable DHCP" drop-down list, select **Enable**.

3. Click **Submit**.

4. To activate the DHCP process, reset the device.

The following shows an example of a configuration file for a Linux DHCP server (dhcpd.conf). The devices are allocated temporary IP addresses in the range 10.31.4.53 to 10.31.4.75. TFTP is assumed to be on the same computer as the DHCP server (alternatively, the "next-server" directive may be used).

```
ddns-update-style ad-hoc;
default-lease-time 60;
max-lease-time 60;
class "gateways" {
    match if(substring(hardware, 1, 3) = 00:90:8f);
}
subnet 10.31.0.0 netmask 255.255.0.0 {
    pool {
```

```
              allow members of "audiocodes";
              range 10.31.4.53 10.31.4.75;
              filename "SIP_F6.60A.217.003.cmp -fb;device.ini";
              option routers                  10.31.0.1;
              option subnet-mask              255.255.0.0;
      }
}
```

> **Notes:**
>
> - If, 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.

32.1.4.1 Provisioning from HTTP Server using DHCP Option 67

Most DHCP servers support the configuration of individual DHCP option values for different devices on the network. The DHCP configuration should be modified so that the device receives a URL to the configuration file in Option 67, along with IP addressing and DNS server information. The DHCP response is processed by the device upon startup and the device automatically downloads the configuration file from the HTTP server specified in the DHCP response. This method is NAT-safe.

Below is an example of a Linux DHCP configuration file (dhcpd.conf) showing the required format of Option 67:

```
ddns-update-style ad-hoc;
default-lease-time 3600;
max-lease-time 3600;
class "audiocodes" {
    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;
          option routers                  10.31.0.1;
          option subnet-mask              255.255.0.0;
          option domain-name-servers      10.1.0.11;
          option bootfile-name
"INI=http://www.corp.com/master.ini";
          option dhcp-parameter-request-list 1,3,6,51,67;
```

```
        }
}
```

32.1.4.2 Provisioning from TFTP Server using DHCP Option 66

This method is suitable when the network in which the device is deployed contains a provisioning TFTP server for all network equipment, without being able to distinguish between AudioCodes and non-AudioCodes devices.

Upon startup, the device searches for Option 66 in the DHCP response from the DHCP server. If Option 66 contains a valid IP address, the device attempts to download, through TFTP, a file that has a filename containing the device's MAC address (e.g., 00908f0130aa.ini). This method requires a provisioning server at the customer premises.

This method loads the configuration file to the device as a one-time action. The download is only repeated if the device is manually restored to factory defaults (by pressing the hardware reset button while the Ethernet cable is not connected) and DHCP is enabled (see note below).

> **Notes:**
>
> - For TFTP configuration using DHCP Option 66, enable DHCP on your device: DHCPEnable = 1 and DHCPRequestTFTPParams = 1.
> - Access to the core network using TFTP is not NAT-safe.
> - The TFTP data block size (packets) when downloading a file from a TFTP server for the Automatic Update mechanism can be configured using the AUPDTftpBlockSize parameter.

32.1.4.3 Provisioning the Device using DHCP Option 160

You can provision the device using DHCP Option 160. DHCP Option 160 provides the device with the URL address of the provisioning server from where it can download its software (.cmp) and configuration (.ini) files. The URL can also include the names of the required files to download and their folder location on the server.

If you enable DHCP client functionality with DHCP Option 160, upon a device reset or power up, the device (as a DHCP client) sends a DHCP request to the DHCP server to obtain networking information (e.g., the device's IP address) and the URL address of the provisioning server.

- The following syntax is supported for defining the URL and configuration/firmware filenames in DHCP Option 160 (on the DHCP server):

- <protocol>://<server IP address or hostname>

- <protocol>://<server IP address or hostname>/<software filename>

- <protocol>://<server IP address or hostname>/;<configuration filename>

- <protocol>://<server IP address or hostname>/<software filename>;<configuration filename>

The protocol can be HTTP, HTTPS, FTP, or TFTP. As shown above, the URL can include both the software and configuration filenames. In this case, they must be separated by a semicolon (;) and without spaces.

If the URL does not specify a configuration filename or the file does not exist on the provisioning server, the device requests from the server a "default" configuration file whose name includes the device's product name and MAC address (<Product><MAC>.ini, for example, "MP114FXS00908f5b1035.ini"). If this "default" file also does not exist on the server, the device attempts to retrieve another "default" configuration file whose name includes only the device's product name (<Product>.ini, for example, "MP114FXS.ini"). The device makes up to three attempts to download the configuration file if a failure occurs (i.e.,

file not exist or any other failure reason). This applies to each of the configuration files, as mentioned previously.

If the URL specifies a software file, the device makes only one attempt to download the file (even if a failure occurs). If the URL does not specify a software file, the device does not make any attempt to download a software file.

Once the device downloads the file(s), it undergoes a reset to apply the configuration and/or software. In addition, once the file(s) has been downloaded, the device ignores all future DHCP Option 160 messages. Only if the device is restored to factory defaults will it process Option 160 again (and download any required files).

➢ **To enable provisioning using DHCP Option 160:**

1. Make sure that the DHCP server is configured with the appropriate information (including the URL address of the provisioning server for Option 160).

2. Make sure that the required configuration and/or software files are located on the provisioning server.

3. Enable DHCP client functionality, as described in DHCP-based Provisioning on page 390.

4. Enable the device to include DHCP Option 160 in the DHCP Parameter Request List field of the DHCP request packet that is sent to the DHCP server. Do this by loading an ini file to the device with the following parameter setting:

   ```
   DhcpOption160Support = 1
   ```

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

32.1.5 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 388or via TFTP at a staging warehouse. The URL is configured using the IniFileURL parameter.

■ Private labeling (preconfigured during the manufacturing process).

■ Using DHCP Option 67 (see Provisioning from HTTP Server using DHCP Option 67 on page 389).

■ Manually on-site, using the RS-232 port or Web interface.

When the device is deployed at the customer site, local DHCP server provides the devices with IP addressing and DNS server information. From the URL provided in the DHCP response, the device can then contact the HTTP server at the core network and automatically download its configuration. The URL can be a simple file name or contain the device's MAC or IP address, e.g.:

■ *http://corp.com/config-<MAC>.ini* - which becomes, for example, http://corp.com/config-00908f030012.ini

■ *http://corp.com/<IP>/config.ini* - which becomes, for example, http://corp.com/192.168.0.7/config.ini

For more information on HTTP-based provisioning, see "HTTP/S-Based Provisioning using the Automatic Update Feature" on page 393.

32.1.5.1 Loading Files Securely by Disabling TFTP

The TFTP protocol is not considered secure and some network operators block it using a firewall. It is possible to disable TFTP completely, using the *ini* file parameter EnableSecureStartup (set to 1). Secure protocols such as HTTPS may be used to fetch the device configuration instead.

➢ **To download the ini file to the device using HTTPS instead of TFTP:**

1. Prepare the device's configuration file on an HTTPS server and obtain a URL to the file (e.g., https://192.168.100.53/gateways.ini).

2. Enable DHCP, if necessary.

3. Enable SSH and connect to it.

4. In the CLI, use the *ini* file parameters IniFileURL (for defining the URL of the configuration file) and EnableSecureStartup (for disabling TFTP), and then restart the device with the new configuration:

```
/conf/scp IniFileURL https://192.168.100.53/gateways.ini
/conf/scp EnableSecureStartup 1
/conf/sar bootp
```

Note: Once Secure Startup has been enabled, it can only be disabled by setting EnableSecureStartup to 0 using the CLI. Loading a new ini file using BootP/TFTP is not possible until EnableSecureStartup is disabled.

32.1.6   FTP- or NFS-based Provisioning

Some networks block access to HTTP(S). The Automatic Update feature provides limited support for FTP/FTPS connectivity. Periodic polling for updates is not possible since these protocols do not support conditional fetching, i.e., updating files only if it is changed on the server.

The only difference between this method and those described in ''HTTP-based Provisioning'' on page 391 and Provisioning from HTTP Server using DHCP Option 67 on page 389 is that the protocol in the URL is "ftp" (instead of "http").

Notes:

- Unlike FTP, NFS is not NAT-safe.
- NFS v2/v3 is also supported.

32.1.7   Provisioning using AudioCodes OVOC

AudioCodes OVOC server functions as a core-network provisioning server. The device's SNMP Manager should be configured with the IP address of the OVOC server, using one of the methods detailed in the previous sections. As soon as a registered device contacts the OVOC server through SNMP, the OVOC 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.

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

The Automatic Update feature can be used for automatic provisioning of the device through HTTP/S. Automatic provisioning is useful for large-scale deployment of devices. In some cases, the devices are shipped to the end customer directly from the manufacturer. In other cases, they may pass through a staging warehouse. Configuration may occur at the staging warehouse or at the end-customer premises.

The device may be preconfigured during the manufacturing process (commonly known as private labeling). Typically, a two-stage configuration process is implemented whereby initial configuration includes only basic configuration, while the final configuration is done only when the device is deployed in the live network.

Warning: If you use the IniFileURL parameter for the Automatic Update feature, do not use the Web interface to configure the device. If you do configure the device through the Web interface and save (burn) the new settings to the device's flash memory, the IniFileURL parameter is automatically set to 0 and Automatic Updates is consequently disabled. To enable Automatic Updates again, you need to re-load the ini file (using the Web interface or BootP) with the correct IniFileURL settings. As a safeguard to an unintended burn-to-flash when resetting the device, if the device is configured for Automatic Updates, the 'Burn To FLASH' field under the Reset Configuration group in the Web interface's Maintenance Actions page is automatically set to No by default.

> **Notes:**
> - For a description of all the Automatic Update parameters, see "Automatic Update Parameters" on page 641.
> - 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>.

32.2.1 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)

32.2.2 File Location for Automatic Update

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

The Automatic Update feature is done per file and configured by specifying the file name and URL address of the provisioning server where the file is located. For a description of the parameters used to configure URLs per file, see "Automatic Update Parameters" on page 641. 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

> **Note:** For configuration files (ini), the file name in the URL can automatically contain the device's MAC address for enabling the device to download a file unique to the device. For more information, see "MAC Address Automatically Inserted in Configuration File Name" on page 398.

32.2.3 Triggers for Automatic Update

The Automatic Update feature can be triggered by the following:

- Upon device startup (reset or power up).
- Upon startup, but before the device is operational, if the Secure Startup feature is enabled (see Loading Files Securely by Disabling TFTP on page 392).
- Periodically:
  - Specified time of day (e.g., 18:00), configured by the ini file parameter AutoUpdatePredefinedTime.
  - Interval between Automatic Updates (e.g., every 60 minutes), configured by the ini file parameter AutoUpdateFrequency.
- Centralized provisioning server request:
  - Upon receipt of an SNMP request from the provisioning server.

- Upon receipt of a special SIP NOTIFY message from the provisioning server. The NOTIFY message includes an Event header with the AudioCodes proprietary value, "check-sync;reboot=false", as shown in the example below:

```
NOTIFY sip:<user>@<dsthost> SIP/2.0
To: sip:<user>@<dsthost>
From: sip:sipsak@<srchost>
CSeq: 10 NOTIFY
Call-ID: 1234@<srchost>
Event: check-sync;reboot=false
```

To enable this feature through the Web interface:

a. Open the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**).

b. Under the **Misc Parameters** group, set the 'SIP Remote Reset' parameter to **Enable**.

c. Click **Submit**.

32.2.4   Access Authentication with HTTP Server

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

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

  - ini file:

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

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

32.2.5   Querying Provisioning Server for Updated Files

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

1. If you have configured the device to authenticate itself to the HTTP/S server for secure access, the device sends the access authentication username and password to the HTTP/S server (for more information, see "Access Authentication with HTTP Server" on page 395). 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. The information can include any user-defined string or the following supported string variable tags (case-sensitive):

- **<NAME>:** product name, according to the installed Software License Key

- **<MAC>:** device's MAC address

- **<VER>:** software version currently installed on the device, e.g., "7.00.200.001"

- **<CONF>:** configuration version, as configured by the ini file parameter, INIFileVersion

The device automatically populates these tag variables with actual values in the sent header. By default, the device sends the following in the User-Agent header:

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

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

```
User-Agent: MyWorld-Mediant;7.00.200.001(00908F1DD0D3)
```

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

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

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

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

If the file on the provisioning server was unchanged (modified) since the date and time specified in the header, the server replies with an HTTP 304 response and the file is not downloaded. If the file was modified, the provisioning server sends an HTTP 200 OK response with the file in the body of the HTTP response. The device downloads the file and compares the version of the file with the currently installed version on its flash memory. If the downloaded file is of a later version, the device installs it after the device resets (which is only done after the device completes all file downloads); otherwise, the device does not reset and does not install the file.

To enable the automatic software (.cmp) file download method based on this timestamp method, use the ini file parameter, AutoCmpFileUrl. The device uses the same configured URL to download the .cmp file for each subsequent Automatic Update process.

You can also enable the device to run a CRC on the downloaded configuration file (ini) to determine whether the file has changed in comparison to the previously downloaded file. Depending on the CRC result, the device can install or discard the downloaded file. For more information, see "Cyclic Redundancy Check on Downloaded Configuration Files" on page 398.

Notes:

- When this method is used, there is typically no need for the provisioning server to check the device's current firmware version using the HTTP-User-Agent header.

- The Automatic Update feature assumes that the Web server conforms to the HTTP standard. If the Web server ignores the If-Modified-Since header or doesn't provide the current date and time during the HTTP 200 OK response, the device may reset itself repeatedly. To overcome this problem, modify the update frequency, using the ini file parameter AutoUpdateFrequency.

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

    Auxiliary Files:
    - ini:
      ```
      CptFileURL =
      'https://www.company.com/call_progress.dat'
      ```

    Software (.cmp) File:
    - ini:
      ```
      CmpFileUrl =
      'https://www.company.com/device/v.6.80A.227.005.cmp'
      ```

Notes:

- For one-time file download, the HTTP Get request sent by the device does not include the If-Modified-Since header. Instead, the HTTP-User-Agent header can be used in the HTTP Get request to determine whether firmware update is required.

- When downloading SSL certificates (Auxiliary file), it is recommended to use HTTPS with mutual authentication for secure transfer of the SSL Private Key.

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

32.2.6   File Download Sequence

Whenever the Automatic Update feature is triggered (see "Triggers for Automatic Update" on page 394), the device attempts to download each file from the configured URLs, in the following order:

1. ini file

2. Periodic software file (.cmp) download

3. One-time software file (.cmp) download

4. Auxiliary file(s)

The following files automatically instruct the device to reset:

■ Periodic software file (.cmp)

■ One-time software file (.cmp)

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

■ ini file: Use the ResetNow in file parameter

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

Notes:

- For ini file downloads, by default, parameters not included in the file are set to defaults. To retain the current settings of these parameters, set the SetDefaultOnINIFileProcess parameter to 0.

- If you have configured one-time software file (.cmp) download (configured by the ini file parameter CmpFileURL), 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.

- 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.

32.2.7 Cyclic Redundancy Check on Downloaded Configuration Files

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

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

For enabling CRC, use the ini file parameter AUPDCheckIfIniChanged. By default, CRC is disabled. For more information on the parameter, see "Automatic Update Parameters" on page 641.

32.2.8 MAC Address Automatically Inserted in Configuration File Name

You can configure the file name of the configuration file (ini) in the URL to automatically include the MAC address of the device. As described in ''File Location for Automatic Update'' on page 394, the file name is included in the configured URL of the provisioning server where the file is located.

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

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

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

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

32.2.9 Automatic Update Configuration Examples

This section provides a few examples on configuring the Automatic Update feature.

32.2.9.1 Automatic Update for Single Device

This simple example describes how to configure the Automatic Update feature for updating a single device. In this example, the device queries the provisioning server for software, configuration and Auxiliary files every 24 hours.

➢ **To set up Automatic Provisioning for single device (example):**

1. Set up an HTTP Web server (e.g., http://www.company.com) and place all the required configuration files on this server.

2. Configure the device with the IP address of the DNS server for resolving the domain name (e.g., http://www.company.com) that is used in the URL of the provisioning server. You configure this in the Interface table:

   - ini File:
     ```
     [ InterfaceTable ]
     FORMAT InterfaceTable_Index =
     InterfaceTable_ApplicationTypes,
     InterfaceTable_InterfaceMode, InterfaceTable_IPAddress,
     InterfaceTable_PrefixLength, InterfaceTable_Gateway,
     InterfaceTable_VlanID, InterfaceTable_InterfaceName,
     InterfaceTable_PrimaryDNSServerIPAddress,
     InterfaceTable_SecondaryDNSServerIPAddress,
     InterfaceTable_UnderlyingDevice;
     InterfaceTable 0 = 6, 10, 10.15.7.95, 16, 10.15.0.1, 1,
     "Voice", 80.179.52.100, 0.0.0.0, "vlan 1";
     [ \InterfaceTable ]
     ```

3. Configure the device with the following Automatic Update settings:

   a. Automatic Update is done every 24 hours (1440 minutes):
      ♦ ini File:
      ```
      AutoUpdateFrequency = 1440
      ```

b. Automatic Update of software file (.cmp):

- ♦ ini File:

```
AutoCmpFileUrl =  'https://www.company.com/sw.cmp'
```

c. Automatic Update of Call Progress Tone file:

- ♦ ini File:

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

d. Automatic Update of ini configuration file:

- ♦ ini File:

IniFileURL = 'https://www.company.com/config.ini'

e. Enable Cyclical Redundancy Check (CRC) on downloaded ini file:

- ♦ ini File:

```
AUPDCheckIfIniChanged = 1
```

4. Power down and then power up the device.

32.2.9.2 Automatic Update from NFS, FTP and HTTP Servers

This example describes how to configure the Automatic Update feature where files are stored and downloaded from different file server types. The example scenario includes the following:

- ■ NFS server (Version 2) at 10.13.2.10 for storing the CPT file.

- ■ FTPS server at ftpserver.corp.com for storing the Voice Prompts (VP) file. The login credentials to the server are username "root" and password "wheel".

- ■ HTTP server at www.company.com for storing the configuration file (ini).

- ■ DNS server at 80.179.52.100 for resolving the domain names of the provisioning servers (FTPS and HTTP).

➢ **To set up Automatic Provisioning for files stored on different server types (example):**

1. CPT file:

a. Set up an NFS server and copy the CPT file to the directory */usr/shared/public* on the NFS server.

b. Configure the device with the NFS server:

```
[ NFSServers ]
FORMAT NFSServers_Index = NFSServers_HostOrIP,
NFSServers_RootPath, NFSServers_NfsVersion,
NFSServers_AuthType, NFSServers_UID, NFSServers_GID,
NFSServers_VlanType;
NFSServers 0 = "10.31.2.10", "/usr/share/public/", 2, 1, 0,
1, 1;
[ \NFSServers ]
```

c. Configure the device with the URL path of the CPT file:

- ♦ ini File:

```
CptFileURL =
'file://10.31.2.10/usr/share/public/usa_tones.dat'
```

2. VP file:

a. Set up an FTPS server and copy the VP file to the server.

b. Configure the device with the URL path of the VP file:

- ♦ ini File:

```
VPFileUrl =
'ftps://root:wheel@ftpserver.corp.com/vp.dat'
```

3. **Software (.cmp) and ini files:**

    a. Set up an HTTP Web server and copy the .cmp and configuration files to the server.

    b. Configure the device with the URL paths of the .cmp and ini files:

        ♦ ini File:
        ```
        AutoCmpFileUrl =
        'http://www.company.com/device/sw.cmp'
        ```
        IniFileURL = 'http://www.company.com/device/inifile.ini'

4. Configure the device with the IP address of the DNS server for resolving the domain names of the FTPS and HTTP servers:

    ```
    [ InterfaceTable ]
    FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes,
    InterfaceTable_InterfaceMode, InterfaceTable_IPAddress,
    InterfaceTable_PrefixLength, InterfaceTable_Gateway,
    InterfaceTable_VlanID, InterfaceTable_InterfaceName,
    InterfaceTable_PrimaryDNSServerIPAddress,
    InterfaceTable_SecondaryDNSServerIPAddress,
    InterfaceTable_UnderlyingDevice;
    InterfaceTable 0 = 6, 10, 10.15.7.95, 16, 10.15.0.1, 1,
    "Voice", 80.179.52.100, 0.0.0.0, "vlan 1";
    [ \InterfaceTable ]
    ```

5. Configure the device to perform the Automatic Update process daily at 03:00 (3 a.m):

    • ini File:
        ```
        AutoUpdatePredefinedTime = '03:00'
        ```

32.2.9.3 Automatic Update for Mass Deployment

This example describes how to configure the Automatic Update feature for updating multiple devices (i.e., mass deployment) using an HTTP provisioning server. In this example, all the devices are configured to download the same "master" configuration file. This file serves as the configuration template and instructs the devices which files to download and how often to perform the Automatic Update process. In addition, the master file also instructs each device to download an ini configuration file whose file name contains the MAC address of the device.

The example scenario is as follows:

■ All devices download a "master" configuration file that contains the following:

   • Common configuration shared by all device's.

   • Specific configuration that instructs each device to download a specific configuration file based on the device's MAC address, using the special string "<MAC>" in the URL, as described in "MAC Address Automatically Inserted in Configuration File Name" on page 398.

■ Device queries the provisioning server daily at 24:00 (midnight) for software, configuration and Auxiliary files.

■ HTTP-based provisioning server at www.company.com for storing the files.

■ DNS server at 80.179.52.100 for resolving the domain name of the provisioning server.

➢ **To set up automatic provisioning for mass provisioning (example):**

1. Create a "master" configuration file template named "master_configuration.ini" with the following settings:

- Common configuration for all devices:
  - ♦ ini file:

    AutoUpdatePredefinedTime = '24:00'
    CptFileURL = 'https://www.company.com/call_progress.dat'
    AutoCmpFileUrl = 'https://www.company.com/sw.cmp'

- Configuration per device based on MAC address:
  - ♦ ini file:

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

    (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'
    ```

- **b.** Configure the device with the IP address of the DNS server for resolving the domain name (e.g., http://www.company.com) that is used in the URL for the provisioning server. This is done in the Interface table:
  - ♦ ini File:

    ```
    [ InterfaceTable ]
    FORMAT InterfaceTable_Index =
    InterfaceTable_ApplicationTypes,
    InterfaceTable_InterfaceMode, InterfaceTable_IPAddress,
    InterfaceTable_PrefixLength, InterfaceTable_Gateway,
    InterfaceTable_VlanID, InterfaceTable_InterfaceName,
    InterfaceTable_PrimaryDNSServerIPAddress,
    InterfaceTable_SecondaryDNSServerIPAddress,
    InterfaceTable_UnderlyingDevice;
    InterfaceTable 0 = 6, 10, 10.15.7.95, 16, 10.15.0.1, 1,
    "Voice", 80.179.52.100, 0.0.0.0, "vlan 1";
    [ \InterfaceTable ]
    ```

4. Power down and then power up the device.

33 Restoring Factory Defaults

You can restore the device's configuration to factory defaults using one of the following methods:

- CLI (see 'Restoring Defaults using CLI' on page)
- Hardware reset pinhole button (see Restoring Defaults using Hardware Reset Button on page)
- Loading an empty *ini* file (see 'Restoring Defaults using an ini File' on page)

33.1 Restoring Defaults using CLI

The device can be restored to factory defaults using CLI, as described in the procedure below.

➤ **To restore factory defaults using CLI:**

1. Access the CLI:
    a. Connect the RS-232 serial port of the device to the communication port on your computer. For serial cabling, refer to the *Hardware Installation Manual*.
    b. Establish serial communication with the device using a serial communication program (such as HyperTerminal™) with the following communication port settings:
        ♦ **Baud Rate:** 9,600 bps for MP-11x; 115,200 bps for MP-124
        ♦ **Data Bits:** 8
        ♦ **Parity:** None
        ♦ **Stop Bits:** 1
        ♦ **Flow Control:** None
2. At the CLI prompt, type the following command to access the configuration mode, and then press Enter:
    ```
    # conf
    ```
3. At the prompt, type the following command to reset the device to default settings, and then press Enter:
    ```
    # RestoreFactorySettings
    ```

33.2 Restoring Defaults using Hardware Reset Button

The device's hardware reset pinhole button can be used to reset the device to default settings.

➤ **To restore default settings using the hardware reset pinhole button:**

- To restore MP-124 to factory default settings:
    a. Disconnect the Ethernet cable from the device.
    b. With a paper clip or any other similar pointed object, press and hold down the reset pinhole button for at least 12 seconds, but no more than 25 seconds.
- To restore MP-11x to factory default settings:
    a. Disconnect the Ethernet cable from the device.
    b. With a paper clip or any other similar pointed object, press and hold down the reset pinhole button for about six seconds; the Fail LED turns red and the device restores to factory default settings.
    c. When the Fail LED turns off, reconnect the Ethernet cable to the device.

33.3   Restoring Defaults using an ini File

You can restore the device to factory default settings by loading an empty *ini* file to the device. This is done using the Web interface's Configuration File page (see 'Backing Up and Loading Configuration File' on page 384). If the *ini* file does include content (e.g., parameters), ensure that they are on lines beginning with comment signs (i.e., semicolons ";") so that the device ignores them.

> **Note:**   The only settings that are not restored to default are the management (OAMP) IP address and the Web interface's login user name and password.

Part VIII

Status, Performance Monitoring and Reporting

34   System Status

This section describes how to view various system statuses.

34.1   Viewing Device Information

The Device Information page displays various hardware and software information of the device. This page also lists any Auxiliary files that have been installed on the device and allows you to remove them.

➢   **To access the Device Information page:**

■   Open the Device Information page (**Status & Diagnostics** tab > **System Status** menu > **Device Information**).

▼ General Settings	
MAC Address:	00908f084f99
Serial Number:	544665
Board Type:	MP-118 FXS_FXO
Device Up Time:	10d:3h:14m:47s:52th
Device Administrative State:	Unlocked
Device Operational State:	Enabled
Flash Size [Mbytes]:	8
RAM Size [Mbytes]:	32
CPU Speed [MHz]:	40

▼ Versions	
Version ID:	6.60A.011.013
DSP Type:	0
DSP Software Version:	66003
DSP Software Name:	204IM
Flash Version:	199

▼ Loaded Files	
Loaded Call Progress Tones:	Default Progress Tones
Loaded Coder Table :	Default CODERTABLE

➢   **To delete a loaded file:**

■   Click the **Delete** button corresponding to the file that you want to delete. Deleting a file takes effect only after device reset (see 'Resetting the Device' on page 359).

34.2   Viewing Ethernet Port Information

The Ethernet Port Information page displays read-only information on the Ethernet port connections.

> **Note:** The Ethernet Port Information page can also be accessed from the Home page (see 'Viewing the Home Page' on page 63).

➢   **To view Ethernet port information:**

■   Open the Ethernet Port Information page (**Status & Diagnostics** tab > **System**

Status menu > **Ethernet Port Information**).
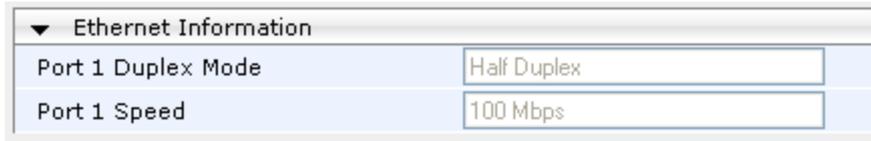
Table 34-1: Ethernet Port Information Parameters

Parameter	Description
Port Duplex Mode	Displays whether the port is in half or duplex mode.
Port Speed	Displays the speed (in Mbps) of the Ethernet port.

35   Carrier-Grade Alarms

This section describes how to view the following types of alarms:

■   Active alarms - see 'Viewing Active Alarms' on page 409

■   Alarm history - see 'Viewing Alarm History' on page 409

35.1   Viewing Active Alarms

The Active Alarms page displays a list of currently active alarms. You can also access this page from the Home page (see 'Viewing the Home Page' on page 63).

> **Note:**
>
> • The alarms in the table are deleted upon a device reset.
>
> • To configure the maximum number of active alarms that can be displayed in the table, see the ini file parameter, ActiveAlarmTableMaxSize.
>
> • For more information on SNMP alarms, refer to the *SNMP Reference Guide* document.

➢   **To view the list of active alarms:**

■   Open the Active Alarms page (**Status & Diagnostics** tab > **System Status** menu > **Carrier-Grade Alarms** > **Active Alarms**)**.**

Sequential number	Severity	Source	Description	Date
3	Major	Board#1	Controller failure alarm BusyOut Line 6 Link failure	16.7.2012 , 17:44:52

For each alarm, the following information is provided:

■   **Severity:** severity level of the alarm:

• Critical (red)

• Major (orange)

• Minor (yellow)

■   **Source:** unit from which the alarm was raised

■   **Description:** brief explanation of the alarm

■   **Date:** date and time that the alarm was generated

You can view the next 20 alarms (if exist), by clicking the **Go to page** button.

35.2   Viewing Alarm History

The Alarms History page displays a list of alarms that have been raised and traps that have been cleared.

➢   **To view the list of history alarms:**

■   Open the Alarms History page (**Status & Diagnostics** tab > **System Status** menu > **Carrier-Grade Alarms** > **Alarms History**)**.**

Sequential number	Severity	Source	Description	Date
1	Major	Board#1	Controller failure alarm Proxy Set 0: Proxy lost. looking for another proxy	6.1.2010 , 14:1:26
2	cleared	Board#1	Alarm cleared: Controller failure alarm Proxy Set 0: Proxy lost. looking for another proxy	6.1.2010 , 14:1:26
3	Major	Board#1	Controller failure alarm Proxy Set ID 0	6.1.2010 , 14:1:26
4	Major	Board#1/WanLink#1	WAN link alarm. FE interface 1 is down.	6.1.2010 , 14:1:29
5	Minor	Board#1/EthernetLink#2	Ethernet link alarm. LAN port number 2 is down.	6.1.2010 , 14:1:29
6	Major	Board#1	NTP server alarm. No connection to NTP server.	6.1.2010 , 14:11:14

For each alarm, the following information is provided:

■ **Severity:** severity level of the alarm:

- Critical (red)
- Major (range)
- Minor (yellow)
- Cleared (green)

■ **Source:** unit from which the alarm was raised

■ **Description:** brief explanation of the alarm

■ **Date:** date and time that the alarm was generated

You can view the next 20 alarms (if exist), by clicking the **Go to page** button.

➢ **To delete all the alarms in the table:**

1. Click the **Delete History Table** button; a confirmation message box appears.
2. Click **OK** to confirm.

36   VoIP Status

This section describes how to view VoIP status and statistics.

36.1   Viewing Analog Port Information

The Home page allows you to view detailed information on selected FXS and FXO analog ports such as RTP/RTCP and voice settings.

➢   **To view information on an analog port:**

1.   Open the Home page.
2.   On the graphical display of the device, click the required analog port; a shortcut menu appears.
3.   From the shortcut menu, choose **Port Settings**; the Basic Channel Information page appears with the **Basic** tab selected (displayed in green):

Figure 36-1: Basic Channel Information Page

◆ SIP ◆ Basic ◆ RTP/RTCP ◆ Voice Settings	
Channel Identifier:	55
Status:	Inactive
Call ID:	0
Endpoint ID:	Not Available
Call Duration [sec]:	0
Call Type:	Voice
Call Destination:	10.13.4.12
Coder:	Transparent

4.   To view additional channel information, click the required tab - **SIP**, **RTP/RTCP**, and **Voice Settings**.

36.2   Viewing Active IP Interfaces

The IP Interface Status page displays the device's active IP interfaces that are listed in the Multiple Interface Table page (see 'Configuring IP Network Interfaces' on page 116).

➢   **To view the active IP network interfaces:**

■   Open the IP Interface Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **IP Interface Status**).

Index	Application Type	Address Type	Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
NA	O+M+C	IPv4	IPv4 Manual	10.13.4.13	16	10.13.0.1	0	O+M+C

VLAN Mode	Disabled
Native VLAN ID	1

36.3 Viewing Performance Statistics

The Basic Statistics page provides read-only, device performance statistics. This page is refreshed every 60 seconds. The duration that the currently displayed statistics has been collected is displayed above the statistics table.

➢ **To view performance statistics:**

■ Open the Basic Statistics page (**Status & Diagnostics** tab > **VoIP Status** menu > **Performance Statistics**)**.**

Figure 36-2: Basic Statistics Page

(Statistics for 759525 seconds)	
Active TDM channels	0
Active DSP resources	0
Active analog channels	0
Active G.711 channels	0
Average voice delay (ms)	5
Average voice jitter (ms)	11
Total RTP packets TX	4250
Total RTP packets RX	4241
Total call attempts	6

The duration that the displayed statistics were collected is displayed in seconds above the table. To reset the performance statistics to zero, click the **Reset Statistics** button.

36.4 Viewing Call Counters

The IP to Tel Calls Count page and Tel to IP Calls Count page provide you with statistical information on incoming (IP-to-Tel) and outgoing (Tel-to-IP) calls. The statistical information is updated according to the release reason that is received after a call is terminated (during the same time as the end-of-call Call Detail Record or CDR message is sent). The release reason can be viewed in the 'Termination Reason' field in the CDR message.

You can reset the statistical data displayed on the page (i.e., refresh the display), by clicking the **Reset Counters** button located below the table.

➢ **To view IP-to-Tel and Tel-to-IP call counters:**

■ Open the Call Counters page that you want to view (**Status & Diagnostics** tab > **VoIP Status** menu > **IP to Tel Calls Count** or **Tel to IP Calls Count**); the figure below shows the IP to Tel Calls Count page.

Figure 36-3: Calls Count Page

Number of Attempted Calls	19
Number of Established Calls	14
Percentage of Successful Calls(ASR)	73.684211
Number of Calls Terminated due to a Busy Line	2
Number of Calls Terminated due to No Answer	0
Number of Calls Terminated due to Forward	0
Number of Failed Calls due to No Route	0
Number of Failed Calls due to No Matched Capabilities	0
Number of Failed Calls due to No Resources	0
Number of Failed Calls due to Other Failures	0
Average Call Duration(ACD)[sec]	25
Attempted Fax Calls Counter	0
Successful Fax Calls Counter	0

The fields in this page are described in the following table:

Table 36-1: Call Counters Description

Counter	Description
Number of Attempted Calls	Indicates the number of attempted calls. It is composed of established and failed calls. The number of established calls is represented by the 'Number of Established Calls' counter. The number of failed calls is represented by the failed-call counters. Only one of the established / failed call counters is incremented every time.
Number of Established Calls	Indicates the number of established calls. It is incremented as a result of one of the following release reasons if the duration of the call is greater than zero: ▪ GWAPP_REASON_NOT_RELEVANT (0) ▪ GWAPP_NORMAL_CALL_CLEAR (16) ▪ GWAPP_NORMAL_UNSPECIFIED (31) And the internal reasons: ▪ RELEASE_BECAUSE_UNKNOWN_REASON ▪ RELEASE_BECAUSE_REMOTE_CANCEL_CALL ▪ RELEASE_BECAUSE_MANUAL_DISC ▪ RELEASE_BECAUSE_SILENCE_DISC ▪ RELEASE_BECAUSE_DISCONNECT_CODE **Note:** When the duration of the call is zero, the release reason GWAPP_NORMAL_CALL_CLEAR increments the 'Number of Failed Calls due to No Answer' counter. The rest of the release reasons increment the 'Number of Failed Calls due to Other Failures' counter.
Percentage of Successful Calls (ASR)	The percentage of established calls from attempted calls.
Number of Calls Terminated due to a Busy Line	Indicates the number of calls that failed as a result of a busy line. It is incremented as a result of the following release reason: GWAPP_USER_BUSY (17)
Number of Calls Terminated due to No Answer	Indicates the number of calls that weren't answered. It's incremented as a result of one of the following release reasons: ▪ GWAPP_NO_USER_RESPONDING (18) ▪ GWAPP_NO_ANSWER_FROM_USER_ALERTED (19) ▪ GWAPP_NORMAL_CALL_CLEAR (16) (when the call duration is zero)
Number of Calls Terminated due to Forward	Indicates the number of calls that were terminated due to a call forward. The counter is incremented as a result of the following release reason: RELEASE_BECAUSE_FORWARD
Number of Failed Calls due to No Route	Indicates the number of calls whose destinations weren't found. It is incremented as a result of one of the following release reasons: ▪ GWAPP_UNASSIGNED_NUMBER (1) ▪ GWAPP_NO_ROUTE_TO_DESTINATION (3)
Number of Failed Calls due to No Matched Capabilities	Indicates the number of calls that failed due to mismatched device capabilities. It is incremented as a result of an internal identification of capability mismatch. This mismatch is reflected to CDR via the value of the parameter DefaultReleaseReason (default is GWAPP_NO_ROUTE_TO_DESTINATION (3)) or by the GWAPP_SERVICE_NOT_IMPLEMENTED_UNSPECIFIED (79) reason.

Counter	Description
Number of Failed Calls due to No Resources	Indicates the number of calls that failed due to unavailable resources or a device lock. The counter is incremented as a result of one of the following release reasons: ▪ GWAPP_RESOURCE_UNAVAILABLE_UNSPECIFIED ▪ RELEASE_BECAUSE_GW_LOCKED
Number of Failed Calls due to Other Failures	This counter is incremented as a result of calls that failed due to reasons not covered by the other counters.
Average Call Duration (ACD) [sec]	The average call duration (ACD) in seconds of established calls. The ACD value is refreshed every 15 minutes and therefore, this value reflects the average duration of all established calls made within a 15 minute period.
Attempted Fax Calls Counter	Indicates the number of attempted fax calls.
Successful Fax Calls Counter	Indicates the number of successful fax calls.

36.5 Viewing Registered Users

The SAS/SBC Registered Users page displays a list of registered SAS users recorded in the device's database.

➢ **To view registered SAS users:**

■ Open the Registration Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **Registered Users**).

Figure 36-4: SAS/SBC Registered Users Page

Address Of Record	Contact
1000@10.8.5.71	<sip:1000@10.8.5.71:5060>;expires=180; Active status: 1
1001@10.8.5.71	<sip:1001@10.8.5.71:5060>;expires=180; Active status: 1
1100@10.8.5.71	<sip:1100@10.8.5.71:5060>;expires=180; Active status: 1
1101@10.8.5.71	<sip:1101@10.8.5.71:5060>;expires=180; Active status: 1
2000@10.8.5.72	<sip:2000@10.8.5.72:5060>;expires=180; Active status: 1

Table 36-2: SAS Registered Users Parameters

Column Name	Description
Address of Record	An address-of-record (AOR) is a SIP or SIPS URI that points to a domain with a location service that can map the URI to another URI (Contact) where the user might be available.
Contact	SIP URI that can be used to contact that specific instance of the User Agent for subsequent requests.

36.6   Viewing Registration Status

The Registration Status page displays whether the device as a whole, its endpoints (FXS / FXO), and SIP Accounts are registered to a SIP Registrar/Proxy server.

➢ **To view the registration status:**

■ Open the Registration Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **Registration Status**).

Figure 36-5: Registration Status

Registered Per Gateway		NO

▼ Ports Registration Status

Gateway Port	Status
Port 1 FXS	NOT REGISTERED
Port 2 FXS	NOT REGISTERED
Port 3 FXS	NOT REGISTERED
Port 4 FXS	NOT REGISTERED
Port 5 FXO	NOT REGISTERED
Port 6 FXO	NOT REGISTERED
Port 7 FXO	NOT REGISTERED
Port 8 FXO	NOT REGISTERED

▼ Accounts Registration Status

Index	Group Type	Group Name	Status

■ **Registered Per Gateway:**
- "YES" = Registration is per device
- "NO"= Registration is not per device

■ Ports Registration Status:
- "REGISTERED" = channel is registered
- "NOT REGISTERED" = channel not registered

■ **Accounts Registration Status:** registration status based on the Accounts table (configured in 'Configuring Account Table' on page 205):
- **Group Type:** type of served group - Hunt Group or IP Group
- **Group Name:** name of the served group, if applicable
- **Status:** indicates whether or not the group is registered ("Registered" or "Unregistered")

> **Note:**   The registration mode (i.e., per device, endpoint, account. or no registration) is configured in the Hunt Group Settings table (see 'Configuring Hunt Group Settings' on page 229) or using the TrunkGroupSettings *ini* file parameter.

36.7 Viewing Call Routing Status

The Call Routing Status page provides you with information on the current routing method used by the device. This information includes the IP address and FQDN (if used) of the Proxy server with which the device currently operates.

➢ **To view call routing status:**

■ Open the Call Routing Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **Call Routing Status**).

Figure 36-6: Call Routing Status Page

Call-Routing Method		Routing Table	
Active Proxy Sets Status			
ID	IP Address		State
0	Not Used (--)		--
1	10.8.230.64 (10.8.230.64)		OK
2	10.9.244.80 (10.9.244.80)		OK
3	10.10.244.80 (10.10.244.80)		OK
4	10.11.244.80 (10.11.244.80)		OK
5	10.12.244.80 (10.12.244.80)		OK
6	Not Used (--)		--
7	Not Used (--)		--
8	Not Used (--)		--
9	10.8.244.81 (10.8.244.81)		OK
10	Not Used (--)		--
11	Not Used (--)		--
12	Not Used (--)		

Table 36-3: Call Routing Status Parameters

Parameter	Description
Call-Routing Method	▪ Proxy/GK = Proxy server is used to route calls. ▪ Routing Table = The Tel to IP Routing table is used to route calls.
IP Address	▪ Not Used = Proxy server isn't defined. ▪ IP address and FQDN (if exists) of the Proxy server with which the device currently operates.
State	▪ N/A = Proxy server isn't defined. ▪ OK = Communication with the Proxy server is in order. ▪ Fail = No response from any of the defined Proxies.

36.8   Viewing IP Connectivity

The IP Connectivity page displays on-line, read-only network diagnostic connectivity information on all destination IP addresses configured in the Tel to IP Routing page (see 'Configuring Tel to IP Routing' on page 250).

Note: The information in columns 'Quality Status' and 'Quality Info' (per IP address) is reset if two minutes elapse without a call to that destination.

➢   **To view IP connectivity information:**

1.   In the Routing General Parameters page, set the 'Enable Alt Routing Tel to IP' parameter (AltRoutingTel2IPMode) to **Enable** or **Status Only** (see 'Configuring General Routing Parameters' on page 249).

2.   Open the IP Connectivity page (**Status & Diagnostics** tab > **VoIP Status** menu > **IP Connectivity**).

Figure 36-7: IP Connectivity Page

	IP Address	Host Name	Connectivity Method	Connectivity Status	Quality Status	Quality Info	DNS Status
1	Unused	---	Ping	---	---	---	---
2	Unused	---	Ping	---	---	---	---
3	Unused	---	Ping	---	---	---	---
4	Unused	---	Ping	---	---	---	---
5	Unused	---	Ping	---	---	---	---
6	Unused	---	Ping	---	---	---	---
7	Unused	---	Ping	---	---	---	---
8	Unused	---	Ping	---	---	---	---
9	Unused	---	Ping	---	---	---	---
10	Unused	---	Ping	---	---	---	---
11	Unused	---	Ping	---	---	---	---
12	Unused	---	Ping	---	---	---	---

Table 36-4: IP Connectivity Parameters

Column Name	Description
IP Address	The IP address can be one of the following: ▪ IP address defined as the destination IP address in the Tel to IP Routing. ▪ IP address resolved from the host name defined as the destination IP address in the Tel to IP Routing.
Host Name	Host name (or IP address) as defined in the Tel to IP Routing.
Connectivity Method	The method according to which the destination IP address is queried periodically (ICMP ping or SIP OPTIONS request).

Column Name	Description
Connectivity Status	The status of the IP address' connectivity according to the method in the 'Connectivity Method' field. ▪ OK = Remote side responds to periodic connectivity queries. ▪ Lost = Remote side didn't respond for a short period. ▪ Fail = Remote side doesn't respond. ▪ Init = Connectivity queries not started (e.g., IP address not resolved). ▪ Disable = The connectivity option is disabled, i.e., parameter 'Alt Routing Tel to IP Mode' (AltRoutingTel2IPMode *ini*) is set to 'None' or 'QoS'.
Quality Status	Determines the QoS (according to packet loss and delay) of the IP address. ▪ Unknown = Recent quality information isn't available. ▪ OK ▪ Poor **Notes:** ▪ This parameter is applicable only if the parameter 'Alt Routing Tel to IP Mode' is set to 'QoS' or 'Both' (AltRoutingTel2IPMode = 2 or 3). ▪ This parameter is reset if no QoS information is received for 2 minutes.
Quality Info.	Displays QoS information: delay and packet loss, calculated according to previous calls. **Notes:** ▪ This parameter is applicable only if the parameter 'Alt Routing Tel to IP Mode' is set to 'QoS' or 'Both' (AltRoutingTel2IPMode = 2 or 3). ▪ This parameter is reset if no QoS information is received for 2 minutes.
DNS Status	DNS status can be one of the following: ▪ DNS Disable ▪ DNS Resolved ▪ DNS Unresolved

37   Reporting Information to External Party

This section describes features for reporting various information to an external party.

37.1   RTP Control Protocol Extended Reports (RTCP XR)

RTP Control Protocol Extended Reports (RTCP XR) is a VoIP management control that defines a set of metrics containing information for assessing VoIP call quality and for diagnosing problems. RTCP XR (RFC 3611) extends the RTCP reports defined in RFC 3550 by providing additional VoIP metrics. RTCP XR information publishing is implemented in the device according to <draft-johnston-sipping-rtcp-summary-07>. This draft defines how a SIP User Agent (UA) publishes the detailed information to a defined collector. RTCP XR measures VoIP call quality such as packet loss, delay, signal / noise / echo levels, estimated R-factor, and mean opinion score (MOS). RTCP XR measures these parameters using metrics as listed in the table below.

> **Note:** RTCP XR is a customer ordered feature and thus, must be included in the Software License Key installed on the device.

RTCP XR messages containing key call-quality-related metrics are exchanged periodically (user-defined) between the device and the SIP UA. This allows an analyzer to monitor these metrics midstream, or a device to retrieve them using SNMP.

You can configure the device to send RTCP XR to an Event State Compositor (ESC) server or for Gateway calls, to a specific IP Group (using the PublicationIPGroupID ini file parameter). If you configure it to send RTCP XR to an IP Group, the RTCP XR is sent to the address configured for the Proxy Set associated with the IP Group.

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 OVOC server when sent to the ESC server. When sent to an IP Group, the Request-URI value contains the name of the IP Group as configured by the 'IP Group Name' parameter (IPGroup_Name).
- Event: "vq-rtcpxr"
- Content-Type: "application/vq-rtcpxr"

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.
- (Gateway Application Only) End of a media segment. A media segment is a change in media, for example, when the coder is changed or when the caller toggles between two called parties (using call hold/retrieve). The RTCP XR sent at the end of a media segment contains information only of that segment. For call hold, the device sends RTCP XR each time the call is placed on hold and each time it is retrieved. In addition, the Start timestamp in the RTCP XR indicates the start of the media segment; the End timestamp indicates the time of the last sent periodic RTCP XR (typically, up to 5 seconds before reported segment ends).

Table 37-1: RTCP XR Published VoIP Metrics

Group	Metric Name
General	Start Timestamp

Group	Metric Name
	Stop Timestamp
	Call-ID
	Local Address (IP, Port & SSRC)
	Remote Address (IP, Port & SSRC)
Session Description	Payload Type
	Payload Description
	Sample Rate
	Frame Duration
	Frame Octets
	Frames per Packets
	Packet Loss Concealment
	Silence Suppression State
Jitter Buffer	Jitter Buffer Adaptive
	Jitter Buffer Rate
	Jitter Buffer Nominal
	Jitter Buffer Max
	Jitter Buffer Abs Max
Packet Loss	Network Packet Loss Rate
	Jitter Buffer Discard Rate
Burst Gap Loss	Burst Loss Density
	Burst Duration
	Gap Loss Density
	Gap Duration
	Minimum Gap Threshold
Delay	Round Trip Delay
	End System Delay
	One Way Delay
	Interarrival Jitter
	Min Absolute Jitter
	Signal
	Signal Level
	Noise Level
	Residual Echo Return Noise
Quality Estimates	Listening Quality R
	RLQ Est. Algorithm
	Conversational Quality R

Group	Metric Name
	RCQ Est. Algorithm
	External R In
	Ext. R In Est. Algorithm
	External R Out
	Ext. R Out Est. Algorithm
	MOS-LQ
	MOS-LQ Est. Algorithm
	MOS-CQ
	MOS-CQ Est. Algorithm
	QoE Est. Algorithm

Below shows an example of a SIP PUBLISH message sent with RTCP XR and QoE information:

```
PUBLISH sip: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,SUB
SCRIBE,UPDATE
Event: vq-rtcpxr
Expires: 3600
User-Agent: device/<swver>
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:
LocalMAC: 00:90:8f:57:d9:71
LocalMetrics:
Timestamps: START=2015-12-16T20:09:45Z STOP=2015-12-16T20:09:52Z
SessionDesc: PT=8 PD=PCMA SR=8000 FD=20 PLC=3 SSUP=Off
JitterBuffer: JBA=3 JBR=0 JBN=7 JBM=10 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
Signal: SL=-34 NL=-67 RERL=17
QualityEst: RLQ=93 MOSLQ=4.1
MOSCQ=4.10
RemoteMetrics:
Timestamps: START=2015-12-16T20:09:45Z STOP=2015-12-16T20:09:52Z
JitterBuffer: JBA=3 JBR=0 JBN=0 JBM=0 JBX=300
PacketLoss: NLR=0.00 JDR=0.00
BurstGapLoss: BLD=0.00 BD=0 GLD=0.00 GD=0 GMIN=16
Delay: RTD=65535 ESD=0
QualityEst:
DialogID: 20328634741612201520943@172.17.116.201;to-
tag=1c1690611502;from-tag=1c2032864069
```

➢ **To configure RTCP XR:**

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

Figure 37-1: RTCP XR Parameters in RTP/RTCP Settings Page

RTCP XR Settings	
Burst Threshold	-1
Delay Threshold	-1
R-Value Delay Threshold	-1
Enable RTCP XR	CE_VQMON_DISABLE
Minimum Gap Size	16
RTCP XR Report Mode	Disable
RTCP XR Packet Interval	0
Disable RTCP XR Interval Randomization	Disable
RTCP XR Collection Server	
RTCP XR Collection Server Transport Type	Not Configured

2. Configure the RTCP XR parameters, as required:

- 'Enable RTCP XR' (*VQMonEnable*) - enables voice quality monitoring and RTCP XR.

- 'Minimum Gap Size' (*VQMonGMin*) - defines the voice quality monitoring - minimum gap size (number of frames).

- '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.

- 'RTCP XR Report Mode' (*RTCPXRReportMode*) - determines whether RTCP XR reports are sent to the ESC and defines the interval in which they are sent.

- '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.

- 'RTCP XR Collection Server' (*RTCPXREscIP*) - defines the IP address of the Event State Compositor (ESC). Alternatively, if you want to send the RTCP XR to a specific IP Group, use the PublicationIPGroupID ini file parameter.

- 'RTCP XR Collection Server Transport Type' (*RTCPXRESCTransportType*) - determines the transport layer for outgoing SIP dialogs initiated by the device to the RTCP XR Collection Server.

3. Click **Submit**.

4. Reset the device for the settings to take effect.

37.2 Generating Call Detail Records

The Call Detail Record (CDR) contains vital statistic information on calls made from the device. The device can be configured to generate and report CDRs for various stages of the call, including SIP messages and/or media. You can configure when CDRs for a call are generated, for example, only at the end of the call or only at the start and end of the call. Once generated, the device sends the CDRs to a user-defined Syslog server.

The CDR Syslog message complies with RFC 3161 and is identified by Facility 17 (local1) and Severity 6 (Informational).

For CDR in RADIUS format, see 'RADIUS Accounting CDR Attributes' on page 431.

37.2.1 Configuring CDR Reporting

The procedure below describes how to configure CDR reporting.

➢ **To configure CDR reporting:**

1. Enable the Syslog feature for sending log messages generated ty the device to a collecting log message server. For more information, see 'Configuring Syslog' on page 444.

2. Open the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**). The CDR parameters appear under the 'CDR and Debug' group, as shown below:

Figure 37-2: CDR Parameters in Advanced Parameters Page

CDR and Debug	
CDR Server IP Address	10.8.6.55
CDR Report Level	Start & End & Connect Call
Media CDR Report Level	End Media

3. Configure the parameters as required. For a description of the parameters, see 'Syslog, CDR and Debug Parameters' on page 494.

4. Click **Submit**.

> **Note:** If the CDR server IP address is not configured, the CDRs are sent to the Syslog server, configured in 'Configuring Syslog' on page 444.

37.2.2   CDR Field Description

This section describes the CDR fields that are generated by the device.

37.2.2.1 CDR Fields for Gateway/IP-to-IP Application

The CDR fields for the Gateway / IP-to-IP application are listed in the table below.

Table 37-2: CDR Fields for Gateway/IP-to-IP Application

Field Name	Description
GWReportType	Report type: ▪ CALL_START ▪ CALL_CONNECT ▪ CALL_END
Cid	Port number
SessionId	SIP session identifier
Trunk	Physical trunk number **Note:** This field is applicable only to the Gateway application.
BChan	Selected B-channel (set to '0', as not applicable) **Note:** This field is applicable only to the Gateway application.
ConId	SIP conference ID **Note:** This field is applicable only to the Gateway application.
TG	Trunk Group ID **Note:** This field is applicable only to the Gateway application.
EPTyp	Endpoint type: ▪ FXO ▪ FXS ▪ EANDM ▪ ISDN ▪ CAS ▪ DAA ▪ IPMEDIA ▪ NETANN ▪ STREAMING ▪ TRANSPARENT ▪ MSCML ▪ VXML ▪ IP2IP
Orig	Call originator: ▪ LCL (Tel side) ▪ RMT (IP side)
SourceIp	Source IP address
DestIp	Destination IP address

Field Name	Description
TON	Source phone number type **Note:** This field is applicable only to the Gateway application.
NPI	Source phone number plan **Note:** This field is applicable only to the Gateway application.
SrcPhoneNum	Source phone number
SrcNumBeforeMap	Source number before manipulation
TON	Destination phone number type **Note:** This field is applicable only to the Gateway application.
NPI	Destination phone number plan **Note:** This field is applicable only to the Gateway application.
DstPhoneNum	Destination phone number
DstNumBeforeMap	Destination number before manipulation
Durat	Call duration
Coder	Selected coder
Intrv	Packet interval
RtpIp	RTP IP address
Port	Remote RTP port
TrmSd	Initiator of call release (IP, Tel, or Unknown)
TrmReason	SIP call termination reason (see 'Release Reasons in CDR' on page 428)
Fax	Fax transaction during call
InPackets	Number of incoming packets
OutPackets	Number of outgoing packets
PackLoss	Local packet loss
RemotePackLoss	Number of outgoing lost packets
SIPCallId	Unique SIP call ID
SetupTime	Call setup time
ConnectTime	Call connect time
ReleaseTime	Call release time
RTPdelay	RTP delay
RTPjitter	RTP jitter
RTPssrc	Local RTP SSRC
RemoteRTPssrc	Remote RTP SSRC
RedirectReason	Redirect reason
TON	Redirection phone number type **Note:** This field is applicable only to the Gateway application.
NPI	Redirection phone number plan **Note:** This field is applicable only to the Gateway application.

Field Name	Description
RedirectPhonNum	Redirection phone number
MeteringPulses	Number of generated metering pulses **Note:** This field is applicable only to the Gateway application.
SrcHost	Source host name
SrcHostBeforeMap	Source host name before manipulation
DstHost	Destination host name
DstHostBeforeMap	Destination host name before manipulation
IPG	IP Group description
LocalRtpIp	Remote RTP IP address
LocalRtpPort	Local RTP port
Amount	0-999999 Data is stored per call and sent in the syslog as follows: ▪ currency-type: amount multiplier for currency charge (euro or usd) ▪ recorded-units: for unit charge (1-999999)
Mult	0,001-1000 (in steps of 10) (See explanation above.)

TrmReasonCategory	Termination reason category:
	▪ Calls with duration 0 (i.e., not connected):
	  ✓ **NO_ANSWER -** GWAPP_NORMAL_CALL_CLEAR, GWAPP_NO_USER_RESPONDING, GWAPP_NO_ANSWER_FROM_USER_ALERTED
	  ✓ **BUSY -** GWAPP_USER_BUSY
	  ✓ **NO_RESOURCES -** GWAPP_RESOUUCE_UNAVAILABLE_UNSPECIFIED, RELEASE_BECAUSE_NO_CONFERENCE_RESOURCES_LEFT, RESOURCE_BECAUSE_NO_TRANSCODING_RESOURCES_LEFT, RELEASE_BECAUSE_GW_LOCKED
	  ✓ **NO_MATCH -** RELEASE_BECAUSE_UNMATCHED_CAPABILITIES
	  ✓ **FORWARDED -** RELEASE_BECAUSE_FORWARD
	  ✓ **GENERAL_FAILED -** any other reason
	▪ Calls with duration:
	  ✓ **NORMAL_CALL_CLEAR -** GWAPP_NORMAL_CALL_CLEAR
	  ✓ **ABNORMALLY_TERMINATED -** Anything else
	▪ **N/A -** Reasons not belonging to above categories

RedirectNumBeforeMap	Redirect number before manipulation
SrdId	SRD ID name
SIPInterfaceId	SIP interface ID
ProxySetId	Proxy Set ID
IpProfileId	IP Profile ID name

Field Name	Description
MediaRealmId	Media Realm name
SigTransportType	SIP signaling transport type (UDP, TCP, or TLS)
TxRTPIPDiffServ	Media IP DiffServ
TxSigIPDiffServ	Signaling IP DiffServ
LocalRFactor	Local R-factor
RemoteRFactor	Remote R-factor
LocalMosCQ	Local MOS for conversation quality
RemoteMosCQ	Remote MOS for conversation quality
SigSourcePort	SIP source port
SigDestPort	SIP destination port
MediaType	Media type - audio, video, or text
SIPTrmReason	SIP call termination reason (BYE, CANCEL, or SIP error codes, e.g., 404)
SipTermDesc	Description of SIP termination reason: ▪ SIP Reason header, if exists, for example: SIP ;cause=200 ;text="Call completed elsewhere". ▪ If no SIP Reason header exists, the description is taken from the reason text, if exists, of the SIP response code, for example: "417 Unknown Resource-Priority". ▪ If no reason text exists in the SIP response code, the description is taken from an internal SIP response mapping mechanism. For example, if the device receives a SIP response "422", it sends in the CDR "422 Session Interval Too Small method" as the description.
PstnTermReason	Q.850 protocol termination reason (0-127).
LatchedRtpIp	Remote IP address of the incoming RTP stream that the device "latched" on to as a result of the RTP latching mechanism for NAT traversal.
LatchedRtpPort	Remote RTP port of the incoming RTP stream that the device "latched" on to as a result of the RTP latching mechanism for NAT traversal.

37.2.2.2 Release Reasons in CDR

The possible reasons for call termination for the Gateway / IP-to-IP application which is represented in the CDR field **TrmReason** are listed below:

- "REASON N/A"
- "RELEASE_BECAUSE_NORMAL_CALL_DROP"
- "RELEASE_BECAUSE_DESTINATION_UNREACHABLE"
- "RELEASE_BECAUSE_DESTINATION_BUSY"
- "RELEASE_BECAUSE_NOANSWER"
- "RELEASE_BECAUSE_UNKNOWN_REASON"
- "RELEASE_BECAUSE_REMOTE_CANCEL_CALL"
- "RELEASE_BECAUSE_UNMATCHED_CAPABILITIES"
- "RELEASE_BECAUSE_UNMATCHED_CREDENTIALS"
- "RELEASE_BECAUSE_UNABLE_TO_HANDLE_REMOTE_REQUEST"

- ■ "RELEASE_BECAUSE_NO_CONFERENCE_RESOURCES_LEFT"
- ■ "RELEASE_BECAUSE_CONFERENCE_FULL"
- ■ "RELEASE_BECAUSE_VOICE_PROMPT_PLAY_ENDED"
- ■ "RELEASE_BECAUSE_VOICE_PROMPT_NOT_FOUND"
- ■ "RELEASE_BECAUSE_TRUNK_DISCONNECTED"
- ■ "RELEASE_BECAUSE_RSRC_PROBLEM"
- ■ "RELEASE_BECAUSE_MANUAL_DISC"
- ■ "RELEASE_BECAUSE_SILENCE_DISC"
- ■ "RELEASE_BECAUSE_RTP_CONN_BROKEN"
- ■ "RELEASE_BECAUSE_DISCONNECT_CODE"
- ■ "RELEASE_BECAUSE_GW_LOCKED"
- ■ "RELEASE_BECAUSE_NORTEL_XFER_SUCCESS"
- ■ "RELEASE_BECAUSE_FAIL"
- ■ "RELEASE_BECAUSE_FORWARD"
- ■ "RELEASE_BECAUSE_ANONYMOUS_SOURCE"
- ■ "RELEASE_BECAUSE_IP_PROFILE_CALL_LIMIT"
- ■ "GWAPP_UNASSIGNED_NUMBER"
- ■ "GWAPP_NO_ROUTE_TO_TRANSIT_NET"
- ■ "GWAPP_NO_ROUTE_TO_DESTINATION"
- ■ "GWAPP_CHANNEL_UNACCEPTABLE"
- ■ "GWAPP_CALL_AWARDED_AND "
- ■ "GWAPP_PREEMPTION"
- ■ "PREEMPTION_CIRCUIT_RESERVED_FOR_REUSE"
- ■ "GWAPP_NORMAL_CALL_CLEAR"
- ■ "GWAPP_USER_BUSY"
- ■ "GWAPP_NO_USER_RESPONDING"
- ■ "GWAPP_NO_ANSWER_FROM_USER_ALERTED"
- ■ "MFCR2_ACCEPT_CALL"
- ■ "GWAPP_CALL_REJECTED"
- ■ "GWAPP_NUMBER_CHANGED"
- ■ "GWAPP_NON_SELECTED_USER_CLEARING"
- ■ "GWAPP_INVALID_NUMBER_FORMAT"
- ■ "GWAPP_FACILITY_REJECT"
- ■ "GWAPP_RESPONSE_TO_STATUS_ENQUIRY"
- ■ "GWAPP_NORMAL_UNSPECIFIED"
- ■ "GWAPP_CIRCUIT_CONGESTION"
- ■ "GWAPP_USER_CONGESTION"
- ■ "GWAPP_NO_CIRCUIT_AVAILABLE"
- ■ "GWAPP_NETWORK_OUT_OF_ORDER"
- ■ "GWAPP_NETWORK_TEMPORARY_FAILURE"
- ■ "GWAPP_NETWORK_CONGESTION"
- ■ "GWAPP_ACCESS_INFORMATION_DISCARDED"
- ■ "GWAPP_REQUESTED_CIRCUIT_NOT_AVAILABLE"
- ■ "GWAPP_RESOURCE_UNAVAILABLE_UNSPECIFIED"

- "GWAPP_PERM_FR_MODE_CONN_OUT_OF_S"
- "GWAPP_PERM_FR_MODE_CONN_OPERATIONAL"
- "GWAPP_PRECEDENCE_CALL_BLOCKED"
  - "RELEASE_BECAUSE_PREEMPTION_ANALOG_CIRCUIT_RESERVED_FOR_ REUSE"
  - "RELEASE_BECAUSE_PRECEDENCE_CALL_BLOCKED"
- "GWAPP_QUALITY_OF_SERVICE_UNAVAILABLE"
- "GWAPP_REQUESTED_FAC_NOT_SUBSCRIBED"
- "GWAPP_BC_NOT_AUTHORIZED"
- "GWAPP_BC_NOT_PRESENTLY_AVAILABLE"
- "GWAPP_SERVICE_NOT_AVAILABLE"
- "GWAPP_CUG_OUT_CALLS_BARRED"
- "GWAPP_CUG_INC_CALLS_BARRED"
- "GWAPP_ACCES_INFO_SUBS_CLASS_INCONS"
- "GWAPP_BC_NOT_IMPLEMENTED"
- "GWAPP_CHANNEL_TYPE_NOT_IMPLEMENTED"
- "GWAPP_REQUESTED_FAC_NOT_IMPLEMENTED"
- "GWAPP_ONLY_RESTRICTED_INFO_BEARER"
- "GWAPP_SERVICE_NOT_IMPLEMENTED_UNSPECIFIED"
- "GWAPP_INVALID_CALL_REF"
- "GWAPP_IDENTIFIED_CHANNEL_NOT_EXIST"
- "GWAPP_SUSPENDED_CALL_BUT_CALL_ID_NOT_EXIST"
- "GWAPP_CALL_ID_IN_USE"
- "GWAPP_NO_CALL_SUSPENDED"
- "GWAPP_CALL_HAVING_CALL_ID_CLEARED"
- "GWAPP_INCOMPATIBLE_DESTINATION"
- "GWAPP_INVALID_TRANSIT_NETWORK_SELECTION"
- "GWAPP_INVALID_MESSAGE_UNSPECIFIED"
- "GWAPP_NOT_CUG_MEMBER"
- "GWAPP_CUG_NON_EXISTENT"
- "GWAPP_MANDATORY_IE_MISSING"
- "GWAPP_MESSAGE_TYPE_NON_EXISTENT"
- "GWAPP_MESSAGE_STATE_INCONSISTENCY"
- "GWAPP_NON_EXISTENT_IE"
- "GWAPP_INVALID_IE_CONTENT"
- "GWAPP_MESSAGE_NOT_COMPATIBLE"
- "GWAPP_RECOVERY_ON_TIMER_EXPIRY"
- "GWAPP_PROTOCOL_ERROR_UNSPECIFIED"
- "GWAPP_INTERWORKING_UNSPECIFIED"
- "GWAPP_UKNOWN_ERROR"
- "RELEASE_BECAUSE_HELD_TIMEOUT"

37.3   Configuring RADIUS Accounting

The the RADIUS Parameters page allows you to enable RADIUS accounting of SIP calls by a RADIUS accounting server. The device can send the accounting messages to the RADIUS server upon call release, call connection and release, or call setup and release.

> **Notes:**
>
> - For RADIUS accounting settings to take effect, you must save the settings to flash memory with a device reset.
> - For a description of the RADIUS accounting parameters, see 'RADIUS Parameters' on page 512.

➢ **To configure RADIUS accounting:**

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

Figure 37-3: RADIUS Accounting Parameters Page

2. Configure the parameters as required.

3. Click **Submit.**

The table below describes the RADIUS Accounting CDR attributes included in the communication packets transmitted between the device and a RADIUS server.

Table 37-3: Supported RADIUS Accounting CDR Attributes

Attribute Number	Attribute Name	Vendor Specific Attribute (VSA) No.	Purpose	Value Format	Example	AAA
Request Attributes						
1	user-name	-	Account number or calling party number or blank	String up to 15 digits long	5421385747	Start Acc Stop Acc
4	nas-ip-address	-	IP address of the requesting device	Numeric	192.168.14.43	Start Acc Stop Acc
6	service-type	-	Type of service requested	Numeric	1: login	Start Acc Stop Acc
26	h323-incoming-conf-id	1	SIP call identifier	Up to 32 octets	-	Start Acc Stop Acc

Attribute Number	Attribute Name	Vendor Specific Attribute (VSA) No.	Purpose	Value Format	Example	AAA
26	h323-remote-address	23	IP address of the remote gateway	Numeric	-	Stop Acc
26	h323-conf-id	24	H.323/SIP call identifier	Up to 32 octets	-	Start Acc Stop Acc
26	h323-setup-time	25	Setup time in NTP format 1	String	-	Start Acc Stop Acc
26	h323-call-origin	26	The call's originator: Answering (IP) or Originator (PSTN)	String	Answer, Originate etc	Start Acc Stop Acc
26	h323-call-type	27	Protocol type or family used on this leg of the call	String	VoIP	Start Acc Stop Acc
26	h323-connect-time	28	Connect time in NTP format	String	-	Stop Acc
26	h323-disconnect-time	29	Disconnect time in NTP format	String	-	Stop Acc
26	H323-Disconnect-Cause	30	Q.931 disconnect cause code	Numeric	-	Stop Acc
26	h323-gw-id	33	Name of the gateway	String	SIPIDString	Start Acc Stop Acc
26	sip-call-id	34	SIP Call ID	String	abcde@ac.com	Start Acc Stop Acc
26	call-terminator	35	The call's terminator: PSTN-terminated call (Yes); IP-terminated call (No).	String	Yes, No	Stop Acc
30	called-station-id	-	Destination phone number	String	8004567145	Start Acc
31	calling-station-id	-	Calling Party Number (ANI)	String	5135672127	Start Acc Stop Acc
40	acct-status-type	-	Account Request Type (start or stop) **Note:** 'start' isn't supported on the Calling Card application.	Numeric	1: start, 2: stop	Start Acc Stop Acc
41	acct-delay-time	-	No. of seconds tried in sending a particular record	Numeric	5	Start Acc Stop Acc

Attribute Number	Attribute Name	Vendor Specific Attribute (VSA) No.	Purpose	Value Format	Example	AAA
42	acct-input-octets	-	Number of octets received for that call duration	Numeric	-	Stop Acc
43	acct-output-octets	-	Number of octets sent for that call duration	Numeric	-	Stop Acc
44	acct-session-id	-	A unique accounting identifier - match start & stop	String	34832	Start Acc Stop Acc
46	acct-session-time	-	For how many seconds the user received the service	Numeric	-	Stop Acc
47	acct-input-packets	-	Number of packets received during the call	Numeric	-	Stop Acc
48	acct-oputput-packets	-	Number of packets sent during the call	Numeric	-	Stop Acc
61	nas-port-type	-	Physical port type of device on which the call is active	String	0: Asynchronous	Start Acc Stop Acc
Response Attributes						
26	h323-return-code	103	The reason for failing authentication (0 = ok, other number failed)	Numeric	0 Request accepted	Stop Acc
44	acct-session-id	-	A unique accounting identifier – match start & stop	String	-	Stop Acc

Below is an example of RADIUS Accounting, where the non-standard parameters are preceded with brackets:

```
Accounting-Request (361)
user-name = 111
acct-session-id = 1
nas-ip-address = 212.179.22.213
nas-port-type = 0
acct-status-type = 2
acct-input-octets = 4841
acct-output-octets = 8800
acct-session-time = 1
acct-input-packets = 122
acct-output-packets = 220
called-station-id = 201
calling-station-id = 202
// Accounting non-standard parameters:
(4923 33) h323-gw-id =
```

```
(4923 23) h323-remote-address = 212.179.22.214
(4923 1) h323-ivr-out = h323-incoming-conf-id:02102944 600a1899
3fd61009 0e2f3cc5
(4923 30) h323-disconnect-cause = 22 (0x16)
(4923 27) h323-call-type = VOIP
(4923 26) h323-call-origin = Originate
(4923 24) h323-conf-id = 02102944 600a1899 3fd61009 0e2f3cc5
```

37.4 Event Notification using X-Detect Header

The device supports the sending of notifications to a remote party notifying the occurrence (or detection) of certain events on the media stream. Event detection and notifications is performed using the SIP X-Detect message header and only when establishing a SIP dialog.

For supporting some events, certain device configurations need to be performed. The table below lists the supported event types (and subtypes) and the corresponding device configurations, if required:

Table 37-4: Supported X-Detect Event Types

Events Type	Subtype	Required Configuration
CPT	SIT-NC SIT-IC SIT-VC SIT-RO Busy Reorder Ringtone	SITDetectorEnable = 1 UserDefinedToneDetectorEnable = 1 **Note:** Ensure that the CPT file is configured with the required tone type.
FAX	CED	(IsFaxUsed ≠ 0) or (IsFaxUsed = 0, and FaxTransportMode ≠ 0)
	modem	VxxModemTransportType = 3
PTT	voice-start voice-end	EnableDSPIPMDetectors = 1

The device can detect and report the following Special Information Tones (SIT) types from the PSTN:

- SIT-NC (No Circuit found)
- SIT-IC (Operator Intercept)
- SIT-VC (Vacant Circuit - non-registered number)
- SIT-RO (Reorder - System Busy)

There are additional three SIT tones that are detected as one of the above SIT tones:

- The NC* SIT tone is detected as NC
- The RO* SIT tone is detected as RO
- The IO* SIT tone is detected as VC

The device can map these SIT tones to a Q.850 cause and then map them to SIP 5xx/4xx responses, using the parameters SITQ850Cause, SITQ850CauseForNC, SITQ850CauseForIC, SITQ850CauseForVC, and SITQ850CauseForRO.

Table 37-5: Special Information Tones (SITs) Reported by the device

Special Information Tones (SITs) Name	Description	First Tone Frequency Duration		Second Tone Frequency Duration		Third Tone Frequency Duration	
		(Hz)	(ms)	(Hz)	(ms)	(Hz)	(ms)
NC1	No circuit found	985.2	380	1428.5	380	1776.7	380
IC	Operator intercept	913.8	274	1370.6	274	1776.7	380
VC	Vacant circuit (non registered number)	985.2	380	1370.6	274	1776.7	380
RO1	Reorder (system busy)	913.8	274	1428.5	380	1776.7	380
NC*	-	913.8	380	1370.6	380	1776.7	380
RO*	-	985.2	274	1370.6	380	1776.7	380
IO*	-	913.8	380	1428.5	274	1776.7	380

For example:

```
INFO sip:5001@10.33.2.36 SIP/2.0
Via: SIP/2.0/UDP 10.33.45.65;branch=z9hG4bKac2042168670
Max-Forwards: 70
From: <sip:5000@10.33.45.65;user=phone>;tag=1c1915542705
To: <sip:5001@10.33.2.36;user=phone>;tag=WQJNIDDPCOKAPIDSCOTG
Call-ID: AIFHPETLLMVVFWPDXUHD@10.33.2.36
CSeq: 1 INFO
Contact: <sip:2206@10.33.45.65>
Supported: em,timer,replaces,path,resource-priority
Content-Type: application/x-detect
Content-Length: 28
Type= CPT
SubType= SIT-IC
```

The X-Detect event notification process is as follows:

1. For IP-to-Tel or Tel-to-IP calls, the device receives a SIP request message (using the X-Detect header) that the remote party wishes to detect events on the media stream. For incoming (IP-to-Tel) calls, the request must be indicated in the initial INVITE and responded to either in the 183 response (for early dialogs) or in the 200 OK response (for confirmed dialogs).

2. Once the device receives such a request, it sends a SIP response message (using the X-Detect header) to the remote party, listing all supported events that can be detected. The absence of the X-Detect header indicates that no detections are available.

3. Each time the device detects a supported event, the event is notified to the remote party by sending an INFO message with the following message body:

- Content-Type: application/X-DETECT
- Type = **[CPT |** FAX | PTT…]
- Subtype = xxx (according to the defined subtypes of each type)

Below is an example of SIP messages using the X-Detect header:

```
INVITE sip:101@10.33.2.53;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
Max-Forwards: 70
From: "anonymous" <sip:anonymous@anonymous.invalid>;tag=1c25298
To: <sip:101@10.33.2.53;user=phone>
Call-ID: 11923@10.33.2.53
CSeq: 1 INVITE
Contact: <sip:100@10.33.2.53>
X- Detect: Request=CPT,FAX
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
From: "anonymous" <sip:anonymous@anonymous.invalid>;tag=1c25298
To: <sip:101@10.33.2.53;user=phone>;tag=1c19282
Call-ID: 11923@10.33.2.53
CSeq: 1 INVITE
Contact: <sip:101@10.33.2.53>
X- Detect: Response=CPT,FAX
INFO sip:101@10.33.2.53;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
Max-Forwards: 70
From: "anonymous" <sip:anonymous@anonymous.invalid>;tag=1c25298
To: <sip:101@10.33.2.53;user=phone>
Call-ID: 11923@10.33.2.53
CSeq: 1 INVITE
Contact: <sip:100@10.33.2.53>
X- Detect: Response=CPT,FAX
Content-Type: Application/X-Detect
Content-Length: xxx
Type = CPT
Subtype = SIT
```

37.5 Querying Device Channel Resources using SIP OPTIONS

The device reports its maximum and available channel resources in SIP 200 OK responses upon receipt of SIP OPTIONS messages. The device sends this information in the SIP X-Resources header with the following parameters:

■ **telchs:** Specifies the total telephone channels and the number of free (available) telephone channels.

■ **mediachs:** Not applicable.

Below is an example of the X-Resources:

```
X-Resources: telchs= 8/4;mediachs=0/0
```

In the example above, "telchs" specifies the number of available channels and the number of occupied channels (4 channels are occupied and 8 channels are available).

Part IX

Diagnostics

38 Syslog and Debug Recordings

Syslog is an event notification protocol that enables a device to send event notification messages across IP networks to event message collectors, also known as Syslog servers. The device contains an embedded Syslog client, which sends error reports / events that it generates to a remote Syslog server using the IP / UDP protocol. This information is a collection of error, warning, and system messages that records every internal operation of the device.

For receiving Syslog messages generated by the device, you can use any of the following Syslog servers:

- **Device's embedded Syslog server:** The device provides an embedded Syslog server, which is accessed through the Web interface. This provides limited Syslog server functionality.
- **Wireshark:** Third-party network protocol analyzer (http://www.wireshark.org).
- **Third-party, Syslog server:** Any third-party Syslog server program that enables filtering of messages according to parameters such as priority, IP sender address, time, and date.

38.1 Syslog Message Format

The Syslog message is sent from the device to a Syslog server as an ASCII (American Standard Code for Information Interchange) message. Syslog uses UDP as its underlying transport layer mechanism. By default, UDP port 514 is assigned to Syslog, but this can be changed (see 'Configuring Syslog' on page 444).

Below is an example of a Syslog message:

```
13:10:57.811 : 10.13.4.12 : NOTICE  : [S=235][SID:1034099026] (
lgr_flow)(63       )  UdpTransportObject#0- Adding socket event
for address 10.33.2.42:5060 [Time: 04-19-2012@18:29:39]
```

Table 38-1: Syslog Message Format Description

Message Item	Description
Message Types	Syslog generates the following types of messages:
	▪ ERROR: Indicates that a problem has been identified that requires immediate handling.
	▪ WARNING: Indicates an error that might occur if measures are not taken to prevent it.
	▪ NOTICE: Indicates that an unusual event has occurred.
	▪ INFO: Indicates an operational message.
	▪ DEBUG: Messages used for debugging.
	Notes:
	▪ The INFO and DEBUG messages are required only for advanced debugging. Therefore, by default, they are not sent by the device.
	▪ When viewing Syslog messages in the Web interface, these message types are color coded.

Message Item	Description
Message Sequence Number [S=<number>]	Syslog messages are sequentially numbered in the format **[S=<number>]**, for example, "**[S=**643]". A skip in the number sequence of messages indicates a loss of message packets. For example, in the below Syslog message generation, messages 238 through 300 were not received. In other words, 63 Syslog messages were lost (the sequential numbers are indicated below in bold font): ``` 18:38:14. 52 : 10.33.45.72 : NOTICE: [S=235][SID:1034099026] (lgr_psbrdex)(619) recv <-- DIGIT(0) Ch:0 OnTime:0 InterTime:100 Direction:0 System:1 [File: Line:-1] 18:38:14. 83 : 10.33.45.72 : NOTICE: [S=236][SID:1034099026] (lgr_flow)(620) #0:DIGIT_EV [File: Line:-1] 18:38:14. 83 : 10.33.45.72 : NOTICE: [S=237][SID:1034099026] (lgr_flow)(621)	#0:DIGIT_EV [File: Line:-1] 18:38:14.958 : 10.33.45.72 : NOTICE: [S=301][SID:1034099026] (lgr_flow)(625)	#0:DIGIT_EV [File: Line:-1] ```
Log Number (lgr)(number)	Ignore this number; it has been replaced by the Message Sequence Number (described previously).
Session ID	Automatically assigned (random), unique session identifier (session-id / SID) number per call in the CDR of sent Syslog messages and debug recording packets. This enables you to filter the information (such as SIP, Syslog, and media) according to the SID. A call session is considered either as a Tel-to-IP leg or an IP-to-Tel leg, where each leg is assigned a unique SID. The benefit of this unique numbering is that it enables you to filter the information (such as SIP, Syslog, and media) according to a specific SID. **Note:** Forked legs and alternative legs share the same SID.
Message Body	Describes the message.
Timestamp	When the Network Time Protocol (NTP) is enabled, a timestamp string **[hour:**minutes:seconds] is added to all Syslog messages.

38.1.1   Event Representation in Syslog Messages

The Syslog message events that the device sends are represented by unique abbreviations. An example of an abbreviated event in a Syslog message indicating packet loss (PL) is shown below:

```
Apr  4 12:00:12 172.30.1.14 PL:5  [Code:3a002] [CID:3294] [Time: 20:17:00]
```

The table below lists these unique event abbreviations:

Table 38-2: Syslog Error Name Descriptions

Error Abbreviation	Error Name Description
AA	Invalid Accumulated Packets Counter
AC	Invalid Channel ID
AL	Invalid Header Length
AO	Invalid Codec Type
AP	Unknown Aggregation Payload Type
AR	Invalid Routing Flag Received
AT	Simple Aggregation Packets Lost
CC	Command Checksum Error
CE	Invalid Cell Coder Code
CS	Command Sequence Error
ES	8 sec Timeout Before Disconnect
HO	Host Received Overrun
IA	Invalid AMR Payload
IC	Invalid CID Error
IG	Invalid G723 Code
IP	Invalid payload length
IR	Invalid RTCP Packet
IS	Invalid SID Length
LC	Transmitter Received Illegal Command
LF	Lost Fax Frames In High Speed Mode
LM	Lost Modem Frames In High Speed Mode
MI	Misalignment Error
MR	Modem Relay Is Not Supported
OR	DSP JB Overrun
PH	Packet Header Error
PL	RTP Packet Loss
RB	Counts the number of BFI Frames Received From The Host
RD	No Available Release Descriptor
RO	RTP Reorder
RP	Unknown RTP Payload Type
RS	RTP SSRC Error
UF	Unrecognized Fax Relay Command
AA	Invalid Accumulated Packets Counter

Error Abbreviation	Error Name Description
AC	Invalid Channel ID
AL	Invalid Header Length
AO	Invalid Codec Type
AP	Unknown Aggregation Payload Type
AR	Invalid Routing Flag Received

38.1.2 Identifying AudioCodes Syslog Messages using Facility Levels

The device's Syslog messages can easily be identified and distinguished from Syslog messages from other equipment, by setting its Facility level. The Facility levels of the device's Syslog messages are numerically coded with decimal values. Facility level may use any of the "local use" facilities (0 through 7), according to RFC 3164. Implementing Facility levels is useful, for example, if you collect the device's as well as other equipments' Syslog messages on the same server. Therefore, in addition to filtering Syslog messages according to IP address, the messages can be filtered according to Facility level.

The Facility level is configured using the SyslogFacility ini file parameter, which provides the following options:

Table 38-3: Syslog Facility Levels

Numerical Value	Facility Level
16 (default)	local use 0 (local0)
17	local use 1 (local1)
18	local use 2 (local2)
19	local use 3 (local3)
20	local use 4 (local4)
21	local use 5 (local5)
22	local use 6 (local6)
23	local use 7 (local7)

Syslog messages begin with a less-than ("**<**") character, followed by a number, which is followed by a greater-than ("**>**") character. This is optionally followed by a single ASCII space. The number is known as the *Priority* and represents both the Facility level and the Severity level. A Syslog message with Facility level 16 is shown below:

```
Facility: LOCAL0 - reserved for local use (16)
```

38.1.3   SNMP Alarms in Syslog Messages

SNMP alerts are sent to the Syslog server using the following formats:

■ **Raised Alarms:** RAISE-ALARM: <Alarm Name>; Textual Description: <Textual Description>; Severity <Alarm Severity>; Source <Alarm Source>; Unique ID: <Alarm Unique ID >.

If additional information exists in the alarm, then these are also added: Additional Info1:/ Additional Info2:/ Additional Info3

The Messages' Severity is as follows:

Table 38-4: Syslog Message Severity

ITU Perceived Severity (SNMP Alarm's Severity)	AudioCodes' Syslog Severity
Critical	RecoverableMsg
Major	RecoverableMsg
Minor	RecoverableMsg
Warning	Notice
Indeterminate	Notice
Cleared	Notice

■ **Cleared Alarms:** CLEAR-ALARM: <Alarm Name>; Textual Description: <Textual Description>; Severity <Alarm Severity>; Source <Alarm Source>; Unique ID: <Alarm Unique ID >; If exists Additional Info1:/ Additional Info2:/ Additional Info3:

38.2   Configuring Syslog Settings

The procedure below describes how to configure Syslog. This includes defining the Syslog server address as well as selecting the activities on the device (for example, a parameter value change) that you want reported to the server.

> **Notes:**
>
> - For configuring CDR reporting, see 'Configuring CDR Reporting' on page 424.
> - For viewing Syslog messages in the Web interface, see 'Viewing Syslog Messages' on page 447.
> - For a detailed description on the Syslog parameters, see 'Syslog, CDR and Debug Parameters' on page 494.

➢ **To configure Syslog :**

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

2. Enable the Syslog feature by setting the 'Enable Syslog' to **Enable**.
3. Define the Syslog server using the 'Syslog Server IP Address' and 'Syslog Server Port' parameters.
4. Configure the debug level using the 'Debug Level' parameter.
5. Under the 'Activity Types to Report ...' group, select the activities to report.
6. Click **Submit** to apply your changes.

38.3   Configuring Debug Recording

The device enables you to activate debug recording and send debug recording packets to a defined capturing server. When the debug recording is activated, the device duplicates all messages that are sent and/or received by it and then sends them to an external IP address. The debug recording can be done for different types of traffic for example, RTP/RTCP, T.38, and SIP.

Debug recording is used for advanced debugging when you need to analyze internal messages and signals. Debug recording is also useful for recording network traffic in environments where hub or port mirroring is unavailable and for recording internal traffic between two endpoints on the same device.

> **Note:**   Debug recording is collected only on the device's OAMP interface.

> ➢   **To configure and activate debug recording:**

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

Figure 38-1: Logging Settings Page

▼ Debug Recording	
Debug Recording Destination IP	10.13.4.22
Debug Recording Destination Port	925
Debug Recording Status	Start ▼

2.   Configure the debug capturing server using the 'Debug Recording Destination IP' and 'Debug Recording Destination Port' parameters.

3.   From the 'Debug Recording Status' drop-down list, select **Start** to start the debug recording or **Stop** to end the recording.

4.   Click **Submit** to apply your changes.

38.4   Filtering Syslog Messages and Debug Recordings

The device can filter Syslog messages and debug recording (DR) packets, sent by the device to a Syslog server and packet capturing application (such as Wireshark) respectively. This can be useful to reduce CPU consumption and minimize negative impact on VoIP performance.

You can configure up to 30 filtering rules, each based on a selected filtering criteria (e.g., an IP Group). Each filtering criteria can be configured with a range. For example, you can filter Syslog messages for IP Groups 1 through 4. For each filter criteria, you can enable or disable Syslog messages and debug recording.

Debug recording can also be filtered using various filtering criteria such as SIP signaling or signaling and media.

➢ **To configure logging filtering rules:**

1. Open the Logging Filters Table page (**Configuration** tab > **System** menu > **Logging** > **Logging Filters Table**).

2. Click the **Add** button; the Add Record dialog box appears:

Figure 38-2: Logging Filters Table - Add Record Dialog Box

Add Record	✕
Index	1
Type	Any Filter ▼
Value	
Syslog	Enable ▼
Capture Type	Signaling + Media ▼
	🖫 Submit ✕ Cancel

3. Configure the logging filter, as required. See the table below for a description of the parameters.

4. Click **Submit** to save your changes.

Notes:

- To configure the Syslog debug level, use the 'Debug Level' parameter (see 'Configuring Syslog' on page 444).
- The Logging Filters table can also be configured using the table ini file parameter, LoggingFilters.

Table 38-5: Logging Filters Table Parameters Description

Parameter	Description
Filter Type CLI: filter-type **[LoggingFilters_Type]**	Defines the filter criteria. ▪ **[1]** Any (default) ▪ **[3]** Trunk Group ID = Filters according to a specified Trunk Group ID. ▪ **[6]** Tel-to-IP = Filters according to a specified Tel-to-IP routing rule listed in the Outbound IP Routing table. ▪ **[7]** IP-to-Tel = Filters according to a specified IP-to-Tel routing rule listed in the Inbound IP Routing table. ▪ **[8]** IP Group = Filters according to a specified IP Group ID listed in the IP Group table. ▪ **[12]** User = Filters according to a specified user defined by username or user@host. ▪ **[13]** IP Trace = Filters according to a specified IP network trace wireshark-like expression. For a detailed description on configuring IP traces, see 'Filtering IP Network Traces' on page 447.

Parameter	Description

Value
CLI: value
[LoggingFilters_Value]
	Defines the value of the selected filtering type in the 'Filter Type' parameter.
	The value can be the following:
	▪ A single value
	▪ A range, using a hyphen "-" between the two values, e.g., "1-3"
	▪ Multiple, non-contiguous values, using commas "," between each value, e.g., "1,3,9"
	▪ FXO/FXS pertaining to a module, using the syntax module number/port or port, for example:
	  ✓ "1/2", means module 1, port 2
	  ✓ "1/[2-4]", means module 1, ports 2 through 4
	▪ Any to indicate all
	▪ For IP trace expressions, see e 'Filtering IP Network Traces' on page 447

Syslog **[LoggingFilters_Syslog]**	Enables Syslog messages for the defined logging filter: ▪ **[0]** Disable (default) ▪ **[1]** Enable
Capture Type **[LoggingFilters_CaptureType]**	Enables debug recordings for the defined logging filter and defines what to record: ▪ **[0]** None (default) ▪ **[1]** Signaling = Information related to signaling such as SIP signaling messages, Syslog, and CDR. ▪ **[2]** Signaling & Media = Signaling and media (RTP/RTCP/T.38). ▪ **[3]** Signaling & Media & PCM = Signaling, media, and PCM (voice signals from and to TDM).

38.4.1   Filtering IP Network Traces

You can filter Syslog and debug recording messages for IP network traces, by setting the 'Filter Type' parameter to **IP Trace** in the Logging Filters table. IP traces are used to record any IP stream that is not associated with media (RTP/RTCP), 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).

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:

Supported Wireshark-like Expressions for 'Value' Parameter

Expression	Description
ip.src, ip.dst	Source and destination IP address
ip.addr	IP address - up to two IP addresses can be entered
ip.proto	IP protocol type (PDU) entered as an enumeration value (e.g., 1 is ICMP, 6 is TCP, 17 is UDP)
udp, tcp, icmp, sip, http, https	Single expressions for protocol type
udp.port, tcp.port	Transport layer

Expression	Description
udp.srcport, tcp.srcport	Transport layer for source port
udp.dstport, tcp.dstport	Transport layer for destination port
and, &&, ==, <, >	Between expressions

Below are examples of configured expressions for the 'Value' parameter:

- udp && ip.addr==10.8.6.55
- ip.src==10.8.6.55 && udp.port>=5000 and udp.port<6000
- ip.dst==10.8.0.1/16
- ip.addr==10.8.6.40

For conditions requiring the "or" / "||" expression, add multiple table rows. For example, the Wireshark condition "(ip.src == 1.1.1.1 or ip.src == 2.2.2.2) and ip.dst == 3.3.3.3" can be configured using the following two table row entries:

1. ip.src == 1.1.1.1 and ip.dst == 3.3.3.3
2. ip.src == 2.2.2.2 and ip.dst == 3.3.3.3

> **Note:** If the 'Value' field is left empty, the device will record all IP traffic types.

38.5   Viewing Syslog Messages

You can use the following tools to view the Syslog messages sent by the device:

■   Web interface's Message Log page (see below).

■   Any third-party Syslog server (e.g., Wireshark).

The procedure below describes how to view Syslog messages in the Web interface.

> **Notes:**
>
> • It's not recommended to keep a Message Log session open for a prolonged period. This may cause the device to overload. For prolonged (and detailed) debugging, use an external Syslog server.
>
> • You can select the Syslog messages in this page, and copy and paste them into a text editor such as Notepad. This text file (*txt*) can then be sent to AudioCodes Technical Support for diagnosis and troubleshooting.

➢   **To activate the Web interface's Message Log:**

1.   Enable Syslog (see 'Configuring Syslog' on page).

2.   Open the Message Log page (**Status & Diagnostics** tab > **System Status** menu > **Message Log**); the Message Log page is displayed and the log is activated.

Figure 38-3: Message Log Page

The displayed logged messages are color-coded as follows:

• Yellow - fatal error message

• Blue - recoverable error message (i.e., non-fatal error)

• Black - notice message

➢   **To stop and clear the Message Log:**

■   Close the Message Log page by accessing any another page in the Web interface.

38.6   Collecting Debug Recording Messages

To collect debug recording packets, the open source program Wireshark is used. AudioCodes proprietary plug-in files for Wireshark, which are shipped in your software kit, are also required.

> **Notes:**
>
> - The default debug recording port is 925. You can change the port in Wireshark (**Edit** menu > **Preferences** > **Protocols** > **AC DR**).
> - The plug-ins are per major software release and are applicable to Wireshark Ver. 1.62.
> - The plug-ins are backward compatible.
> - From Wireshark Ver. 99.08, the tpncp.dat file must be located in the folder, ...WireShark\tpncp.

➢ **To install Wireshark and the plug-ins for debug recording:**

1. Install Wireshark on your computer. The Wireshark program can be downloaded from http://www.wireshark.org.

2. Copy the supplied AudioCodes plug-in files to the directory in which you installed Wireshark, as follows:

Copy this file	To this folder
...\dtds\cdr.dtd	Wireshark\dtds\
...\plugins\1.6.2*.dll	Wireshark\plugins\1.6.2
...\tpncp\tpncp.dat	Wireshark\tpncp

3. Start Wireshark.

4. 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:

Figure 38-4: Wireshark Page
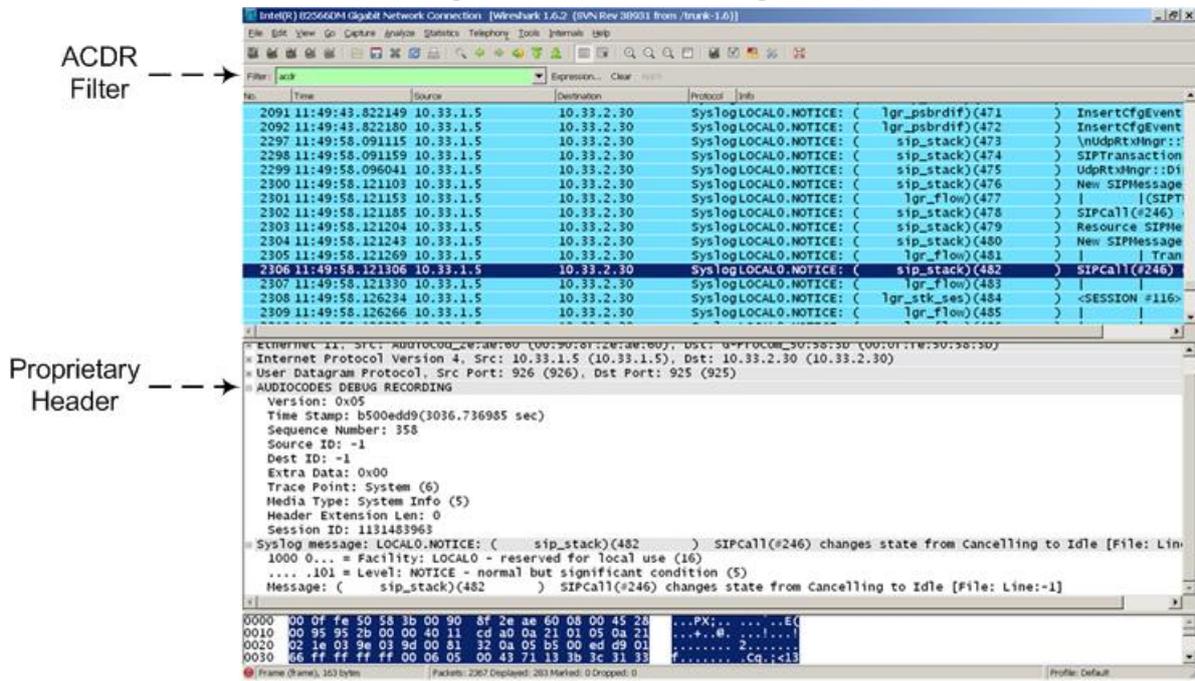
This page is intentionally left blank.

39 Self-Testing

The device features the following self-testing modes to identify faulty hardware components:

■ **Detailed Test (Configurable):** This test verifies the correct functioning of the different hardware components on the device. This test is done when the device is taken out of service (i.e., not in regular service for processing calls). The test is performed on startup when initialization of the device completes.

To enable this test, set the ini file parameter, EnableDiagnostics to 1 or 2, and then reset the device. The Ready and Fail LEDs are lit while this test is running. Upon completion of the test and if the test fails, the device sends information on the test results of each hardware component to the Syslog server.

The following hardware components are tested:

- Flash memory - when EnableDiagnostics = 1 or 2
- DSPs - when EnableDiagnostics = 1 or 2
- Physical Ethernet ports - when EnableDiagnostics = 1 or 2
- Analog interfaces - when EnableDiagnostics = 1 or 2

Notes:

- To return the device to regular operation and service, disable the test by setting the ini file parameter, EnableDiagnostics to 0, and then reset the device.
- While the test is enabled, ignore errors sent to the Syslog server.

■ **Startup Test (automatic):** This hardware test has minor impact in real-time. While this test is executed, the regular operation of the device is disabled. If an error is detected, an error message is sent to the Syslog.

This page is intentionally left blank.

40   Line Testing

40.1   FXS Line Testing

The device can test the telephone lines connected to its FXS ports, using the SNMP acAnalogFxsLineTestTable table. These tests provide various line measurements. In addition to these tests, a keep-alive test is also done every 100 msec on each of the analog ports to detect communication problems with the analog equipment and overheating of the FXS ports.

■   Hardware revision number

■   Temperature (above or below limit, only if a thermometer is installed)

■   Hook state

■   Coefficients checksum

■   Message waiting indication status

■   Ring state

■   Reversal polarity state

For MP-118 and MP-124 only, you can also use the following Command Shell commands to view line status and electrical measurements per FXS port or phone number:

```
/SIP>LineTesting Port <port number> <test type>
```

- or -

```
/SIP>LineTesting Phone <phone number> <test type>
```

Where <test type> can be one of the following values:

■   0 = Line status, which includes the following:

 • Hook status – on-hook (0) or off-hook (1)

 • Message Waiting Indication (MWI) – off (0) or on (1)

 • Ring – off (0) or on (1)

 • Reversal polarity – off (0) or on (1)

■   Line electrical measurements:

 • 1 = DC Voltage Tip-Ring [V]

 • 2 = DC Voltage Tip-Ground [V]

 • 3 = DC Voltage Ring-Ground [V]

 • 4 = AC Voltage Transmit(Tel2IP) [dbm]

 • 5 = AC Voltage Receive (IP2Tel) [dbm]

 • 6 = AC Voltage Transmit & Receive [dbm]

 • 7 = Current [mA]

 • 8 = Resistance Tip-Ring [Ohm]

 • 9 = Resistance Tip-Ground [Ohm]

 • 10 = Resistance Ring-Ground [Ohm]

 • 11 = Capacity Tip-Ring [F]

 • 12 = Capacity Tip-Ground [F]

 • 13 = Capacity Ring-Ground [F]

 • 14 = AC Voltage Tip-Ring [V]

- 15 = AC Voltage Tip-Ground [V]
- 16 = AC Voltage Ring-Ground [V]
- 17 = All the above

> **Notes:**
> - Use the Analog Line testing mechanism only for monitoring and never when there are calls in progress.
> - For MP-118, the line status and electrical measurement tests are supported only if they are done when the grounding reference is relative to the device. However, if the grounding is done directly to the earth, the tests are supported only on specific hardware models. For more information, please contact your AudioCodes' sales representative.
> - Line electrical measurements are supported only on certain MP-124 hardware assemblies. For more information, contact your AudioCodes' sales representative.

40.2 FXO Line Testing

The device can test the telephone lines connected to its FXO ports, using the SNMP acAnalogFxoLineTestTable table. These tests provide various line measurements. In addition to these tests (detailed below), a keep-alive test is also done every 100 msec on each of the analog ports to detect communication problems with the analog equipment.

- Line Current (mA)
- Line Voltage (V)
- Hook (0 = on-hook; 1 = off-hook)
- Ring (0 - Off; 1 - On)
- Line Connected (0 = Disconnected; 1 = Connected)
- Polarity state (0 = Normal; 1 = Reversed, 2 = N\A)
- Line polarity (0 = Positive; 1 = Negative)
- Message Waiting Indication (0 = Off; 1 = On)

> **Note:** Use the Analog Line testing mechanism only for monitoring and never when there are calls in progress.

41 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, and 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, DTMF signals, termination reasons, as well as voice quality statistics.

41.1 Configuring Test Call Endpoints

The Test Call table enables you to test the SIP signaling (setup and registration) of calls and media (DTMF signals) between a simulated phone on the device and a remote endpoint. These tests involve both incoming and outgoing calls, where the test endpoint can be configured as the caller or called party. Test calls can be dialed automatically at a user-defined interval and/or manually when required. The simulated phone and remote endpoints are defined as SIP URIs (user@host) and the remote destination can be defined as an IP Group, IP address, or according to an Outbound IP Routing rule. You can also enable automatic registration of the endpoint.

When a SIP test call is initiated, the device generates a SIP INVITE towards the remote endpoint (e.g., a SIP proxy server or softswitch). It simulates the SIP call setup process, managing SIP 1xx responses and completing the SIP transaction with a 200 OK.

> **Notes:**
> - By default, you can configure up to five test calls. This maximum can be increased by installing the relevant Software License Key. For more information, contact your AudioCodes sales representative.
> - The Test Call Endpoint table can also be configured using the table ini file parameter Test_Call (see 'SIP Test Call Parameters' on page 494).

➢ **To configure test calls:**

1. Open the Test Call Table page (**Configuration** tab > **System** menu > **Test Call** > **Test Call Table**).
2. Click the **Add** button; the following dialog box appears:

Figure 41-1: General Tab of Test Call Table

| General | Authentication | Test Settings |

Index	0
Endpoint URI	
Called URI	
Route By	GW Tel2IP
IP Group ID	-1
Destination Address	
Destination Transport Type	
SRD	0
Application Type	GW & IP2IP

Submit   Cancel

3. Configure the test endpoint parameters as desired. See the table below for a description of these parameters.
4. Click **Submit** to apply your settings.

Test Call Table Parameters

Parameter	Description
General Tab	
Endpoint URI **[Test_Call_EndpointURI]**	Defines the endpoint's URI. This can be defined as a user or user@host. The device identifies this endpoint only by the URI's user part. The URI's host part is used in the SIP From header in REGISTER requests. The valid value is a string of up to 150 characters. By default, this parameter is not configured.
Called URI **[Test_Call_CalledURI]** CLI: called-uri	Defines the destination (called) URI (user@host). The valid value is a string of up to 150 characters. By default, this parameter is not configured.
Route By **[Test_Call_DestType]**	Defines the type of routing method. This applies to incoming and outgoing calls. ▪ **[0]** GW Tel2IP = (Default) Calls are matched by (or routed to) an SRD and Application type (defined in the SRD and Application Type parameters below). ▪ **[1]** IP Group = Calls are matched by (or routed to) an IP Group ID. ▪ **[2]** Dest Address = Calls are matched by (or routed to) an SRD and application type. **Notes:** ▪ For REGISTER messages, the option **[0]** cannot be used as the routing method. ▪ For REGISTER messages, if option **[1]** is used, only Server-type IP Groups can be used.
IP Group ID **[Test_Call_IPGroupID]**	Defines the IP Group ID to which the test call is sent or from which it is received. **Notes:** ▪ This parameter is applicable only if option **[1]** is configured for the 'Route By' parameter. ▪ This IP Group is used for incoming and outgoing calls.
Destination Address **[Test_Call_DestAddress]**	Defines the destination host. This can be defined as an IP address**[:port]** or DNS name**[:port]**. **Note:** This parameter is applicable only if the 'Route By' parameter is set to **[2]** (Dest Address).
Destination Transport Type **[Test_Call_DestTransportType]**	Defines the transport type for outgoing calls. ▪ **[-1]** Not configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** This parameter is applicable only if the 'Route By' parameter is set to **[2]** (Dest Address).
Application Type **[Test_Call_ApplicationType]**	Defines the application type for the endpoint. ▪ **[0]** GW & IP2IP (default)

Parameter	Description
Authentication Tab **Note:** These parameters are applicable only if the test endpoint is set to **Caller** (see the 'Call Party' parameter).	
Auto Register **[Test_Call_AutoRegister]**	Enables automatic registration of the endpoint. The endpoint can register to the device itself or to the 'Destination Address' or 'IP Group ID' parameter settings (see above). ▪ **[0]** False (default) ▪ **[1]** True
User Name **[Test_Call_UserName]**	Defines the authentication username. By default, no username is defined.
Password **[Test_Call_Password]**	Defines the authentication password. By default, no password is defined.
Test Settings Tab	
Call Party **[Test_Call_CallParty]**	Defines whether the test endpoint is the initiator or receiving side of the test call. ▪ **[0]** Caller (default) ▪ **[1]** Called
Maximum Channels for Session **[Test_Call_MaxChannels]**	Defines the maximum number of concurrent channels for the test session. For example, if you have configured an endpoint "101" and you set this parameter to "3", the device automatically creates three simulated endpoints - "101", "102" and "103" (i.e., consecutive endpoint URIs are assigned). The default is 1.
Call Duration **[Test_Call_CallDuration]**	Defines the call duration (in seconds). The valid value is -1 to 100000. The default is 20. A value of 0 means infinite. A value of -1 means that the parameter value is automatically calculated according to the values of the 'Calls per Second' and 'Maximum Channels for Session' parameters. **Note:** This parameter is applicable only if 'Call Party' is set to **Caller**.
Calls per Second **[Test_Call_CallsPerSecond]**	Defines the number of calls per second. **Note:** This parameter is applicable only if 'Call Party' is set to **Caller**.

Test Mode
[Test_Call_TestMode]
	Defines the test session mode.
	▪ **[0]** Once = (Default) The test runs until the lowest value between the following is reached:
	  ✓ Maximum channels is reached for the test session, configured by 'Maximum Channels for Session'.
	  ✓ Call duration ('Call Duration') multiplied by calls per second ('Calls per Second').
	  ✓ Test duration expires, configured by 'Test Duration'.
	▪ **[1]** Continuous = The test runs until the configured test duration is reached. If it reaches the maximum channels configured for the test session (in the 'Maximum Channels for Session'), it waits until the configured call duration of a currently established tested call expires before making the next test call. In this way, the test session stays within the configured maximum channels.
	Note: This parameter is applicable only if 'Call Party' is set to **Caller**.

Parameter	Description
Test Duration **[Test_Call_TestDuration]**	Defines the test duration (in minutes). The valid value is 0 to 100000. The default is 0 (i.e., unlimited). **Note:** This parameter is applicable only if 'Call Party' is set to **Caller**.
Play **[Test_Call_Play]**	Enables playing a user-defined DTMF signal to the answered side of the call. ▪ **[0]** Disable (default) ▪ **[1]** DTMF To configure the played DTMF signal, see 'Configuring DTMF Tones for Test Calls' on page 462. **Note:** To configure the DTMF signaling type (e.g., out-of-band or in-band) use the 'DTMF Transport Type' parameter (see 'Configuring DTMF Transport Types' on page 173).
Schedule Interval **[Test_Call_ScheduleInterval]**	Defines the interval (in minutes) between automatic outgoing test calls. The valid value range is 0 to 100000. The default is 0 (i.e., scheduling is disabled). **Note:** This parameter is applicable only if 'Call Party' is set to **Caller**.

41.1.1   Starting, Stopping and Restarting Test Calls

The procedure below describes how to start, stop, and restart test calls.

➢ **To start, stop, and restart a test call:**

1. In the Test Call table, select the required test call entry; the **Actions** button appears above the table.
2. From the **Actions** drop-down list, choose the required command:
   - **Dial**: starts the test call (this action is applicable only if the test call party is the caller).
   - **Drop Call**: stops the test call.
   - **Restart**: ends all established calls and then starts the test call session again.

The status of the test call is displayed in the 'Test Status' field of the Test Call table:

- ■ "Idle": test call is not active.
- ■ "Scheduled": test call is planned to run (according to 'Schedule Interval' parameter settings)
- ■ "Running": test call has been started (i.e., the **Dial** command was clicked)
- ■ "Receiving": test call has been automatically activated by calls received for the test call endpoint from the remote endpoint (when all these calls end, the status returns to "Idle")
- ■ "Terminating": test call is in the process of terminating the currently established calls (this occurs if the **Drop Call** command is clicked to stop the test)
- ■ "Done": test call has been successfully completed (or was prematurely stopped by clicking the **Drop Call** command)

A more detailed description of this field is displayed below the table when you click the **Show/Hide** button (see 'Viewing Test Call Statistics' on page 461.

41.1.2   Viewing Test Call Statistics

In addition to viewing a brief status description of the test call in the 'Test Status' field (as described in 'Starting, Stopping and Restarting Test Calls' on page 460), you can also view a more detailed status description which includes test call statistics.

➢ **To view statistics of a test call:**

1. Open the Test Call Table page (**Configuration** tab > **System** menu > **Test Call** > **Test Call Table**).

2. Select the test call table entry whose call statistics you want to view.

3. Click the **Show/Hide** button; the call statistics are displayed in the **Test Statistics** pane located below the table, as shown in the figure below:

Figure 41-2: Viewing Test Call Statistics

The 'Test Statistics' pane displays the following test session information:

■ **Elapsed Time:** Duration of the test call since it was started (or restarted).

■ **Active Calls:** The number of currently active test calls.

■ **Call Attempts:** The number of calls that were attempted.

■ **Total Established Calls:** The total number of calls that were successfully established.

■ **Total Failed Attempts:** The total number of calls that failed to be established.

■ **Remote Disconnections Count:** Number of calls that were disconnected by the remote side.

■ **Average CPS:** The average calls per second.

■ **Test Status:** Displays the status (brief description) as displayed in the 'Test Status' field (see 'Starting, Stopping and Restarting Test Calls' on page 460).

■ **Detailed Status:** Displays a detailed description of the test call status::

• "Idle": The test call is currently not active.

• "Scheduled - Established Calls: <established calls>, ASR: <%>": The test call is planned to run (according to 'Schedule Interval' parameter settings) and also shows the following summary of completed test calls:

♦ Total number of test calls that were established.

♦ Number of successfully answered calls out of the total number of calls attempted (ASR).

- "Running (Calls: <number of active calls>, ASR: <%>)": The test call has been started (i.e., the **Dial** command was clicked) and shows the following:
  - ♦ Number of currently active test calls.
  - ♦ Number of successfully answered calls out of the total number of calls attempted (Answer Seizure Ratio or ASR).
- "Receiving (<number of active calls>)": The test call has been automatically activated by calls received for this configured test call endpoint from the configured remote endpoint. When all these calls terminate, the status returns to "Idle".
- "Terminating (<number of active calls>)": The **Drop Call** command has been clicked to stop the test call and the test call is in the process of terminating the currently active test calls.
- "Done - Established Calls: <established calls>, ASR: <%>": The test call has been successfully completed (or was prematurely stopped by clicking the **Drop Call** command) and shows the following:
  - ♦ Total number of test calls that were established.
  - ♦ Number of successfully answered calls out of the total number of calls attempted (ASR).

> **Note:** On the receiving side, when the first call is accepted in "Idle" state, statistics are reset.

41.2   Configuring DTMF Tones for Test Calls

By default, no DTMF signal is played to an answered test call (incoming or outgoing). However, you can enable this per configured test call in the Test Call table (see 'Configuring Test Call Endpoints' on page 457). If enabled, the default DTMF signal that is played is "3212333". You can change this as described below.

> **Notes:**
>
> - The DTMF signaling type (e.g., out-of-band or in-band) can be configured using the 'DTMF Transport Type' parameter. For more information, see 'Configuring DTMF Transport Types' on page 173.
> - To generate DTMF tones, the device's DSP resources are required.

> **To configure the played DTMF signal to answered test call:**

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

Test Call DTMF String	3212333

2. In the 'Test Call DTMF String' field, enter the DTMF string (up to 15 digits).
3. Click **Submit**.

41.3   Configuring Basic Test Call

The Basic Test Call feature tests incoming Gateway / IP-to-IP calls from a remote SIP endpoint to a simulated test endpoint on the device. The only required configuration is to assign a prefix number (*test call ID*) to the simulated endpoint. All incoming calls with this called (destination) prefix number is identified as a test call and sent to the simulated endpoint. The figure below displays a basic test call example.

Figure 41-3: Incoming Test Call Example
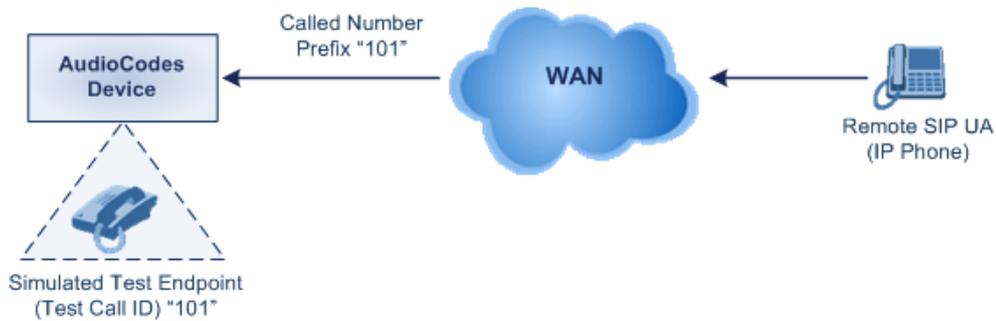
➢   **To configure basic call testing:**

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

Figure 41-4: Test Call Settings Page

2.   In the 'Test Call ID' field, enter a prefix for the simulated endpoint.
3.   Click **Submit** to apply your settings.

Notes:

- The Basic Test Call feature tests incoming calls only and is initiated only upon receipt of incoming calls with the configured prefix.
- For a full description of this parameter, see 'SIP Test Call Parameters' on page 494.

41.4 Test Call Configuration Examples

Below are a few examples of test call configurations.

■ **Single Test Call Scenario:** This example describes the configuration of a simple test call scenario that includes a single test call between a simulated test endpoint on the device and a remote endpoint.

Figure 41-5: Single Test Call Example
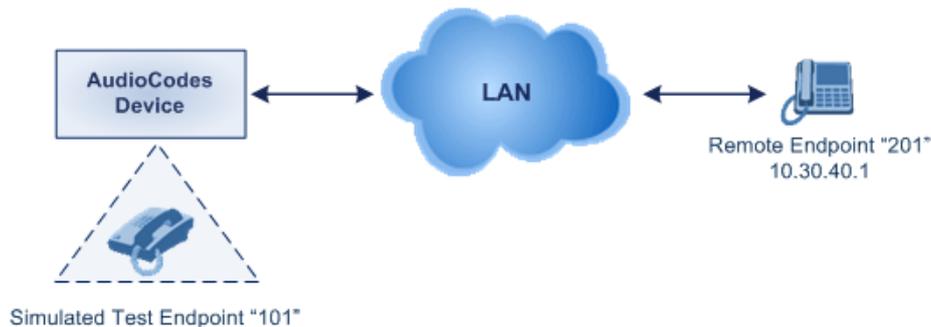
- Test Call table configuration:
  - ♦ Endpoint URI: "101"
  - ♦ Called URI: "201"
  - ♦ Route By: Dest Address
  - ♦ Destination Address: "10.30.40.01"
  - ♦ Call Party: Caller
  - ♦ Test Mode: Once (default)

Alternatively, if you want to route the test call using the Outbound IP Routing table for the Gateway / IP-to-IP application, configure the following:

- Test Call table configuration:
  - ♦ Endpoint URI: 101@10.0.0.1
  - ♦ Route By: GW Tel2IP
  - ♦ Called URI: 201@10.30.40.1
  - ♦ Call Party: Caller
- Outbound IP Routing table configuration:
  - ♦ Dest. Phone Prefix: 201 (i.e., the Called URI user-part)
  - ♦ Source Phone Prefix: 101 (i.e., the Endpoint URI user-part)
  - ♦ Dest. IP Address: 10.30.40.1

■ **Batch Test Call Scenario:** This example describes the configuration of a batch test call setup for scheduled and continuous call testing of multiple endpoints. The test call is done between two AudioCodes devices - Device A and Device B - with simulated test endpoints. This eliminates the need for phone users, who would otherwise need to answer and end calls many times for batch testing. The calls are initiated from Device A, where Device B serves as the remote answering endpoint.

Figure 41-6: Batch Test Call Example
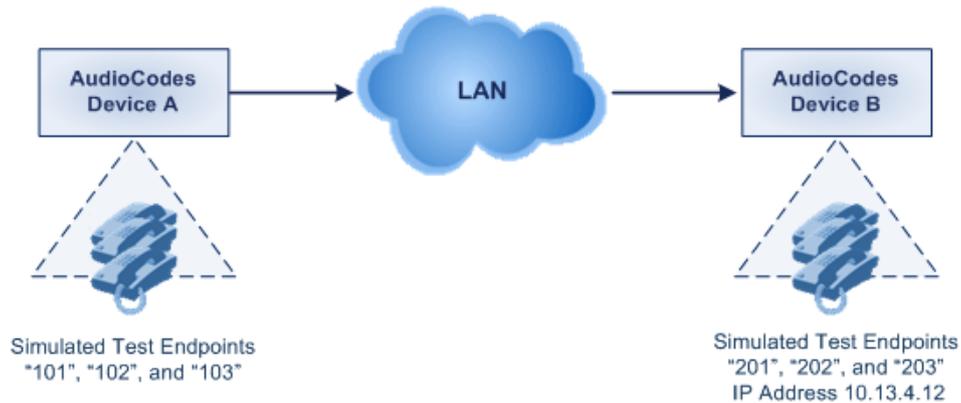
- Test Call table configuration at Device A:
  ♦ Endpoint URI: "101"
  ♦ Called URI: "201"
  ♦ Route By: Dest Address
  ♦ Destination Address: "10.13.4.12"
  ♦ Call Party: Caller
  ♦ Maximum Channels for Session: "3" (this setting configures three endpoints - "101", "102" and "103)
  ♦ Call Duration: "5" (seconds)
  ♦ Calls per Sec: "1"
  ♦ Test Mode: Continuous
  ♦ Test Duration: "3" (minutes)
  ♦ Schedule Interval: "180" (minutes)
- Test Call table configuration at Device B:
  ♦ Endpoint URI: "201"
  ♦ Call Party: Caller
  ♦ Maximum Channels for Session: "3" (this setting 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 41-7: Test Call Registration Example
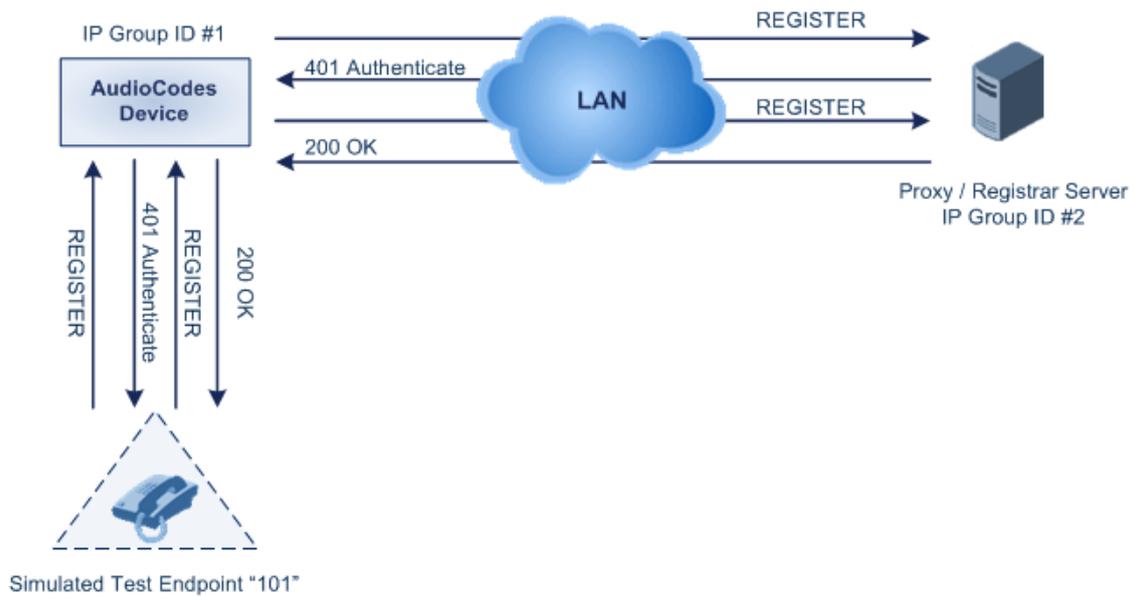
This example assumes that you have configured your device for communication between LAN phone users such as IP Groups to represent the device (10.13.4.12) and the proxy server, and IP-to-IP routing rules to route calls between these IP Groups.

- Test Call table configuration:
    - Endpoint URI: "101"
    - Called URI: "itsp"
    - Route By: Dest Address
    - Destination Address: "10.13.4.12" (this is the IP address of the device itself)
    - Auto Register: Enable
    - User Name: "testuser"
    - Password: "12345"
    - Call Party: Caller

Part X

Appendix

42   Dialing Plan Notation for Routing and Manipulation

The device supports flexible dialing plan notations for denoting the prefix and/or suffix source and/or destination numbers and SIP URI user names in the routing and manipulation tables.

Table 42-1: Dialing Plan Notations for Prefixes and Suffixes

Notation	Description
x (letter "x")	Denotes any single digit.
# (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. ▪ When used as part of a number, it denotes the asterisk key.
$ (dollar sign)	Denotes an empty prefix for incoming IP calls that do not have a user part in the Request-URI, or for incoming Tel calls that do not have a called or calling number. This is used for the following matching criteria: ▪ Source and Destination Phone Prefix ▪ Source and Destination Username ▪ Source and Destination Calling Name Prefix

Range of Digits

Notes:

▪ Dial plans denoting a prefix that is a range must be enclosed in square brackets, e.g., **[4-8]** or **23xx[456]**.
▪ Dial plans denoting a prefix that is not a range is not enclosed, e.g., **12345#**.
▪ Dial plans denoting a suffix must be enclosed in parenthesis, e.g., **(4)** and **(4-8)**.
▪ Dial plans denoting a suffix that include multiple ranges, the range must be enclosed in square brackets, e.g., **(23xx[4,5,6])**.
▪ An example for entering a combined prefix and suffix dial plan - assume you want to match a rule whose destination phone prefix is 4 to 8, and suffix is 234, 235, or 236. The entered value would be the following: **[4-8](23[4,5,6])**.

[n-m] or **(n-m)**	Represents a range of numbers, for example:
	▪ To depict numbers from 5551200 to 5551300:
	  ✓ Prefix: **[5551200-5551300]#**
	  ✓ Suffix: **(5551200-5551300)**
	▪ To depict numbers from 123100 to 123200:
	  ✓ Prefix: **123[100-200]**
	  ✓ Suffix: **(123[100-200])**
	▪ To depict prefix and suffix numbers together:
	  ✓ 03(100): for any number that starts with 03 and ends with 100.
	  ✓ **[100-**199](100,101,105): for a number that starts with 100 to 199 and ends with 100, 101 or 105.
	  ✓ 03(abc): for any number that starts with 03 and ends with abc.

Notation	Description
	✓ 03(5xx): for any number that starts with 03 and ends with 5xx. ✓ 03(400,401,405): for any number that starts with 03 and ends with 400 or 401 or 405. **Notes:** ▪ The value *n* must be less than the value *m*. ▪ Only numerical ranges are supported (not alphabetical letters). ▪ For suffix ranges, the starting (*n*) and ending (*m*) numbers in the range must have the same number of digits. For example, (23-34) is correct, but (3-12) is not.
[n,m,...] or **(n,m,...)**	Represents multiple numbers. For example, to depict a one-digit number starting with 2, 3, 4, 5, or 6: ▪ Prefix: **[2,3,4,5,6]#** ▪ Suffix: **(2,3,4,5,6)** ▪ Prefix with Suffix: **[2,3,4,5,6](8,7,6)** - prefix is denoted in square brackets; suffix in parenthesis For **prefix only**, the notations *d[n,m]e* and *d[n-m]e* can also be used: ▪ To depict a five-digit number that starts with 11, 22, or 33: **[11,22,33]xxx#** ▪ To depict a six-digit number that starts with 111 or 222: **[111,222]xxx#**
[n1-m1,n2-m2,a,b,c,n3-m3] or **(n1-m1,n2-m2,a,b,c,n3-m3)**	Represents a mixed notation of single numbers and multiple ranges. For example, to depict numbers 123 to 130, 455, 766, and 780 to 790: ▪ Prefix: **[123-130,455,766,780-790]** ▪ Suffix: **(123-130,455,766,780-790)** **Note:** The ranges and the single numbers used in the dial plan must have the same number of digits. For example, each number range and single number in the dialing plan example above consists of three digits.

> **Note:** When configuring phone numbers or prefixes in the Web interface, enter them only as digits without any other characters. For example, if you wish to enter the phone number 555-1212, it must be entered as 5551212 without the hyphen (-). If the hyphen is entered, the entry is invalid.

43   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.

43.1   Networking Parameters

This subsection describes the device's networking parameters.

43.1.1   Ethernet Parameters

The Ethernet parameters are described in the table below.

Table 43-1: Ethernet Parameters

Parameter	Description
OVOC: Physical Configuration **[EthernetPhyConfiguration]**	Defines the Ethernet connection mode type. ▪ **[0]** = 10Base-T half-duplex (Not applicable) ▪ **[1]** = 10Base-T full-duplex ▪ **[2]** = 100Base-TX half-duplex ▪ **[3]** = 100Base-TX full-duplex ▪ **[4]** = (Default) Auto-negotiate **Note:** For this parameter to take effect, a device reset is required.

43.1.2   Multiple VoIP Network Interfaces and VLAN Parameters

The IP network interfaces and VLAN parameters are described in the table below.

Table 43-2: IP Network Interfaces and VLAN Parameters

Parameter	Description
Multiple Interface Table	
Web: Multiple Interface Table OVOC: IP Interface Settings **[InterfaceTable]**	This table parameter configures the Multiple Interface table. The format of this parameter is as follows: [InterfaceTable] FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes, InterfaceTable_InterfaceMode, InterfaceTable_IPAddress, InterfaceTable_PrefixLength, InterfaceTable_Gateway, InterfaceTable_VlanID, InterfaceTable_InterfaceName, InterfaceTable_PrimaryDNSServerIPAddress, InterfaceTable_SecondaryDNSServerIPAddress, InterfaceTable_UnderlyingInterface; [\InterfaceTable]

Parameter	Description
	For example: InterfaceTable 0 = 0, 0, 192.168.85.14, 16, 0.0.0.0, 1, Management; InterfaceTable 1 = 2, 0, 200.200.85.14, 24, 0.0.0.0, 200, Control; InterfaceTable 2 = 1, 0, 211.211.85.14, 24, 211.211.85.1, 211, Media; The above example, configures three network interfaces (OAMP, Control, and Media). **Notes:** - For this parameter to take effect, a device reset is required. - For a description of this parameter, see 'Configuring IP Network Interfaces' on page 116.
Single IP Network Parameters	
Web: IP Address OVOC: Local IP Address **[LocalOAMIPAddress]**	Defines the device's source IP address of the operations, administration, maintenance, and provisioning (OAMP) interface when operating in a single interface scenario without a Multiple Interface table. The default is 0.0.0.0. **Note:** For this parameter to take effect, a device reset is required.
Web: Subnet Mask OVOC: OAM Subnet Mask **[LocalOAMSubnetMask]**	Defines the device's subnet mask of the OAMP interface when operating in a single interface scenario without a Multiple Interface table. The default subnet mask is 0.0.0.0. **Note:** For this parameter to take effect, a device reset is required.
Web: Default Gateway Address OVOC: Local Def GW **[LocalOAMDefaultGW]**	Defines the Default Gateway of the OAMP interface when operating in a single interface scenario without a Multiple Interface table.
VLAN Parameters	
Web/OVOC: VLAN Mode **[VLANMode]**	Enables VLANs tagging (IEEE 802.1Q). **[0]** Disable (default) **[1]** Enable **Notes:** - For this parameter to take effect, a device reset is required. - To operate with multiple network interfaces, VLANs must be enabled. - VLANs are available only when booting the device from flash. When booting using BootP/DHCP protocols, VLANs are disabled to allow easier maintenance access. In this scenario, multiple network interface capabilities are unavailable.
Web/OVOC: Native VLAN ID **[VLANNativeVLANID]**	Defines the Native VLAN ID. This is the VLAN ID to which untagged incoming traffic is assigned. Outgoing packets sent to this VLAN are sent only with a priority tag (VLAN ID = 0). When the Native VLAN ID is equal to one of the VLAN IDs listed in the Multiple Interface table (and VLANs are enabled), untagged incoming traffic is considered as incoming traffic for

Parameter	Description
	that interface. Outgoing traffic sent from this interface is sent with the priority tag (tagged with VLAN ID = 0).
	When the Native VLAN ID is different to any value in the 'VLAN ID' column in the table, untagged incoming traffic is discarded and all outgoing traffic is tagged.
	The default Native VLAN ID is 1.
	Note: If this parameter is not configured (i.e., default is 1) and one of the interfaces has a VLAN ID set to 1, this interface is still considered the 'Native' VLAN. If you do not wish to have a 'Native' VLAN ID and want to use VLAN ID 1, set this parameter to a value other than any VLAN ID in the table.
[EnableNTPasOAM]	Defines the application type for Network Time Protocol (NTP) services. ▪ **[1]** = OAMP (default) ▪ **[0]** = Control **Note:** For this parameter to take effect, a device reset is required.
[VLANSendNonTaggedOnNative]	Determines whether to send non-tagged packets on the native VLAN. ▪ **[0]** = (Default) Sends priority tag packets. ▪ **[1]** = Sends regular packets (with no VLAN tag). **Note:** For this parameter to take effect, a device reset is required.

43.1.3   Routing Parameters

The IP network routing parameters are described in the table below.

Table 43-3: IP Network Routing Parameters

Parameter	Description
Web: Disable ICMP Redirects **[DisableICMPRedirects]**	Determines whether the device accepts or ignores ICMP Redirect messages. ▪ **[0]** Disable = (Default) ICMP Redirect messages are handled by the device. ▪ **[1]** Enable = ICMP Redirect messages are ignored.
Static IP Routing Table	
Web/OVOC: IP Routing Table **[StaticRouteTable]**	Defines up to 30 static IP routing rules for the device. These rules can be associated with IP interfaces defined in the Multiple Interface table (InterfaceTable parameter). The routing decision for sending the outgoing IP packet is based on the source subnet/VLAN. If not associated with an IP interface, the static IP rule is based on destination IP address.
	When the destination of an outgoing IP packet does not match one of the subnets defined in the Multiple Interface table, the device searches this table for an entry that matches the requested destination host/network. If such an entry is found, the device sends the packet to the indicated router (i.e., next hop). If no explicit entry is found, the

Parameter	Description
	packet is sent to the default gateway according to the source interface of the packet (if defined). The format of this parameter is as follows: [StaticRouteTable] FORMAT StaticRouteTable_Index = StaticRouteTable_InterfaceName, StaticRouteTable_Destination, StaticRouteTable_PrefixLength, StaticRouteTable_Gateway, StaticRouteTable_Description; [\StaticRouteTable] **Note:** For a description of this parameter, see 'Configuring Static IP Routing' on page 127.

43.1.4   Quality of Service Parameters

The Quality of Service (QoS) parameters are described in the table below.

Table 43-4: QoS Parameters

Parameter	Description
Layer-2 Class Of Service (CoS) Parameters (VLAN Tag Priority Field)	
Web: Network Priority OVOC: Network Service Class Priority **[VLANNetworkServiceClassPriority]**	Defines the VLAN priority (IEEE 802.1p) for Network Class of Service (CoS) content. The valid range is 0 to 7. The default is 7.
Web: Media Premium OVOC: Premium Service Class Media Priority Priority **[VLANPremiumServiceClassMediaPriority]**	Defines the VLAN priority (IEEE 802.1p) for the Premium CoS content and media traffic. The valid range is 0 to 7. The default is 6.
Web: Control Premium Priority OVOC: Premium Service Class Control Priority **[VLANPremiumServiceClassControlPriority]**	Defines the VLAN priority (IEEE 802.1p) for the Premium CoS content and control traffic. The valid range is 0 to 7. The default is 6.
Web: Gold Priority OVOC: Gold Service Class Priority **[VlanGoldServiceClassPriority]**	Defines the VLAN priority (IEEE 802.1p) for the Gold CoS content. The valid range is 0 to 7. The default is 4.
Web: Bronze Priority OVOC: Bronze Service Class Priority **[VLANBronzeServiceClassPriority]**	Defines the VLAN priority (IEEE 802.1p) for the Bronze CoS content. The valid range is 0 to 7. The default is 2.
Layer-3 Class of Service (TOS/DiffServ) Parameters	
Web: Network QoS OVOC: Network Service Class Diff Serv **[NetworkServiceClassDiffServ]**	Defines the Differentiated Services (DiffServ) value for Network CoS content. The valid range is 0 to 63. The default is 48. **Note:** For this parameter to take effect, a device reset is required.
Web: Media Premium QoS OVOC: Premium Service Class Media Diff Serv **[PremiumServiceClassMediaDiffServ]**	Defines the DiffServ value for Premium Media CoS content (only if IPDiffServ is not set in the selected IP Profile). The valid range is 0 to 63. The default is 46.

Parameter	Description
	Note: The value for the Premium Control DiffServ is determined by the following (according to priority): - IPDiffServ value in the selected IP Profile (IPProfile parameter). - PremiumServiceClassMediaDiffServ.
Web: Control Premium QoS OVOC: Premium Service Class Control Diff Serv **[PremiumServiceClassControlDiffServ]**	Defines the DiffServ value for Premium Control CoS content (Call Control applications) - only if ControlIPDiffserv is not set in the selected IP Profile. The valid range is 0 to 63. The default is 40. **Notes:** - The value for the Premium Control DiffServ is determined by the following (according to priority): ✓ SigIPDiffserv value in the selected IP Profile (IPProfile parameter). ✓ PremiumServiceClassControlDiffServ. - The same value must be configured for this parameter and the parameter MLPPDiffServ. - Outgoing calls are tagged according to this parameter.
Web: Gold QoS OVOC: Gold Service Class Diff Serv **[GoldServiceClassDiffServ]**	Defines the DiffServ value for the Gold CoS content (Streaming applications). The valid range is 0 to 63. The default is 26.
Web: Bronze QoS OVOC: Bronze Service Class Diff Serv **[BronzeServiceClassDiffServ]**	Defines the DiffServ value for the Bronze CoS content (OAMP applications). The valid range is 0 to 63. The default is 10.

43.1.5   NAT and STUN Parameters

The Network Address Translation (NAT) and Simple Traversal of UDP through NAT (STUN) parameters are described in the table below.

Table 43-5: NAT and STUN Parameters

Parameter	Description
STUN Parameters	
Web: Enable STUN OVOC: STUN Enable **[EnableSTUN]**	Enables Simple Traversal of UDP through NATs (STUN). **[0]** Disable (default) **[1]** Enable When enabled, the device functions as a STUN client and communicates with a STUN server located in the public Internet. STUN is used to discover whether the device is located behind a NAT and the type of NAT. It is also used to determine the IP addresses and port numbers that the NAT assigns to outgoing signaling messages (using SIP) and media streams (using RTP, RTCP and T.38). STUN works with many existing NAT types and does not require any special behavior from them.

Parameter	Description
	Notes:
	▪ For this parameter to take effect, a device reset is required.
	▪ For defining the STUN server domain name, use the parameter STUNServerDomainName.
	▪ For more information on STUN, see Configuring STUN on page 140.
Web: STUN Server Primary IP OVOC: Primary Server IP **[STUNServerPrimaryIP]**	Defines the IP address of the primary STUN server. The valid range is the legal IP addresses. The default is 0.0.0.0. **Note:** For this parameter to take effect, a device reset is required.
Web: STUN Server Secondary IP OVOC: Secondary Server IP **[STUNServerSecondaryIP]**	Defines the IP address of the secondary STUN server. The valid range is the legal IP addresses. The default is 0.0.0.0. **Note:** For this parameter to take effect, a device reset is required.
[STUNServerDomainName]	Defines the domain name for the Simple Traversal of User Datagram Protocol (STUN) server's address (used for retrieving all STUN servers with an SRV query). The STUN client can perform the required SRV query to resolve this domain name to an IP address and port, sort the server list, and use the servers according to the sorted list. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ Use either the STUNServerPrimaryIP or the STUNServerDomainName parameter, with priority to the first one.
NAT Parameters	
NAT Mode disable-NAT-traversal [NATMode]	Enables the NAT feature for media when the device communicates with UAs located behind NAT. ▪ **[0]** Auto-Detect = NAT is performed only if necessary. If the UA is identified as being located behind NAT, the device sends the media packets to the public IP address:port obtained from the source address of the first media packet received from the UA. Otherwise, the packets are sent using the IP address:port obtained from the address in the first received SIP message. Note that if the SIP session is established (ACK) and the device (not the UA) sends the first packet, it sends it to the address obtained from the SIP message and only after the device receives the first packet from the UA, does it determine whether the UA is behind NAT. ▪ **[1]** NAT Is Not Used = (Default) NAT feature is disabled. The device always sends the media packets to the remote UA using the IP address:port obtained from the first received SIP message. ▪ **[2]** NAT Is Used = NAT is always performed. The device always sends the media packets to the remote UA using the source address obtained from the first media packet from the UA. In this mode, the device does not send any packets

Parameter	Description
	until it receives the first packet from the UA (in order to obtain the IP address).
Web: NAT IP Address OVOC: Static NAT IP Address **[StaticNatIP]**	Defines the global (public) IP address of the device to enable static NAT between the device and the Internet. **Note:** For this parameter to take effect, a device reset is required.
OVOC: Binding Life Time **[NATBindingDefaultTimeout]**	Defines the default NAT binding lifetime in seconds. STUN refreshes the binding information after this time expires. The valid range is 0 to 2,592,000. The default is 30. **Note:** For this parameter to take effect, a device reset is required.
[EnableIPAddrTranslation]	Enables IP address translation for RTP, RTCP, and T.38 packets. ▪ **[0]** = Disable IP address translation. ▪ **[1]** = (Default) Enable IP address translation. ▪ **[2]** = Enable IP address translation for RTP Multiplexing (ThroughPacket™). ▪ **[3]** = Enable IP address translation for all protocols (RTP, RTCP, T.38 and RTP Multiplexing). When enabled, the device compares the source IP address of the first incoming packet to the remote IP address stated in the opening of the channel. If the two IP addresses don't match, the NAT mechanism is activated. Consequently, the remote IP address of the outgoing stream is replaced by the source IP address of the first incoming packet. **Notes:** ▪ The NAT mechanism must be enabled for this parameter to take effect (parameter NATMode). ▪ For information on RTP Multiplexing, see RTP Multiplexing (ThroughPacket).
[EnableUDPPortTranslation]	Enables UDP port translation. ▪ **[0]** = (Default) Disables UDP port translation. ▪ **[1]** = Enables UDP port translation. The device compares the source UDP port of the first incoming packet to the remote UDP port stated in the opening of the channel. If the two UDP ports don't match, the NAT mechanism is activated. Consequently, the remote UDP port of the outgoing stream is replaced by the source UDP port of the first incoming packet. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The NAT mechanism and the IP address translation must be enabled for this parameter to take effect (i.e., parameter NATMode and the parameter EnableIpAddrTranslation to 1).

43.1.6 NFS Parameters

The Network File Systems (NFS) configuration parameters are described in the table below.

Table 43-6: NFS Parameters

Parameter	Description
[NFSBasePort]	Defines the start of the range of numbers used for local UDP ports used by the NFS client. The maximum number of local ports is maximum channels plus maximum NFS servers. The valid range is 0 to 65535. The default is 47000.
NFS Table	
Web: NFS Table OVOC: NFS Settings **[NFSServers]**	This table parameter defines up to 16 NFS file systems so that the device can access a remote server's shared files and directories for loading cmp, ini, and auxiliary files (using the Automatic Update mechanism). The format of this table ini file parameter is as follows: [NFSServers] FORMAT NFSServers_Index = NFSServers_HostOrIP, NFSServers_RootPath, NFSServers_NfsVersion, NFSServers_AuthType, NFSServers_UID, NFSServers_GID, NFSServers_VlanType; [\NFSServers] For example: NFSServers 1 = 101.1.13, /audio1, 3, 1, 0, 1, 1; **Note:** For a detailed description of this table, see 'Configuring NFS Settings' on page 137.

43.1.7 DNS Parameters

The Domain name System (DNS) parameters are described in the table below.

Table 43-7: DNS Parameters

Parameter	Description
Internal DNS Table	
Web: Internal DNS Table OVOC: DNS Information **[DNS2IP]**	This table parameter defines the internal DNS table for resolving host names into IP addresses. Up to four different IP addresses (in dotted-decimal notation) can be assigned to a host name. The format of this parameter is as follows: [Dns2Ip] FORMAT Dns2Ip_Index = Dns2Ip_DomainName, Dns2Ip_FirstIpAddress, Dns2Ip_SecondIpAddress, Dns2Ip_ThirdIpAddress, Dns2Ip_FourthIpAddress; [\Dns2Ip] For example: Dns2Ip 0 = DnsName, 1.1.1.1, 2.2.2.2, 3.3.3.3, 4.4.4.4; **Note:** For a detailed description of this table parameter, see 'Configuring the Internal DNS Table' on page 134.
Internal SRV Table	

Parameter	Description
Web: Internal SRV Table OVOC: DNS Information **[SRV2IP]**	This table parameter defines the internal SRV table for resolving host names into DNS A-Records. Three different A-Records can be assigned to a host name. Each A-Record contains the host name, priority, weight, and port. The format of this parameter is as follows: [SRV2IP] FORMAT SRV2IP_Index = SRV2IP_InternalDomain, SRV2IP_TransportType, SRV2IP_Dns1, SRV2IP_Priority1, SRV2IP_Weight1, SRV2IP_Port1, SRV2IP_Dns2, SRV2IP_Priority2, SRV2IP_Weight2, SRV2IP_Port2, SRV2IP_Dns3, SRV2IP_Priority3, SRV2IP_Weight3, SRV2IP_Port3; [\SRV2IP] For example: SRV2IP 0 = SrvDomain,0,Dnsname1,1,1,500,Dnsname2,2,2,501,$$,0,0,0; **Note**: For a detailed description of this table parameter, see 'Configuring the Internal SRV Table' on page 135.

43.1.8   DHCP and LLDP Parameters

The Dynamic Host Control Protocol (DHCP) and LLDP parameters are described in the table below.

Table 43-8: DHCP Parameters

Parameter	Description
LLDP Parameters	
[EnableLLDP]	Enables the device to use the discovery protocol, Link Layer Discovery Protocol (LLDP) to obtain (over the Layer-2 data link layer) a VLAN ID for its OAMP interface (per IEEE 802.1, IEEE 802.3 and TR-41) upon device startup (reset or power up). ▪ [0] = (Default) Disabled ▪ [1] = Enabled For more information on LLDP, see Section VLAN ID Discovery using LLDP on page 387.
DHCP Parameters	
Web: Enable DHCP OVOC: DHCP Enable **[DHCPEnable]**	Enables Dynamic Host Control Protocol (DHCP) functionality. ▪ **[0]** Disable (default) ▪ **[1]** Enable After the device powers up, it attempts to communicate with a BootP server. If a BootP server does not respond and DHCP is enabled, then the device attempts to obtain its IP address and other networking parameters from the DHCP server. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ After you enable the DHCP server, do the following: **a.** Enable DHCP and save the configuration.

Parameter	Description
	b. Perform a cold reset using the device's hardware reset button (soft reset using the Web interface doesn't trigger the BootP/DHCP procedure and this parameter reverts to 'Disable'). ▪ Throughout the DHCP procedure, the BootP/TFTP application must be deactivated; otherwise the device receives a response from the BootP server instead of from the DHCP server. ▪ This parameter is a special 'Hidden' parameter. Once defined and saved in flash memory, its assigned value doesn't revert to its default even if the parameter doesn't appear in the *ini* file.
[DhcpOption160Support]	Enables the use of DHCP Option 160. ▪ [0] = (Default) Disable ▪ [1] = Enable For more information, see "Provisioning the Device using DHCP Option 160" on page 390. **Note:** For the parameter to take effect, a device reset is required.
OVOC: DHCP Speed Factor **[DHCPSpeedFactor]**	Defines the DHCP renewal speed. ▪ **[0]** = Disable ▪ **[1]** = (Default) Normal ▪ **[2]** to **[10]** = Fast When set to 0, the DHCP lease renewal is disabled. Otherwise, the renewal time is divided by this factor. Some DHCP-enabled routers perform better when set to 4. **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
[DHCPRequestTFTPParams]	Determines whether the device includes DHCP options 66 and 67 in DHCP Option 55 (Parameter Request List) for requesting the DHCP server for TFTP provisioning parameters. ▪ **[0]** = (Default) Disable ▪ **[1]** = Enable **Note:** For this parameter to take effect, a device reset is required.
[DhcpHasLocalIpAddrDuringInit]	Defines whether the device includes DHCP Option 50 and the local IP address to the initial DHCP request upon device restart. ▪ [0] = After the device restarts, it first sends a DHCP Discover packet without DHCP Option 50 to the DHCP server. In addition, it sends outgoing DHCP Request packets with a local IP address of zero value (0.0.0.0), before it gets a new IP address from the DHCP server. The device adds Option 50 to the DHCP Request packet only after it receives a DHCP Offer (with the offered IP address) from the DHCP sever. The maximum timeout between sent DHCP Discover and receiving a DHCP Ack (with IP address) from the server is 120 sec. ▪ [1] = (Default) After the device restarts, it first sends a DHCP Request with DHCP Option 50 to the DHCP server. In addition, it sends outgoing DHCP Request packets with a non-zero value as its' local IP address. The maximum timeout between sent DHCP requests and receiving a DHCP Ack from the server is 60 sec. **Note:** For this parameter to take effect, a device reset is required.

43.1.9   NTP and Daylight Saving Time Parameters

The Network Time Protocol (NTP) and daylight saving time parameters are described in the table below.

Table 43-9: NTP and Daylight Saving Time Parameters

Parameter	Description
NTP Parameters **Note:** For more information on Network Time Protocol (NTP), see 'Simple Network Time Protocol Support' on page 111.	
Web: NTP Server DN/IP OVOC: Server IP Address **[NTPServerIP]**	Defines the IP address (in dotted-decimal notation or as an FQDN) of the NTP server. The advantage of using an FQDN is that multiple IP addresses can be resolved from the DNS server, providing NTP server redundancy. The default IP address is 0.0.0.0 (i.e., internal NTP client is disabled).
Web: NTP Secondary Server IP **[NTPSecondaryServerIP]**	Defines a second NTP server's address as an FQDN or an IP address (in dotted-decimal notation). This NTP is used for redundancy; if the primary NTP server fails, then this NTP server is used.

Parameter	Description
	The default IP address is 0.0.0.0.
Web: NTP UTC Offset OVOC: UTC Offset **[NTPServerUTCOffset]**	Defines the Universal Time Coordinate (UTC) offset (in seconds) from the NTP server. The default offset is 0. The offset range is -43200 to 43200.
Web: NTP Update Interval OVOC: Update Interval **[NTPUpdateInterval]**	Defines the time interval (in seconds) that the NTP client requests for a time update. The default interval is 86400 (i.e., 24 hours). The range is 0 to 214783647. **Note:** It is not recommend to set this parameter to beyond one month (i.e., 2592000 seconds).
Daylight Saving Time Parameters	
Web: Day Light Saving Time OVOC: Mode **[DayLightSavingTimeEnable]**	Enables daylight saving time. ▪ **[0]** Disable (default) ▪ **[1]** Enable

Web: Start Time or Day of Month Start
OVOC: Start
[DayLightSavingTimeStart]
	Defines the date and time when daylight saving begins. This value can be configured using any of the following formats:
	▪ Day of year - *mm:dd:hh:mm*, where:
	  ✔ *mm* denotes month
	  ✔ *dd* denotes date of the month
	  ✔ *hh* denotes hour
	  ✔ *mm* denotes minutes
	For example, "05:01:08:00" denotes daylight saving starting from May 1 at 8 A.M.
	▪ Day of month - *mm:day/wk:hh:mm*, where:
	  ✔ *mm* denotes month (e.g., 04)
	  ✔ *day* denotes day of week (e.g., FRI)
	  ✔ *wk* denotes week of the month (e.g., 03)
	  ✔ *hh* denotes hour (e.g., 23)
	  ✔ *mm* denotes minutes (e.g., 10)
	For example, "04:FRI/03:23:00" denotes Friday, the third week of April, at 11 P.M. The week field can be 1-5, where 5 denotes the last occurrence of the specified day in the specified month. For example, "04:FRI/05:23:00" denotes the last Friday of April, at 11 P.M.

Web: End Time or Day of Month End OVOC: End **[DayLightSavingTimeEnd]**	Defines the date and time when daylight saving ends. For a description of the format of this value, see the DayLightSavingTimeStart parameter.
Web/OVOC: Offset **[DayLightSavingTimeOffset]**	Defines the daylight saving time offset (in minutes). The valid range is 0 to 120. The default is 60.

43.2   Management Parameters

This subsection describes the device's Web and Telnet parameters.

43.2.1   General Parameters

The general management parameters are described in the table below.

Table 43-10: General Management Parameters

Parameter	Description
Web: Web and Telnet Access List Table OVOC: Web Access Addresses **[WebAccessList_x]**	This table configures up to ten IP addresses that are permitted to access the device's Web interface and Telnet interfaces. Access from an undefined IP address is denied. When no IP addresses are defined in this table, this security feature is inactive (i.e., the device can be accessed from any IP address). The default is 0.0.0.0 (i.e., the device can be accessed from any IP address). For example: WebAccessList_0 = 10.13.2.66 WebAccessList_1 = 10.13.77.7 For a description of this parameter, see 'Configuring Web and Telnet Access List' on page 75.
[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., 1S3p+fno=). ▪ [1] = Enable. All passwords are hidden and replaced by an asterisk (*).

43.2.2   Web Parameters

The Web parameters are described in the table below.

Table 43-11: Web Parameters

Parameter	Description
Web: Password Change Interval **[WebUserPassChangeInterval]**	Defines the duration (in minutes) of the validity of Web login passwords. When this duration expires, the password of the Web user must be changed. The valid value is 0 to 100000, where 0 means that the password is always valid. The default is 1140. **Note:** This parameter is applicable only when using the Web Users table, where the default value of the 'Password Age' parameter in the Web Users table inherits this parameter's value.
Web: User inactivity timer **[UserInactivityTimer]**	Defines the duration (in days) for which a user has not logged in to the Web interface, after which the status of the user becomes inactive and can no longer access the Web interface. These users can only log in to the Web interface if their status

Parameter	Description
	is changed (to New or Valid) by a System Administrator or Master user. The valid value is 0 to 10000, where 0 means inactive. The default is 90. **Note:** This parameter is applicable only when using the Web Users table.
Web: Session Timeout **[WebSessionTimeout]**	Defines the duration (in minutes) of Web inactivity of a logged-in user, after which the user is automatically logged off the Web interface. The valid value is 0-100000, where 0 means no timeout. The default is 15. **Note:** This parameter can apply to all users, or per user when set in the Web Users table.
Web: Deny Access On Fail Count **[DenyAccessOnFailCount]**	Defines the maximum number of failed login attempts, after which the requesting IP address is blocked. The valid value range is 0 to 10. The values 0 and 1 mean immediate block. The default is 3.
Web: Deny Authentication Timer OVOC: WEB Deny Authentication Timer **[DenyAuthenticationTimer]**	Defines the duration (in seconds) for which login to the Web interface is denied from a specific IP address (for all users) when the number of failed login attempts has exceeded the maximum. This maximum is defined by the DenyAccessOnFailCount parameter. Only after this time expires can users attempt to login from this same IP address. The valid value is 0 to 100000, where 0 means that login is not denied regardless of number of failed login attempts. The default is 60.
Web: Display Login Information **[DisplayLoginInformation]**	Enables display of user's login information on each successful login attempt. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
[EnableMgmtTwoFactorAuthentication]	Enables Web login authentication using a third-party, smart card. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable When enabled, the device retrieves the Web user's login username from the smart card, which is automatically displayed (read-only) in the Web Login screen; the user is then required to provide only the login password. Typically, a TLS connection is established between the smart card and the device's Web interface, and a RADIUS server is implemented to authenticate the password with the username. Thus, this feature implements a two-factor authentication - what the user has (the physical card) and what the user knows (i.e., the login password).
OVOC: HTTPS Port **[HTTPport]**	Defines the LAN HTTP port for Web management (default is 80). To enable Web management from the LAN, configure the desired port. **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
OVOC: Disable WEB Config **[DisableWebConfig]**	Determines whether the entire Web interface is read-only. ▪ **[0]** = (Default) Enables modifications of parameters. ▪ **[1]** = Web interface is read-only. When in read-only mode, parameters can't be modified. In addition, the following pages can't be accessed: 'Web User Accounts', 'Certificates', 'Regional Settings', 'Maintenance Actions' and all file-loading pages ('Load Auxiliary Files', 'Software Upgrade Wizard', and 'Configuration File'). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ To return to read/write after you have applied read-only using this parameter (set to 1), you need to reboot your device with an ini file that doesn't include this parameter, using the AcBootP utility.

[ResetWebPassword]	Enables the <device> to restore the default management users:
	
	▪ Security Administrator user (username "Admin"; password "Admin")
	▪ Monitor user (username "User"; password "User")
	
	In addition, all other users that may have been configured (in the Web Users table) are deleted.
	
	▪ [0] = (Default) Disabled. Currently configured users (usernames and passwords) are retained.
	▪ [1] = Enabled. Default users are restored (see description above) and all other configured users are deleted.
	
	Notes:
	▪ For the parameter to take effect, a device reset is required.
	▪ In addition to the ini file (see above), you can also restore the default user accounts through the following management platforms:
	  ✓ SNMP (restores default users and retains other configured users:
	    1) Set acSysGenericINILine to WEBPasswordControlViaSNMP = 1, and reset the device with a flash burn (set acSysActionSetResetControl to 1 and acSysActionSetReset to 1).
	    2) Change the username and password in the acSysWEBAccessEntry table. Use the following format:
	      Username acSysWEBAccessUserName: old/pass/new
	      Password acSysWEBAccessUserCode: username/old/new

[ScenarioFileName]	Defines the file name of the Scenario file to be loaded to the device. The file name must have the .dat extension and can be up to 47 characters. For loading a Scenario using the Web interface, see Loading a Scenario to the Device on page 57.

Parameter	Description
[WelcomeMessage]	Enables and defines a Welcome message that appears on the Web Login page for logging in to the Web interface. The format of this parameter is as follows: [WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text [\WelcomeMessage] For Example: FORMAT WelcomeMessage_Index = WelcomeMessage_Text WelcomeMessage 1 = "**********************************" ; WelcomeMessage 2 = "********* This is a Welcome message ***" ; WelcomeMessage 3 = "**********************************" ; **Notes:** ▪ Each index row represents a line of text in the Welcome message box. Up to 20 lines (or rows) of text can be defined. ▪ The configured text message must be enclosed in double quotation marks (i.e., "..."). ▪ If this parameter is not configured, no Welcome message is displayed.
Local Users Table	
Local Users `configure system > create-users-table` [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] For more information, see Advanced User Accounts Configuration on page 69.

43.2.3 Telnet Parameters

The Telnet parameters are described in the table below.

Table 43-12: Telnet Parameters

Parameter	Description
Web: Embedded Telnet Server OVOC: Server Enable **[TelnetServerEnable]**	Enables the device's embedded Telnet server. Telnet is disabled by default for security. ▪ **[0]** Disable (default) ▪ **[1]** Enable Unsecured ▪ **[2]** Enable Secured (SSL) **Note:** Only the primary Web User Account (which has Security Administration access level) can access the device using Telnet (see 'Configuring Web User Accounts' on page 66).

Parameter	Description
Web: Telnet Server TCP Port OVOC: Server Port **[TelnetServerPort]**	Defines the port number for the embedded Telnet server. The valid range is all valid port numbers. The default port is 23.
Web: Telnet Server Idle Timeout OVOC: Server Idle Disconnect **[TelnetServerIdleDisconnect]**	Defines the timeout (in minutes) for disconnection of an idle Telnet session. When set to zero, idle sessions are not disconnected. The valid range is any value. The default is 0. **Note:** For this parameter to take effect, a device reset is required.

43.2.4   SNMP Parameters

The SNMP parameters are described in the table below.

Table 43-13: SNMP Parameters

Parameter	Description
Web: Enable SNMP **[DisableSNMP]**	Enables SNMP. ▪ **[0]** Enable = (Default) SNMP is enabled. ▪ **[1]** Disable = SNMP is disabled and no traps are sent. **Note:** For this parameter to take effect, a device reset is required.
[SNMPPort]	Defines the device's local (LAN) UDP port used for SNMP Get/Set commands. The range is 100 to 3999. The default port is 161. **Note:** For this parameter to take effect, a device reset is required.
OVOC: Keep Alive Trap Port **[KeepAliveTrapPort]**	Defines the port to which keep-alive traps are sent. The valid range is 0 - 65534. The default is port 162.
[SendKeepAliveTrap]	Enables keep-alive traps and sends them every 9/10 of the time as defined by the NATBindingDefaultTimeout parameter. ▪ **[0]** = Disable ▪ **[1]** = Enable **Note:** For this parameter to take effect, a device reset is required.
[SNMPSysOid]	Defines the base product system OID. The default is eSNMP_AC_PRODUCT_BASE_OID_D. **Note:** For this parameter to take effect, a device reset is required.
[SNMPTrapEnterpriseOid]	Defines the Trap Enterprise OID. The default is eSNMP_AC_ENTERPRISE_OID. The inner shift of the trap in the AcTrap subtree is added to the end of the OID in this parameter. **Note:** For this parameter to take effect, a device reset is required.
[acUserInputAlarmDescription]	Defines the description of the input alarm.

Parameter	Description
[acUserInputAlarmSeverity]	Defines the severity of the input alarm.
[AlarmHistoryTableMaxSize]	Defines the maximum number of rows in the Alarm History table. This parameter can be controlled by the Config Global Entry Limit MIB (located in the Notification Log MIB).
	The valid range is 50 to 100. The default is 100.
	Note: For this parameter to take effect, a device reset is required.
[ActiveAlarmTableMaxSize]	Defines the maximum number of currently active alarms that can be displayed in the Active Alarms table. When the table reaches this user-defined maximum capacity (i.e., full), the device sends the SNMP trap event, acActiveAlarmTableOverflow. If the table is full and a new alarm is raised by the <device>, the new alarm is not displayed in the table.
	The valid range is 15 (MP-11x) or 20 (MP-124) to 40 (MP-11x) or 100 (MP-124). The default is 20 (MP-11x) or 60 (MP-124).
	For more information on the Active Alarms table, see Viewing Active Alarms on page 409.
	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.
No Alarm For Disabled Port [NoAlarmForDisabledPort]	Enables the device to not send the SNMP trap acBoardControllerFailureAlarm, which indicates a "disabled" (non-configured) telephony port. A disabled port is one that is not configured at all or that is configured but without a Trunk Group ID (i.e., Trunk Group ID is 0), in the Endpoint Phone Number table.
	▪ [0] Disable = (Default) The device sends the SNMP trap for non-configured ports.
	▪ [1] Enable = The device does not send the SNMP trap for non-configured ports.
	Note: The parameter is applicable to all telephony (analog) port types.
[SNMPEngineIDString]	Defines the SNMP engine ID for SNMPv2/SNMPv3 agents. This is used for authenticating a user attempting to access the SNMP agent on the device.
	The ID can be a string of up to 36 characters. The default is 00:00:00:00:00:00:00:00:00:00:00:00 (12 Hex octets characters). The provided key must be set with 12 Hex values delimited by a colon (":") in the format xx:xx:...:xx. For example, 00:11:22:33:44:55:66:77:88:99:aa:bb
	Notes:
	▪ For this parameter to take effect, a device reset is required.
	▪ Before setting this parameter, all SNMPv3 users must be deleted; otherwise, the parameter setting is ignored.

Parameter	Description
	▪ If the supplied key does not pass validation of the 12 Hex values input or it is set with the default value, the engine ID is generated according to RFC 3411.
Web: SNMP Trap Destination Parameters OVOC: Network > SNMP Managers Table **Note:** Up to five SNMP trap managers can be defined.	
SNMP Manager **[SNMPManagerIsUsed_x]**	Determines the validity of the parameters (IP address and port number) of the corresponding SNMP Manager used to receive SNMP traps. ▪ **[0]** (Check box cleared) = Disabled (default) ▪ **[1]** (Check box selected) = Enabled
Web: IP Address OVOC: Address **[SNMPManagerTableIP_x]**	Defines the IP address of the remote host used as an SNMP Manager. The device sends SNMP traps to this IP address. Enter the IP address in dotted-decimal notation, e.g., 108.10.1.255.
Web: Trap Port OVOC: Port **[SNMPManagerTrapPort_x]**	Defines the port number of the remote SNMP Manager. The device sends SNMP traps to this port. The valid SNMP trap port range is 100 to 4000. The default port is 162.
Web: Trap Enable **[SNMPManagerTrapSendingEnable_x]**	Enables the sending of traps to the corresponding SNMP manager. ▪ **[0]** Disable = Sending is disabled. ▪ **[1]** Enable = (Default) Sending is enabled.
Web: Trap User **[SNMPManagerTrapUser_x]**	Defines the SNMPv3 USM user or SNMPv2 user to associate with the trap destination. This determines the trap format, authentication level, and encryption level. By default, it is associated with the SNMPv2 user (SNMP trap community string). The valid value is a string.
Web: Trap Manager Host Name **[SNMPTrapManagerHostName]**	Defines an FQDN of the remote host used as an SNMP manager. The resolved IP address replaces the last entry in the Trap Manager table (defined by the SNMPManagerTableIP parameter) and the last trap manager entry of snmpTargetAddrTable in the snmpTargetMIB. For example: 'mngr.corp.mycompany.com'. The valid range is a string of up to 99 characters.
SNMP Community String Parameters	
Community String **[SNMPReadOnlyCommunityString_x]**	Defines up to five read-only SNMP community strings (up to 19 characters each). The default string is 'public'.

Parameter	Description
Community String **[SNMPReadWriteCommunityString_x]**	Defines up to five read/write SNMP community strings (up to 19 characters each). The default string is 'private'.
Trap Community String **[SNMPTrapCommunityString]**	Defines the Community string used in traps (up to 19 characters). The default string is 'trapuser'.
SNMP Trusted Managers Table	
Web: SNMP Trusted Managers **[SNMPTrustedMgr_x]**	Defines up to five IP addresses of remote trusted SNMP managers from which the SNMP agent accepts and processes SNMP Get and Set requests. **Notes:** ▪ By default, the SNMP agent accepts SNMP Get and Set requests from any IP address, as long as the correct community string is used in the request. Security can be enhanced by using Trusted Managers, which is an IP address from which the SNMP agent accepts and processes SNMP requests. ▪ If no values are assigned to these parameters any manager can access the device. ▪ Trusted managers can work with all community strings.
SNMP V3 Users Table	
Web/OVOC: SNMP V3 Users **[SNMPUsers]**	This *para*meter table defines SNMP v3 users. The format of this parameter is as follows: [SNMPUsers] FORMAT SNMPUsers_Index = SNMPUsers_Username, SNMPUsers_AuthProtocol, SNMPUsers_PrivProtocol, SNMPUsers_AuthKey, SNMPUsers_PrivKey, SNMPUsers_Group; [\SNMPUsers] For example: SNMPUsers 1 = v3admin1, 1, 0, myauthkey, -, 1; The example above configures user 'v3admin1' with security level authNoPriv(2), authentication protocol MD5, authentication text password 'myauthkey', and ReadWriteGroup2. **Note:** For a description of this table, see 'Configuring SNMP V3 Users' on page 91.

43.2.5   Serial Parameters

The RS-232 serial parameters are described in the table below.

Table 43-14: Serial Parameters

Parameter	Description
[DisableRS232]	Enables the device's RS-232 (serial) port. ▪ **[0]** = (Default) Enabled ▪ **[1]** = Disabled

Parameter	Description
	The RS-232 serial port can be used to change the networking parameters and view error/notification messages. For how to establish a serial communication with the device, refer to the *Installation Manual*. **Note:** For this parameter to take effect, a device reset is required.
OVOC: Baud Rate **[SerialBaudRate]**	Defines the RS-232 baud rate. The valid values include the following: 1200, 2400, 9600 (default), 14400, 19200, 38400, 57600, or 115200. **Note:** For this parameter to take effect, a device reset is required.
OVOC: Data **[SerialData]**	Defines the RS-232 data bit. ▪ **[7]** = 7-bit ▪ **[8]** = (Default) 8-bit **Note:** For this parameter to take effect, a device reset is required.
OVOC: Parity **[SerialParity]**	Defines the RS-232 polarity. ▪ **[0]** = (Default) None ▪ **[1]** = Odd ▪ **[2]** = Even **Note:** For this parameter to take effect, a device reset is required.
OVOC: Stop **[SerialStop]**	Defines the RS-232 stop bit. ▪ **[1]** = (Default) 1-bit (default) ▪ **[2]** = 2-bit **Note:** For this parameter to take effect, a device reset is required.
OVOC: Flow Control **[SerialFlowControl]**	Defines the RS-232 flow control. ▪ **[0]** = (Default) None ▪ **[1]** = Hardware **Note:** For this parameter to take effect, a device reset is required.

43.3   Debugging and Diagnostics Parameters

This subsection describes the device's debugging and diagnostic parameters.

43.3.1   General Parameters

The general debugging and diagnostic parameters are described in the table below.

Table 43-15: General Debugging and Diagnostic Parameters

Parameter	Description
OVOC: Enable Diagnostics **[EnableDiagnostics]**	Determines the method for verifying correct functioning of the different hardware components on the device. On completion of the check and if the test fails, the device sends information on the test results of each hardware component to the Syslog server. ▪ **[0]** = (Default) Rapid and Enhanced self-test mode. ▪ **[1]** = Detailed self-test mode (full test of DSPs, PCM, Switch, LAN, PHY and Flash).

Parameter	Description
	▪ **[2]** = A quicker version of the Detailed self-test mode (full test of DSPs, PCM, Switch, LAN, PHY, but partial test of Flash). **Note:** For this parameter to take effect, a device reset is required.
Web: Enable LAN Watchdog **[EnableLanWatchDog]**	Enables the LAN watchdog feature. ▪ **[0]** Disable (default) ▪ **[1]** Enable When LAN watchdog is enabled, the device's overall communication integrity is checked periodically. If no communication is detected for about three minutes, the device performs a self test: ▪ If the self-test succeeds, the problem is a logical link down (i.e., Ethernet cable disconnected on the switch side) and the Busy Out mechanism is activated if enabled (i.e., the parameter EnableBusyOut is set to 1). Lifeline is activated only if it is enabled (using the parameter LifeLineType). ▪ If the self-test fails, the device restarts to overcome internal fatal communication error. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ Enable LAN watchdog is relevant only if the Ethernet connection is full duplex. ▪ LAN watchdog is not applicable to MP-118.

Parameter	Description
[LifeLineType]	Defines the condition(s) upon which the Lifeline analog (FXS) feature is activated. The Lifeline feature can be activated upon a power outage, physical disconnection of the LAN cable, or network failure (i.e., loss of IP connectivity). Upon any of these conditions, the Lifeline feature provides PSTN connectivity and thus call continuity for the FXS phone users.
	If the device is in Lifeline mode and the scenario that caused it to enter Lifeline (e.g., power outage) no longer exists (e.g., power returns), the device exists Lifeline and operates as normal.
	▪ [0] = (Default) Lifeline is activated upon power outage.
	▪ [1] = Lifeline is activated upon power outage or when the link is down (i.e., physically disconnected).
	▪ [2] = Lifeline is activated upon a power outage, when the link is down (physically disconnected), network failure (logical link disconnection), or when the Trunk Group is in Busy Out state (see the EnableBusyOut parameter).
	The Lifeline (FXS) phone is connected to the following port:
	▪ MP-11x FXS-only device: FXS Port 1
	▪ MP-118 FXS/FXO device: FXS Ports 1 to 4
	For the FXS-only device, FXS Port 1 connects to the POTS (Lifeline) phone as well as to the PSTN / PBX, using a splitter cable. For the combined FXS / FXO device, the FXS ports are provided with a lifeline by their corresponding FXO ports which are connected to the PSTN / PBX (i.e., FXO Port 5 provides a lifeline for FXS Port 1, FXO Port 6 provides a lifeline for FXS Port 2, and so on).
	Notes:
	▪ For this parameter to take effect, a device reset is required.
	▪ This parameter is applicable only to FXS interfaces.
	▪ For optional value [2], lifeline activation upon an IP network failure or Busy Out state is not supported by MP-118.
	▪ To enable Lifeline upon a network failure, the LAN watch dog must be activated (i.e., set the parameter EnableLANWatchDog to 1).
	▪ For information on Lifeline cabling, refer to the Installation Manual.
Web: Delay After Reset **[sec]** **[GWAppDelayTime]**	Defines the time interval (in seconds) that the device's operation is delayed after a reset.
	The valid range is 0 to 45. The default is 7 seconds.
	Note: This feature helps overcome connection problems caused by some LAN routers or IP configuration parameters' modifications by a DHCP server.
[EnableAutoRAITransmitBER]	Enables the device to send a remote alarm indication (RAI) when the bit error rate (BER) is greater than 0.001.
	▪ **[0]** Disable (default)
	▪ **[1]** Enable

43.3.2   SIP Test Call Parameters

The SIP Signaling Test Call parameters are described in the table below.

Table 43-16: SIP Test Call Parameters

Parameter	Description
Web: Test Call DTMF String **[TestCallDtmfString]**	Defines the DTMF tone that is played for answered test calls (incoming and outgoing). The DTMF string can be up to 15 strings. The default is "3212333". An empty string means that no DTMF is played.
Web: Test Call ID **[TestCallID]**	Defines the test call prefix number (*ID*) of the simulated phone on the device. Incoming calls received with this called prefix number are identified as test calls. This can be any string of up to 15 characters. By default, no number is defined. **Note:** This parameter is only for testing incoming calls destined to this prefix number.
Test Call Table	
Web: Test Call Table **[Test_Call]**	Defines the local and remote endpoints to be tested. FORMAT Test_Call_Index = Test_Call_EndpointURI, Test_Call_CalledURI, Test_Call_RouteBy, Test_Call_IPGroupID, Test_Call_DestAddress, Test_Call_DestTransportType, Test_Call_SRD, Test_Call_ApplicationType, Test_Call_AutoRegister, Test_Call_UserName, Test_Call_Password, Test_Call_CallParty, Test_Call_MaxChannels, Test_Call_CallDuration, Test_Call_CallsPerSecond, Test_Call_TestMode, Test_Call_TestDuration, Test_Call_Play, Test_Call_ScheduleInterval; **Note:** For a description of this table, see 'Configuring Test Calls' on page 457.

43.3.3   Syslog, CDR and Debug Parameters

The Syslog, CDR and debug parameters are described in the table below.

Table 43-17: Syslog, CDR and Debug Parameters

Parameter	Description
Web: Enable Syslog OVOC: Syslog enable **[EnableSyslog]**	Determines whether the device sends logs and error messages (e.g., CDRs) generated by the device to a Syslog server. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ If you enable Syslog, you must enter an IP address of the Syslog server (using the SyslogServerIP parameter). ▪ Syslog messages may increase the network traffic. ▪ To configure Syslog SIP message logging levels, use the GwDebugLevel parameter. ▪ By default, logs are also sent to the RS-232 serial port. For how to establish serial communication with the device, refer to the Installation Manual.

Parameter	Description
Web/OVOC: Syslog Server IP Address **[SyslogServerIP]**	Defines the IP address (in dotted-decimal notation) of the computer on which the Syslog server is running. The Syslog server is an application designed to collect the logs and error messages generated by the device. The default IP address is 0.0.0.0.
Web: Syslog Server Port OVOC: Syslog Server Port Number **[SyslogServerPort]**	Defines the UDP port of the Syslog server. The valid range is 0 to 65,535. The default port is 514.
[MaxBundleSyslogLength]	Defines the maximum size (in bytes) threshold of logged Syslog messages bundled into a single UDP packet, after which they are sent to a Syslog server. The valid value range is 0 to 1220 (where 0 indicates that no bundling occurs). The default is 1220. **Note:** This parameter is applicable only if the GWDebugLevel parameter is set to 7.
Web: CDR Server IP Address OVOC: IP Address of CDR Server **[CDRSyslogServerIP]**	Defines the destination IP address to where CDR logs are sent. The default is a null string, which causes CDR messages to be sent with all Syslog messages to the Syslog server. **Notes:** ▪ The CDR messages are sent to UDP port 514 (default Syslog port). ▪ This mechanism is active only when Syslog is enabled (i.e., the parameter EnableSyslog is set to 1).
Web/OVOC: CDR Report Level **[CDRReportLevel]**	Enables media- and signaling-related CDRs to be sent to a Syslog server and determines the call stage at which they are sent. ▪ **[0]** None = (Default) CDRs are not used. ▪ **[1]** End Call = CDR is sent to the Syslog server at the end of each call. ▪ **[2]** Start & End Call = CDR report is sent to Syslog at the start and end of each call. ▪ **[3]** Connect & End Call = CDR report is sent to Syslog at connection and at the end of each call. ▪ **[4]** Start & End & Connect Call = CDR report is sent to Syslog at the start, at connection, and at the end of each call. **Notes:** ▪ The CDR Syslog message complies with RFC 3161 and is identified by: Facility = 17 (local1) and Severity = 6 (Informational). ▪ This mechanism is active only when Syslog is enabled (i.e., the parameter EnableSyslog is set to 1).
Web/OVOC: Debug Level **[GwDebugLevel]**	Defines the Syslog debug logging level. ▪ **[0]** 0 = (Default) Debug is disabled. ▪ **[1]** 1 = Flow debugging is enabled. ▪ **[5]** 5 = Flow, device interface, stack interface, session manager, and device interface expanded debugging are enabled. ▪ **[7]** 7 = This option is recommended when the device is running under "heavy" traffic. In this mode:

Parameter	Description
	✓ The Syslog debug level automatically changes between level 5, level 1, and level 0, depending on the device's CPU consumption so that VoIP traffic isn't affected. ✓ Syslog messages are bundled into a single UDP packet, after which they are sent to a Syslog server (bundling size is determined by the MaxBundleSyslogLength parameter). Bundling reduces the number of UDP Syslog packets, thereby improving CPU utilization. Note that when this option is used, in order to read Syslog messages with Wireshark, a special plug-in (i.e., acsyslog.dll) must be used. Once the plug-in is installed, the Syslog messages are decoded as "AC SYSLOG" and are dispalyed using the 'acsyslog' filter instead of the regular 'syslog' filter. **Notes:** ▪ This parameter is typically set to 5 if debug traces are required. However, in cases of heavy traffic, option 7 is recommended. ▪ Options 2, 3, 4, and 6 are not recommended.
Web: Syslog Facility Number OVOC: SyslogFacility **[SyslogFacility]**	Defines the Facility level (0 through 7) of the device's Syslog messages, according to RFC 3164. This allows you to identify Syslog messages generated by the device. This is useful, for example, if you collect the device's and other equipments' Syslog messages, at one single server. The device's Syslog messages can easily be identified and distinguished from other Syslog messages by its Facility level. Therefore, in addition to filtering Syslog messages according to IP address, the messages can be filtered according to Facility level. ▪ **[16]** = (Default) local use 0 (local0) ▪ **[17]** = local use 1 (local1) ▪ **[18]** = local use 2 (local2) ▪ **[19]** = local use 3 (local3) ▪ **[20]** = local use 4 (local4) ▪ **[21]** = local use 5 (local5) ▪ **[22]** = local use 6 (local6) ▪ **[23]** = local use 7 (local7)
Web: Activity Types to Report via Activity Log Messages **[ActivityListToLog]**	Defines the Activity Log mechanism of the device, which sends log messages to a Syslog server for reporting certain types of Web operations according to the below user-defined filters. ▪ **[pvc]** Parameters Value Change = Changes made on-the-fly to parameters. Note that the *ini* file parameter, EnableParametersMonitoring can also be used to set this option, using values **[0]** (disable) or **[1]** (enable). ▪ **[afl]** Auxiliary Files Loading = Loading of auxiliary files. ▪ **[dr]** Device Reset = Reset of device via the 'Maintenance Actions page. **Note:** For this option to take effect, a device reset is required. ▪ **[fb]** Flash Memory Burning = Burning of files or parameters to flash (in 'Maintenance Actions page). ▪ **[swu]** Device Software Update = cmp file loading via the Software Upgrade Wizard. ▪ **[ard]** Access to Restricted Domains = Access to restricted domains, which include the following Web pages: ✓ (1) ini parameters (AdminPage) ✓ (2) General Security Settings

Parameter	Description
	✔ (3) Configuration File ✔ (4) IP Security Proposal / IP Security Associations Tables ✔ (5) Software Upgrade Key Status ✔ (6) Firewall Settings ✔ (7) Web & Telnet Access List ✔ (8) WEB User Accounts ▪ **[naa]** Non-Authorized Access = Attempt to access the Web interface with a false or empty user name or password. ▪ **[spc]** Sensitive Parameters Value Change = Changes made to sensitive parameters: ✔ (1) IP Address ✔ (2) Subnet Mask ✔ (3) Default Gateway IP Address ✔ (4) ActivityListToLog ▪ **[ll]** Login and Logout = Every login and logout attempt. For example: ActivityListToLog = 'pvc', 'afl', 'dr', 'fb', 'swu', 'ard', 'naa', 'spc' **Note:** For the *ini* file, values must be enclosed in single quotation marks.
Web: Debug Recording Destination IP **[DebugRecordingDestIP]**	Defines the IP address of the server for capturing debug recording.
Web: Debug Recording Destination Port **[DebugRecordingDestPort]**	Defines the UDP port of the server for capturing debug recording. The default is 925.
Debug Recording Status **[DebugRecordingStatus]**	Activates or de-activates debug recording. ▪ **[0]** Stop (default) ▪ **[1]** Start
Logging Filters Table	
Web: Logging Filters Table **[LoggingFilters]**	This table parameter defines logging filtering rules for Syslog messages and debug recordings. The format of this parameter is as follows: [LoggingFilters] FORMAT LoggingFilters_Index = LoggingFilters_Type, LoggingFilters_Value, LoggingFilters_Syslog, LoggingFilters_CaptureType; [\LoggingFilters] **Note:** For a detailed description of this table, see 'Filtering Syslog Messages and Debug Recordings' on page 445.

43.3.4   Resource Allocation Indication Parameters

The Resource Allocation Indication (RAI) parameters are described in the table below.

Table 43-18: RAI Parameters

Parameter	Description
[EnableRAI]	Enables RAI alarm generation if the device's busy endpoints exceed a user-defined threshold. ▪ **[0]** = (Default) Disable RAI (Resource Available Indication) service. ▪ **[1]** = RAI service enabled and an SNMP 'acBoardCallResourcesAlarm' Alarm Trap is sent. **Note:** For this parameter to take effect, a device reset is required
[RAIHighThreshold]	Defines the high threshold percentage of total calls that are active (busy endpoints). When the percentage of the device's busy endpoints exceeds this high threshold, the device sends the SNMP acBoardCallResourcesAlarm alarm trap with a 'major' alarm status. The range is 0 to 100. The default is 90. **Note:** The percentage of busy endpoints is calculated by dividing the number of busy endpoints by the total number of "enabled" endpoints.
[RAILowThreshold]	Defines the low threshold percentage of total calls that are active (busy endpoints). When the percentage of the device's busy endpoints falls below this low threshold, the device sends an SNMP acBoardCallResourcesAlarm alarm trap with a 'cleared' alarm status. The range is 0 to 100%. The default is 90%.
[RAILoopTime]	Defines the time interval (in seconds) that the device periodically checks call resource availability. The valid range is 1 to 200. The default is 10.

43.3.5 BootP Parameters

The BootP parameters are described in the table below. The BootP parameters are special 'hidden' parameters. Once defined and saved in the device's flash memory, they are used even if they don't appear in the *ini* file.

Table 43-19: BootP Parameters

Parameter	Description	
[BootPRetries]	**Note:** For this parameter to take effect, a device reset is required. This parameter is used to:	
	Defines the number of BootP requests that the device sends during start-up. The device stops sending BootP requests when either BootP reply is received or number of retries is reached. ▪ **[1]** = 1 BootP retry, 1 sec. ▪ **[2]** = 2 BootP retries, 3 sec. ▪ **[3]** = (Default) 3 BootP retries, 6 sec. ▪ **[4]** = 10 BootP retries, 30 sec. ▪ **[5]** = 20 BootP retries, 60 sec. ▪ **[6]** = 40 BootP retries, 120 sec. ▪ **[7]** = 100 BootP retries, 300 sec.	Defines the number of DHCP packets that the device sends. If after all packets are sent there's still no reply, the device loads from flash. ▪ **[1]** = 4 DHCP packets ▪ **[2]** = 5 DHCP packets ▪ **[3]** = (Default) 6 DHCP packets ▪ **[4]** = 7 DHCP packets ▪ **[5]** = 8 DHCP packets ▪ **[6]** = 9 DHCP packets ▪ **[7]** = 10 DHCP packets ▪ **[15]** = 18 DHCP packets

Parameter	Description
	▪ **[15]** = BootP retries indefinitely.
[BootPSelectiveEnable]	Enables the Selective BootP mechanism. ▪ **[1]** = Enabled ▪ **[0]** = Disabled (default) The Selective BootP mechanism (available from Boot version 1.92) enables the device's integral BootP client to filter unsolicited BootP/DHCP replies (accepts only BootP replies that contain the text 'AUDC' in the vendor specific information field). This option is useful in environments where enterprise BootP/DHCP servers provide undesired responses to the device's BootP requests. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When working with DHCP (i.e., the parameter DHCPEnable is set to 1), the selective BootP feature must be disabled.
[BootPDelay]	Defines the interval between the device's startup and the first BootP/DHCP request that is issued by the device. ▪ **[1]** = (Default) 1 second ▪ **[2]** = 3 second ▪ **[3]** = 6 second ▪ **[4]** = 30 second ▪ **[5]** = 60 second **Note:** For this parameter to take effect, a device reset is required.
[ExtBootPReqEnable]	Determines whether the device uses the Vendor Specific Information field in the BootP request to provide device-related initial startup information. ▪ **[0]** = (Default) Disabled. ▪ **[1]** = Enables extended information to be sent in BootP requests. The device uses the Vendor Specific Information field in the BootP request to provide device-related initial startup information such as device type, current IP address, software version. For a full list of the Vendor Specific Information fields, refer to the *AcBootP Utility User's Guide*. The AcBootP utility displays this information in the 'Client Info' column. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This option is not available on DHCP servers.

43.4   Security Parameters

This subsection describes the device's security parameters.

43.4.1   General Parameters

The general security parameters are described in the table below.

Table 43-20: General Security Parameters

Parameter	Description
Web: Voice Menu Password **[VoiceMenuPassword]**	Defines the password for accessing the device's FXS Voice menu used for configuring and monitoring the device. The default is 12345. **Notes:** ▪ To activate the menu, connect a POTS telephone to an FXS port and dial *** (three stars) followed by the password. ▪ To disable the Voice menu, do any of the following: ✔ Set the VoiceMenuPassword parameter to 'disable'. ✔ Change the Web login password for the Admin user from its default value (i.e., 'Admin') to any other value, and then reset the device. ▪ This parameter is applicable only to FXS interfaces. ▪ For more information on the Voice menu, see FXS Voice Menu Guidance on page 32.
[EnableSecureStartup]	Enables the Secure Startup mode. In this mode, downloading the ini file to the device is restricted to a URL provided in initial configuration (see the parameter IniFileURL) or using DHCP. ▪ **[0]** Disable (default). ▪ **[1]** Enable = disables TFTP and allows secure protocols such as HTTPS to fetch the device configuration. **Note:** For this parameter to take effect, a device reset is required.
Firewall Table	
Web/OVOC: Internal Firewall Parameters **[AccessList]**	This table parameter defines the device's access list (firewall), which defines network traffic filtering rules. The format of this parameter is as follows: [AccessList] FORMAT AccessList_Index = AccessList_Source_IP, AccessList_Source_Port, AccessList_PrefixLen, AccessList_Source_Port, AccessList_Start_Port, AccessList_End_Port, AccessList_Protocol, AccessList_Use_Specific_Interface, AccessList_Interface_ID, AccessList_Packet_Size, AccessList_Byte_Rate, AccessList_Byte_Burst, AccessList_Allow_Type; [\AccessList] For example: AccessList 10 = mgmt.customer.com, , , 32, 0, 80, tcp, 1, OAMP, 0, 0, 0, allow; AccessList 22 = 10.4.0.0, , , 16, 4000, 9000, any, 0, , 0, 0, 0, block; In the example above, Rule #10 allows traffic from the host 'mgmt.customer.com' destined to TCP ports 0 to 80 on interface OAMP (OAMP). Rule #22 blocks traffic from the subnet 10.4.xxx.yyy destined to ports 4000 to 9000. **Note:** For a description of this table, see 'Configuring Firewall Settings' on page 145.

43.4.2 HTTPS Parameters

The Secure Hypertext Transport Protocol (HTTPS) parameters are described in the table below.

Table 43-21: HTTPS Parameters

Parameter	Description
Web: Secured Web Connection (HTTPS) OVOC: HTTPS Only **[HTTPSOnly]**	Determines the protocol used to access the Web interface. ▪ **[0]** HTTP and HTTPS (default). ▪ **[1]** HTTPs Only = Unencrypted HTTP packets are blocked. **Note:** For this parameter to take effect, a device reset is required.
OVOC: HTTPS Port **[HTTPSPort]**	Defines the local Secured HTTPS port of the device. This parameter allows secure remote device Web management from the LAN. To enable secure Web management from the LAN, configure the desired port. The valid range is 1 to 65535 (other restrictions may apply within this range). The default port is 443. **Note:** For this parameter to take effect, a device reset is required.
Web/OVOC: HTTPS Cipher String **[HTTPSCipherString]**	Defines the Cipher string for HTTPS (in OpenSSL cipher list format). For the valid range values, refer to URL http://www.openssl.org/docs/apps/ciphers.html. The default is 'RC4:EXP' (Export encryption algorithms). For example, use 'ALL' for all ciphers suites (e.g., for ARIA encryption for TLS). The only ciphers available are RC4 and DES, and the cipher bit strength is limited to 56 bits. **Note:** For this parameter to take effect, a device reset is required.
Web: HTTP Authentication Mode OVOC: Web Authentication Mode **[WebAuthMode]**	Determines the authentication mode used for the Web interface. ▪ **[0]** Basic Mode = Basic authentication (clear text) is used. ▪ **[1]** Web Based Authentication = (Default) Digest authentication (MD5) is used. **Note:** If you enable RADIUS login (i.e., the WebRADIUSLogin parameter is set to 1), you must set the WebAuthMode parameter to Basic Mode [0].
Web: Requires Client Certificates for HTTPS connection **[HTTPSRequireClientCertificate]**	Determines whether client certificates are required for HTTPS connection. ▪ **[0]** Disable = (Default) Client certificates are not required. ▪ **[1]** Enable = Client certificates are required. The client certificate must be preloaded to the device and its matching private key must be installed on the managing PC. Time and date must be correctly set on the device for the client certificate to be verified. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For a description on implementing client certificates, see 'Client Certificates' on page 106.

Parameter	Description
[HTTPSRootFileName]	Defines the name of the HTTPS trusted root certificate file to be loaded using TFTP. The file must be in base64-encoded PEM (Privacy Enhanced Mail) format. The valid range is a 47-character string. **Note:** This parameter is applicable only when the device is loaded using BootP/TFTP.
[HTTPSPkeyFileName]	Defines the name of a private key file (in unencrypted PEM format) to be loaded from the TFTP server.
[HTTPSCertFileName]	Defines the name of the HTTPS server certificate file to be loaded using TFTP. The file must be in base64-encoded PEM format. The valid range is a 47-character string. **Note:** This parameter is only applicable when the device is loaded using BootP/TFTP.

43.4.3 SRTP Parameters

The Secure Real-Time Transport Protocol (SRTP) parameters are described in the table below.

Table 43-22: SRTP Parameters

Parameter	Description

Web: Media Security
OVOC: Enable Media Security
[EnableMediaSecurity]
	Enables Secure Real-Time Transport Protocol (SRTP).
	▪ **[0]** Disable = (Default) SRTP is disabled.
	▪ **[1]** Enable = SRTP is enabled.
	Notes:
	▪ For this parameter to take effect, a device reset is required.
	▪ SRTP reduces the number of available channels.
	   ✓ MP-124: 18 available channels
	   ✓ MP-118: 6 available channels
	   ✓ MP-114: 3 available channels
	   ✓ MP-112: No reduction

Web/OVOC: Media Security Behavior **[MediaSecurityBehaviour]**	Determines the device's mode of operation when SRTP is used (i.e., when the parameter EnableMediaSecurity is set to 1). ▪ **[0]** Preferable = (Default) The device initiates encrypted calls. However, if negotiation of the cipher suite fails, an unencrypted call is established. Incoming calls that don't include encryption information are accepted. ▪ **[1]** Mandatory = The device initiates encrypted calls, but if negotiation of the cipher suite fails, the call is terminated. Incoming calls that don't include encryption information are rejected. ▪ **[2]** Disable = The IP Profile for which this parameter is set does not support encrypted calls (i.e., SRTP). ▪ **[3]** Preferable - Single Media = The device sends SDP with a single media ('m=') line only (e.g., m=audio 6000 RTP/AVP 4 0 70 96) with RTP/AVP and crypto keys. The remote UA can respond with SRTP or RTP parameters:

Parameter	Description
	✓ If the remote SIP UA does not support SRTP, it uses RTP and ignores the crypto lines. ✓ In the opposite direction, if the device receives an SDP offer with a single media (as shown above), it responds with SRTP (RTP/SAVP) if the EnableMediaSecurity parameter is set to 1. If SRTP is not supported (i.e., EnableMediaSecurity is set to 0), it responds with RTP. **Notes:** ▪ Before configuring this parameter, set the EnableMediaSecurity parameter to 1. ▪ If this parameter is set to Preferable **[3]** and two 'm=' lines are received in the SDP offer, the device prefers the SAVP (secure audio video profile) regardless of the order in the SDP. ▪ Option **[2]** Disable is applicable only to IP Profiles. ▪ This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page 217).
Web: Master Key Identifier (MKI) Size OVOC: Packet MKI Size **[SRTPTxPacketMKISize]**	Defines the size (in bytes) of the Master Key Identifier (MKI) in SRTP Tx packets. The range is 0 to 4. The default is 0 (i.e., new keys are generated without MKI). **Notes:** ▪ The device only initiates the MKI size. ▪ You can also configure MKI size in an IP Profile.

Web: Symmetric MKI Negotiation
OVOC: Enable Symmetric MKI
[EnableSymmetricMKI]
	Enables symmetric MKI negotiation.
	▪ **[0]** Disable = (Default) The device includes the MKI in its 200 OK response according to the SRTPTxPacketMKISize parameter (if set to 0, then it is not included; if set to any other value, it is included with this value).
	▪ **[1]** Enable = The answer crypto line contains (or excludes) an MKI value according to the selected crypto line in the offer. For example, assume that the device receives an INVITE containing the following two crypto lines in SDP:
	<pre>a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:TAaxNnQt8/qLQMnDuG4vxYfWl6K7eBK/ufk04p R4|2^31|1:1
	a=crypto:3 AES_CM_128_HMAC_SHA1_80 inline:bnuYZnMxSfUiGitviWJZmzr7OF3AiRO0l5Vnh0 kH|2^31</pre>
	The first crypto line includes the MKI parameter "1:1". In the 200 OK response, the device selects one of the crypto lines (i.e., '2' or '3'). Typically, it selects the first line that supports the crypto suite. If the device selects crypto line '2', it includes the MKI parameter in its answer SDP, for example:
	<pre>a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:R1VyA1xV/qwBjkEklu4kSJyl3wCtYeZLq1/QFu xw|2^31|1:1</pre>
	If the device selects a crypto line that does not contain the MKI parameter, then the MKI parameter is not included in the crypto line in the SDP answer (even if the

Parameter	Description
	SRTPTxPacketMKISize parameter is set to any value other than 0). **Notes:** ▪ To enable symmetric MKI, the SRTPTxPacketMKISize parameter must be set to any value other than 0. ▪ You can also enable MKI negotiation per IP Profile.
Web/OVOC: SRTP offered Suites **[SRTPofferedSuites]**	Defines the offered crypto suites (cipher encryption algorithms) for SRTP. ▪ **[0]** = (Default) All available crypto suites. ▪ **[1]** CIPHER SUITES AES CM 128 HMAC SHA1 80 = device uses AES-CM encryption with a 128-bit key and HMAC-SHA1 message authentication with a 80-bit tag. ▪ **[2]** CIPHER SUITES AES CM 128 HMAC SHA1 32 = device uses AES-CM encryption with a 128-bit key and HMAC-SHA1 message authentication with a 32-bit tag. **Note:** This parameter also affects the selection of the crypto in the device's answer. For example, if the device receives an offer with two crypto lines containing HMAC_SHA1_80 and HMAC_SHA_32, it uses the HMAC_SHA_32 key in its SIP 200 OK response if the parameter is set to 2.
Web: Disable Authentication On Transmitted RTP Packets OVOC: RTP AuthenticationDisable Tx **[RTPAuthenticationDisableTx]**	Enables authentication on transmitted RTP packets in a secured RTP session. ▪ **[0]** Enable (default) ▪ **[1]** Disable
Web: Disable Encryption On Transmitted RTP Packets OVOC: RTP EncryptionDisable Tx **[RTPEncryptionDisableTx]**	Enables encryption on transmitted RTP packets in a secured RTP session. ▪ **[0]** Enable (default) ▪ **[1]** Disable
Web: Disable Encryption On Transmitted RTCP Packets OVOC: RTCP EncryptionDisable Tx **[RTCPEncryptionDisableTx]**	Enables encryption on transmitted RTCP packets in a secured RTP session. ▪ **[0]** Enable (default) ▪ **[1]** Disable
[ResetSRTPStateUponRekey]	Enables synchronization of the SRTP state between the device and a server when a new SRTP key is generated upon a SIP session expire. This feature ensures that the roll-over counter (ROC), one of the parameters used in the SRTP encryption/decryption process of the SRTP packets, is synchronized on both sides for transmit and receive packets. ▪ **[0]** = (Default) Disabled. ROC is not reset on the device side. ▪ **[1]** = Enabled. If the session expires causing a session refresh through a re-INVITE, the device or server generates a new key and the device resets the ROC index (and other SRTP fields) as done by the server, resulting in a synchronized SRTP. **Notes:** ▪ This feature can also be configured for an IP Profile. ▪ If this feature is disabled and the server resets the ROC upon a re-key generation, one-way voice may occur.

43.4.4   TLS Parameters

The Transport Layer Security (TLS) parameters are described in the table below.

Table 43-23: TLS Parameters

Parameter	Description
Web/OVOC: TLS Version **[TLSVersion]**	Defines the supported SSL/TLS protocol version. Clients attempting to communicate with the device using a different TLS version are rejected. ▪ [0] Any - Including SSLv3 = (Default) SSL 3.0 and all TLS versions are supported. ▪ [1] TLSv1.0 = Only TLS 1.0. ▪ [2] TLSv1.1 = Only TLS 1.1. ▪ [3] TLSv1.0 and TLSv1.1 = Only TLS 1.0 and TLS 1.1. ▪ [4] TLSv1.2 = Only TLS 1.2. ▪ [5] TLSv1.0 and TLSv1.2 = Only TLS 1.0 and TLS 1.2. ▪ [6] TLSv1.1 and TLSv1.2 = Only TLS 1.1 and TLS 1.2. [7] TLSv1.0 TLSv1.1 and TLSv1.2 = Only TLS 1.0, TLS 1.1 and TLS 1.2 (excludes SSL 3.0). **Note:** For this parameter to take effect, a device reset is required.
Web: TLS Client Re-Handshake Interval OVOC: TLS Re Handshake Interval **[TLSReHandshakeInterval]**	Defines the time interval (in minutes) between TLS Re-Handshakes initiated by the device. The interval range is 0 to 1,500 minutes. The default is 0 (i.e., no TLS Re-Handshake).
Web: TLS Mutual Authentication OVOC: SIPS Require Client Certificate **[SIPSRequireClientCertificate]**	Determines the device's behavior when acting as a server for TLS connections. ▪ **[0]** Disable = (Default) The device does not request the client certificate. ▪ **[1]** Enable = The device requires receipt and verification of the client certificate to establish the TLS connection. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The SIPS certificate files can be changed using the parameters HTTPSCertFileName and HTTPSRootFileName.
Web/OVOC: Peer Host Name Verification Mode **[PeerHostNameVerificationMode]**	Determines whether the device verifies the Subject Name of a remote certificate when establishing TLS connections. ▪ **[0]** Disable (default). ▪ **[1]** Server Only = Verify Subject Name only when acting as a client for the TLS connection. ▪ **[2]** Server & Client = Verify Subject Name when acting as a server or client for the TLS connection. When a remote certificate is received and this parameter is not disabled, the value of SubjectAltName is compared with the list of available Proxies. If a match is found for any of the configured Proxies, the TLS connection is established.

Parameter	Description
	The comparison is performed if the SubjectAltName is either a DNS name (DNSName) or an IP address. If no match is found and the SubjectAltName is marked as 'critical', the TLS connection is not established. If DNSName is used, the certificate can also use wildcards ('*') to replace parts of the domain name.
	If the SubjectAltName is not marked as 'critical' and there is no match, the CN value of the SubjectName field is compared with the parameter TLSRemoteSubjectName. If a match is found, the connection is established. Otherwise, the connection is terminated.
	Note: If you set this parameter to **[2]** (Server & Client), for this functionality to operate, you also need to set the SIPSRequireClientCertificate parameter to **[1]** (Enable).
Web: TLS Client Verify Server Certificate OVOC: Verify Server Certificate **[VerifyServerCertificate]**	Determines whether the device, when acting as a client for TLS connections, verifies the Server certificate. The certificate is verified with the Root CA information. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** If Subject Name verification is necessary, the parameter PeerHostNameVerificationMode must be used as well.
Web: Strict Certificate Extension Validation **[RequireStrictCert]**	Enables the validation of the extensions (keyUsage and extentedKeyUsage) of peer certificates. This validation ensures that the signing CA is authorized to sign certificates and that the end-entity certificate is authorized to negotiate a secure TLS connection. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web/OVOC: TLS Remote Subject Name **[TLSRemoteSubjectName]**	Defines the Subject Name that is compared with the name defined in the remote side certificate when establishing TLS connections. If the SubjectAltName of the received certificate is not equal to any of the defined Proxies Host names/IP addresses and is not marked as 'critical', the Common Name (CN) of the Subject field is compared with this value. If not equal, the TLS connection is not established. If the CN uses a domain name, the certificate can also use wildcards ('*') to replace parts of the domain name. The valid range is a string of up to 49 characters. **Note:** This parameter is applicable only if the parameter PeerHostNameVerificationMode is set to 1 or 2.
Web: Client Cipher String **[TLSClientCipherString]**	Defines the cipher-suite string for TLS clients. The valid value is up to 255 strings. The default is "ALL:!ADH". For example: TLSClientCipherString = 'EXP' This parameter complements the HTTPSCipherString parameter (which affects TLS servers). For possible values and additional details, visit the OpenSSL website at https://www.openssl.org/docs/man1.0.2/apps/ciphers.html.

Parameter	Description
[TLSPkeySize]	Defines the key size (in bits) for RSA public-key encryption for newly self-signed generated keys for SSH. ▪ **[512]** ▪ **[768]** ▪ **[1024]** (default) ▪ **[2048]**
Web: TLS FIPS 140 Mode **[TLS_Fips140_Mode]**	Enables FIPS 140-2 conformance mode for TLS. ▪ **[0]** Disable (default) ▪ **[1]** Enable

43.4.5   SSH Parameters

Secure Shell (SSH) parameters are described in the table below.

Table 43-24: SSH Parameters

Parameter	Description
Web/OVOC: Enable SSH Server **[SSHServerEnable]**	Enables the device's embedded SSH server. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web/OVOC: Server Port **[SSHServerPort]**	Defines the port number for the embedded SSH server. Range is any valid port number. The default port is 22.
Web/OVOC: SSH Admin Key **[SSHAdminKey]**	Defines the RSA public key for strong authentication for logging in to the SSH interface (if enabled). The value should be a base64-encoded string. The value can be a maximum length of 511 characters.
Web: Require Public Key OVOC: OVOC: SSH Require Public Key **[SSHRequirePublicKey]**	Enables RSA public keys for SSH. ▪ **[0]** = (Default) RSA public keys are optional if a value is configured for the parameter SSHAdminKey. ▪ **[1]** = RSA public keys are mandatory. **Note:** To define the key size, use the TLSPkeySize parameter.
Web: Max Payload Size OVOC: SSH Max Payload Size **[SSHMaxPayloadSize]**	Defines the maximum uncompressed payload size (in bytes) for SSH packets. The valid value is 550 to 32768. The default is 32768.
Web: Max Binary Packet Size OVOC: SSH Max Binary Packet Size **[SSHMaxBinaryPacketSize]**	Defines the maximum packet size (in bytes) for SSH packets. The valid value is 582 to 35000. The default is 35000.
OVOC: Telnet SSH Max Sessions **[SSHMaxSessions]**	Defines the maximum number of simultaneous SSH sessions. The valid range is 1 to 2. The default is 2 sessions.
Web: Enable Last Login Message **[SSHEnableLastLoginMessage]**	Enables message display in SSH sessions of the time and date of the last SSH login. The SSH login message displays the number of unsuccessful login attempts since the last successful login. ▪ **[0]** Disable

Parameter	Description
	▪ **[1]** Enable (default)
	Note: The last SSH login information is cleared when the device is reset.
Web: Max Login Attempts **[SSHMaxLoginAttempts]**	Defines the maximum SSH login attempts allowed for entering an incorrect password by an administrator before the SSH session is rejected.
	The valid range is 1 to 3. The default is 3.
	Note: The new setting takes effect only for new subsequent SSH connections

43.4.6 IPSec Parameters

The Internet Protocol security (IPSec) parameters are described in the table below.

Table 43-25: IPSec Parameters

Parameter	Description
IPSec Parameters	
Web: Enable IP Security OVOC: IPSec Enable **[EnableIPSec]**	Enables IPSec on the device. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: IKE Certificate Ext Validate **[IKEcertificateExtValidate]**	Enables the validation of the extensions (keyUsage and extentedKeyUsage) of peer certificates. This validation ensures that the signing CA is authorized to sign certificates and that the end-entity certificate is authorized to negotiate a secure IPSec connection. ▪ **[[0]** Disable (default) ▪ **[1]** Enable
IPSec Associations Table	
Web: IP Security Associations Table OVOC: IPSec SA Table **[IPSecSATable]**	This table parameter defines the IPSec SA table. This table allows you to configure the Internet Key Exchange (IKE) and IP Security (IPSec) protocols. You can define up to 20 IPSec peers. The format of this parameter is as follows: [IPsecSATable] FORMAT IPsecSATable_Index = IPsecSATable_RemoteEndpointAddressOrName, IPsecSATable_AuthenticationMethod, IPsecSATable_SharedKey, IPsecSATable_SourcePort, IPsecSATable_DestPort, IPsecSATable_Protocol, IPsecSATable_Phase1SaLifetimeInSec, IPsecSATable_Phase2SaLifetimeInSec, IPsecSATable_Phase2SaLifetimeInKB, IPsecSATable_DPDmode, IPsecSATable_IPsecMode, IPsecSATable_RemoteTunnelAddress, IPsecSATable_RemoteSubnetIPAddress, IPsecSATable_RemoteSubnetPrefixLength, IPsecSATable_InterfaceName; [\IPsecSATable] For example: IPsecSATable 1 = 0, 10.3.2.73, 0, 123456789, 0, 0, 0, 0, 28800, 3600, ;

Parameter	Description
	In the above example, a single IPSec/IKE peer (10.3.2.73) is configured. Pre-shared key authentication is selected, with the pre-shared key set to 123456789. In addition, a lifetime of 28800 seconds is selected for IKE and a lifetime of 3600 seconds is selected for IPSec. **Note:** For a detailed description of this table, see 'Configuring IP Security Associations Table' on page 153.
IPSec Proposal Table	
Web: IP Security Proposal Table OVOC: IPSec Proposal Table **[IPSecProposalTable]**	This table parameter defines up to four IKE proposal settings, where each proposal defines an encryption algorithm, an authentication algorithm, and a Diffie-Hellman group identifier. [IPsecProposalTable] FORMAT IPsecProposalTable_Index = IPsecProposalTable_EncryptionAlgorithm, IPsecProposalTable_AuthenticationAlgorithm, IPsecProposalTable_DHGroup; [\IPsecProposalTable] For example: IPsecProposalTable 0 = 3, 2, 1; IPsecProposalTable 1 = 2, 2, 1; In the example above, two proposals are defined: ▪ Proposal 0: AES, SHA1, DH group 2 ▪ Proposal 1: 3DES, SHA1, DH group 2 **Note:** For a detailed description of this table, see 'Configuring IP Security Proposal Table' on page 151.

43.4.7   802.1X Parameters

The 802.1X parameters are described in the table below.

Table 43-26: 802.1X Parameters

Parameter	Description
Web: 802.1x Mode OVOC: Mode **[802.**1xMode]	Enables support for IEEE 802.1x physical port security. The device can function as an IEEE 802.1X supplicant. IEEE 802.1X is a standard for port-level security on secure Ethernet switches; when a unit is connected to a secure port, no traffic is allowed until the identity of the unit is authenticated. ▪ **[0]** Disabled (default) ▪ **[1]** EAP-MD5 = Authentication is performed using a user name and password configured by the parameters 802.1xUsername and 802.1xPassword. ▪ **[2]** Protected EAP = Authentication is performed using a user name and password configured by the parameters 802.1xUsername and 802.1xPassword. In addition, the protocol used is MSCHAPv2 over an encrypted TLS tunnel. ▪ **[3]** EAP-TLS = The device's certificate is used to establish a mutually-authenticated TLS session with the Access Server. This requires prior configuration of the server certificate and root CA (see Configuring the Certificates on page 103). The parameter 802.1xUsername is used to identify the device, however 802.1xPassword is ignored. **Note:** The configured mode must match the configuration of the Access server (e.g., RADIUS server).
Web: 802.1x Username OVOC: User Name **[802.**1xUsername]	Defines the username for IEEE 802.1x support. The valid value is a string of up to 256 characters. The default is an empty string.
Web: 802.1x Password OVOC: Password **[802.**1xPassword]	Defines the password for IEEE 802.1x support. The valid value is a string of up to 64 characters. The default is an empty string.
Web: 802.1x Verify Peer Certificate OVOC: Verify Peer Certificate **[802.**1xVerifyPeerCertificate]	Determines whether the device verifies the Peer Certificate for IEEE 802.1x support. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: 802.1x Max Fragment Size [802.1MaxFragmentSize]	Defines the maximum fragment size for IEEE 802.1x protocol (max. bytes on the wire). The valid value range is 64 to 1518 bytes. The default is 1500 bytes.

43.4.8   OCSP Parameters

The Online Certificate Status Protocol (OCSP) parameters are described in the table below.

Table 43-27: OCSP Parameters

Parameter	Description
Web: Enable OCSP Server OVOC: OCSP Enable **[OCSPEnable]**	Enables or disables certificate checking using OCSP. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: Primary Server IP OVOC: OCSP Server IP **[OCSPServerIP]**	Defines the IP address of the OCSP server. The default IP address is 0.0.0.0.
Web: Secondary Server IP **[OCSPSecondaryServerIP]**	Defines the IP address (in dotted-decimal notation) of the secondary OCSP server (optional). The default IP address is 0.0.0.0.
Web: Server Port OVOC: OCSP Server Port **[OCSPServerPort]**	Defines the OCSP server's TCP port number. The default port number is 2560.
Web: Default Response When Server Unreachable OVOC: OCSP Default Response **[OCSPDefaultResponse]**	Determines the default OCSP behavior when the server cannot be contacted. ▪ **[0]** Reject = (Default) Rejects peer certificate. ▪ **[1]** Allow = Allows peer certificate.

43.5 RADIUS Parameters

The RADIUS parameters are described in the table below. For supported RADIUS attributes, see 'RADIUS Accounting CDR Attributes' on page .

Table 43-28: RADIUS Parameters

Parameter	Description
RADIUS Accounting Parameters	
Web: Enable RADIUS Access Control **[EnableRADIUS]**	Enables the RADIUS application. ▪ **[0]** Disable (Default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Accounting Server IP Address **[RADIUSAccServerIP]**	Defines the IP address of the RADIUS accounting server.
Web: Accounting Port **[RADIUSAccPort]**	Defines the port of the RADIUS accounting server. The default is 1646.
Web/OVOC: RADIUS Accounting Type **[RADIUSAccountingType]**	Determines when the RADIUS accounting messages are sent to the RADIUS accounting server. ▪ **[0]** At Call Release = (Default) Sent at call release only. ▪ **[1]** At Connect & Release = Sent at call connect and release. ▪ **[2]** At Setup & Release = Sent at call setup and release.
Web: AAA Indications OVOC: Indications **[AAAIndications]**	Determines the Authentication, Authorization and Accounting (AAA) indications. ▪ **[0]** None = (Default) No indications. ▪ **[3]** Accounting Only = Only accounting indications are used.
General RADIUS Parameters	

Web: Use RADIUS for Web/Telnet Login
OVOC: Web Use Radius Login
[WebRADIUSLogin]
	Enables RADIUS queries for Web and Telnet login authentication. When enabled, logging into the device's Web and Telnet embedded servers is done through a RADIUS server. The device communicates with a user-defined RADIUS server and verifies the given username and password against a remote database, in a secure manner.
	▪ **[0]** Disable (default)
	▪ **[1]** Enable
	Notes:
	▪ For RADIUS login authentication to function, you also need to set the following parameters:
	  ✓ EnableRADIUS = 1 (Enable)
	  ✓ WebAuthMode = 0 (Basic Mode)
	▪ RADIUS authentication requires HTTP basic authentication, where the username and password are transmitted in clear text over the network. Therefore, it's recommended to set the HTTPSOnly parameter to 1 in order to force the use of HTTPS, since the transport is encrypted.
	▪ If using RADIUS authentication to log into the CLI, only the primary Web User Account, which has Security Administration access level, can access the device's CLI (see 'Configuring Web User Accounts' on page 66).

Parameter	Description
Web: RADIUS Authentication Server IP Address OVOC: RADIUS Auth Server IP **[RADIUSAuthServerIP]**	Defines the IP address of the RADIUS authentication server. **Note:** For this parameter to take effect, a device reset is required.
Web: RADIUS Authentication Server Port OVOC: RADIUS Auth Server Port **[RADIUSAuthPort]**	Defines the port of the RADIUS Authentication Server. **Note:** For this parameter to take effect, a device reset is required.
Web: RADIUS Shared Secret OVOC: RADIUS Auth Server Secret **[SharedSecret]**	Defines the 'Secret' used to authenticate the device to the RADIUS server. This should be a cryptically strong password.
RADIUS Authentication Parameters	
Web: Default Access Level **[DefaultAccessLevel]**	Defines the default access level for the device when the RADIUS (authentication) response doesn't include an access level attribute. The valid range is 0 to 255. The default is 200 (i.e., Security Administrator).
Web: Local RADIUS Password Cache Mode **[RadiusLocalCacheMode]**	Determines the device's mode of operation regarding the timer (configured by the parameter RadiusLocalCacheTimeout) that determines the validity of the user name and password (verified by the RADIUS server). ▪ **[0]** Absolute Expiry Timer = When you access a Web page, the timeout doesn't reset, instead it continues decreasing. ▪ **[1]** Reset Timer Upon Access = (Default) Upon each access to a Web page, the timeout always resets (reverts to the initial value configured by RadiusLocalCacheTimeout).
Web: Local RADIUS Password Cache Timeout **[RadiusLocalCacheTimeout]**	Defines the time (in seconds) the locally stored user name and password (verified by the RADIUS server) are valid. When this time expires, the user name and password become invalid and a must be re-verified with the RADIUS server. The valid range is 1 to 0xFFFFFF. The default is 300 (5 minutes). ▪ **[-1]** = Never expires. ▪ **[0]** = Each request requires RADIUS authentication.
Web: RADIUS VSA Vendor ID **[RadiusVSAVendorID]**	Defines the vendor ID that the device accepts when parsing a RADIUS response packet. The valid range is 0 to 0xFFFFFFFF. The default is 5003.
Web: RADIUS VSA Access Level Attribute **[RadiusVSAAccessAttribute]**	Defines the code that indicates the access level attribute in the Vendor Specific Attributes (VSA) section of the received RADIUS packet. The valid range is 0 to 255. The default is 35.
[MaxRADIUSSessions]	Defines the number of concurrent calls that can communicate with the RADIUS server (optional). The valid range is 0 to 240. The default is 240.
OVOC: RADIUS Auth Number of Retries **[RADIUSRetransmission]**	Defines the number of retransmission retries. The valid range is 1 to 10. The default is 3.

Parameter	Description
[RadiusTO]	Defines the time interval (measured in seconds) that the device waits for a response before a RADIUS retransmission is issued. The valid range is 1 to 30. The default is 10.

43.6   SIP Media Realm Parameters

The Media Realm parameters are described in the table below.

Table 43-29: Media Realm Parameters

Parameter	Description
Media Realm Table	
Web: Media Realm Table OVOC: VoIP > Media > Media Realm **[CpMediaRealm]**	This table parameter defines the Media Realm table. The Media Realm table allows you to divide a Media-type interface (defined in the Multiple Interface table) into several realms, where each realm is specified by a UDP port range. The format of this parameter is as follows: [CpMediaRealm] FORMAT CpMediaRealm_Index = CpMediaRealm_MediaRealmName, CpMediaRealm_IPv4IF, CpMediaRealm_IPv6IF, CpMediaRealm_PortRangeStart, CpMediaRealm_MediaSessionLeg, CpMediaRealm_PortRangeEnd, CpMediaRealm_TransRateRatio, CpMediaRealm_IsDefault; [\CpMediaRealm] For example, CpMediaRealm 1 = Mrealm1, Voice, , 6600, 20, 6790, , 1; CpMediaRealm 2 = Mrealm2, Voice, , 6800, 10, 6890; , 0; **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For a detailed description of this table, see 'Configuring Media Realms' on page 180.
Quality of Experience Parameters	
Web: Server IP CLI: server-ip **[QOEServerIP]**	Defines the IP address of AudioCodes One Voice Operations Center (OVOC)server to where the quality experience reports are sent. **Note:** For this parameter to take effect, a device reset is required.
Web: Port **[QOEPort]**	Defines the port of the OVOC server. The valid value range is 0 to 65534. The default is 5000.
Web: Interface Name **[QOEInterfaceName]**	Defines the IP network interface on which the quality experience reports are sent. The default is the OAMP interface. **Note:** For this parameter to take effect, a device reset is required.
Media Realm > Quality of Experience Table	
Web: Media Realm > Quality Of Experience OVOC: Media > Media Realm > Voice Quality	This table configures Quality of Experience parameters per Media Realm. [QOERules] ORMAT QOERules_Index = QOERules_MediaRealmIndex, QOERules_RuleIndex, QOERules_MonitoredParam,

Parameter	Description
Rules **[QOERules]**	QOERules_Direction, QOERules_Profile, QOERules_GreenYellowThreshold, QOERules_GreenYellowHysteresis, QOERules_YellowRedThreshold, QOERules_YellowRedHysteresis, QOERules_GreenYellowOperation, QOERules_GreenYellowOperationDetails, QOERules_YellowRedOperation, QOERules_YellowRedOperationDetails; [\QOERules]
	Note: For a detailed description of this table, see Configuring Quality of Experience Parameters per Media Realm on page 182.

43.7   Control Network Parameters

43.7.1   IP Group, Proxy, Registration and Authentication Parameters

The proxy server, registration and authentication SIP parameters are described in the table below.

Table 43-30: Proxy, Registration and Authentication SIP Parameters

Parameter	Description
IP Group Table	
Web: IP Group Table OVOC: Endpoints > IP Group **[IPGroup]**	This table configures IP Groups. The ini file format of this parameter is as follows: [IPGroup] FORMAT IPGroup_Index = IPGroup_Type, IPGroup_Description, IPGroup_ProxySetId, IPGroup_SIPGroupName, IPGroup_ContactUser, IPGroup_EnableSurvivability, IPGroup_ServingIPGroup, IPGroup_SipReRoutingMode, IPGroup_AlwaysUseRouteTable, IPGroup_RoutingMode, IPGroup_SRD, IPGroup_MediaRealm, IPGroup_ClassifyByProxySet, IPGroup_ProfileId, IPGroup_MaxNumOfRegUsers, IPGroup_InboundManSet, IPGroup_OutboundManSet, IPGroup_RegistrationMode, IPGroup_AuthenticationMode, IPGroup_MethodList, IPGroup_EnableSBCClientForking, IPGroup_SourceUriInput, IPGroup_DestUriInput, IPGroup_ContactName; [/IPGroup] **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For a description of this table, see 'Configuring IP Groups' on page 197.
Authentication per Port Table	
Web: Authentication Table OVOC: SIP Endpoints > Authentication **[Authentication]**	This table parameter defines a user name and password for authenticating each device port. The format of this parameter is as follows: [Authentication] FORMAT Authentication_Index = Authentication_UserId,

Parameter	Description
	Authentication_UserPassword; [\Authentication]
	Where,
	▪ Index = port number, where 0 denotes the Port 1
	For example: Authentication 1 = lee,1552; (user name "lee" with password 1552 for authenticating Port 2)
	Note: For a description o this table, see Configuring Authentication on page 302.
Account Table	
Web: Account Table OVOC: SIP Endpoints > Account **[Account]**	This table parameter configures the Account table for registering and/or authenticating (digest) Hunt Groups(e.g., an IP-PBX) to another IP Group (e.g., an Internet Telephony Service Provider - ITSP). The format of this parameter is as follows:
	[Account] FORMAT Account_Index = Account_ServedTrunkGroup, Account_ServedIPGroup, Account_ServingIPGroup, Account_Username, Account_Password, Account_HostName, Account_Register, Account_ContactUser, Account_ApplicationType; [\Account]
	For example: Account 1 = 1, -1, 1, user, 1234, acl, 1, ITSP1;
	Note: For a detailed description of this table, see 'Configuring Account Table' on page 205.
Proxy Registration Parameters	
Web: Use Default Proxy OVOC: Proxy Used **[IsProxyUsed]**	Enables the use of a SIP proxy server.
	▪ **[0]** No = (Default) Proxy isn't used and instead, the internal routing table is used.
	▪ **[1]** Yes = Proxy server is used. Define the IP address of the proxy server in the Proxy Sets table (see 'Configuring Proxy Sets Table' on page 200).
	Note: If you are not using a proxy server, you must define outbound IP call routing rules in the Tel to IP Routing (described in Configuring Tel to IP Routing on page 250).
Web/OVOC: Proxy Name **[ProxyName]**	Defines the Home Proxy domain name. If specified, this name is used as the Request-URI in REGISTER, INVITE and other SIP messages, and as the host part of the To header in INVITE messages. If not specified, the Proxy IP address is used instead.
	The valid value is a string of up to 49 characters.
	Note: This parameter functions together with the UseProxyIPasHost parameter.
Web: Use Proxy IP as Host **[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

Parameter	Description
	If this parameter is disabled and the device registers to an IP Group (i.e., proxy server), it uses the string configured by the ProxyName parameter as the host name in the REGISTER's Request-URI and uses the string configured by the IP Group table parameter, SIPGroupName as the host name in the To and From headers. If the IP Group is configured with a Proxy Set that has multiple IP addresses, all the REGISTER messages sent to these proxies are sent with the same host name. **Note:** If this parameter is disabled and the ProxyName parameter is not configured, the proxy's IP address is used as the host name in the REGISTER Request-URI.
Web: Redundancy Mode OVOC: Proxy Redundancy Mode **[ProxyRedundancyMode]**	Determines whether the device switches back to the primary Proxy after using a redundant Proxy. ▪ **[0]** Parking = (Default) The device continues working with a redundant (now active) Proxy until the next failure, after which it works with the next redundant Proxy. ▪ **[1]** Homing = The device always tries to work with the primary Proxy server (i.e., switches back to the primary Proxy whenever it's available). **Note:** To use this Proxy Redundancy mechanism, you need to enable the keep-alive with Proxy option, by setting the parameter EnableProxyKeepAlive to 1 or 2.
Web: Proxy IP List Refresh Time OVOC: IP List Refresh Time **[ProxyIPListRefreshTime]**	Defines the time interval (in seconds) between each Proxy IP list refresh. The range is 5 to 2,000,000. The default interval is 60.
Web: Enable Fallback to Routing Table OVOC: Fallback Used **[IsFallbackUsed]**	Determines whether the device falls back to the Tel to IP Routing for call routing when Proxy servers are unavailable. ▪ **[0]** Disable = (Default) Fallback is not used. ▪ **[1]** Enable = The Tel to IP Routing is used when Proxy servers are unavailable. When the device falls back to the Tel to IP Routing, it continues scanning for a Proxy. When the device locates an active Proxy, it switches from internal routing back to Proxy routing. **Note:** To enable the redundant Proxies mechanism, set the parameter EnableProxyKeepAlive to 1 or 2.
Web/OVOC: Prefer Routing Table **[PreferRouteTable]**	Determines whether the device's internal routing table takes precedence over a Proxy for routing calls. ▪ **[0]** No = (Default) Only a Proxy server is used to route calls. ▪ **[1]** Yes = The device checks the routing rules in the Tel to IP Routing for a match with the Tel-to-IP call. Only if a match is not found is a Proxy used.
Web/OVOC: Always Use Proxy **[AlwaysSendToProxy]**	Determines whether the device sends SIP messages and responses through a Proxy server. ▪ **[0]** Disable = (Default) Use standard SIP routing rules. ▪ **[1]** Enable = All SIP messages and responses are sent to the Proxy server.

Parameter	Description
	Note: This parameter is applicable only if a Proxy server is used (i.e., the parameter IsProxyUsed is set to 1).
Web: SIP ReRouting Mode OVOC: SIP Re-Routing Mode **[SIPReroutingMode]**	Determines the routing mode after a call redirection (i.e., a 3xx SIP response is received) or transfer (i.e., a SIP REFER request is received). ▪ **[0]** Standard = (Default) INVITE messages that are generated as a result of Transfer or Redirect are sent directly to the URI, according to the Refer-To header in the REFER message, or Contact header in the 3xx response. ▪ **[1]** Proxy = Sends a new INVITE to the Proxy. Note: This option is applicable only if a Proxy server is used and the parameter AlwaysSendtoProxy is set to 0. ▪ **[2]** Routing Table = Uses the Routing table to locate the destination and then sends a new INVITE to this destination. **Notes:** ▪ When this parameter is set to **[1]** and the INVITE sent to the Proxy fails, the device re-routes the call according to the Standard mode **[0].** ▪ When this parameter is set to **[2]** and the INVITE fails, the device re-routes the call according to the Standard mode **[0].** If DNS resolution fails, the device attempts to route the call to the Proxy. If routing to the Proxy also fails, the Redirect/Transfer request is rejected. ▪ When this parameter is set to **[2],** the XferPrefix parameter can be used to define different routing rules for redirect calls. ▪ This parameter is disregarded if the parameter AlwaysSendToProxy is set to 1.
Web/OVOC: DNS Query Type **[DNSQueryType]**	Enables the use of DNS Naming Authority Pointer (NAPTR) and Service Record (SRV) queries to resolve Proxy and Registrar servers and to resolve all domain names that appear in the SIP Contact and Record-Route headers. ▪ **[0]** A-Record (default) ▪ **[1]** SRV ▪ **[2]** NAPTR If set to A-Record **[0],** no NAPTR or SRV queries are performed. If set to SRV **[1]** and the Proxy/Registrar IP address parameter, Contact/Record-Route headers, or IP address defined 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. If set to NAPTR **[2],** 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. If the Proxy/Registrar IP address parameter, the domain name in the Contact/Record-Route headers, or the IP

Parameter	Description
	address defined in the Routing tables contain a domain name with port definition, the device performs a regular DNS A-record query. If a specific Transport Type is defined, a NAPTR query is not performed. **Note:** To enable NAPTR/SRV queries for Proxy servers only, use the parameter ProxyDNSQueryType.
Web: Proxy DNS Query Type **[ProxyDNSQueryType]**	Enables the use of DNS Naming Authority Pointer (NAPTR) and Service Record (SRV) queries to discover Proxy servers. ▪ **[0]** A-Record (default) ▪ **[1]** SRV ▪ **[2]** NAPTR If set to A-Record **[0],** no NAPTR or SRV queries are performed. If set to SRV **[1]** and 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. Therefore, if the first SRV query returns two domain names and the A-record queries return two IP addresses each, no additional searches are performed. If set to NAPTR **[2],** 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. If the Proxy IP address parameter contains a domain name with port definition (e.g., ProxyIP = domain.com:5080), the device performs a regular DNS A-record query. If a specific Transport Type is defined, a NAPTR query is not performed. **Note:** When enabled, NAPTR/SRV queries are used to discover Proxy servers even if the parameter DNSQueryType is disabled.
Web/OVOC: Use Gateway Name for OPTIONS **[UseGatewayNameForOptions]**	Determines whether the device uses its IP address or gateway name in keep-alive SIP OPTIONS messages. ▪ **[0]** No = (Default) Use the device's IP address in keep-alive OPTIONS messages. ▪ **[1]** Yes = Use 'Gateway Name' (SIPGatewayName) in keep-alive OPTIONS messages. ▪ **[2]** Server = Device's IP address is used in the From and To headers in keep-alive OPTIONS messages. The OPTIONS Request-URI host part contains either the device's IP address or a string defined by the parameter SIPGatewayName. The device uses the OPTIONS request as a keep-alive message to its primary and redundant Proxies (i.e., the parameter EnableProxyKeepAlive is set to 1).

Parameter	Description
Web/OVOC: User Name **[UserName]**	Defines the user name used for registration and Basic/Digest authentication with a Proxy/Registrar server. The default is an empty string. **Notes:** ▪ This parameter is applicable only if single device registration is used (i.e., the parameter AuthenticationMode is set to authentication per gateway). ▪ Instead of configuring this parameter, the Authentication table can be used (see Authentication on page 302).
Web/OVOC: Password **[Password]**	Defines the password for Basic/Digest authentication with a Proxy/Registrar server. A single password is used for all device ports. The default is 'Default_Passwd'. **Note:** Instead of configuring this parameter, the Authentication table can be used (see Authentication on page 302).
Web/OVOC: Cnonce **[Cnonce]**	Defines the Cnonce string used by the SIP server and client to provide mutual authentication. The value is free format, i.e., 'Cnonce = 0a4f113b'. The default is 'Default_Cnonce'.
Web/OVOC: Mutual Authentication Mode **[MutualAuthenticationMode]**	Determines the device's mode of operation when Authentication and Key Agreement (AKA) Digest Authentication is used. ▪ **[0]** Optional = (Default) Incoming requests that don't include AKA authentication information are accepted. ▪ **[1]** Mandatory = Incoming requests that don't include AKA authentication information are rejected.
Web/OVOC: Challenge Caching Mode **[SIPChallengeCachingMode]**	Determines the mode for Challenge Caching, which reduces the number of SIP messages transmitted through the network. The first request to the Proxy is sent without authorization. The Proxy sends a 401/407 response with a challenge. This response is saved for further uses. A new request is re-sent with the appropriate credentials. Subsequent requests to the Proxy are automatically sent with credentials (calculated from the saved challenge). If the Proxy doesn't accept the new request and sends another challenge, the old challenge is replaced with the new one. ▪ **[0]** None = (Default) Challenges are not cached. Every new request is sent without preliminary authorization. If the request is challenged, a new request with authorization data is sent. ▪ **[1]** INVITE Only = Challenges issued for INVITE requests are cached. This prevents a mixture of REGISTER and INVITE authorizations. ▪ **[2]** Full = Caches all challenges from the proxies. **Note:** Challenge Caching is used with all proxies and not only with the active one.
Proxy IP Table	

Parameter	Description
Web: Proxy IP Table OVOC: Proxy IP **[ProxyIP]**	This table parameter configures the Proxy Set table with Proxy Set IDs, each with up to five Proxy server IP addresses (or fully qualified domain name/FQDN). Each Proxy Set can be defined with a transport type (UDP, TCP, or TLS). The format of this parameter is as follows: [ProxyIP] FORMAT ProxyIp_Index = ProxyIp_IpAddress, ProxyIp_TransportType, ProxyIp_ProxySetId; [\ProxyIP] For example: ProxyIp 0 = 10.33.37.77, -1, 0; ProxyIp 1 = 10.8.8.10, 0, 2; ProxyIp 2 = 10.5.6.7, -1, 1; **Notes:** ▪ To assign various attributes (such as Proxy Load Balancing) per Proxy Set ID, use the parameter ProxySet. ▪ For a description of this table, see 'Configuring Proxy Sets Table' on page 200.
Proxy Set Table	
Web: Proxy Set Table OVOC: Proxy Set **[ProxySet]**	This table parameter configures the Proxy Set ID table. It is used in conjunction with the ProxyIP table ini file parameter, which defines the IP addresses per Proxy Set ID. The ProxySet table ini file parameter defines additional attributes per Proxy Set ID. This includes, for example, Proxy keep-alive and load balancing and redundancy mechanisms (if a Proxy Set contains more than one proxy address). The format of this parameter is as follows: [ProxySet] FORMAT ProxySet_Index = ProxySet_EnableProxyKeepAlive, ProxySet_ProxyKeepAliveTime, ProxySet_ProxyLoadBalancingMethod, ProxySet_IsProxyHotSwap, ProxySet_SRD, ProxySet_ClassificationInput, ProxySet_ProxyRedundancyMode, ProxySet_KeepAliveFailureResp, ProxySet_HomingSuccessDetectionRetries; [\ProxySet] For example: ProxySet 0 = 0, 60, 0, 0, 0, , 1,0; ProxySet 1 = 1, 60, 1, 0, 1, , 0,0; **Notes:** ▪ For configuring the Proxy Set IDs and their IP addresses, use the parameter ProxyIP. ▪ For a description of this table, see 'Configuring Proxy Sets Table' on page 200.
Registrar Parameters	
Web: Enable Registration OVOC: Is Register Needed **[IsRegisterNeeded]**	Enables the device to register to a Proxy/Registrar server. ▪ **[0]** Disable = (Default) The device doesn't register to Proxy/Registrar server.

Parameter	Description
	▪ **[1]** Enable = The device registers to Proxy/Registrar server when the device is powered up and at every user-defined interval (configured by the parameter RegistrationTime).
	Note: The device sends a REGISTER request for each channel or for the entire device (according to the AuthenticationMode parameter).
Web/OVOC: Registrar Name **[RegistrarName]**	Defines the Registrar domain name. If specified, the name is used as the Request-URI in REGISTER messages. If it isn't specified (default), the Registrar IP address, or Proxy name or IP address is used instead. The valid range is up to 100 characters.
Web: Registrar IP Address OVOC: Registrar IP **[RegistrarIP]**	Defines the IP address (or FQDN) and port number (optional) of the Registrar server. The IP address is in dotted-decimal notation, e.g., 201.10.8.1:<5080>. **Notes:** ▪ If not specified, the REGISTER request is sent to the primary Proxy server. ▪ When a port number is specified, DNS NAPTR/SRV queries aren't performed, even if the parameter DNSQueryType is set to 1 or 2. ▪ If the parameter RegistrarIP is set to an FQDN and is resolved to multiple addresses, the device also provides real-time switching (hotswap mode) between different Registrar IP addresses (the parameter IsProxyHotSwap is set to 1). If the first Registrar doesn't respond to the REGISTER message, the same REGISTER message is sent immediately to the next Proxy. To allow this mechanism, the parameter EnableProxyKeepAlive must be set to 0. ▪ When a specific transport type is defined using the parameter RegistrarTransportType, a DNS NAPTR query is not performed even if the parameter DNSQueryType is set to 2.
Web/OVOC: Registrar Transport Type **[RegistrarTransportType]**	Determines the transport layer used for outgoing SIP dialogs initiated by the device to the Registrar. ▪ **[-1**] Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When set to 'Not Configured', the value of the parameter SIPTransportType is used.
Web/OVOC: Registration Time **[RegistrationTime]**	Defines the time interval (in seconds) for registering to a Proxy server. The value is used in the SIP Expires header. This parameter also defines the time interval between Keep-Alive messages when the parameter EnableProxyKeepAlive is set to 2 (REGISTER). Typically, the device registers every 3,600 sec (i.e., one hour). The device resumes registration according to the parameter RegistrationTimeDivider. The valid range is 10 to 2,000,000. The default is 180.

Parameter	Description
Web: Re-registration Timing **[%]** OVOC: Time Divider **[RegistrationTimeDivider]**	Defines the re-registration timing (in percentage). The timing is a percentage of the re-register timing set by the Registrar server. The valid range is 50 to 100. The default is 50. For example: If this parameter is set to 70% and the Registration Expires time is 3600, the device re-sends its registration request after 3600 x 70% (i.e., 2520 sec). **Note**: This parameter may be overridden if the parameter RegistrationTimeThreshold is greater than 0.
Web/OVOC: Registration Retry Time **[RegistrationRetryTime]**	Defines the time interval (in seconds) after which a registration request is re-sent if registration fails with a 4xx response or if there is no response from the Proxy/Registrar server. The default is 30 seconds. The range is 10 to 3600.
Web: Registration Time Threshold OVOC: Time Threshold **[RegistrationTimeThreshold]**	Defines a threshold (in seconds) for re-registration timing. If this parameter is greater than 0, but lower than the computed re-registration timing (according to the parameter RegistrationTimeDivider), the re-registration timing is set to the following: timing set by the Registration server in the SIP Expires header minus the value of the parameter RegistrationTimeThreshold. The valid range is 0 to 2,000,000. The default is 0.
Web: Re-register On INVITE Failure OVOC: Register On Invite Failure **[RegisterOnInviteFailure]**	Enables immediate re-registration if no response is received for an INVITE request sent by the device. ▪ **[0]** Disable (default) ▪ **[1]** Enable When enabled, the device immediately expires its re-registration timer and commences re-registration to the same Proxy upon any of the following scenarios: ▪ The response to an INVITE request is 407 (Proxy Authentication Required) without an authentication header included. ▪ The remote SIP UA abandons a call before the device has received any provisional response (indicative of an outbound proxy server failure). ▪ The remote SIP UA abandons a call and the only provisional response the device has received for the call is 100 Trying (indicative of a home proxy server failure, i.e., the failure of a proxy in the route after the outbound proxy). ▪ The device terminates a call due to the expiration of RFC 3261 Timer B or due to the receipt of a 408 (Request Timeout) response and the device has not received any provisional response for the call (indicative of an outbound proxy server failure). ▪ The device terminates a call due to the receipt of a 408 (Request Timeout) response and the only provisional response the device has received for the call is the 100 Trying provisional response (indicative of a home proxy server failure).

Parameter	Description
Web: ReRegister On Connection Failure OVOC: Re Register On Connection Failure **[ReRegisterOnConnectionFailure]**	Enables the device to perform SIP re-registration upon TCP/TLS connection failure. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: Gateway Registration Name OVOC: Name **[GWRegistrationName]**	Defines the user name that is used in the From and To headers in SIP REGISTER messages. If no value is specified (default) for this parameter, the UserName parameter is used instead. **Note:** This parameter is applicable only for single registration per device (i.e., AuthenticationMode is set to 1). When the device registers each channel separately (i.e., AuthenticationMode is set to 0), the user name is set to the channel's phone number.
Web/OVOC: Registration Mode **[AuthenticationMode]**	Determines the device's registration and authentication method. ▪ **[0]** Per Endpoint = Registration and authentication is performed separately for each endpoint. This is typically used for FXS interfaces, where each endpoint registers (and authenticates) separately with its user name and password. ▪ **[1]** Per Gateway = (Default) Single registration and authentication for the entire device. This is typically used for FXO interfaces. ▪ **[3]** Per FXS = Registration and authentication for FXS endpoints.
Web: Set Out-Of-Service On Registration Failure OVOC: Set OOS On Registration Fail **[OOSOnRegistrationFail]**	Enables setting the endpoint or entire device (i.e., all endpoints) to out-of-service if registration fails. ▪ **[0]** Disable (default) ▪ **[1]** Enable If the registration is per endpoint (i.e., AuthenticationMode is set to 0) or per Account (see Configuring Hunt Group Settings on page 229) and a specific endpoint/Account registration fails (SIP 4xx or no response), then that endpoint is set to out-of-service until a success response is received in a subsequent registration request. When the registration is per the entire device (i.e., AuthenticationMode is set to 1) and registration fails, all endpoints are set to out-of-service. **Note:** The out-of-service method is configured using the FXSOOSBehavior parameter.
[UnregistrationMode]	Enables the device to perform explicit unregisters. ▪ **[0]** Disable (default) ▪ **[1]** Enable = The device sends an asterisk ("*") value in the SIP Contact header, instructing the Registrar server to remove all previous registration bindings. The device removes SIP User Agent (UA) registration bindings in a Registrar, according to RFC 3261. Registrations are soft state and expire unless refreshed, but they can also be explicitly removed. A client can attempt to influence the expiration interval selected by the Registrar. A UA requests the immediate removal of a binding by specifying an expiration interval of "0" for that contact address in a

Parameter	Description
	REGISTER request. UA's should support this mechanism so that bindings can be removed before their expiration interval has passed. Use of the "*" Contact header field value allows a registering UA to remove all bindings associated with an address-of-record (AOR) without knowing their precise values. **Note:** The REGISTER-specific Contact header field value of "*" applies to all registrations, but it can only be used if the Expires header field is present with a value of "0".

Web/OVOC: Add Empty Authorization Header **[EmptyAuthorizationHeader]**	Enables the inclusion of the SIP Authorization header in initial registration (REGISTER) requests sent by the device. **[0]** Disable (default) **[1]** Enable The Authorization header carries the credentials of a user agent (UA) in a request to a server. The sent REGISTER message populates the Authorization header with the following parameters: - username - set to the value of the private user identity - realm - set to the domain name of the home network - uri - set to the SIP URI of the domain name of the home network - nonce - set to an empty value - response - set to an empty value For example:
	`Authorization: Digest`
	`username=alice_private@home1.net,`
	`realm="home1.net", nonce="",`
	`response="e56131d19580cd833064787ecc"`
	Note: This registration header is according to the IMS 3GPP TS24.229 and PKT-SP-24.220 specifications.

Web: Add initial Route Header **[InitialRouteHeader]**	Enables the inclusion of the SIP Route header in initial registration or re-registration (REGISTER) requests sent by the device. **[0]** Disable (default) **[1]** Enable When the device sends a REGISTER message, the Route header includes either the Proxy's FQDN, or IP address and port according to the configured Proxy Set, for example:
	`Route: <sip:10.10.10.10;lr;transport=udp>`
	or
	`Route: <sip: pcscf-`
	`gm.ims.rr.com;lr;transport=udp>`

OVOC: Ping Pong Keep Alive **[UsePingPongKeepAlive]**	Enables the use of the carriage-return and line-feed sequences (CRLF) Keep-Alive mechanism, according to RFC 5626 "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)" for reliable, connection-orientated transport types such as TCP. **[0]** Disable (default) **[1]** Enable

Parameter	Description
	The SIP user agent/client (i.e., device) uses a simple periodic message as a keep-alive mechanism to keep their flow to the proxy or registrar alive (used for example, to keep NAT bindings open). For connection-oriented transports such as TCP/TLS this is based on CRLF. This mechanism uses a client-to-server "ping" keep-alive and a corresponding server-to-client "pong" message. This ping-pong sequence allows the client, and optionally the server, to tell if its flow is still active and useful for SIP traffic. If the client does not receive a pong in response to its ping, it declares the flow "dead" and opens a new flow in its place. In the CRLF Keep-Alive mechanism the client periodically (defined by the PingPongKeepAliveTime parameter) sends a double-CRLF (the "ping") then waits to receive a single CRLF (the "pong"). If the client does not receive a "pong" within an appropriate amount of time, it considers the flow failed. **Note:** The device sends a CRLF message to the Proxy Set only if the Proxy Keep-Alive feature (EnableProxyKeepAlive parameter) is enabled and its transport type is set to TCP or TLS. The device first sends a SIP OPTION message to establish the TCP/TLS connection and if it receives any SIP response, it continues sending the CRLF keep-alive sequences.
OVOC: Ping Pong Keep Alive Time **[PingPongKeepAliveTime]**	Defines the periodic interval (in seconds) after which a "ping" (double-CRLF) keep-alive is sent to a proxy/registrar, using the CRLF Keep-Alive mechanism. The default range is 5 to 2,000,000. The default is 120. The device uses the range of 80-100% of this user-defined value as the actual interval. For example, if the parameter value is set to 200 sec, the interval used is any random time between 160 to 200 seconds. This prevents an "avalanche" of keep-alive by multiple SIP UAs to a specific server.

43.8   General SIP Parameters

The general SIP parameters are described in the table below.

Table 43-31: General SIP Parameters

Parameter	Description
[IgnoreAuthorizationStale]	Enables the device to retry registering even if the last SIP 401\407 response included "stale=false". When the device initiates a REGISTER request with an Authorization header (according to the relevant configured credentials), and it receives a SIP 401\407 response with the stale parameter set to "false", by default the device doesn't try to send another REGISTER message. When the parameter is enabled, the device retries registering even if the last 401\407 response had "stale=false". ▪ [0] = (Default) If the device receives a SIP 401\407 response with "stale=true" or no stale parameter at all, it sends another REGISTER message. If "stale=false", the device doesn't send another REGISTER message.

Parameter	Description
	▪ [1] = If the device receives a SIP 401\407 response with "stale=false", it sends another REGISTER message. **Note:** This parameter is applicable only to REGISTER requests that the device initiates; it's not for REGISTER requests that the device forwards from the user to the registrar server.

[GwSDPConnectionMode]	Defines how the device displays the Connection ("c=") line ("c=") in the SDP Offer/Answer model.
	▪ [0] = (Default) The Connection ("c=") line is displayed as follows:
	   ✓ Offer: In the session description only.
	   ✓ Answer: In the session description and in each media ("m=") description.
	▪ [1] = For Offer and Answer, the Connection ("c=") line is displayed only in the session description; not in any media ("m=") descriptions.
	▪ [2] = The Connection ("c=") line is displayed only in media ("m=") descriptions.

Web: SIP Remote Reset CLI: sip-remote-reset **[EnableSIPRemoteReset]**	Enables a specific device action upon the receipt of a SIP NOTIFY request, where the action depends on the value received in the Event header. ▪ **[0]** Disable (default) ▪ **[1]** Enable The action depends on the Event header value: ▪ 'check-sync;reboot=false': triggers the regular Automatic Update feature (if Automatic Update has been enabled on the device) ▪ 'check-sync;reboot=true': triggers a device reset ▪ If the 'reboot' parameter is not specified, it defaults to 'true'. **Note:** The Event header value is proprietary to AudioCodes.
Web/OVOC: Max SIP Message Length **[KB]** **[MaxSIPMessageLength]**	Defines the maximum size (in Kbytes) for each SIP message that can be sent over the network. The device rejects messages exceeding this user-defined size. The valid value range is 1 to 50. The default is 50.
[SIPForceRport]	Determines whether the device sends SIP responses to the UDP port from where SIP requests are received even if the 'rport' parameter is not present in the SIP Via header. ▪ **[0]** = (Default) Disabled. The device sends the SIP response to the UDP port defined in the Via header. If the Via header contains the 'rport' parameter, the response is sent to the UDP port from where the SIP request is received. ▪ **[1]** = Enabled. SIP responses are sent to the UDP port from where SIP requests are received even if the 'rport' parameter is not present in the Via header.
Web: Reject Cancel after Connect CLI: reject-cancel-after-connect **[RejectCancelAfterConnect]**	Determines whether the device accepts or rejects a SIP CANCEL request received after the receipt of a 200 OK, during an established call. ▪ **[0]** = (Default) Accepts the CANCEL, by responding with a 200 OK and terminating the call session. ▪ **[1]** = Rejects the CANCEL, by responding with a SIP 481 Call/Transaction Does Not Exist, and maintaining the call session.

Parameter	Description
Web: Verify Received RequestURI CLI: verify-rcvd-requri **[VerifyReceevedRequestUri]**	Enables the device to reject SIP requests (such as ACK, BYE, or re-INVITE) whose user part in the Request-URI is different from the user part received in the Contact header of the last sent SIP request. ▪ **[0]** Disable = (Default) Even if the user is different, the device accepts the SIP request. ▪ **[1]** Enable = If the user is different, the device rejects the SIP request (BYE is responded with 481; re-INVITE is responded with 404; ACK is ignored).
Web: Max Number of Active Calls OVOC: Maximum Concurrent 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 is the maximum available channels (i.e., no restriction on the maximum number of calls).

Web: QoS statistics in SIP Release Call
[QoSStatistics]
	Enables the device to include call quality of service (QoS) statistics in SIP BYE and SIP 200 OK response to BYE, using the proprietary SIP header X-RTP-Stat.
	▪ **[0]** = Disable (default)
	▪ **[1]** = Enable
	The X-RTP-Stat header provides the following statistics:
	▪ Number of received and sent voice packets
	▪ Number of received and sent voice octets
	▪ Received packet loss, jitter (in ms), and latency (in ms)
	The X-RTP-Stat header contains the following fields:
	▪ PS=<voice packets sent>
	▪ OS=<voice octets sent>
	▪ PR=<voice packets received>
	▪ OR=<voice octets received>
	▪ PL=<receive packet loss>
	▪ JI=<jitter in ms>
	▪ LA=<latency in ms>
	Below is an example of the X-RTP-Stat header in a SIP BYE message:
	<pre>BYE sip:302@10.33.4.125 SIP/2.0
	Via: SIP/2.0/UDP
	10.33.4.126;branch=z9hG4bKac2127550866
	Max-Forwards: 70
	From:
	<sip:401@10.33.4.126;user=phone>;tag=1c2113553324
	To: <sip:302@company.com>;tag=1c991751121
	Call-ID: 991750671245200001912@10.33.4.125
	CSeq: 1 BYE
	X-RTP-Stat:
	PS=207;OS=49680;;PR=314;OR=50240;PL=0;JI=600;LA=40
	;
	Supported: em,timer,replaces,path,resource-
	priority
	Allow:
	REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRAC
	K,REFER,INFO,SUBSCRIBE,UPDATE
	User-Agent: Sip-Gateway-/v.6.2A.008.006</pre>

Parameter	Description
	`Reason: Q.850 ;cause=16 ;text="local"` `Content-Length: 0`
Web/OVOC: PRACK Mode **[PrackMode]**	Determines the PRACK (Provisional Acknowledgment) mechanism mode for SIP 1xx reliable responses. ▪ **[0]** Disable ▪ **[1]** Supported (default) ▪ **[2]** Required **Notes:** ▪ The Supported and Required headers contain the '100rel' tag. ▪ The device sends PRACK messages if 180/183 responses are received with '100rel' in the Supported or Required headers.
Web/OVOC: Enable Early Media **[EnableEarlyMedia]**	Enables the Early Media feature. Enables the device to send a 183 Session Progress response with SDP instead of a 180 Ringing, allowing the media stream to be established prior to the answering of the call. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ To send a 183 response, you must also set the parameter ProgressIndicator2IP to 1. If it is equal to 0, 180 Ringing response is sent. ▪ This feature can also be configured as an IP Profile and/or Tel Profile.
Web: 183 Message Behavior OVOC: SIP 183 Behaviour **[SIP183Behaviour]**	Defines the response of the device upon receipt of a SIP 183 response. ▪ **[0]** Progress = (Default) A 183 response (without SDP) does not cause the device to play a ringback tone. ▪ **[1]** Alert = 183 response is handled by the device as if a 180 Ringing response is received, and the device plays a ringback tone.
Web: Session-Expires Time OVOC: Sip Session Expires **[SIPSessionExpires]**	Defines the numerical value sent in the Session-Expires header in the first INVITE request or response (if the call is answered). The valid range is 1 to 86,400 sec. The default is 0 (i.e., the Session-Expires header is disabled).
Web: Minimum Session-Expires OVOC: Minimal Session Refresh Value **[MinSE]**	Defines the time (in seconds) that is used in the Min-SE header. This header defines the minimum time that the user agent refreshes the session. The valid range is 10 to 100,000. The default is 90.
Web/OVOC: Session Expires Disconnect Time CLI: session-exp-disconnect-time **[SessionExpiresDisconnectTime]**	Defines a session expiry timeout. The device disconnects the session (sends a SIP BYE) if the refresher does not send a refresh request before one-third (1/3) of the session expires time, or before the time configured by this parameter (the minimum of the two). The valid range is 0 to 32 (in seconds). The default is 32.

Parameter	Description
Web/OVOC: Session Expires Method **[SessionExpiresMethod]**	Determines the SIP method used for session-timer updates. ▪ **[0]** Re-INVITE = (Default) Uses Re-INVITE messages for session-timer updates. ▪ **[1]** UPDATE = Uses UPDATE messages. **Notes:** ▪ The device can receive session-timer refreshes using both methods. ▪ The UPDATE message used for session-timer is excluded from the SDP body.
[RemoveToTagInFailure Response]	Determines whether the device removes the 'to' header tag from final SIP failure responses to INVITE transactions. ▪ **[0]** = (Default) Do not remove tag. ▪ **[1]** = Remove tag.
[EnableRTCPAttribute]	Enables the use of the 'rtcp' attribute in the outgoing SDP. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
OVOC: Options User Part **[OPTIONSUserPart]**	Defines the user part value of the Request-URI for outgoing SIP OPTIONS requests. If no value is configured, the endpoint number 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. The default is an empty string ('').

Web: Fax Signaling Method OVOC: Fax Used **[IsFaxUsed]**	Determines the SIP signaling method for establishing and transmitting a fax session after a fax is detected.
	▪ **[0]** No Fax = (Default) No fax negotiation using SIP signaling. Fax transport method is according to the parameter FaxTransportMode.
	▪ **[1]** T.38 Relay = Initiates T.38 fax relay.
	▪ **[2]** G.711 Transport = Initiates fax/modem using the coder G.711 A-law/Mu-law with adaptations (see Note below).
	▪ **[3]** Fax Fallback = Initiates T.38 fax relay. If the T.38 negotiation fails, the device re-initiates a fax session using the coder G.711 A-law/μ-law with adaptations (see the Note below).
	Notes:
	▪ Fax adaptations (for options 2 and 3):
	  ✓ Echo Canceller = On
	  ✓ Silence Compression = Off
	  ✓ Echo Canceller Non-Linear Processor Mode = Off
	  ✓ Dynamic Jitter Buffer Minimum Delay = 40
	  ✓ Dynamic Jitter Buffer Optimization Factor = 13
	▪ If the device initiates a fax session using G.711 (option 2 and possibly 3), a 'gpmd' attribute is added to the SDP in the following format:
	  ✓ **For A-law:** 'a=gpmd:8 vbd=yes;ecan=on'
	  ✓ **For μ-law:** 'a=gpmd:0 vbd=yes;ecan=on'
	▪ When this parameter is set to 1, 2, or 3, the parameter FaxTransportMode is ignored.
	▪ When this parameter is set to 0, T.38 might still be used without the control protocol's involvement. To completely disable T.38, set FaxTransportMode to a value other than 1.

Parameter	Description
	▪ This parameter can also be configured per IP Profile (using the IPProfile parameter). ▪ For more information on fax transport methods, see 'Fax/Modem Transport Modes' on page 161.
[HandleG711asVBD]	Enables the handling of G.711 as G.711 VBD coder. ▪ **[0]** = (Default) Disable. The device negotiates G.711 as a regular audio coder and sends an answer only with G.729 coder. For example, if the device is configured with G.729 and G.711 VBD coders and it receives an INVITE with an SDP offer containing G.729 and "regular" G.711 coders, it sends an SDP answer containing only the G.729 coder. ▪ **[1]** = Enable. The device assumes that the G.711 coder received in the INVITE SDP offer is a VBD coder. For example, if the device is configured with G.729 and G.711 VBD coders and it receives an INVITE with an SDP offer containing G.729 and "regular" G.711 coders, it sends an SDP answer containing G.729 and G.711 VBD coders, allowing a subsequent bypass (passthrough) session if fax/modem signals are detected during the call. **Note:** This parameter is applicable only if G.711 VBD coder(s) with regular G.711 payload types 0 or 8 are configured for the device (using the CodersGroup parameter).
[FaxVBDBehavior]	Determines the device's fax transport behavior when G.711 VBD coder is negotiated at call start. ▪ **[0]** = (Default) If the device is configured with a VBD coder (see the CodersGroup parameter) and is negotiated OK at call start, then both fax and modem signals are sent over RTP using the bypass payload type (and no mid-call VBD or T.38 Re-INVITEs occur). ▪ **[1]** = If the IsFaxUsed parameter is set to 1, the channel opens with the FaxTransportMode parameter set to 1 (relay). This is required to detect mid-call fax tones and to send T.38 Re-INVITE messages upon fax detection. If the remote party supports T.38, the fax is relayed over T.38. **Notes:** ▪ If VBD coder negotiation fails at call start and if the IsFaxUsed parameter is set to 1 (or 3), then the channel opens with the FaxTransportMode parameter set to 1 (relay) to allow future detection of fax tones and sending of T.38 Re-INVITES. In such a scenario, the FaxVBDBehavior parameter has no effect. ▪ This feature can be used only if the remote party supports T.38 fax relay; otherwise, the fax fails.
[NoAudioPayloadType]	Defines the payload type of the outgoing SDP offer. The valid value range is 96 to 127 (dynamic payload type). The default is 0 (i.e. NoAudio is not supported). For example, if set to 120, the following is added to the INVITE SDP: `a=rtpmap:120 NoAudio/8000\r\n` **Note:** For incoming SDP offers, NoAudio is always supported.
Web: SIP Transport Type OVOC: Transport Type **[SIPTransportType]**	Determines the default transport layer for outgoing SIP calls initiated by the device. ▪ **[0]** UDP (default) ▪ **[1]** TCP ▪ **[2]** TLS (SIPS)

Parameter	Description
	Notes: ▪ It's recommended to use TLS for communication with a SIP Proxy and not for direct device-to-device communication. ▪ For received calls (i.e., incoming), the device accepts all these protocols. ▪ The value of this parameter is also used by the SAS application as the default transport layer for outgoing SIP calls.
Web: SIP UDP Local Port OVOC: Local SIP Port **[LocalSIPPort]**	Defines the local UDP port for SIP messages. The valid range is 1 to 65534. The default is 5060.
Web: SIP TCP Local Port OVOC: TCP Local SIP Port **[TCPLocalSIPPort]**	Defines the local TCP port for SIP messages. The valid range is 1 to 65535. The default is 5060.
Web: SIP TLS Local Port OVOC: TLS Local SIP Port **[TLSLocalSIPPort]**	Defines the local TLS port for SIP messages. The valid range is 1 to 65535. The default is 5061. **Note:** The value of this parameter must be different from the value of the parameter TCPLocalSIPPort.
Web/OVOC: Enable SIPS **[EnableSIPS]**	Enables secured SIP (SIPS URI) connections over multiple hops. ▪ **[0]** Disable (default) ▪ **[1]** Enable When the SIPTransportType parameter is set to 2 (i.e., TLS) and the parameter EnableSIPS is disabled, TLS is used for the next network hop only. When the parameter SIPTransportType is set to 2 or 1 (i.e., TCP or TLS) and EnableSIPS is enabled, TLS is used through the entire connection (over multiple hops). **Note:** If this parameter is enabled and the parameter SIPTransportType is set to 0 (i.e., UDP), the connection fails.
Web/OVOC: Enable TCP Connection Reuse **[EnableTCPConnectionReuse]**	Enables the reuse of the same TCP connection for all calls to the same destination. ▪ **[0]** Disable = Uses a separate TCP connection for each call. ▪ **[1]** Enable = (Default) Uses the same TCP connection for all calls. **Note:** For the SAS application, this feature is configured using the SASConnectionReuse parameter.
Web: Fake TCP alias **[FakeTCPalias]**	Enables the re-use of the same TCP/TLS connection for sessions with the same user, even if the "alias" parameter is not present in the SIP Via header of the first INVITE. ▪ **[0]** Disable = (Default) TCP/TLS connection reuse is done only if the "alias" parameter is present in the Via header of the first INVITE. ▪ **[1]** Enable **Note:** To enable TCP/TLS connection re-use, set the EnableTCPConnectionReuse parameter to 1.
Web/OVOC: Reliable Connection Persistent Mode **[ReliableConnectionPersistentMode]**	Enables setting of all TCP/TLS connections as persistent and therefore, not released. ▪ **[0]** = (Default) Disable. All TCP connections (except those that are set to a proxy IP) are released if not used by any SIP dialog\transaction.

Parameter	Description
	▪ **[1]** = Enable - TCP connections to all destinations are persistent and not released unless the device reaches 70% of its maximum TCP resources. While trying to send a SIP message connection, reuse policy determines whether live connections to the specific destination are re-used. Persistent TCP connection ensures less network traffic due to fewer setting up and tearing down of TCP connections and reduced latency on subsequent requests due to avoidance of initial TCP handshake. For TLS, persistent connection may reduce the number of costly TLS handshakes to establish security associations, in addition to the initial TCP connection set up. **Note:** If the destination is a Proxy server, the TCP/TLS connection is persistent regardless of the settings of this parameter.
Web/OVOC: TCP Timeout **[SIPTCPTimeout]**	Defines the Timer B (INVITE transaction timeout timer) and Timer F (non-INVITE transaction timeout timer), as defined in RFC 3261, when the SIP Transport Type is TCP. The valid range is 0 to 40 sec. The default is 64 multiplied by the SipT1Rtx parameter value. For example, if SipT1Rtx is set to 500 msec, then the default of SIPTCPTimeout is 32 sec.
Web: SIP Destination Port OVOC: Destination Port **[SIPDestinationPort]**	Defines the SIP destination port for sending initial SIP requests. The valid range is 1 to 65534. The default port is 5060. **Note:** SIP responses are sent to the port specified in the Via header.
Web: Use user=phone in SIP URL OVOC: Is User Phone **[IsUserPhone]**	Determines whether the 'user=phone' string is added to the SIP URI and SIP To header. ▪ **[0]** No = 'user=phone' string is not added. ▪ **[1]** Yes = (Default) 'user=phone' string is part of the SIP URI and SIP To header.
Web: Use user=phone in From Header OVOC: Is User Phone In From **[IsUserPhoneInFrom]**	Determines whether the 'user=phone' string is added to the From and Contact SIP headers. ▪ **[0]** No = (Default) Doesn't add 'user=phone' string. ▪ **[1]** Yes = 'user=phone' string is part of the From and Contact headers.
Web: Use Tel URI for Asserted Identity **[UseTelURIForAssertedID]**	Determines the format of the URI in the P-Asserted-Identity and P-Preferred-Identity headers. ▪ **[0]** Disable = (Default) 'sip:' ▪ **[1]** Enable = 'tel:'
Web: Tel to IP No Answer Timeout OVOC: IP Alert Timeout **[IPAlertTimeout]**	Defines the time (in seconds) that the device waits for a 200 OK response from the called party (IP side) after sending an INVITE message. If the timer expires, the call is released. The valid range is 0 to 3600. The default is 180.
Web: Enable Remote Party ID OVOC: Enable RPI Header **[EnableRPIheader]**	Enables Remote-Party-Identity headers for calling and called numbers for Tel-to-IP calls. ▪ **[0]** Disable (default). ▪ **[1]** Enable = Remote-Party-Identity headers are generated in SIP INVITE messages for both called and calling numbers.
Web: Enable History-Info Header OVOC: Enable History	Enables usage of the History-Info header. ▪ **[0]** Disable (default)

Parameter	Description

Info
[EnableHistoryInfo]
	• **[1]** Enable
	User Agent Client (UAC) Behavior:
	• Initial request: The History-Info header is equal to the Request-URI. If a PSTN Redirect number is received, it is added as an additional History-Info header with an appropriate reason.
	• Upon receiving the final failure response, the device copies the History-Info as is, adds the reason of the failure response to the last entry, and concatenates a new destination to it (if an additional request is sent). The order of the reasons is as follows:
	  a. Q.850 Reason
	  b. SIP Reason
	  c. SIP Response code
	• Upon receiving the final response (success or failure), the device searches for a Redirect reason in the History-Info (i.e., 3xx/4xx SIP reason). If found, it is passed to ISDN according to the following table:
	
	SIP Reason Code / ISDN Redirecting Reason
	302 - Moved Temporarily → Call Forward Universal (CFU)
	408 - Request Timeout → Call Forward No Answer (CFNA)
	480 - Temporarily Unavailable
	487 - Request Terminated
	486 - Busy Here → Call Forward Busy (CFB)
	600 - Busy Everywhere
	
	• If history reason is a Q.850 reason, it is translated to the SIP reason (according to the SIP-ISDN tables) and then to ISDN Redirect reason according to the table above.
	User Agent Server (UAS) Behavior:
	• The History-Info header is sent only in the final response.
	• Upon receiving a request with History-Info, the UAS checks the policy in the request. If a 'session', 'header', or 'history' policy tag is found, the (final) response is sent without History-Info; otherwise, it is copied from the request.

Web: Use Tgrp Information OVOC: Use SIP Tgrp **[UseSIPTgrp]**	Determines whether the SIP 'tgrp' parameter is used. This SIP parameter specifies the Hunt Group to which the call belongs (according to RFC 4904). For example, the SIP message below indicates that the call belongs to Hunt Group ID 1: `INVITE sip::+16305550100;tgrp=1;trunk-context=example.com@10.1.0.3;user=phone SIP/2.0` • **[0]** Disable = (Default) The 'tgrp' parameter isn't used. • **[1]** Send Only = The Hunt Group number or name (configured in the Hunt Group Settings) is added to the 'tgrp' parameter value in the Contact header of outgoing SIP messages. If a Hunt Group number / name is not associated with the call, the 'tgrp' parameter isn't included. If a 'tgrp' value is specified in incoming messages, it is ignored. • **[2]** Send and Receive = The functionality of outgoing SIP messages is identical to the functionality described for option **[1].** In addition, for incoming SIP INVITEs, if the Request-URI includes a 'tgrp' parameter, the device routes the call according to that value (if possible). The Contact header in the outgoing SIP INVITE (Tel-to-IP

Parameter	Description
	call) contains "tgrp=<source trunk group ID>;trunk-context=<gateway IP address>". The <source trunk group ID> is the Hunt Group ID where incoming calls from Tel is received. For IP-Tel calls, the SIP 200 OK device's response contains "tgrp=<destination trunk group ID>;trunk-context=<gateway IP address>". The <destination trunk group ID> is the Hunt Group ID used for outgoing Tel calls. The <gateway IP address> in "trunk-context" can be configured using the SIPGatewayName parameter. **Note:** IP-to-Tel configuration (using the PSTNPrefix parameter) overrides the 'tgrp' parameter in incoming INVITE messages.
Web/OVOC: TGRP Routing Precedence **[TGRProutingPrecedence]**	Determines the precedence method for routing IP-to-Tel calls - according to the IP to Hunt Group Routing Table or according to the SIP 'tgrp' parameter. ▪ **[0]** = (Default) IP-to-Tel routing is determined by the IP to Hunt Group Routing Table (PSTNPrefix parameter). If a matching rule is not found in this table, the device uses the Hunt Group parameters for routing the call. ▪ **[1]** = The device first places precedence on the 'tgrp' parameter for IP-to-Tel routing. If the received INVITE Request-URI does not contain the 'tgrp' parameter or if the Hunt Group number is not defined, then the IP to Hunt Group Routing Table is used for routing the call. Below is an example of an INVITE Request-URI with the 'tgrp' parameter, indicating that the IP call should be routed to Hunt Group 7: INVITE sip:200;tgrp=7;trunk-context=example.com@10.33.2.68;user=phone SIP/2.0 **Notes:** ▪ For enabling routing based on the 'tgrp' parameter, the UseSIPTgrp parameter must be set to 2. ▪ For IP-to-Tel routing based on the 'dtg' parameter (instead of the 'tgrp' parameter), use the parameter UseBroadsoftDTG.
[UseBroadsoftDTG]	Determines whether the device uses the 'dtg' parameter for routing IP-to-Tel calls to a specific Hunt Group. ▪ **[0]** Disable (default) ▪ **[1]** Enable When this parameter is enabled, if the Request-URI in the received SIP INVITE includes the 'dtg' parameter, the device routes the call to the Hunt Group according to its value. This parameter is used instead of the 'tgrp/trunk-context' parameters. The 'dtg' parameter appears in the INVITE Request-URI (and in the To header). For example, the received SIP message below routes the call to Hunt Group ID 56: INVITE sip:123456@192.168.1.2;dtg=56;user=phone SIP/2.0 **Note:** If the Hunt Group is not found based on the 'dtg' parameter, the IP to Hunt Group Routing Table is used instead for routing the call to the appropriate Hunt Group.
Web/OVOC: 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)

Parameter	Description

	▪ **[1]** Enable
	A GRUU is a SIP URI that routes to an instance-specific UA and can be reachable from anywhere. There are a number of contexts in which it is desirable to have an identifier that addresses a single UA (using GRUU) rather than the group of UA's indicated by an Address of Record (AOR). For example, in call transfer where user A is talking to user B, and user A wants to transfer the call to user C. User A sends a REFER to user C:
	<pre>REFER sip:C@domain.com SIP/2.0
	From: sip:A@domain.com;tag=99asd
	To: sip:C@domain.com
	Refer-To: (URI that identifies B's UA)</pre>
	The Refer-To header needs to contain a URI that user C can use to place a call to user B. This call needs to route to the specific UA instance that user B is using to talk to user A. User B should provide user A with a URI that has to be usable by anyone. It needs to be a GRUU.
	
	▪ Obtaining a GRUU: The mechanism for obtaining a GRUU is through registrations. A UA can obtain a GRUU by generating a REGISTER request containing a Supported header field with the value "gruu". The UA includes a "+sip.instance" Contact header parameter of each contact for which the GRUU is desired. This Contact parameter contains a globally unique ID that identifies the UA instance. The global unique ID is created from one of the following:
	  ✓ If the REGISTER is per the device's client (endpoint), it is the MAC address concatenated with the phone number of the client.
	  ✓ If the REGISTER is per device, it is the MAC address only.
	  ✓ When using TP, "User Info" can be used for registering per endpoint. Thus, each endpoint can get a unique id – its phone number. The globally unique ID in TP is the MAC address concatenated with the phone number of the endpoint.
	
	If the remote server doesn't support GRUU, it ignores the parameters of the GRUU. Otherwise, if the remote side also supports GRUU, the REGISTER responses contain the "gruu" parameter in each Contact header. This parameter contains a SIP or SIPS URI that represents a GRUU corresponding to the UA instance that registered the contact. The server provides the same GRUU for the same AOR and instance-id when sending REGISTER again after registration expiration. RFC 5627 specifies that the remote target is a GRUU target if its' Contact URL has the "gr" parameter with or without a value.
	
	▪ Using GRUU: The UA can place the GRUU in any header field that can contain a URI. It must use the GRUU in the following messages: INVITE request, its 2xx response, SUBSCRIBE request, its 2xx response, NOTIFY request, REFER request and its 2xx response.

OVOC: Is CISCO Sce Mode **[IsCiscoSCEMode]**	Determines whether a Cisco gateway exists at the remote side. ▪ **[0]** = (Default) No Cisco gateway exists at the remote side. ▪ **[1]** = A Cisco gateway exists at the remote side. When a Cisco gateway exists at the remote side, the device must set the value of the 'annexb' parameter of the fmtp attribute in the SDP to 'no'. This logic is used if Silence Suppression for the used coder is configured to 2 (enable without adaptation). In this case, Silence

Parameter	Description
	Suppression is used on the channel but not declared in the SDP. **Note:** The IsCiscoSCEMode parameter is applicable only when the selected coder is G.729.

Web: User-Agent Information
OVOC: User Agent Display Info
[UserAgentDisplayInfo]
	Defines the string that is used in the SIP User-Agent and Server response headers. When configured, the string <UserAgentDisplayInfo value>/software version' is used, for example:
	`User-Agent: myproduct/v.6.40.010.006`
	If not configured, the default string, <AudioCodes product-name>/software version' is used, for example:
	`User-Agent: Audiocodes-Sip-Gateway-`
	`MediaPack/v.6.40.010.006`
	The maximum string length is 50 characters.
	Note: The software version number and preceding forward slash (/) cannot be modified. Therefore, it is recommended not to include a forward slash in the parameter's value (to avoid two forward slashes in the SIP header, which may cause problems).

Web/OVOC: SDP Session Owner
[SIPSDPSessionOwner]
	Defines the value of the Owner line ('o' field) in outgoing SDP messages.
	The valid range is a string of up to 39 characters. The default is 'AudiocodesGW'.
	For example:
	`o=AudiocodesGW 1145023829 1145023705 IN IP4`
	`10.33.4.126`

[EnableSDPVersionNegotiation]	Enables the device to ignore new SDP re-offers (from the media negotiation perspective) in certain scenarios (such as session expires). According to RFC 3264, once an SDP session is established, a new SDP offer is considered a new offer only when the SDP origin value is incremented. In scenarios such as session expires, SDP negotiation is irrelevant and thus, the origin field is not changed. Even though some SIP devices don't follow this behavior and don't increment the origin value even in scenarios where they want to re-negotiate, the device can assume that the remote party operates according to RFC 3264, and in cases where the origin field is not incremented, the device does not re-negotiate SDP capabilities. ▪ **[0]** Disable = (Default) The device negotiates any new SDP re-offer, regardless of the origin field. ▪ **[1]** Enable = The device negotiates only an SDP re-offer with an incremented origin field.
Web/OVOC: Subject **[SIPSubject]**	Defines the Subject header value in outgoing INVITE messages. If not specified, the Subject header isn't included (default). The maximum length is up to 50 characters.
[CoderPriorityNegotiation]	Defines the priority for coder negotiation in the incoming SDP offer, between the device's or remote UA's coder list. ▪ **[0]** = (Default) Coder negotiation is given higher priority to the remote UA's list of supported coders. ▪ **[1]** = Coder negotiation is given higher priority to the device's (local) supported coders list. ▪ Note: This parameter is applicable only to the Gateway/IP-to-IP application.

Parameter	Description
Web: Send All Coders on Retrieve **[SendAllCodersOnRetrieve]**	Enables coder re-negotiation in the sent re-INVITE for retrieving an on-hold call. ▪ **[0]** Disable = (Default) Sends only the initially chosen coder when the call was first established and then put on-hold. ▪ **[1]** Enable = Includes all supported coders in the SDP of the re-INVITE sent to the call made un-hold (retrieved). The used coder is therefore, re-negotiated. This parameter is useful in the following call scenario example: **1** Party A calls party B and coder G.711 is chosen. **2** Party B is put on-hold while Party A blind transfers Party B to Party C. **3** Party C answers and Party B is made un-hold. However, as Party C supports only G.729 coder, re-negotiation of the supported coder is required.
Web: Multiple Packetization Time Format OVOC: Multi Ptime Format **[MultiPtimeFormat]**	Determines whether the 'mptime' attribute is included in the outgoing SDP. ▪ **[0]** None = (Default) Disabled. ▪ **[1]** PacketCable = Includes the 'mptime' attribute in the outgoing SDP - PacketCable-defined format. The mptime' attribute enables the device to define a separate packetization period for each negotiated coder in the SDP. The 'mptime' attribute is only included if this parameter is enabled even if the remote side includes it in the SDP offer. Upon receipt, each coder receives its 'ptime' value in the following precedence: from 'mptime' attribute, from 'ptime' attribute, and then from default value.
OVOC: Enable P Time **[EnablePtime]**	Determines whether the 'ptime' attribute is included in the SDP. ▪ **[0]** = Remove the 'ptime' attribute from SDP. ▪ **[1]** = (Default) Include the 'ptime' attribute in SDP.
Web/OVOC: 3xx Behavior **[3xxBehavior]**	Determines the device's behavior regarding call identifiers when a 3xx response is received for an outgoing INVITE request. The device can either use the same call identifiers (Call-ID, To, and From tags) or change them in the new initiated INVITE. ▪ **[0]** Forward = (Default) Use different call identifiers for a redirected INVITE message. ▪ **[1]** Redirect = Use the same call identifiers.
Web/OVOC: Enable P-Charging Vector **[EnablePChargingVector]**	Enables the inclusion of the P-Charging-Vector header to all outgoing INVITE messages. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web/OVOC: Retry-After Time **[RetryAfterTime]**	Defines the time (in seconds) used in the Retry-After header when a 503 (Service Unavailable) response is generated by the device. The time range is 0 to 3,600. The default is 0.
Web/OVOC: Fake Retry After **[sec]** **[FakeRetryAfter]**	Determines whether the device, upon receipt of a SIP 503 response without a Retry-After header, behaves as if the 503 response included a Retry-After header and with the period (in seconds) specified by this parameter. ▪ **[0]** Disable (default) ▪ Any positive value (in seconds) for defining the period

Parameter	Description
	When enabled, this feature allows the device to operate with Proxy servers that do not include the Retry-After SIP header in SIP 503 (Service Unavailable) responses to indicate an unavailable service. The Retry-After header is used with the 503 (Service Unavailable) response to indicate how long the service is expected to be unavailable to the requesting SIP client. The device maintains a list of available proxies, by using the Keep-Alive mechanism. The device checks the availability of proxies by sending SIP OPTIONS every keep-alive timeout to all proxies. If the device receives a SIP 503 response to an INVITE, it also marks that the proxy is out of service for the defined "Retry-After" period.
Web/OVOC: Enable P-Associated-URI Header **[EnablePAssociatedURI Header]**	Determines the device usage of the P-Associated-URI header. This header can be received in 200 OK responses to REGISTER requests. When enabled, the first URI in the P-Associated-URI header is used in subsequent requests as the From/P-Asserted-Identity headers value. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** P-Associated-URIs in registration responses is handled only if the device is registered per endpoint (using the User Information file).

Web/OVOC: Source Number Preference **[SourceNumberPreferen ce]**	Determines from which SIP header the source (calling) number is obtained in incoming INVITE messages.
	▪ If not configured (i.e., empty string) or if any string other than "From" or "Pai2" is configured, the calling number is obtained from a specific header using the following logic:
	  a. P-Preferred-Identity header.
	  b. If the above header is not present, then the first P-Asserted-Identity header is used.
	  c. If the above header is not present, then the Remote-Party-ID header is used.
	  d. If the above header is not present, then the From header is used.
	▪ "**From**" = The calling number is obtained from the From header.
	▪ "**Pai2**" = The calling number is obtained using the following logic:
	  a. If a P-Preferred-Identity header is present, the number is obtained from it.
	  b. If no P-Preferred-Identity header is present and two P-Asserted-Identity headers are present, the number is obtained from the second P-Asserted-Identity header.
	  c. If only one P-Asserted-Identity header is present, the calling number is obtained from it.
	Notes:
	▪ The "From" and "Pai2" values are not case-sensitive.
	▪ Once a URL is selected, all the calling party parameters are set from this header. If P-Asserted-Identity is selected and the Privacy header is set to 'id', the calling number is assumed restricted.

[SelectSourceHeaderFor CalledNumber]	Determines the SIP header used for obtaining the called number (destination) for IP-to-Tel calls. ▪ **[0]** Request-URI header = (Default) Obtains the destination number from the user part of the Request-URI. ▪ **[1]** To header = Obtains the destination number from the user part of the To header.

Parameter	Description
	▪ **[2]** P-Called-Party-ID header = Obtains the destination number from the P-Called-Party-ID header.
Web/OVOC: Forking Handling Mode **[ForkingHandlingMode]**	Determines how the device handles the receipt of multiple SIP 18x forking responses for Tel-to-IP calls. The forking 18x response is the response with a different SIP to-tag than the previous 18x response. These responses are typically generated (initiated) by Proxy / Application servers that perform call forking, sending the device's originating INVITE (received from SIP clients) to several destinations, using the same CallID.
	▪ **[0]** Parallel handling = (Default) If SIP 18x with SDP is received, the device opens a voice stream according to the received SDP and disregards any subsequently received 18x forking responses (with or without SDP). If the first response is 180 without SDP, the device responds according to the PlayRBTone2TEL parameter and disregards the subsequent forking 18x responses.
	▪ **[1]** Sequential handling = If 18x with SDP is received, the device opens a voice stream according to the received SDP. The device re-opens the stream according to subsequently received 18x responses with SDP, or plays a ringback tone if 180 response without SDP is received. If the first received response is 180 without SDP, the device responds according to the PlayRBTone2TEL parameter and processes the subsequent 18x forking responses.
	Note: Regardless of this parameter setting, once a SIP 200 OK response is received, the device uses the RTP information and re-opens the voice stream, if necessary.
Web: Forking Timeout **[ForkingTimeOut]**	Defines the timeout (in seconds) that is started after the first SIP 2xx response has been received for a User Agent when a Proxy server performs call forking (Proxy server forwards the INVITE to multiple SIP User Agents). The device sends a SIP ACK and BYE in response to any additional SIP 2xx received from the Proxy within this timeout. Once this timeout elapses, the device ignores any subsequent SIP 2xx.
	The number of supported forking calls per channel is 4. In other words, for an INVITE message, the device can receive up to 4 forking responses from the Proxy server.
	The valid range is 0 to 30. The default is 30.
[ForkingDelayTimeForInvite]	Defines the interval (in seconds) to wait before sending INVITE messages to the other members of the forking group. The INVITE is immediately sent to the first member.
	The valid value range is 0 to 40. The default is 0 (i.e., sends immediately).
Web: Tel2IP Call Forking Mode **[Tel2IPCallForkingMode]**	Enables Tel-to-IP call forking, whereby a Tel call can be routed to multiple IP destinations.
	▪ **[0]** Disable (default)
	▪ **[1]** Enable
	Note: Once enabled, routing rules must be assigned Forking Groups in the Outbound IP Routing table.
Web/OVOC: Enable Reason Header **[EnableReasonHeader]**	Enables the usage of the SIP Reason header.
	▪ **[0]** Disable
	▪ **[1]** Enable (default)

Parameter	Description
Web/OVOC: Gateway Name CLI: gw-name **[SIPGatewayName]**	Defines a name for the device (e.g., device123.com). This name is used as the host part of the SIP URI in the From header. If not specified, the device's IP address is used instead (default). **Notes:** ▪ Ensure that the parameter value is the one with which the Proxy has been configured with to identify the device. ▪ This parameter can also be configured for an IP Group (in the IP Group table).
[ZeroSDPHandling]	Determines the device's response to an incoming SDP that includes an IP address of 0.0.0.0 in the SDP's Connection Information field (i.e., "c=IN IP4 0.0.0.0"). ▪ **[0]** = (Default) Sets the IP address of the outgoing SDP's c= field to 0.0.0.0. ▪ **[1]** = Sets the IP address of the outgoing SDP c= field to the IP address of the device. If the incoming SDP doesn't contain the "a=inactive" line, the returned SDP contains the "a=recvonly" line.
Web/OVOC: Enable Delayed Offer **[EnableDelayedOffer]**	Determines whether the device sends the initial INVITE message with or without an SDP. Sending the first INVITE without SDP is typically done by clients for obtaining the far-end's full list of capabilities before sending their own offer. (An alternative method for obtaining the list of supported capabilities is by using SIP OPTIONS, which is not supported by every SIP agent.) ▪ **[0]** Disable = (Default) The device sends the initial INVITE message with an SDP. ▪ **[1]** Enable = The device sends the initial INVITE message without an SDP.
[DisableCryptoLifeTimeInSDP]	Enables the device to send "a=crypto" lines without the lifetime parameter in the SDP. For example, if the SDP contains "a=crypto:12 AES_CM_128_HMAC_SHA1_80 inline:hhQe10yZRcRcpIFPkH5xYY9R1de37ogh9G1MpvNp|2^31", it removes the lifetime parameter "2^31". ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web/OVOC: Enable 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
[AnonymousMode]	Determines whether the device's IP address is used as the URI host part instead of "anonymous.invalid" in the INVITE's From header for Tel-to-IP calls. ▪ **[0]** = (Default) If the device receives a call from the Tel with blocked caller ID, it sends an INVITE with From: "anonymous"<anonymous@anonymous.invalid> ▪ **[1]** = The device's IP address is used as the URI host part instead of "anonymous.invalid". This parameter may be useful, for example, for service providers who identify their SIP Trunking customers by their source phone number or IP address, reflected in the From header of the SIP INVITE. Therefore, even customers blocking their Caller ID can be identified by the service provider. Typically, if the device receives a call with blocked Caller ID

Parameter	Description
	from the PSTN side (e.g., Trunk connected to a PBX), it sends an INVITE to the IP with a From header as follows: From: "anonymous" <anonymous@anonymous.invalid>. This is in accordance with RFC 3325. However, when this parameter is set to 1, the device replaces the "anonymous.invalid" with its IP address.
OVOC: P Asserted User Name **[PAssertedUserName]**	Defines a 'representative number' (up to 50 characters) that is used as the user part of the Request-URI in the P-Asserted-Identity header of an outgoing INVITE for Tel-to-IP calls. The default is null.
OVOC: Use URL In Refer To Header **[UseAORInReferToHeader]**	Defines the source for the SIP URI set in the Refer-To header of outgoing REFER messages. ▪ **[0]** = (Default) Use SIP URI from Contact header of the initial call. ▪ **[1]** = Use SIP URI from To/From header of the initial call.
Web: Enable User-Information Usage **[EnableUserInfoUsage]**	Enables the usage of the User Information, which is loaded to the device in the User Information auxiliary file. For a description on User Information, see 'Loading Auxiliary Files' on page 365. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
[HandleReasonHeader]	Determines whether the device uses the value of the incoming SIP Reason header for Release Reason mapping. ▪ **[0]** = Disregard Reason header in incoming SIP messages. ▪ **[1]** = (Default) Use the Reason header value for Release Reason mapping.
[EnableSilenceSuppInSDP]	Determines the device's behavior upon receipt of SIP Re-INVITE messages that include the SDP's 'silencesupp:off' attribute. ▪ **[0]** = (Default) Disregard the 'silecesupp' attribute. ▪ **[1]** = Handle incoming Re-INVITE messages that include the 'silencesupp:off' attribute in the SDP as a request to switch to the Voice-Band-Data (VBD) mode. In addition, the device includes the attribute 'a=silencesupp:off' in its SDP offer. **Note:** This parameter is applicable only if the G.711 coder is used.
[EnableRport]	Enables the usage of the 'rport' parameter in the Via header. ▪ **[0]** = Disabled (default) ▪ **[1]** = Enabled The device adds an 'rport' parameter to the Via header of each outgoing SIP message. The first Proxy that receives this message sets the 'rport' value of the response to the actual port from where the request was received. This method is used, for example, to enable the device to identify its port mapping outside a NAT. If the Via header doesn't include the 'rport' parameter, the destination port of the response is obtained from the host part of the Via header. If the Via header includes the 'rport' parameter without a port value, the destination port of the response is the source port of the incoming request. If the Via header includes 'rport' with a port value (e.g., rport=1001), the destination port of the response is the port indicated in the 'rport' parmeter.

Parameter	Description

OVOC: X Channel Header **[XChannelHeader]**	Determines whether the SIP X-Channel header is added to SIP messages for providing information on the physical channel on which the call is received or placed.
	▪ **[0]** Disable = (Default) X-Channel header is not used.
	▪ **[1]** Enable = X-Channel header is generated by the device and sent in INVITE messages and 180, 183, and 200 OK SIP responses. The header includes the channel, and the device's IP address. For example, 'x-channel: DS/DS1-1/8;IP=192.168.13.1', where:
	   ✔ 'DS/DS-1' is a constant string
	   ✔ '1' is a constant string
	   ✔ '8' is the channel (port)
	   ✔ 'IP=192.168.13.1' is the device's IP address

Web/OVOC: Progress Indicator to IP **[ProgressIndicator2IP]**	For Analog (FXS/FXO) interfaces: ▪ **[-1]** Not Configured = (Default) Default values are used. The default for FXO interfaces is 1; The default for FXS interfaces is 0. ▪ **[0]** No PI = For IP-to-Tel calls, the device sends a 180 Ringing response to IP after placing a call to a phone (FXS) or PBX (FXO). ▪ **[1]** PI = 1, **[8]** PI = 8: For IP-to-Tel calls, if the parameter EnableEarlyMedia is set to 1, the device sends a 183 Session Progress message with SDP immediately after a call is placed to a phone/PBX. This is used to cut-through the voice path before the remote party answers the call. This allows the originating party to listen to network Call Progress Tones (such as ringback tone or other network announcements). **Note:** This parameter can also be configured per IP Profile (using the IPProfile parameter) and Tel Profile (using the TelProfile parameter).
[EnableRekeyAfter181]	Enables the device to send a re-INVITE with a new (different) SRTP key (in the SDP) if a SIP 181 response is received ("call is being forwarded"). The re-INVITE is sent immediately upon receipt of the 200 OK (when the call is answered). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Note:** This parameter is applicable only if SRTP is used.
[NumberOfActiveDialogs]	Defines the maximum number of concurrent, outgoing SIP REGISTER dialogs. This parameter is used to control the registration rate. The valid range is 1 to 5. The default is 5. **Notes:** ▪ Once a 200 OK is received in response to a REGISTER message, the REGISTER message is not considered in this maximum count limit. ▪ This parameter applies only to outgoing REGISTER messages (i.e., incoming is unlimited).
Web/OVOC: Default Release Cause **[DefaultReleaseCause]**	Defines the default Release Cause (sent to IP) for IP-to-Tel calls when the device initiates a call release and an explicit matching cause for this release is not found. The default release cause is NO_ROUTE_TO_DESTINATION (3). Other common values include NO_CIRCUIT_AVAILABLE (34), DESTINATION_OUT_OF_ORDER (27), etc. **Notes:**

Parameter	Description
	▪ The default release cause is described in the Q.931 notation and is translated to corresponding SIP 40x or 50x values (e.g., 3 to SIP 404, and 34 to SIP 503). ▪ For information on mapping PSTN release causes to SIP responses, see Mapping PSTN Release Cause to SIP Response. ▪ For a list of SIP responses-Q.931 release cause mapping, see 'Alternative Routing to Trunk upon Q.931 Call Release Cause Code' on page 265.
Web: 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 EnableMicrosofExt is enabled, the device modifies the called number by adding an "e" as the prefix, removing the "ext=" parameter, and adding the extension number as the suffix (e.g., **e**622125519100**104**). Once modified, the device can then manipulate the number further, using the Number Manipulation tables to leave only the last 3 digits (for example) for sending to a PBX.
OVOC: Use SIP URI For Diversion Header **[UseSIPURIForDiversionHeader]**	Defines the URI format in the SIP Diversion header. ▪ **[0]** = 'tel:' (default) ▪ **[1]** = 'sip:'
[TimeoutBetween100And18x]	Defines the timeout (in msec) between receiving a 100 Trying response and a subsequent 18x response. If a 18x response is not received within this timeout period, the call is disconnected. The valid range is 0 to 180,000 (i.e., 3 minutes). The default is 32000 (i.e., 32 sec).
[IgnoreRemoteSDPMKI]	Determines whether the device ignores the Master Key Identifier (MKI) if present in the SDP received from the remote side. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: Comfort Noise Generation Negotiation OVOC: Comfort Noise Generation **[ComfortNoiseNegotiation]**	Enables negotiation and usage of Comfort Noise (CN). ▪ **[0]** Disable ▪ **[1]** Enable (default) The use of CN is indicated by including a payload type for CN on the media description line of the SDP. The device can use CN with a codec whose RTP time stamp clock rate is 8,000 Hz (G.711/G.726). The static payload type 13 is used. The use of CN is negotiated between sides. Therefore, if the remote side doesn't support CN, it is not used. Regardless of the device's settings, it always attempts to adapt to the remote SIP UA's request for CNG, as described below. To determine CNG support, the device uses the ComfortNoiseNegotiation parameter and the codec's SCE (silence suppression setting) using the CodersGroup parameter.

Parameter	Description
	If the ComfortNoiseNegotiation parameter is enabled, then the following occurs: ▪ If the device is the initiator, it sends a "CN" in the SDP only if the SCE of the codec is enabled. If the remote UA responds with a "CN" in the SDP, then CNG occurs; otherwise, CNG does not occur. ▪ If the device is the receiver and the remote SIP UA does not send a "CN" in the SDP, then no CNG occurs. If the remote side sends a "CN", the device attempts to be compatible with the remote side and even if the codec's SCE is disabled, CNG occurs. If the ComfortNoiseNegotiation parameter is disabled, then the device does not send "CN" in the SDP. However, if the codec's SCE is enabled, then CNG occurs.
[SDPEcanFormat]	Defines the echo canceller format in the outgoing SDP. The 'ecan' attribute is used in the SDP to indicate the use of echo cancellation. ▪ **[0]** = (Default) The 'ecan' attribute appears on the 'a=gpmd' line. ▪ **[1]** = The 'ecan' attribute appears as a separate attribute. ▪ **[2]** = The 'ecan' attribute is not included in the SDP. ▪ **[3]** = The 'ecan' attribute and the 'vbd' parameter are not included in the SDP. **Note:** This parameter is applicable only when the IsFaxUsed parameter is set to 2, and for re-INVITE messages generated by the device as result of modem or fax tone detection.
Web/OVOC: First Call Ringback Tone ID **[FirstCallRBTId]**	Defines the index of the first ringback tone in the CPT file. This option enables an Application server to request the device to play a distinctive ringback tone to the calling party according to the destination of the call. The tone is played according to the Alert-Info header received in the 180 Ringing SIP response (the value of the Alert-Info header is added to the value of this parameter). The valid range is -1 to 1,000. The default is -1 (i.e., play standard ringback tone). **Notes:** ▪ It is assumed that all ringback tones are defined in sequence in the CPT file. ▪ In case of an MLPP call, the device uses the value of this parameter plus 1 as the index of the ringback tone in the CPT file (e.g., if this value is set to 1, then the index is 2, i.e., 1 + 1).
Web: Reanswer Time OVOC: Regret Time **[RegretTime]**	Defines the time interval from when the user hangs up the phone until the call is disconnected (FXS). This allows the user to hang up and then pick up the phone (before this timeout) to continue the call conversation. Thus, it's also referred to as regret time. The valid range is 0 to 255 (in seconds). The default is 0.
Web: Enable Reanswering Info **[EnableReansweringINFO]**	Enables the device to send a SIP INFO message with the On-Hook/Off-Hook parameter when the FXS phone goes on-hook during an ongoing call and then off-hook again, within the user-defined regret timeout (configured by the parameter RegretTime). Therefore, the device notifies the far-end that the call has been re-answered. ▪ **[0]** Disable (default) ▪ **[1]** Enable

Parameter	Description
	This parameter is typically implemented for incoming IP-to-Tel collect calls to the FXS port. If the FXS user does not wish to accept the collect call, the user disconnects the call by on-hooking the phone. The device notifies the softswitch (or Application server) of the unanswered collect call (on-hook) by sending a SIP INFO message. As a result, the softswitch disconnects the call (sends a BYE message to the device). If the call is a regular incoming call and the FXS user on-hooks the phone without intending to disconnect the call, the softswitch does not disconnect the call (during the regret time).
	The INFO message format is as follows:
	INFO sip:12345@10.50.228.164:5082 SIP/2.0 Via: SIP/2.0/UDP 127.0.0.1;branch=z9hG4bK_05_905924040-90579 From: <sip:+551137077803@ims.acme.com.br:5080;user=phone>;tag=008277765 To: <sip:notavailable@unknown.invalid>;tag=svw-0-1229428367 Call-ID: ConorCCR-0-LU-1229417827103300@dtas-stdn.fs5000group0-000.l CSeq: 1 INFO Contact: sip:10.20.7.70:5060 Content-Type: application/On-Hook (application/Off-Hook) Content-Length: 0
	Notes: ▪ This parameter is applicable only if the parameter RegretTime is configured. ▪ This parameter is applicable only to FXS interfaces.
Web: PSTN Alert Timeout OVOC: Trunk PSTN Alert Timeout **[PSTNAlertTimeout]**	Defines the Alert Timeout (in seconds) for calls to the Tel side. This timer is used between the time a ring is generated (FXS) or a line is seized (FXO), until the call is connected. For example: If the FXS device receives an INVITE, it generates a ring to the phone and sends a SIP 180 Ringing response to the IP. If the phone is not answered within the time interval set by this parameter, the device cancels the call by sending a SIP 408 response. The valid value range is 1 to 600 (in seconds). The default is 180.
Web/OVOC: RTP Only Mode **[RTPOnlyMode]**	Enables the device to send and receive RTP packets to and from remote endpoints without the need to establish a SIP session. The remote IP address is determined according to the Outbound IP Routing table (Prefix parameter). The port is the same port as the local RTP port (configured by the BaseUDPPort parameter and the channel on which the call is received). ▪ **[0]** Disable (default) ▪ **[1]** Transmit & Receive = Send and receive RTP packets. ▪ **[2]** Transmit Only= Send RTP packets only. ▪ **[3]** Receive Only= Receive RTP packets only. **Notes:** ▪ To configure the RTP Only mode per trunk, use the RTPOnlyModeForTrunk_ID parameter. ▪ If per trunk configuration (using the RTPOnlyModeForTrunk_ID parameter) is set to a value other than the default, the RTPOnlyMode parameter value is ignored.

Parameter	Description
Web/OVOC: SIT Q850 Cause **[SITQ850Cause]**	Defines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when a Special Information Tone (SIT) is detected on an IP-to-Tel call. The valid range is 0 to 127. The default is 34. **Notes:** ▪ For mapping specific SIT tones, you can use the SITQ850CauseForNC, SITQ850CauseForIC, SITQ850CauseForVC, and SITQ850CauseForRO parameters. ▪ This parameter is applicable only to FXO interfaces.
Web/OVOC: SIT Q850 Cause For NC **[SITQ850CauseForNC]**	Defines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when SIT-NC (No Circuit Found Special Information Tone) is detected from the TelPSTN for IP-to-Tel calls. The valid range is 0 to 127. The default is 34. **Notes:** ▪ When not configured (i.e., default), the SITQ850Cause parameter is used. ▪ This parameter is applicable only to FXO interfaces.
Web/OVOC: SIT Q850 Cause For IC **[SITQ850CauseForIC]**	Defines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when SIT-IC (Operator Intercept Special Information Tone) is detected from the Tel for IP-to-Tel calls. The valid range is 0 to 127. The default is -1 (not configured). **Notes:** ▪ When not configured (i.e., default), the SITQ850Cause parameter is used. ▪ This parameter is applicable only to FXO interfaces.
Web/OVOC: SIT Q850 Cause For VC **[SITQ850CauseForVC]**	Defines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when SIT-VC (Vacant Circuit - non-registered number Special Information Tone) is detected from the Tel for IP-to-Tel calls. The valid range is 0 to 127. The default is -1 (not configured). **Notes:** ▪ When not configured (i.e., default), the SITQ850Cause parameter is used. ▪ This parameter is applicable only to FXO interfaces.
Web/OVOC: SIT Q850 Cause For RO **[SITQ850CauseForRO]**	Defines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when SIT-RO (Reorder - System Busy Special Information Tone) is detected from the Tel for IP-to-Tel calls. The valid range is 0 to 127. The default is -1 (not configured). **Notes:** ▪ When not configured (i.e., default), the SITQ850Cause parameter is used. ▪ This parameter is applicable only to FXO interfaces.
Out-of-Service (Busy Out) Parameters	
Web/OVOC: Enable Busy Out **[EnableBusyOut]**	Enables the Busy Out feature. ▪ **[0]** Disable (Default) ▪ **[1]** Enable

Parameter	Description
	When Busy Out is enabled and certain scenarios exist, the device does the following:
	The FXS port behaves according to the settings of the FXSOOSBehavior parameter such as playing a reorder tone when the phone is off-hooked, or changing the line polarity.
	These behaviors are done upon one of the following scenarios:
	▪ The device is physically disconnected from the network (i.e., Ethernet cable is disconnected).
	▪ The Ethernet cable is connected, but the device is unable to communicate with any host. For this scenario, the LAN Watch-Dog must be activated (i.e., set the EnableLANWatchDog parameter to 1).
	▪ The device can't communicate with the proxy (according to the Proxy Keep-Alive mechanism) and no other alternative route exists to send the call.
	▪ The IP Connectivity mechanism is enabled (using the AltRoutingTel2IPEnable parameter) and there is no connectivity to any destination IP address.
	Notes:
	▪ The FXSOOSBehavior parameter determines the behavior of the FXS endpoints when a Busy Out or Graceful Lock occurs.
	▪ FXO endpoints during Busy Out and Lock are inactive.
	▪ See the LifeLineType parameter for complementary optional behavior.
Web: Out-Of-Service Behavior OVOC:FXS OOS Behavior **[FXSOOSBehavior]**	Determines the behavior of FXS endpoints when a Busy Out condition exists.
	▪ **[0]** None = Silence is heard when the FXS endpoint goes off-hook.
	▪ **[1]** Reorder Tone = (Default) The device plays a reorder tone to the connected phone / PBX.
	▪ **[2]** Polarity Reversal = The device reverses the polarity of the endpoint making it unusable (relevant, for example, for PBX DID lines).
	▪ **[3]** Reorder Tone + Polarity Reversal = Same as options [1] and [2].
	▪ **[4]** Current Disconnect = The device disconnects the current to the FXS endpoint.
	Notes:
	▪ A device reset is required for this parameter to take effect when it is set to [2], [3], or [4].
	▪ This parameter is applicable only to FXS interfaces.
Retransmission Parameters	
Web: SIP T1 Retransmission Timer **[msec]** OVOC: T1 RTX **[SipT1Rtx]**	Defines the time interval (in msec) between the first transmission of a SIP message and the first retransmission of the same message.
	The default is 500.
	Note: The time interval between subsequent retransmissions of the same SIP message starts with SipT1Rtx. For INVITE requests, it is multiplied by two for each new retransmitted message. For all other SIP messages, it is multiplied by two until SipT2Rtx. For example, assuming SipT1Rtx = 500 and SipT2Rtx = 4000:
	▪ The first retransmission is sent after 500 msec.
	▪ The second retransmission is sent after 1000 (2*500) msec.

Parameter	Description
	▪ The third retransmission is sent after 2000 (2*1000) msec. ▪ The fourth retransmission and subsequent retransmissions until SIPMaxRtx are sent after 4000 (2*2000) msec.
Web: SIP T2 Retransmission Timer **[msec]** OVOC: T2 RTX **[SipT2Rtx]**	Defines the maximum interval (in msec) between retransmissions of SIP messages (except for INVITE requests). The default is 4000. **Note:** The time interval between subsequent retransmissions of the same SIP message starts with SipT1Rtx and is multiplied by two until SipT2Rtx.
Web: SIP Maximum RTX OVOC: Max RTX **[SIPMaxRtx]**	Defines the maximum number of UDP transmissions of SIP messages (first transmission plus retransmissions). The range is 1 to 30. The default is 7.
Web: Number of RTX Before Hot-Swap OVOC: Proxy Hot Swap Rtx **[HotSwapRtx]**	Defines the number of retransmitted INVITE/REGISTER messages before the call is routed (hot swap) to another Proxy/Registrar. The valid range is 1 to 30. The default is 3. **Note:** This parameter is also used for alternative routing. If a domain name in the Tel to IP Routing is resolved into two IP addresses, and if there is no response for HotSwapRtx retransmissions to the INVITE message that is sent to the first IP address, the device immediately initiates a call to the second IP address.

43.9   Coders and Profile Parameters

The profile parameters are described in the table below.

Table 43-32: Profile Parameters

Parameter	Description
Coders Table / Coder Groups Table	

Web: Coders Table/Coder Group Settings OVOC: Coders Group **[CodersGroup0]** **[CodersGroup1]** **[CodersGroup2]** **[CodersGroup3]** **[CodersGroup4]** **[CodersGroup5]** **[CodersGroup6]** **[CodersGroup7]** **[CodersGroup8]** **[CodersGroup9]**	This table parameter defines the device's coders. Each group can consist of up to 10 coders. The first Coder Group is the default coder list and the default Coder Group.
	The format of this parameter is as follows:
	[CodersGroup<0-9>]
	FORMAT CodersGroup0_Index = CodersGroup0_Name, CodersGroup0_pTime, CodersGroup0_rate, CodersGroup0_PayloadType, CodersGroup0_Sce;
	[\CodersGroup<0-9]
	For example, below are defined two Coder Groups (0 and 1):
	
	`[ CodersGroup0 ]`
	`FORMAT CodersGroup0_Index = CodersGroup0_Name,`
	`CodersGroup0_pTime, CodersGroup0_rate,`
	`CodersGroup0_PayloadType, CodersGroup0_Sce;`
	`CodersGroup0 0 = g711Alaw64k, 20, 0, 255, 0;`
	`CodersGroup0 1 = eg711Ulaw, 10, 0, 71, 0;`
	`CodersGroup0 2 = eg711Ulaw, 10, 0, 71, 0;`
	`[ \CodersGroup0 ]`

Parameter	Description
| | ```
[ CodersGroup1 ]
FORMAT CodersGroup1_Index = CodersGroup1_Name,
CodersGroup1_pTime, CodersGroup1_rate,
CodersGroup1_PayloadType, CodersGroup1_Sce;
CodersGroup1 0 = Transparent, 20, 0, 56, 0;
CodersGroup1 1 = g726, 20, 0, 23, 0;
[ \CodersGroup1 ]
```<br>**Notes:**<br>▪ For a list of supported coders and a detailed description of this table, see Configuring Coders on page 211.<br>▪ The coder name is case-sensitive. |
| **IP Profile Table** | |
| Web: IP Profile Settings
OVOC: Protocol Definition > IP Profile
[IPProfile] | This table parameter configures the IP Profile table. Each IP Profile ID includes a set of parameters (which are typically configured separately using their individual "global" parameters). You can later assign these IP Profiles to Tel-to-IP routing rules (Prefix parameter), IP-to-Tel routing rules and IP Groups.
The format of this parameter is as follows:
[IPProfile]
FORMAT IpProfile_Index = IpProfile_ProfileName, IpProfile_IpPreference, IpProfile_CodersGroupID, IpProfile_IsFaxUsed, IpProfile_JitterBufMinDelay, IpProfile_JitterBufOptFactor, IpProfile_IPDiffServ, IpProfile_SigIPDiffServ, IpProfile_SCE, IpProfile_RTPRedundancyDepth, IpProfile_RemoteBaseUDPPort, IpProfile_CNGmode, IpProfile_VxxTransportType, IpProfile_NSEMode, IpProfile_IsDTMFUsed, IpProfile_PlayRBTone2IP, IpProfile_EnableEarlyMedia, IpProfile_ProgressIndicator2IP, IpProfile_EnableEchoCanceller, IpProfile_CopyDest2RedirectNumber, IpProfile_MediaSecurityBehaviour, IpProfile_CallLimit, IpProfile_DisconnectOnBrokenConnection, IpProfile_FirstTxDtmfOption, IpProfile_SecondTxDtmfOption, IpProfile_RxDTMFOption, IpProfile_EnableHold, IpProfile_InputGain, IpProfile_VoiceVolume, IpProfile_AddIEInSetup, IpProfile_SBCExtensionCodersGroupID, IpProfile_MediaIPVersionPreference, IpProfile_TranscodingMode, IpProfile_SBCAllowedCodersGroupID, IpProfile_SBCAllowedCodersMode, IpProfile_SBCMediaSecurityBehaviour, IpProfile_SBCRFC2833Behavior, IpProfile_SBCAlternativeDTMFMethod, IpProfile_SBCAssertIdentity, IpProfile_AMDSensitivityParameterSuit, IpProfile_AMDSensitivityLevel, IpProfile_AMDMaxGreetingTime, IpProfile_AMDMaxPostSilenceGreetingTime, IpProfile_SBCDiversionMode, IpProfile_SBCHistoryInfoMode, IpProfile_EnableQSIGTunneling, IpProfile_SBCFaxCodersGroupID, IpProfile_SBCFaxBehavior, IpProfile_SBCFaxOfferMode, IpProfile_SBCFaxAnswerMode, IpProfile_SbcPrackMode, IpProfile_SBCSessionExpiresMode, IpProfile_SBCRemoteUpdateSupport, IpProfile_SBCRemoteReinviteSupport, IpProfile_SBCRemoteDelayedOfferSupport, IpProfile_SBCRemoteReferBehavior, IpProfile_SBCRemote3xxBehavior, IpProfile_SBCRemoteMultiple18xSupport, IpProfile_SBCRemoteEarlyMediaResponseType, IpProfile_SBCRemoteEarlyMediaSupport, IpProfile_EnableSymmetricMKI, IpProfile_MKISize, IpProfile_SBCEnforceMKISize, IpProfile_SBCRemoteEarlyMediaRTP, IpProfile_SBCRemoteSupportsRFC3960, IpProfile_SBCRemoteCanPlayRingback, IpProfile_EnableEarly183, IpProfile_EarlyAnswerTimeout, IpProfile_SBC2833DTMFPayloadType, |

Parameter	Description
	IpProfile_SBCUserRegistrationTime, IpProfile_ResetSRTPStateUponRekey, IpProfile_AmdMode, IpProfile_SBCReliableHeldToneSource, IpProfile_SBCPlayHeldTone, IpProfile_SBCRemoteHoldFormat; [\IPProfile] **Note:** For a description of this table, see 'Configuring IP Profiles' on page 217.
Tel Profile Table	
Web: Tel Profile Settings OVOC: Protocol Definition > Telephony Profile **[TelProfile]**	This table parameter configures the Tel Profile table. Each Tel Profile ID includes a set of parameters (which are typically configured separately using their individual, "global" parameters). You can later assign these Tel Profile IDs to other elements such as in the <trunkgrouptableMP>Endpoint Phone Number table (TrunkGroup parameter). Therefore, Tel Profiles allow you to apply the same settings of a group of parameters to multiple channels, or apply specific settings to different channels. The format of this parameter is as follows: [TelProfile] FORMAT TelProfile_Index = TelProfile_ProfileName, TelProfile_TelPreference, TelProfile_CodersGroupID, TelProfile_IsFaxUsed, TelProfile_JitterBufMinDelay, TelProfile_JitterBufOptFactor, TelProfile_IPDiffServ, TelProfile_SigIPDiffServ, TelProfile_DtmfVolume, TelProfile_InputGain, TelProfile_VoiceVolume, TelProfile_EnableReversePolarity, TelProfile_EnableCurrentDisconnect, TelProfile_EnableDigitDelivery, TelProfile_EnableEC, TelProfile_MWIAnalog, TelProfile_MWIDisplay, TelProfile_FlashHookPeriod, TelProfile_EnableEarlyMedia, TelProfile_ProgressIndicator2IP, TelProfile_TimeForReorderTone, TelProfile_EnableDIDWink, TelProfile_IsTwoStageDial, TelProfile_DisconnectOnBusyTone, TelProfile_EnableVoiceMailDelay, TelProfile_DialPlanIndex, TelProfile_Enable911PSAP, TelProfile_SwapTelToIpPhoneNumbers, TelProfile_EnableAGC, TelProfile_ECNlpMode, TelProfile_DigitalCutThrough, TelProfile_EnableFXODoubleAnswer, TelProfile_CallPriorityMode; [\TelProfile] **Notes:** ▪ For a description of this parameter, see Configuring Tel Profiles on page 215. ▪ For a detailed description of each parameter, see its corresponding "global" parameter. <table><tr><td>TelProfile Field</td><td>Web Name</td><td>Global Parameter</td></tr><tr><td>TelProfile_ProfileName</td><td>Profile Name</td><td>-</td></tr><tr><td>TelProfile_TelPreference</td><td>Profile Preference</td><td>-</td></tr><tr><td>TelProfile_CodersGroupID</td><td>Coder Group</td><td>CodersGroup0</td></tr><tr><td>TelProfile_IsFaxUsed</td><td>Fax Signaling Method</td><td>IsFaxUsed</td></tr><tr><td>TelProfile_JitterBufMinDelay</td><td>Dynamic Jitter Buffer Minimum Delay</td><td>DJBufMinDelay</td></tr></table>

Parameter	Description		
	TelProfile_JitterBufOptFactor	Dynamic Jitter Buffer Optimization Factor	DJBufOptFactor
	TelProfile_IPDiffServ	RTP IP DiffServ	PremiumServiceClassMediaDiffServ
	TelProfile_SigIPDiffServ	Signaling DiffServ	PremiumServiceClassControlDiffServ
	TelProfile_DtmfVolume	DTMF Volume	DTMFVolume
	TelProfile_InputGain	Input Gain	InputGain
	TelProfile_VoiceVolume	Voice Volume	VoiceVolume
	TelProfile_EnableReversePolarity	Enable Polarity Reversal	EnableReversalPolarity
	TelProfile_EnableCurrentDisconnect	Enable Current Disconnect	EnableCurrentDisconnect
	TelProfile_EnableDigitDelivery	Enable Digit Delivery	EnableDigitDelivery
	TelProfile_EnableEC	Echo Canceler	EnableEchoCanceller
	TelProfile_MWIAnalog	MWI Analog Lamp	MWIAnalogLamp
	TelProfile_MWIDisplay	MWI Display	MWIDisplay
	TelProfile_FlashHookPeriod	Flash Hook Period	FlashHookPeriod
	TelProfile_EnableEarlyMedia	Enable Early Media	EnableEarlyMedia
	TelProfile_ProgressIndicator2IP	Progress Indicator to IP	ProgressIndicator2IP
	TelProfile_TimeForReorderTone	Time For Reorder Tone	TimeForReorderTone
	TelProfile_EnableDIDWink	Enable DID Wink	EnableDIDWink
	TelProfile_IsTwoStageDial	Dialing Mode	IsTwoStageDial
	TelProfile_DisconnectOnBusyTone	Disconnect Call on Detection of Busy Tone	DisconnectOnBusyTone
	TelProfile_EnableVoiceMailDelay	Enable Voice Mail Delay	-
	TelProfile_DialPlanIndex	Dial Plan Index	DialPlanIndex
	TelProfile_Enable911PSAP	Enable 911 PSAP	Enable911PSAP
	TelProfile_SwapTelToIpPhoneNumbers	Swap Tel To IP Phone Numbers	SwapTEl2IPCalled&CallingNumbers

Parameter	Description		
	TelProfile_EnableAGC	Enable AGC	EnableAGC
	TelProfile_ECNlpMode	EC NLP Mode	ECNLPMode
	TelProfile_DigitalCutThrough	-	DigitalCutThrough
	TelProfile_EnableFXODoubleAnswer	-	EnableFXODoubleAnswer
	TelProfile_CallPriorityMode	-	CallPriorityMode
The parameter IpPreference parameter EnableVo.			

43.10 Channel Parameters

This subsection describes the device's channel parameters.

43.10.1 Voice Parameters

The voice parameters are described in the table below.

Table 43-33: Voice Parameters

Parameter	Description
Web/OVOC: Input Gain **[InputGain]**	Defines the pulse-code modulation (PCM) input gain control (in decibels). This parameter sets the level for the received (Tel-to-IP) signal. The valid range is -32 to 31 dB. The default is 0 dB. **Note:** This parameter can also be configured in an IP Profile and/or a Tel Profile.
Web: Voice Volume OVOC: Volume (dB) **[VoiceVolume]**	Defines the voice gain control (in decibels). This parameter sets the level for the transmitted (IP-to-Tel) signal. The valid range is -32 to 31 dB. The default is 0 dB. **Note:** This parameter can also be configured in an IP Profile and/or a Tel Profile.
OVOC: Payload Format **[VoicePayloadFormat]**	Determines the bit ordering of the G.726/G.727 voice payload format. ▪ **[0]** = (Default) Little Endian ▪ **[1]** = Big Endian **Notes:** ▪ To ensure high voice quality when using G.726/G.727, both communicating ends should use the same endianness format. Therefore, when the device communicates with a third-party entity that uses the G.726/G.727 voice coder and voice quality is poor, change the settings of this parameter (between Big Endian and Little Endian). ▪ The G.727 coder is not supported by MP-124 Rev. E.
Web: MF Transport Type **[MFTransportType]**	Currently, not supported.
Web: Enable Answer Detector **[EnableAnswerDetector]**	Currently, not supported.

Parameter	Description
Web: 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. **Note:** AD is not supported by MP-124 Rev. E.
Web: Answer Detector Silence Time **[AnswerDetectorSilenceTime]**	Currently, not supported.
Web: Answer Detector Redirection **[AnswerDetectorRedirection]**	Currently, not supported.
Web: Answer Detector Sensitivity OVOC: Sensitivity **[AnswerDetectorSensitivity]**	Defines the Answer Detector sensitivity. The range is 0 (most sensitive) to 2 (least sensitive). The default is 0. **Note:** AD is not supported by MP-124 Rev. E.
Web: Echo Canceler OVOC: Echo Canceller Enable **[EnableEchoCanceller]**	Enables echo cancellation (i.e., echo from voice calls is removed). ▪ **[0]** Disable ▪ **[1]** Enable (default) **Note:** This parameter can also be configured in an IP Profile and/or a Tel Profile.
OVOC: Echo Canceller Hybrid Loss **[ECHybridLoss]**	Defines the four-wire to two-wire worst-case Hybrid loss, the ratio between the signal level sent to the hybrid and the echo level returning from the hybrid. ▪ **[0]** = (Default) 6 dB ▪ **[1]** = N/A ▪ **[2]** = 0 dB ▪ **[3]** = 3 dB
OVOC: ECN lp Mode **[ECNLPMode]**	Defines the echo cancellation Non-Linear Processing (NLP) mode. ▪ **[0]** = (Default) NLP adapts according to echo changes ▪ **[1]** = Disables NLP ▪ **[2]** = Silence output NLP **Note:** This parameter can also be configured in a Tel Profile.
[EchoCancellerAggressiveNLP]	Enables the Aggressive NLP at the first 0.5 second of the call. ▪ **[0]** = Disable ▪ **[1]** = (Default) Enable. The echo is removed only in the first half of a second of the incoming IP signal. **Note:** For this parameter to take effect, a device reset is required.
[RTPSIDCoeffNum]	Defines the number of spectral coefficients added to an SID packet being sent according to RFC 3389. The valid values are **[0]** (default), **[4]**, **[6]**, **[8]** and **[10]**.

43.10.2 Coder Parameters

The coder parameters are described in the table below.

Table 43-34: Coder Parameters

Parameter	Description
OVOC: AMR Coder Header Format **[AMRCoderHeaderFormat]**	Determines the payload format of the AMR header. ▪ **[0]** = Non-standard multiple frames packing in a single RTP frame. Each frame has a CMR and TOC header. ▪ **[1]** = AMR frame according to RFC 3267 bundling. ▪ **[2]** = AMR frame according to RFC 3267 interleaving. ▪ **[3]** = AMR is passed using the AMR IF2 format. **Note:** Bandwidth Efficient mode is not supported; the mode is always Octet-aligned.
Web: DSP Version Template Number OVOC: Version Template Number **[DSPVersionTemplateNumber]**	Determines the DSP template used by the device. Each DSP template supports specific coders, channel capacity, and features. The default is DSP Template 0. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For a list of supported DSP templates, see DSP Templates on page 645.

43.10.3 DTMF Parameters

The dual-tone multi-frequency (DTMF) parameters are described in the table below.

Table 43-35: DTMF Parameters

Parameter	Description
Web/OVOC: DTMF Transport Type **[DTMFTransportType]**	Determines the DTMF transport type. ▪ **[0]** Mute DTMF = DTMF digits are removed from the voice stream and are not relayed to remote side. ▪ **[2]** Transparent DTMF = DTMF digits remain in the voice stream. ▪ **[3]** RFC 2833 Relay DTMF = (Default) DTMF digits are removed from the voice stream and are relayed to remote side according to RFC 2833. ▪ **[7]** RFC 2833 Relay Decoder Mute = DTMF digits are sent according to RFC 2833 and muted when received. **Note:** This parameter is automatically updated if the parameters TxDTMFOption or RxDTMFOption are configured.
Web: DTMF Volume (-31 to 0 dB) OVOC: DTMF Volume (dBm) **[DTMFVolume]**	Defines the DTMF gain control value (in decibels) to the or analog side. The valid range is -31 to 0 dB. The default is -11 dB. **Note:** This parameter can also be configured in a Tel Profile.

Parameter	Description
Web: DTMF Generation Twist OVOC: DTMF Twist Control **[DTMFGenerationTwist]**	Defines the range (in decibels) between the high and low frequency components in the DTMF signal. Positive decibel values cause the higher frequency component to be stronger than the lower one. Negative values cause the opposite effect. For any parameter value, both components change so that their average is constant. The valid range is -10 to 10 dB. The default is 0 dB. **Note:** For this parameter to take effect, a device reset is required.
OVOC: DTMF Inter Interval (msec) **[DTMFInterDigitInterval]**	Defines the time (in msec) between generated DTMF digits to PSTN side (if TxDTMFOption = 1, 2 or 3). The default is 100 msec. The valid range is 0 to 32767.
OVOC: DTMF Length (msec) **[DTMFDigitLength]**	Defines the time (in msec) for generating DTMF tones to the PSTN side (if TxDTMFOption = 1, 2 or 3). It also configures the duration that is sent in INFO (Cisco) messages. The valid range is 0 to 32767. The default is 100.
OVOC: Rx DTMF Relay Hang Over Time (msec) **[RxDTMFHangOverTime]**	Defines the Voice Silence time (in msec) after playing DTMF or MF digits to the Tel/PSTN side that arrive as Relay from the IP side. Valid range is 0 to 2,000 msec. The default is 1,000 msec.
OVOC: Tx DTMF Relay Hang Over Time (msec) **[TxDTMFHangOverTime]**	Defines the Voice Silence time (in msec) after detecting the end of DTMF or MF digits at the Tel/PSTN side when the DTMF Transport Type is either Relay or Mute. Valid range is 0 to 2,000 msec. The default is 1,000 msec.
Web/OVOC: NTE Max Duration **[NTEMaxDuration]**	Defines the maximum time for sending Named Telephony Events / NTEs (RFC 4733/2833 DTMF relay) to the IP side, regardless of the DTMF signal duration on the TDM side. The range is -1 to 200,000,000 msec. The default is -1 (i.e., NTE stops only upon detection of an End event).

43.10.4 RTP, RTCP and T.38 Parameters

The RTP, RTCP and T.38 parameters are described in the table below.

Table 43-36: RTP/RTCP and T.38 Parameters

Parameter	Description
Web: Dynamic Jitter Buffer Minimum Delay OVOC: Minimal Delay (dB) **[DJBufMinDelay]**	Defines the minimum delay (in msec) for the Dynamic Jitter Buffer. The valid range is 0 to 150. The default delay is 10. **Notes:** ▪ This parameter can also be configured in an IP Profile and/or a Tel Profile. ▪ For more information on Jitter Buffer, see Dynamic Jitter Buffer Operation on page 171.
Web: Dynamic Jitter Buffer Optimization Factor OVOC: Opt Factor **[DJBufOptFactor]**	Defines the Dynamic Jitter Buffer frame error/delay optimization factor. The valid range is 0 to 12. The default factor is 10.

Parameter	Description
	Notes:
	▪ For data (fax and modem) calls, set this parameter to 12.
	▪ This parameter can also be configured in an IP Profile and/or a Tel Profile.
	▪ For more information on Jitter Buffer, see Dynamic Jitter Buffer Operation on page 171.
Web/OVOC: Analog Signal Transport Type **[AnalogSignalTransportType]**	Determines the analog signal transport type.
	▪ **[0]** Ignore Analog Signals = (Default) Ignore.
	▪ **[1]** RFC 2833 Analog Signal Relay = Transfer hookflash using RFC 2833.
Web: RTP Redundancy Depth OVOC: Redundancy Depth **[RTPRedundancyDepth]**	Enables the device to generate RFC 2198 redundant packets. This can be used for packet loss where the missing information (audio) can be reconstructed at the receiver's end from the redundant data that arrives in subsequent packets. This is required, for example, in wireless networks where a high percentage (up to 50%) of packet loss can be experienced.
	▪ **[0]** 0 = (Default) Disable.
	▪ **[1]** 1 = Enable - previous voice payload packet is added to current packet.
	Notes:
	▪ When enabled, you can configure the payload type, using the RFC2198PayloadType parameter.
	▪ The RTP redundancy dynamic payload type can be included in the SDP, by using the EnableRTPRedundancyNegotiation parameter.
	▪ This parameter can also be configured in an IP Profile.
Web: Enable RTP Redundancy Negotiation **[EnableRTPRedundancyNegotiation]**	Enables the device to include the RTP redundancy dynamic payload type in the SDP, according to RFC 2198.
	▪ **[0]** Disable (default)
	▪ **[1]** Enable
	When enabled, the device includes in the SDP message the RTP payload type "RED" and the payload type configured by the parameter RFC2198PayloadType.
	`a=rtpmap:<PT> RED/8000`
	Where <PT> is the payload type as defined by RFC2198PayloadType. The device sends the INVITE message with "a=rtpmap:<PT> RED/8000" and responds with a 18x/200 OK and "a=rtpmap:<PT> RED/8000" in the SDP.
	Notes:
	▪ For this feature to be functional, you must also set the parameter RTPRedundancyDepth to 1 (i.e., enabled).
	▪ Currently, the negotiation of "RED" payload type is not supported and therefore, it should be configured to the same PT value for both parties.
Web: RFC 2198 Payload Type OVOC: Redundancy Payload Type **[RFC2198PayloadType]**	Defines the RTP redundancy packet payload type according to RFC 2198.
	The range is 96 to 127. The default is 104.

Parameter	Description
	Note: This parameter is applicable only if the parameter RTPRedundancyDepth is set to 1.
Web: Packing Factor OVOC: Packetization Factor **[RTPPackingFactor]**	N/A. Controlled internally by the device according to the selected coder.
Web/OVOC: Basic RTP Packet Interval **[BasicRTPPacketInterval]**	N/A. Controlled internally by the device according to the selected coder.
Web: RTP Directional Control **[RTPDirectionControl]**	N/A. Controlled internally by the device according to the selected coder.
Web/OVOC: RFC 2833 TX Payload Type **[RFC2833TxPayloadType]**	Defines the Tx RFC 2833 DTMF relay dynamic payload type. The valid range is 96 to 99, and 106 to 127. The default is 96. The 100, 102 to 105 range is allocated for proprietary usage. **Notes:** ▪ Certain vendors (e.g., Cisco) use payload type 101 for RFC 2833. ▪ When RFC 2833 payload type negotiation is used (i.e., the parameter TxDTMFOption is set to 4), this payload type is used for the received DTMF packets. If negotiation isn't used, this payload type is used for receive and for transmit.
Web/OVOC: RFC 2833 RX Payload Type **[RFC2833RxPayloadType]**	Defines the Rx RFC 2833 DTMF relay dynamic payload type. The valid range is 96 to 99, and 106 to 127. The default is 96. The 100, 102 to 105 range is allocated for proprietary usage. **Notes:** ▪ Certain vendors (e.g., Cisco) use payload type 101 for RFC 2833. ▪ When RFC 2833 payload type negotiation is used (i.e., the parameter TxDTMFOption is set to 4), this payload type is used for the received DTMF packets. If negotiation isn't used, this payload type is used for receive and for transmit.
[EnableDetectRemoteMACChange]	Determines whether the device changes the RTP packets according to the MAC address of received RTP packets and according to Gratuitous Address Resolution Protocol (GARP) messages. ▪ **[0]** = Nothing is changed. ▪ **[1]** = If the device receives RTP packets with a different source MAC address (than the MAC address of the transmitted RTP packets), then it sends RTP packets to this MAC address and removes this IP entry from the device's ARP cache table. ▪ **[2]** = (Default) The device uses the received GARP packets to change the MAC address of the transmitted RTP packets. ▪ **[3]** = Options 1 and 2 are used. **Notes:**

Parameter	Description
	▪ For this parameter to take effect, a device reset is required. ▪ If the device is located in a network subnet which is connected to other gateways using a router that uses Virtual Router Redundancy Protocol (VRRP) for redundancy, then set this parameter to 0 or 2.
Web: RTP Base UDP Port OVOC: Base UDP Port **[BaseUDPport]**	Defines the lower boundary of the UDP port used for RTP, RTCP (RTP port + 1) and T.38 (RTP port + 2). For example, if the Base UDP Port is set to 6000, then one channel may use the ports RTP 6000, RTCP 6001, and T.38 6002, while another channel may use RTP 6010, RTCP 6011, and T.38 6012, and so on. The range of possible UDP ports is 6,000 to 64,000. The default base UDP port is 6000. Once this parameter is configured, the UDP port range (lower to upper boundary) is calculated as follows: ▪ MP-112/MP-114: BaseUDPport to (BaseUDPport + 3*10) ▪ MP-118: BaseUDPport to (BaseUDPport + 7*10) ▪ MP-124: BaseUDPport to (BaseUDPport + 23*10) **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ You can define a UDP port range per Media Realm (see Configuring Media Realms on page 180). ▪ The UDP ports are allocated randomly to channels. ▪ If RTP Base UDP Port is not a factor of 10, the following message is generated: 'invalid local RTP port'.
OVOC: No Op Enable **[NoOpEnable]**	Enables the transmission of RTP or T.38 No-Op packets. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable This mechanism ensures that the NAT binding remains open during RTP or T.38 silence periods.
OVOC: No Op Interval **[NoOpInterval]**	Defines the time interval in which RTP or T.38 No-Op packets are sent in the case of silence (no RTP/T.38 traffic) when No-Op packet transmission is enabled. The valid range is 20 to 65,000 msec. The default is 10,000. **Note:** To enable No-Op packet transmission, use the NoOpEnable parameter.

Parameter	Description
OVOC: No Op Payload Type **[RTPNoOpPayloadType]**	Defines the payload type of No-Op packets. The valid range is 96 to 127 (for the range of Dynamic RTP Payload Type for all types of non hard-coded RTP Payload types, refer to RFC 3551). The default is 120. **Note:** When defining this parameter, ensure that it doesn't cause collision with other payload types.
[RTCPActivationMode]	Disables RTCP traffic when there is no RTP traffic. This feature is useful, for example, to stop RTCP traffic that is typically sent when calls are put on hold (by an INVITE with 'a=inactive' in the SDP). ▪ **[0]** Active Always = (Default) RTCP is active even during inactive RTP periods, i.e., when the media is in 'recvonly' or 'inactive' mode. ▪ **[1]** Inactive Only If RTP Inactive = No RTCP is sent when RTP is inactive.
RTP Control Protocol Extended Reports (RTCP XR) Parameters	
Web: Enable RTCP XR OVOC: RTCP XR Enable **[VQMonEnable]**	Enables voice quality monitoring and RTCP XR, according to Internet-Draft draft-ietf-sipping-rtcp-summary-13. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Minimum Gap Size OVOC: GMin **[VQMonGMin]**	Defines the voice quality monitoring - minimum gap size (number of frames). The default is 16.
Web/OVOC: Burst Threshold **[VQMonBurstHR]**	Defines the voice quality monitoring - excessive burst alert threshold. The default is -1 (i.e., no alerts are issued).
Web/OVOC: Delay Threshold **[VQMonDelayTHR]**	Defines the voice quality monitoring - excessive delay alert threshold. The default is -1 (i.e., no alerts are issued).
Web: R-Value Delay Threshold OVOC: End of Call Rval Delay Threshold **[VQMonEOCRValTHR]**	Defines the voice quality monitoring - end of call low quality alert threshold. The default is -1 (i.e., no alerts are issued).
Web: RTCP XR Packet Interval OVOC: Packet Interval **[RTCPInterval]**	Defines the time interval (in msec) between adjacent RTCP reports. The valid value range is 0 to 65,535. The default is 5,000.
Web: Disable RTCP XR Interval Randomization OVOC: Disable 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
OVOC: RTCP XR Collection Server Transport Type **[RTCPXRESCTransportType]**	Defines the transport layer used for outgoing SIP dialogs initiated by the device to the RTCP XR Collection Server. ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP

Parameter	Description
	▪ **[1]** TCP ▪ **[2]** TLS **Note:** When set to **[-1**], the value of the SIPTransportType parameter is used.
Web: RTCP XR Collection Server OVOC: Esc IP **[RTCPXREscIP]**	Defines the IP address of the Event State Compositor (ESC). The device sends RTCP XR reports to this server, using SIP PUBLISH messages. The address can be configured as a numerical IP address or as a domain name. **Note:** Instead of sending RTCP XR to an ESC server, you can send RTCP XR to an IP Group (see the PublicationIPGroupID parameter).
Web: RTCP XR Report Mode OVOC: Report Mode **[RTCPXRReportMode]**	Enables the device to send RTCP XR in SIP PUBLISH messages to the Event State Compositor (ESC) server and defines the interval at which they are sent. ▪ [0] Disable = (Default) RTCP XR is not sent. ▪ [1] End Call = RTCP XR is sent at the end of the call. ▪ [2] End Call & Periodic = RTCP XR is sent at the end of the call and periodically according to the RTCPInterval parameter. ▪ [3] End Call & End Segment = RTCP XR is sent at the end of the call and at the end of each media segment of the call. A media segment is a change in media, for example, when the coder is changed or when the caller toggles between two called parties (using call hold/retrieve). The RTCP XR sent at the end of a media segment contains information only of that segment. If the segment does not contain RTP/RTCP content, the RTCP XR is not sent. For call hold, the device sends an RTCP XR each time the call is placed on hold and each time it is retrieved. In addition, the Start timestamp in the RTCP XR indicates the start of the media segment; the End timestamp indicates the time of the last sent periodic RTCP XR (typically, up to 5 seconds before reported segment ends).
publication-ip-group-id [PublicationIPGroupID]	Defines the IP Group to where the RTCP XR is sent. If the value is -1 (default) or 0, the RTCP XR is sent to the ESC server, as configured by the RTCPXREscIP parameter. **Note:** The parameter is applicable only to the Gateway application.

43.11 Gateway and IP-to-IP Parameters

43.11.1 Fax and Modem Parameters

The fax and modem parameters are described in the table below.

Table 43-37: Fax and Modem Parameters

Parameter	Description
Web: Fax Transport Mode OVOC: Transport Mode **[FaxTransportMode]**	Determines the fax transport mode used by the device. ▪ **[0]** Disable = transparent mode ▪ **[1]** T.38 Relay (default) ▪ **[2]** Bypass ▪ **[3]** Events Only **Note:** This parameter is overridden by the parameter IsFaxUsed. If the parameter IsFaxUsed is set to 1 (T.38 Relay) or 3 (Fax Fallback), then FaxTransportMode is always set to 1 (T.38 relay).
Web: V.21 Modem Transport Type OVOC: V21 Transport **[V21ModemTransportType]**	Determines the V.21 modem transport type. ▪ **[0]** Disable = (Default) Disable (Transparent) ▪ **[2]** Enable Bypass ▪ **[3]** Events Only = Transparent with Events **Note:** This parameter can also be configured in an IP Profile.
Web: V.22 Modem Transport Type OVOC: V22 Transport **[V22ModemTransportType]**	Determines the V.22 modem transport type. ▪ **[0]** Disable = Disable (Transparent) ▪ **[2]** Enable Bypass (default) ▪ **[3]** Events Only = Transparent with Events **Note:** This parameter can also be configured in an IP Profile.
Web: V.23 Modem Transport Type OVOC: V23 Transport **[V23ModemTransportType]**	Determines the V.23 modem transport type. ▪ **[0]** Disable = Disable (Transparent) ▪ **[2]** Enable Bypass (default) ▪ **[3]** Events Only = Transparent with Events **Note:** This parameter can also be configured in an IP Profile.
Web: V.32 Modem Transport Type OVOC: V32 Transport **[V32ModemTransportType]**	Determines the V.32 modem transport type. ▪ **[0]** Disable = Disable (Transparent) ▪ **[2]** Enable Bypass (default) ▪ **[3]** Events Only = Transparent with Events **Notes:** ▪ This parameter applies only to V.32 and V.32bis modems. ▪ This parameter can also be configured in an IP Profile.
Web: V.34 Modem Transport Type OVOC: V34 Transport **[V34ModemTransportType]**	Determines the V.90/V.34 modem transport type. ▪ **[0]** Disable = Disable (Transparent) ▪ **[2]** Enable Bypass (default) ▪ **[3]** Events Only = Transparent with Events **Note:** This parameter can also be configured in an IP Profile.
OVOC: Bell Transport Type **[BellModemTransportType]**	Determines the Bell modem transport method. ▪ **[0]** = Transparent (default) ▪ **[2]** = Bypass

Parameter	Description
	▪ **[3]** = Transparent with events
Web/OVOC: Fax CNG Mode **[FaxCNGMode]**	Determines the device's handling of fax relay upon detection of a fax CNG tone from originating faxes. ▪ **[0]** Doesn't send T.38 Re-INVITE = (Default) SIP re-INVITE is not sent. ▪ **[1]** Sends on CNG tone = Sends a SIP re-INVITE with T.38 parameters in SDP to the terminating fax upon detection of a fax CNG tone, if the CNGDetectorMode parameter is set to 1. **Notes:** ▪ This feature is applicable only if the IsFaxUsed parameter is set to **[1]** or **[3].** ▪ The device also sends T.38 re-INVITE if the CNGDetectorMode parameter is set to **[2],** regardless of the FaxCNGMode parameter settings.
Web/OVOC: CNG Detector Mode **[CNGDetectorMode]**	Determines whether the device detects the fax calling tone (CNG). ▪ **[0]** Disable = (Default) The originating device doesn't detect CNG; the CNG signal passes transparently to the remote side. ▪ **[1]** Relay = CNG is detected on the originating side. CNG packets are sent to the remote side according to T.38 (if IsFaxUsed = 1) and the fax session is started. A SIP Re-INVITE message isn't sent and the fax session starts by the terminating device. This option is useful, for example, 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 (i.e., originating device). To also send a Re-INVITE message upon detection of a fax CNG tone in this mode, set the parameter FaxCNGMode to 1. ▪ **[2]** Events Only = CNG is detected on the originating side and a fax session is started by the originating side using the Re-INVITE message. Usually, T.38 fax session starts when the 'preamble' signal is detected by the answering side. Some SIP devices don't support the detection of this fax signal on the answering side and thus, in these cases it is possible to configure the device to start the T.38 fax session when the CNG tone is detected by the originating side. However, this mode is not recommended. **Note:** This parameter can also be configured in an IP Profile.
Web: Fax Relay Enhanced Redundancy Depth OVOC: Enhanced Relay Redundancy Depth **[FaxRelayEnhancedRedundancyDepth]**	Defines the number of times that control packets are retransmitted when using the T.38 standard. The valid range is 0 to 4. The default is 2.

Parameter	Description
Web: Fax Relay Redundancy Depth OVOC: Relay Redundancy Depth **[FaxRelayRedundancyDepth]**	Defines the number of times that each fax relay payload is retransmitted to the network. • **[0]** = (Default) No redundancy • **[1]** = One packet redundancy • **[2]** = Two packet redundancy **Note:** This parameter is applicable only to non-V.21 packets.
Web: Fax Relay Max Rate (bps) OVOC: Relay Max Rate **[FaxRelayMaxRate]**	Defines the maximum rate (in bps) at which fax relay messages are transmitted (outgoing calls). • **[0]** 2400 = 2.4 kbps • **[1]** 4800 = 4.8 kbps • **[2]** 7200 = 7.2 kbps • **[3]** 9600 = 9.6 kbps • **[4]** 12000 = 12.0 kbps • **[5]** 14400 = 14.4 kbps (default) **Note:** The rate is negotiated between both sides (i.e., the device adapts to the capabilities of the remote side). Negotiation of the T.38 maximum supported fax data rate is provided in SIP's SDP T38MaxBitRate parameter. The negotiated T38MaxBitRate is the minimum rate supported between the local and remote endpoints.
Web: Fax Relay ECM Enable OVOC: Relay ECM Enable **[FaxRelayECMEnable]**	Enables Error Correction Mode (ECM) mode during fax relay. • **[0]** Disable • **[1]** Enable (default)
Web: Fax/Modem Bypass Coder Type OVOC: Coder Type **[FaxModemBypassCoderType]**	Determines the coder used by the device when performing fax/modem bypass. Typically, high-bit-rate coders such as G.711 should be used. • **[0]** G.711Alaw= (Default) G.711 A-law 64 • **[1]** G.711Mulaw = G.711 µ-law
Web: Fax/Modem Bypass Packing Factor OVOC: Packetization Period **[FaxModemBypassM]**	Defines the number (20 msec) of coder payloads used to generate a fax/modem bypass packet. The valid range is 1, 2, or 3 coder payloads. The default is 1 coder payload.
[FaxModemNTEMode]	Determines whether the device sends RFC 2833 ANS/ANSam events upon detection of fax and/or modem Answer tones (i.e., CED tone). • **[0]** = Disabled (default) • **[1]** = Enabled **Note:** This parameter is applicable only when the fax or modem transport type is set to bypass or Transparent-with-Events.
Web/OVOC: Fax Bypass Payload Type **[FaxBypassPayloadType]**	Defines the fax bypass RTP dynamic payload type. The valid range is 0 to 127. The default is 102.
OVOC: Modem Bypass Payload Type **[ModemBypassPayloadType]**	Defines the modem bypass dynamic payload type. The range is 0 to 127. The default is 103.

Parameter	Description
OVOC: Relay Volume (dBm) **[FaxModemRelayVolume]**	Defines the fax gain control. The range is -18 to -3, corresponding to -18 dBm to -3 dBm in 1-dB steps. The default is -6 dBm fax gain control.
Web/OVOC: Fax Bypass Output Gain **[FaxBypassOutputGain]**	Defines the fax bypass output gain control. The range is -31 to +31 dB, in 1-dB steps. The default is 0 (i.e., no gain).
Web/OVOC: Modem Bypass Output Gain **[ModemBypassOutputGain]**	Defines the modem bypass output gain control. The range is -31 dB to +31 dB, in 1-dB steps. The default is 0 (i.e., no gain).
OVOC: Basic Packet Interval **[FaxModemBypassBasicRTPPacketInterval]**	Defines the basic frame size used during fax/modem bypass sessions. • **[0]** = (Default) Determined internally • **[1]** = 5 msec (not recommended) • **[2]** = 10 msec • **[3]** = 20 msec **Note:** When set to 5 msec (1), the maximum number of simultaneous channels supported is 120.
OVOC: Dynamic Jitter Buffer Minimal Delay (dB) **[FaxModemBypasDJBufMinDelay]**	Defines the Jitter Buffer delay (in milliseconds) during fax and modem bypass session. The range is 0 to 150 msec. The default is 40.
OVOC: Enable Inband Network Detection **[EnableFaxModemInbandNetworkDetection]**	Enables in-band network detection related to fax/modem. • **[0]** = (Default) Disable. • **[1]** = Enable. When this parameter is enabled on Bypass and transparent with events mode (VxxTransportType is set to 2 or 3), a detection of an Answer Tone from the network triggers a switch to bypass mode in addition to the local Fax/Modem tone detections. However, only a high bit-rate coder voice session effectively detects the Answer Tone sent by a remote endpoint. This can be useful when, for example, the payload of voice and bypass is the same, allowing the originator to switch to bypass mode as well.

Parameter	Description

OVOC: NSE Mode
[NSEMode]
	Enables Cisco compatible fax and modem bypass mode.
	▪ **[0]** = (Default) NSE disabled
	▪ **[1]** = NSE enabled
	In NSE bypass mode, the device starts using G.711 A-Law (default) or G.711μ-Law according to the FaxModemBypassCoderType parameter. The payload type used with these G.711 coders is a standard one (8 for G.711 A-Law and 0 for G.711 μ-Law). The parameters defining payload type for the 'old' Bypass mode FaxBypassPayloadType and ModemBypassPayloadType are not used with NSE Bypass. The bypass packet interval is selected according to the FaxModemBypassBasicRtpPacketInterval parameter.
	Notes:
	▪ This feature can be used only if the VxxModemTransportType parameter is set to 2 (Bypass).
	▪ If NSE mode is enabled, the SDP contains the following line:
	'a=rtpmap:100 X-NSE/8000'.
	▪ To use this feature:
	  ✔ The Cisco gateway must include the following definition:
	    'modem passthrough nse payload-type 100 codec g711alaw'.
	  ✔ Set the Modem transport type to Bypass mode (VxxModemTransportType is set to 2) for all modems.
	  ✔ Configure the gateway parameter NSEPayloadType = 100.
	▪ This parameter can also be configured in an IP Profile.

OVOC: NSE Payload Type **[NSEPayloadType]**	Defines the NSE payload type for Cisco Bypass compatible mode. The valid range is 96-127. The default is 105. **Note:** Cisco gateways usually use NSE payload type of 100.
OVOC: T38 Use RTP Port **[T38UseRTPPort]**	Defines the port (with relation to RTP port) for sending and receiving T.38 packets. ▪ **[0]** = (Default) Use the RTP port +2 to send/receive T.38 packets. ▪ **[1]** = Use the same port as the RTP port to send/receive T.38 packets. **Notes:** ▪ For this parameter to take effect, you must reset the device. ▪ When the device is configured to use V.152 to negotiate audio and T.38 coders, the UDP port published in SDP for RTP and for T38 must be different. Therefore, set the T38UseRTPPort parameter to 0.
Web/OVOC: T.38 Max Datagram Size **[T38MaxDatagramSize]**	Defines the maximum size of a T.38 datagram that the device can receive. This value is included in the outgoing SDP when T.38 is used. The valid range is 120 to 600. The default is 238.

Parameter	Description
Web/OVOC: T38 Fax Max Buffer **[T38FaxMaxBufferSize]**	Defines the maximum size (in bytes) of the device's T.38 buffer. This value is included in the outgoing SDP when T.38 is used for fax relay over IP. The valid range is 500 to 3000. The default is 1024.
Web: Detect Fax on Answer Tone OVOC: Enables Detection of FAX on Answer Tone **[DetFaxOnAnswerTone]**	Determines when the device initiates a T.38 session for fax transmission. ▪ **[0]** Initiate T.38 on Preamble = (Default) The device to which the called fax is connected initiates a T.38 session on receiving HDLC Preamble signal from the fax. ▪ **[1]** Initiate T.38 on CED = The device to which the called fax is connected initiates a T.38 session on receiving a CED answer tone from the fax. This option can only be used to relay fax signals, as the device sends T.38 Re-INVITE on detection of any fax/modem Answer tone (2100 Hz, amplitude modulated 2100 Hz, or 2100 Hz with phase reversals). The modem signal fails when using T.38 for fax relay. **Note:** This parameters is applicable only if the parameter IsFaxUsed is set to 1 (T.38 Relay) or 3 (Fax Fallback).
Web: T38 Fax Session Immediate Start **[T38FaxSessionImmediateStart]**	Enables fax transmission of T.38 "no-signal" packets to the terminating fax machine. ▪ **[0]** Disable (default) ▪ **[1]** Immediate Start on Fax = Device activates T.38 fax relay upon receipt of a re-INVITE with T.38 only in the SDP. ▪ **[2]** Immediate Start on Fax & Voice = Device activates T.38 fax relay upon receipt of a re-INVITE with T.38 and audio media in the SDP. This parameter is used for transmission from fax machines connected to the device and located inside a NAT. Generally, the firewall blocks T.38 (and other) packets received from the WAN, unless the device behind NAT sends at least one IP packet from the LAN to the WAN through the firewall. If the firewall blocks T.38 packets sent from the termination IP fax, the fax fails. To overcome this, the device sends No-Op ("no-signal") packets to open a pinhole in the NAT for the answering fax machine. The originating fax does not wait for an answer, but immediately starts sending T.38 packets to the terminating fax machine. **Note:** To enable No-Op packet transmission, use the NoOpEnable and NoOpInterval parameters.

43.11.2 DTMF and Hook-Flash Parameters

The DTMF and hook-flash parameters are described in the table below.

Table 43-38: DTMF and Hook-Flash Parameters

Parameter	Description
Hook-Flash Parameters	
Web/OVOC: Hook-Flash Code **[HookFlashCode]**	Defines the digit pattern that when received from the Tel side, indicates a Hook Flash event. The valid range is a 25-character string. The default is a null string. **Note:** This parameter can also be configured in a Tel Profile.

Web/OVOC: Hook-Flash Option **[HookFlashOption]**	Determines the hook-flash transport type (i.e., method by which hook-flash is sent and received).
	▪ **[0]** Not Supported = (Default) Hook-Flash indication is not sent.
	▪ **[1]** INFO = Sends proprietary INFO message with Hook-Flash indication.
	▪ **[4]** RFC 2833
	▪ **[5]** INFO (Lucent) = Sends proprietary SIP INFO message with Hook-Flash indication.
	▪ **[6]** INFO (NetCentrex) = Sends proprietary SIP INFO message with Hook-Flash indication. The device sends the INFO message as follows:
	`Content-Type: application/dtmf-relay`
	`Signal=16`
	Where 16 is the DTMF code for hook flash.
	▪ **[7]** INFO (HUAWEI) = Sends a SIP INFO message with Hook-Flash indication. The device sends the INFO message as follows:
	`Content-Length: 17`
	`Content-Type: application/sscc`
	`event=flashhook`
	Notes:
	▪ FXO interfaces support only the receipt of RFC 2833 Hook-Flash signals and INFO **[1]** type.
	▪ FXS interfaces send Hook-Flash signals only if the EnableHold parameter is set to 0.

Web: Min. Flash-Hook Detection Period **[msec]** OVOC: Min Flash Hook Time **[MinFlashHookTime]**	Defines the minimum time (in msec) for detection of a hook-flash event. Detection is guaranteed for hook-flash periods of at least 60 msec (when setting the minimum time to 25). Hook-flash signals that last a shorter period of time are ignored. The valid range is 25 to 300. The default is 300. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to FXS interfaces. ▪ It's recommended to reduce the detection time by 50 msec from the desired value. For example, if you want to set the value to 200 msec, then enter 150 msec (i.e., 200 minus 50).

Parameter	Description

Web: Max. Flash-Hook Detection Period **[msec]**
 OVOC: Flash Hook Period **[FlashHookPeriod]**
	Defines the hook-flash period (in msec) for both Tel and IP sides (per device). For the IP side, it defines the hook-flash period that is reported to the IP.
	
	 For the analog side, it defines the following:
	 ▪ FXS interfaces:
	   ✓ Maximum hook-flash detection period. A longer signal is considered an off-hook or on-hook event.
	   ✓ Hook-flash generation period upon detection of a SIP INFO message containing a hook-flash signal.
	 ▪ FXO interfaces: Hook-flash generation period.
	 The valid range is 25 to 3,000. The default is 700.
	 Notes:
	 ▪ For this parameter to take effect, you need to reset the device.
	 ▪ For FXO interfaces, a constant of 100 msec must be added to the required hook-flash period. For example, to generate a 450 msec hook-flash, set this parameter to 550.
	 ▪ This parameter can also be configured in a Tel Profile.

DTMF Parameters	
OVOC: Use End of DTMF **[MGCPDTMFDetectionPoint]**	Determines when the detection of DTMF events is notified. ▪ **[0]** = DTMF event is reported at the end of a detected DTMF digit. ▪ **[1]** = (Default) DTMF event is reported at the start of a detected DTMF digit.
Web: Declare RFC 2833 in SDP OVOC: Rx DTMF Option **[RxDTMFOption]**	Defines the supported receive DTMF negotiation method. ▪ **[0]** No = Don't declare RFC 2833 telephony-event parameter in SDP. ▪ **[3]** Yes = (Default) Declare RFC 2833 telephony-event parameter in SDP. The device is always receptive to RFC 2833 DTMF relay packets. Therefore, it is always correct to include the 'telephony-event' parameter as default in the SDP. However, some devices use the absence of the 'telephony-event' in the SDP to decide to send DTMF digits in-band using G.711 coder. If this is the case, you can set this parameter to 0. **Note:** This parameter can also be configured in an IP Profile.
Tx DTMF Option Table	
Web/OVOC: Tx DTMF Option **[TxDTMFOption]**	This table parameter configures up to two preferred transmit DTMF negotiation methods. The format of this parameter is as follows: [TxDTMFOption] FORMAT TxDTMFOption_Index = TxDTMFOption_Type; [\TxDTMFOption] Where Type is: ▪ **[0]** Not Supported = (Default) No negotiation - DTMF digits are sent according to the parameters DTMFTransportType and RFC2833PayloadType. ▪ **[1]** INFO (Nortel) = Sends DTMF digits according to IETF Internet-Draft draft-choudhuri-sip-info-digit-00.

Parameter	Description
	[2] NOTIFY = Sends DTMF digits according to IETF Internet-Draft draft-mahy-sipping-signaled-digits-01. **[3]** INFO (Cisco) = Sends DTMF digits according to Cisco format. **[4]** RFC 2833. **[5]** INFO (Korea) = Sends DTMF digits according to Korea Telecom format. For example: TxDTMFOption 0 = 1; TxDTMFOption 1 = 3; **Notes:** - DTMF negotiation methods are prioritized according to the order of their appearance. - When out-of-band DTMF transfer is used (**[1], [2], [3],** or **[5]),** the parameter DTMFTransportType is automatically set to 0 (DTMF digits are erased from the RTP stream). - When RFC 2833 (4) is selected, the device: <blockquote>a. Negotiates RFC 2833 payload type using local and remote SDPs. b. Sends DTMF packets using RFC 2833 payload type according to the payload type in the received SDP. c. Expects to receive RFC 2833 packets with the same payload type as configured by the parameter RFC2833PayloadType. d. Removes DTMF digits in transparent mode (as part of the voice stream).</blockquote> - When TxDTMFOption is set to 0, the RFC 2833 payload type is set according to the parameter RFC2833PayloadType for both transmit and receive. - The table ini file parameter TxDTMFOption can be repeated twice for configuring the DTMF transmit methods. - This parameter can also be configured in an IP Profile.
[DisableAutoDTMFMute]	Enables the automatic muting of DTMF digits when out-of-band DTMF transmission is used. **[0]** = (Default) Automatic mute is used. **[1]** = No automatic mute of in-band DTMF. When this parameter is set to 1, the DTMF transport type is set according to the parameter DTMFTransportType and the DTMF digits aren't muted if out-of-band DTMF mode is selected (TxDTMFOption set to 1, 2 or 3). This enables the sending of DTMF digits in-band (transparent of RFC 2833) in addition to out-of-band DTMF messages. **Note:** Usually this mode is not recommended.
Web/OVOC: Enable Digit Delivery to IP **[EnableDigitDelivery2IP]**	Enables the Digit Delivery feature whereby DTMF digits are sent to the destination IP address after the Tel-to-IP call is answered. **[0]** Disable (default). **[1]** Enable = Enable digit delivery to IP. To enable this feature, modify the called number to include at least one 'p' character. The device uses the digits before the 'p' character in the initial INVITE message. After the call is answered, the device waits for the required time (number of 'p'

Parameter	Description
	multiplied by 1.5 seconds), and then sends the rest of the DTMF digits using the method chosen (in-band or out-of-band). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The called number can include several 'p' characters (1.5 seconds pause), for example, 1001pp699, 8888p9p300.
Web: Enable Digit Delivery to Tel OVOC: Enable Digit Delivery **[EnableDigitDelivery]**	Enables the Digit Delivery feature, which sends DTMF digits of the called number to the device's port (phone line) after the call is answered (i.e., line is off-hooked for FXS, or seized for FXO) for IP-to-Tel calls. ▪ **[0]** Disable (default). ▪ **[1]** Enable = Enable Digit Delivery feature for the FXO/FXS device. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The called number can include characters 'p' (1.5 seconds pause) and 'd' (detection of dial tone). If character 'd' is used, it must be the first 'digit' in the called number. The character 'p' can be used several times. For example (for FXS/FXO interfaces), the called number can be as follows: d1005, dpp699, p9p300. To add the 'd' and 'p' digits, use the usual number manipulation rules. ▪ To use this feature with FXO interfaces, configure the device to operate in one-stage dialing mode. ▪ If this parameter is enabled, it is possible to configure the FXS/FXO interface to wait for dial tone per destination phone number (before or during dialing of destination phone number). Therefore, the parameter IsWaitForDialTone (configurable for the entire device) is ignored. ▪ The FXS interface send SIP 200 OK responses only after the DTMF dialing is complete. ▪ This parameter can also be configured in a Tel Profile.
[ReplaceNumberSignWithEscapeChar]	Determines whether to replace the number sign (#) with the escape character (%23) in outgoing SIP messages for Tel-to-IP calls. ▪ **[0]** Disable (default). ▪ **[1]** Enable = All number signs #, received in the dialed DTMF digits are replaced in the outgoing SIP Request-URI and To headers with the escape sign %23. **Note:** This parameter is applicable only if the parameter IsSpecialDigits is set 1.
Web: Special Digit Representation OVOC: Use Digit For Special DTMF **[UseDigitForSpecialDTMF]**	Defines the representation for 'special' digits ('*' and '#') that are used for out-of-band DTMF signaling (using SIP INFO/NOTIFY). ▪ **[0]** Special = (Default) Uses the strings '*' and '#'. ▪ **[1]** Numeric = Uses the numerical values 10 and 11.
[AdditionalOutOfBandDtmfFormat]	Enables the device to simultaneously send DTMF tones (signals) in SIP messages, e.g., INFO (out-of-band) and in RTP media streams (in-band) with a special payload type (as defined in RFC 2833), when the FirstTxDTMFOption parameter is configured to 4.

Parameter	Description
	▪ [0] unknown = (Default) DTMF is sent according to FirstTxDTMFOption. ▪ [1] Nortel ▪ [2] cisco ▪ [3] threecom ▪ [4] korea

43.11.3 Digit Collection and Dial Plan Parameters

The digit collection and dial plan parameters are described in the table below.

Table 43-39: Digit Collection and Dial Plan Parameters

Parameter	Description
Web/OVOC: Dial Plan Index **[DialPlanIndex]**	Defines the Dial Plan index to use in the external Dial Plan file. The Dial Plan file is loaded to the device as a .dat file (converted using the DConvert utility). The Dial Plan index can be defined globally or per Tel Profile. The valid value range is 0 to 7, where 0 denotes PLAN1, 1 denotes PLAN2, and so on. The default is -1, indicating that no Dial Plan file is used. **Notes:** ▪ If this parameter is configured to select a Dial Plan index, the settings of the parameter DigitMapping are ignored. ▪ If this parameter is configured to select a Dial Plan index from an external Dial Plan file, the device first attempts to locate a matching digit pattern in the Dial Plan file, and if not found, then attempts to locate a matching digit pattern in the Digit Map rules configured by the DigitMapping parameter. ▪ This parameter can also be configured in a Tel Profile. ▪ For more information on the Dial Plan file, see 'Dialing Plans for Digit Collection' on page 372.
[Tel2IPSourceNumberMappingDialPlanIndex]	Defines the Dial Plan index in the external Dial Plan file for the Tel-to-IP Source Number Mapping feature. The valid value range is 0 to 7, defining the Dial Plan index **[Plan x]** in the Dial Plan file. The default is -1 (disabled). For more information on this feature, see Modifying ISDN-to-IP Calling Party Number.
Web: Digit Mapping Rules OVOC: Digit Map Pat terns **[DigitMapping]**	Defines the digit map pattern. If the digit string (i.e., dialed number) matches one of the patterns in the digit map, the device stops collecting digits and establishes a call with the collected number. The digit map pattern can contain up to 52 options (rules), each separated by a vertical bar (|). The maximum length of the entire digit pattern is 152 characters. The available notations include the following: ▪ **[n-m]:** Range of numbers (not letters). ▪ **.** (single dot): Repeat digits until next notation (e.g., T). ▪ **x:** Any single digit. ▪ **T:** Dial timeout (configured by the TimeBetweenDigits parameter).

Parameter	Description
	▪ **S:** Short timer (configured by the TimeBetweenDigits parameter; default is two seconds) that can be used when a specific rule is defined after a more general rule. For example, if the digit map is 99|998, then the digit collection is terminated after the first two 9 digits are received. Therefore, the second rule of 998 can never be matched. But when the digit map is 99s|998, then after dialing the first two 9 digits, the device waits another two seconds within which the caller can enter the digit 8.
	An example of a digit map is shown below: 11xS|00T|**[1-7]**xxx|8xxxxxxx|#xxxxxxx|*xx|91xxxxxxxxxx|9011x.T In the example above, the last rule can apply to International numbers: 9 for dialing tone, 011 Country Code, and then any number of digits for the local number ('x.').
	Notes:
	▪ If the DialPlanIndex parameter is configured (to select a Dial Plan index), then the device first attempts to locate a matching digit pattern in the Dial Plan file, and if not found, then attempts to locate a matching digit pattern in the Digit Map rules configured by the DigitMapping parameter.
	▪ For more information on digit mapping, see 'Digit Mapping' on page 269.
Web: Max Digits in Phone Num OVOC: Max Digits in Phone Number **[MaxDigits]**	Defines the maximum number of collected destination number digits that can be received (i.e., dialed) from the Tel side. When the number of collected digits reaches this maximum, the device uses these digits for the called destination number.
	The valid range is 1 to 49. The default is 5.
	Notes**:**
	▪ Instead of using this parameter, Digit Mapping rules can be configured.
	▪ Dialing ends when any of the following scenarios occur: ✓ Maximum number of digits is dialed ✓ Interdigit Timeout (TimeBetweenDigits) expires ✓ Pound (#) key is pressed ✓ Digit map pattern is matched
Web: Inter Digit Timeout for Overlap Dialing **[sec]** OVOC: Interdigit Timeout (Sec) **[TimeBetweenDigits]**	Defines the time (in seconds) that the device waits between digits that are dialed by the user.
	When this inter-digit timeout expires, the device uses the collected digits to dial the called destination number.
	The valid range is 1 to 10. The default is 4.
Web: Enable Special Digits OVOC: Use '#' For Dial Termination **[IsSpecialDigits]**	Determines whether the asterisk (*) and pound (#) digits can be used in DTMF.
	▪ **[0]** Disable = Use '*' or '#' to terminate number collection (refer to the parameter UseDigitForSpecialDTMF). (Default.)
	▪ **[1]** Enable = Allows '*' and '#' for telephone numbers dialed by a user or for the endpoint telephone number.
	Note: These symbols can always be used as the first digit of a dialed number even if you disable this parameter.

43.11.4 Voice Mail Parameters

The voice mail parameters are described in the table below. For more information on the Voice Mail application, refer to the *CPE Configuration Guide for Voice Mail*.

> **Note:** Voice Mail is applicable only to FXO interfaces.

Table 43-40: Voice Mail Parameters

Parameter	Description
Web/OVOC: Voice Mail Interface **[VoiceMailInterface]**	Enables the device's Voice Mail application and determines the communication method used between the PBX and the device. ▪ **[0]** None (default) ▪ **[1]** DTMF ▪ **[2]** SMDI **Note:** To disable voice mail per Hunt Group, you can use a Tel Profile with the EnableVoiceMailDelay parameter set to disabled (0). This eliminates the phenomenon of call delay on lines not implementing voice mail when voice mail is enabled using this global parameter.
Web: Enable VoiceMail URI OVOC: Enable VMURI **[EnableVMURI]**	Enables the interworking of target and cause for redirection from Tel to IP and vice versa, according to RFC 4468. ▪ **[0]** Disable (default) ▪ **[1]** Enable
[WaitForBusyTime]	Defines the time (in msec) that the device waits to detect busy and/or reorder tones. This feature is used for semi-supervised PBX call transfers (i.e., the LineTransferMode parameter is set to 2). The valid value range is 0 to 20000 (i.e., 20 sec). The default is 2000 (i.e., 2 sec).

Web/OVOC: Line Transfer Mode **[LineTransferMode]**	Defines the call transfer method used by the device. This parameter is applicable to FXO call transfer.
	▪ **[0]** None = (Default) IP.
	▪ **[1]** Blind = PBX blind transfer:
	  ✓ After receiving a SIP REFER message from the IP side, the device (FXO) sends a hook-flash to the PBX, dials the digits (that are received in the Refer-To header), and then immediately releases the line (i.e., on-hook). The PBX performs the transfer internally.
	▪ **[2]** Semi Supervised = PBX semi-supervised transfer:
	  ✓ After receiving a SIP REFER message from the IP side, the device sends a hook-flash to the PBX, and then dials the digits (that are received in the Refer-To header). If no busy or reorder tones are detected (within the device completes the call transfer by releasing the line. If these tones are detected, the transfer is cancelled, the device sends a SIP NOTIFY message with a failure reason in the NOTIFY body (such as 486 if busy tone detected), and generates an

Parameter	Description
	additional hook-flash toward the FXO line to restore connection to the original call.
	▪ **[3]** Supervised = PBX Supervised transfer:
	✓ After receiving a SIP REFER message from the IP side, the device sends a hook-flash to the PBX, and then dials the digits (that are received in the Refer-To header). The device waits for connection of the transferred call and then completes the call transfer by releasing the line. If speech is not detected, the transfer is cancelled, the device sends a SIP NOTIFY message with a failure reason in the NOTIFY body (such as 486 if busy tone detected) and generates an additional hook-flash toward the FXO line to restore connection to the original call.
SMDI Parameters	
Web/OVOC: Enable SMDI **[SMDI]**	Enables Simplified Message Desk Interface (SMDI) interface on the device. ▪ **[0]** Disable = (Default) Normal serial ▪ **[1]** Enable (Bellcore) ▪ **[2]** Ericsson MD-110 ▪ **[3]** NEC (ICS) **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When the RS-232 connection is used for SMDI messages (Serial SMDI), it cannot be used for other applications, for example, to access the Command Line Interface (CLI).
Web/OVOC: SMDI Timeout **[SMDITimeOut]**	Defines the time (in msec) that the device waits for an SMDI Call Status message before or after a Setup message is received. This parameter synchronizes the SMDI and analog CAS interfaces. If the timeout expires and only an SMDI message is received, the SMDI message is dropped. If the timeout expires and only a Setup message is received, the call is established. The valid range is 0 to 10000 (i.e., 10 seconds). The default is 2000.
Message Waiting Indication (MWI) Parameters	
Web: MWI Off Digit Pattern OVOC: MWI Off Code **[MWIOffCode]**	Defines the digit code used by the device to notify the PBX that there are no messages waiting for a specific extension. This code is added as prefix to the dialed number. The valid range is a 25-character string.
Web: MWI On Digit Pattern OVOC: MWI On Code **[MWIOnCode]**	Defines the digit code used by the device to notify the PBX of messages waiting for a specific extension. This code is added as prefix to the dialed number. The valid range is a 25-character string.
Web: MWI Suffix Pattern OVOC: MWI Suffix Code **[MWISuffixCode]**	Defines the digit code used by the device as a suffix for 'MWI On Digit Pattern' and 'MWI Off Digit Pattern'. This suffix is added to the generated DTMF string after the extension number. The valid range is a 25-character string.

Parameter	Description
Web: MWI Source Number OVOC: MWI Source Name **[MWISourceNumber]**	Defines the calling party's phone number used in the Q.931 MWI Setup message to PSTN. If not configured, the channel's phone number is used as the calling number.

[MWISubscribeIPGroupID]	Defines the IP Group ID used when subscribing to an MWI server. The 'The SIP Group Name' field value of the IP Group table is used as the Request-URI host name in the outgoing MWI SIP SUBSCRIBE message. The request is sent to the IP address defined for the Proxy Set that is associated with the IP Group. The Proxy Set's capabilities such as proxy redundancy and load balancing are also applied to the message.
	
	For example, if the 'SIP Group Name' field of the IP Group is set to "company.com", the device sends the following SUBSCRIBE message:
	`SUBSCRIBE sip:company.com...`
	Instead of:
	`SUBSCRIBE sip:10.33.10.10...`
	
	Note: If this parameter is not configured, the MWI SUBSCRIBE message is sent to the MWI server as defined by the MWIServerIP parameter.

Digit Patterns The following digit pattern parameters apply only to voice mail applications that use the DTMF communication method. For the available pattern syntaxes, refer to the *CPE Configuration Guide for Voice Mail*.	
Web: Forward on Busy Digit Pattern (Internal) OVOC: Digit Pattern Forward On Busy **[DigitPatternForwardOnBusy]**	Defines the digit pattern used by the PBX to indicate 'call forward on busy' when the original call is received from an internal extension. The valid range is a 120-character string.
Web: Forward on No Answer Digit Pattern (Internal) OVOC: Digit Pattern Forward On No Answer **[DigitPatternForwardOnNoAnswer]**	Defines the digit pattern used by the PBX to indicate 'call forward on no answer' when the original call is received from an internal extension. The valid range is a 120-character string.
Web: Forward on Do Not Disturb Digit Pattern (Internal) OVOC: Digit Pattern Forward On DND **[DigitPatternForwardOnDND]**	Defines the digit pattern used by the PBX to indicate 'call forward on do not disturb' when the original call is received from an internal extension. The valid range is a 120-character string.
Web: Forward on No Reason Digit Pattern (Internal) OVOC: Digit Pattern Forward No Reason **[DigitPatternForwardNoReason]**	Defines the digit pattern used by the PBX to indicate 'call forward with no reason' when the original call is received from an internal extension. The valid range is a 120-character string.
Web: Forward on Busy Digit Pattern (External) OVOC: VM Digit Pattern On Busy External **[DigitPatternForwardOnBusyExt]**	Defines the digit pattern used by the PBX to indicate 'call forward on busy' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.

Parameter	Description
Web: Forward on No Answer Digit Pattern (External) OVOC: VM Digit Pattern On No Answer Ext **[DigitPatternForwardOnNoAnswerExt]**	Defines the digit pattern used by the PBX to indicate 'call forward on no answer' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Web: Forward on Do Not Disturb Digit Pattern (External) OVOC: VM Digit Pattern On DND External **[DigitPatternForwardOnDNDExt]**	Defines the digit pattern used by the PBX to indicate 'call forward on do not disturb' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Web: Forward on No Reason Digit Pattern (External) OVOC: VM Digit Pattern No Reason External **[DigitPatternForwardNoReasonExt]**	Defines the digit pattern used by the PBX to indicate 'call forward with no reason' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Web: Internal Call Digit Pattern OVOC: Digit Pattern Internal Call **[DigitPatternInternalCall]**	Defines the digit pattern used by the PBX to indicate an internal call. The valid range is a 120-character string.
Web: External Call Digit Pattern OVOC: Digit Pattern External Call **[DigitPatternExternalCall]**	Defines the digit pattern used by the PBX to indicate an external call. The valid range is a 120-character string.
Web: Disconnect Call Digit Pattern OVOC: Tel Disconnect Code **[TelDisconnectCode]**	Defines a digit pattern that when received from the Tel side, indicates the device to disconnect the call. The valid range is a 25-character string.
Web: Digit To Ignore Digit Pattern OVOC: Digit To Ignore **[DigitPatternDigitToIgnore]**	Defines a digit pattern that if received as Src (S) or Redirect (R) numbers is ignored and not added to that number. The valid range is a 25-character string.

43.11.5 Supplementary Services Parameters

This subsection describes the device's supplementary telephony services parameters.

43.11.5.1 Caller ID Parameters

The caller ID parameters are described in the table below.

Table 43-41: Caller ID Parameters

Parameter	Description
Caller ID Permissions Table	
Web: Caller ID Permissions Table OVOC: SIP Endpoints > Caller ID **[EnableCallerID]**	This table parameter enables (per port) Caller ID generation (for FXS interfaces) and detection (for FXO interfaces). The format of this parameter is as follows: [EnableCallerID] FORMAT EnableCallerID_Index = EnableCallerID_IsEnabled; [\EnableCallerID] Where, • Index = Port number (where 0 denotes Port 1). For example: EnableCallerID 0 = 1; (caller ID enabled on Port 1) EnableCallerID 1 = 0; (caller ID disabled on Port 2) **Note:** For a detailed description of this table, see Configuring Caller ID Permissions on page 308.
Caller Display Information Table	
Web: Caller Display Information Table OVOC: SIP Endpoints > Caller ID **[CallerDisplayInfo]**	This table parameter enables the device to send Caller ID information to the IP side when a call is made. The called party can use this information for caller identification. The information configured in this table is sent in the SIP INVITE message's From header. The format of this parameter is as follows: [CallerDisplayInfo] FORMAT CallerDisplayInfo_Index = CallerDisplayInfo_DisplayString, CallerDisplayInfo_IsCidRestricted; [\CallerDisplayInfo] Where, • Index = Port number, where 0 denotes Port 1. For example: CallerDisplayInfo 0 = Susan C.,0; ("Susan C." is sent as the Caller ID for Port 1) CallerDisplayInfo 1 = Mark M.,0; ("Mark M." is sent as Caller ID for Port 2) **Note:** For a detailed description of this table, see Configuring Caller Display Information on page 305.

Parameter	Description
Web/OVOC: Enable Caller ID **[EnableCallerID]**	Enables Caller ID. • **[0]** Disable (default) • **[1]** Enable If the Caller ID service is enabled, then for FXS interfaces, calling number and Display text (from IP) are sent to the device's port. For FXO interfaces, the Caller ID signal is detected and sent to IP in the SIP INVITE message (as 'Display' element). For information on the Caller ID table, see Configuring Caller Display Information on page 305. To disable/enable caller ID generation per port, see Configuring Call Forward on page 307.
Web: Caller ID Type OVOC: Caller id Types **[CallerIDType]**	Determines the standard used for detection (FXO) and generation (FXS) of Caller ID, and detection (FXO) / generation (FXS) of MWI (when specified) signals: • **[0]** Standard Bellcore = (Default) Caller ID and MWI • **[1]** Standard ETSI = Caller ID and MWI • **[2]** Standard NTT • **[4]** Standard BT = Britain • **[16]** Standard DTMF Based ETSI • **[17]** Standard Denmark = Caller ID and MWI • **[18]** Standard India • **[19]** Standard Brazil **Notes:** • Typically, the Caller ID signals are generated / detected between the first and second rings. However, sometimes the Caller ID is detected before the first ring signal. In such a scenario, set the RingsBeforeCallerID parameter to 0. • Caller ID detection for Britain **[4]** is not supported on the device's FXO ports. Only FXS ports can generate the Britain **[4]** Caller ID. • To select the Bellcore Caller ID sub standard, use the BellcoreCallerIDTypeOneSubStandard parameter. To select the ETSI Caller ID substandard, use the ETSICallerIDTypeOneSubStandard parameter. • To select the Bellcore MWI sub standard, use the BellcoreVMWITypeOneStandard parameter. To select the ETSI MWI sub standard, use the ETSIVMWITypeOneStandard parameter. • If you define Caller ID Type as NTT **[2],** you need to define the NTT DID signaling form (FSK or DTMF) using the NTTDIDSignallingForm parameter.

Parameter	Description
Web: Enable FXS Caller ID Category Digit For Brazil Telecom **[AddCPCPrefix2BrazilCallerID]**	Enables the interworking of Calling Party Category (cpc) code from SIP INVITE messages to FXS Caller ID first digit. ▪ **[0]** Disable (default) ▪ **[1]** Enable When this parameter is enabled, the device sends the Caller ID number (calling number) with the cpc code (received in the SIP INVITE message) to the device's FXS port. The cpc code is added as a prefix to the caller ID (after IP-to-Tel calling number manipulation). For example, assuming that the incoming INVITE contains the following From (or P-Asserted-Id) header: From:<sip:+551137077801;cpc=payphone@10.20.7.35>;tag=53700 The calling number manipulation removes "+55" (leaving 10 digits), and then adds the prefix 7, the cpc code for payphone user. Therefore, the Caller ID number that is sent to the FXS port, in this example is 71137077801. If the incoming INVITE message doesn't contain the 'cpc' parameter, nothing is added to the Caller ID number. | CPC Value in Received INVITE	CPC Code Prefixed to Caller ID (Sent to FXS Endpoint)	Description
---	---	---
cpc=unknown	1	Unknown user
cpc=subscribe	1	-
cpc=ordinary	1	Ordinary user
cpc=priority	2	Pre-paid user
cpc=test	3	Test user
cpc=operator	5	Operator
cpc=data	6	Data call
cpc=payphone	7	Payphone user	 **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ For this parameter to be enabled, you must also set the parameter EnableCallingPartyCategory to 1.
[EnableCallerIDTypeTwo]	Disables the generation of Caller ID type 2 when the phone is off-hooked. Caller ID type 2 (also known as off-hook Caller ID) is sent to a currently busy telephone to display the caller ID of the waiting call. ▪ **[0]** = Caller ID type 2 isn't played. ▪ **[1]** = (Default) Caller ID type 2 is played.

Parameter	Description
OVOC: Caller ID Timing Mode **[AnalogCallerIDTimingMode]**	Determines when Caller ID is generated. ▪ **[0]** = (Default) Caller ID is generated between the first two rings. ▪ **[1]** = The device attempts to find an optimized timing to generate the Caller ID according to the selected Caller ID type. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ If this parameter is set to 1 and used with distinctive ringing, the Caller ID signal doesn't change the distinctive ringing timing. ▪ For this parameter to take effect, a device reset is required.
OVOC: Bellcore Caller ID Type One Sub Standard **[BellcoreCallerIDTypeOneSubStandard]**	Determines the Bellcore Caller ID sub-standard. ▪ **[0]** = (Default) Between rings. ▪ **[1]** = Not ring related. **Note:** For this parameter to take effect, a device reset is required.
OVOC: ETSI Caller ID Type One Sub Standard **[ETSICallerIDTypeOneSubStandard]**	Determines the ETSI FSK Caller ID Type 1 sub-standard (FXS only). ▪ **[0]** = (Default) ETSI between rings. ▪ **[1]** = ETSI before ring DT_AS. ▪ **[2]** = ETSI before ring RP_AS. ▪ **[3]** = ETSI before ring LR_DT_AS. ▪ **[4]** = ETSI not ring related DT_AS. ▪ **[5]** = ETSI not ring related RP_AS. ▪ **[6]** = ETSI not ring related LR_DT_AS. **Note:** For this parameter to take effect, a device reset is required.
Web: Asserted Identity Mode OVOC: Asserted ID Mode **[AssertedIdMode]**	Determines whether the SIP header P-Asserted-Identity or P-Preferred-Identity is used in the generated SIP INVITE, 200 OK, or UPDATE request for Caller ID (or privacy). These headers are used to present the originating party's Caller ID. The Caller ID is composed of a Calling Number and a Calling Name (optional). ▪ **[0]** Disabled = (Default) P-Asserted-Identity nor P-Preferred-Identity headers are not added. ▪ **[1]** Add P-Asserted-Identity ▪ **[2]** Add P-Preferred-Identity The header used also depends on the calling Privacy (allowed or restricted). These headers are used together with the Privacy header. If Caller ID is restricted (i.e., P-Asserted-Identity is not sent), the Privacy header includes the value 'id' ('Privacy: id'). Otherwise, for allowed Caller ID, 'Privacy: none' is used. If Caller ID is restricted (received from PSTN / Tel or configured in the device), the From header is set to <anonymous@anonymous.invalid>.

Parameter	Description
Web/OVOC: Use Destination As Connected Number **[UseDestinationAsConnectedNumber]**	Enables the device to include the Called Party Number, from outgoing Tel calls (after number manipulation), in the SIP P-Asserted-Identity header. The device includes the SIP P-Asserted-Identity header in 180 Ringing and 200 OK responses for IP-to-Tel calls. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ For this feature, you must also enable the device to include the P-Asserted-Identity header in 180/200 OK responses, by setting the parameter AssertedIDMode to **Add P-Asserted-Identity**. ▪ This parameter is applicable to FXO interfaces.
Web: Caller ID Transport Type OVOC: Transport Type **[CallerIDTransportType]**	Determines the device's behavior for Caller ID detection. ▪ **[0]** Disable = The caller ID signal is not detected - DTMF digits remain in the voice stream. ▪ **[1]** Relay = (Currently not applicable.) ▪ **[3]** Mute = (Default) The caller ID signal is detected from the Tel side and then erased from the voice stream. **Note:** Caller ID detection is applicable only to FXO interfaces.
Reject Anonymous Calls Per Port Table	
[RejectAnonymousCallPerPort]	This table parameter determines whether the device rejects incoming anonymous calls. If enabled, when a device's FXS interface receives an anonymous call, it rejects the call and responds with a SIP 433 (Anonymity Disallowed) response. The format of this parameter is as follows: [RejectAnonymousCallPerPort] FORMAT RejectAnonymousCallPerPort_Index = RejectAnonymousCallPerPort_Enable; [\RejectAnonymousCallPerPort] Where, ▪ Enable = accept **[0]** (default) or reject **[1]** incoming anonymous calls. For example: RejectAnonymousCallPerPort 0 = 0; RejectAnonymousCallPerPort 1 = 1; **Note:** This parameter is applicable only to FXS interfaces.

43.11.5.2   Call Waiting Parameters

The call waiting parameters are described in the table below.

Table 43-42: Call Waiting Parameters

Parameter	Description
Web/OVOC: Enable Call Waiting **[EnableCallWaiting]**	Enables the Call Waiting feature. ▪ **[0]** Disable ▪ **[1]** Enable (Default) If enabled, when an FXS interface receives a call on a busy endpoint, it responds with a 182 response (and not with a 486 busy). The device plays a call waiting indication signal. When hook-flash is detected, the device switches to the waiting call. The device that initiated the waiting call plays a call waiting ringback tone to the calling party after a 182 response is received. **Notes:** ▪ The device's Call Progress Tones (CPT) file must include a Call Waiting ringback tone (caller side) and a call waiting tone (called side, FXS only). ▪ The EnableHold parameter must be enabled on both the calling and the called side. ▪ You can use the table parameter CallWaitingPerPort to enable Call Waiting per port. ▪ For information on the Call Waiting feature, see Call Waiting on page 282.
OVOC: Send 180 For Call Waiting **[Send180ForCallWaiting]**	Determines the SIP response code for indicating Call Waiting. ▪ **[0]** = (Default) Use 182 Queued response to indicate call waiting. ▪ **[1]** = Use 180 Ringing response to indicate call waiting.
Call Waiting Table	
Web: Call Waiting Table OVOC: SIP Endpoints > Call Waiting **[CallWaitingPerPort]**	This table parameter configures call waiting per FXS port. The format of this parameter is as follows: [CallWaitingPerPort] FORMAT CallWaitingPerPort_Index = CallWaitingPerPort_IsEnabled; [\CallWaitingPerPort] For example: CallWaitingPerPort 0 = 0; (call waiting disabled for Port 1) CallWaitingPerPort 1 = 1; (call waiting enabled for Port 2) **Notes:** ▪ Index denotes port number, where 0 denotes Port 1. ▪ This parameter is applicable only to FXS ports. ▪ For a detailed description of this table, see Configuring Call Waiting on page 309.

Parameter	Description
Web: Number of Call Waiting Indications OVOC: Call Waiting Number of Indications **[NumberOfWaitingIndications]**	Defines the number of call waiting indications that are played to the called telephone that is connected to the device for Call Waiting. The valid range is 1 to 100 indications. The default is 2. **Note:** This parameter is applicable only to FXS ports.
Web: Time Between Call Waiting Indications OVOC: Call Waiting Time Between Indications **[TimeBetweenWaitingIndications]**	Defines the time (in seconds) between consecutive call waiting indications for call waiting. The valid range is 1 to 100. The default is 10. **Note:** This parameter is applicable only to FXS ports.
Web/OVOC: Time Before Waiting Indications **[TimeBeforeWaitingIndications]**	Defines the interval (in seconds) before a call waiting indication is played to the port that is currently in a call. The valid range is 0 to 100. The default time is 0 seconds. **Note:** This parameter is applicable only to FXS ports.
Web/OVOC: Waiting Beep Duration **[WaitingBeepDuration]**	Defines the duration (in msec) of call waiting indications that are played to the port that is receiving the call. The valid range is 100 to 65535. The default is 300. **Note:** This parameter is applicable only to FXS ports.

OVOC: First Call Waiting Tone ID
[FirstCallWaitingToneID]
	Defines the index of the first Call Waiting Tone in the CPT file. This feature enables the called party to distinguish between different call origins (e.g., external versus internal calls).
	There are three ways to use the distinctive call waiting tones:
	▪ Playing the call waiting tone according to the SIP Alert-Info header in the received 180 Ringing SIP response. The value of the Alert-Info header is added to the value of the FirstCallWaitingToneID parameter.
	▪ Playing the call waiting tone according to PriorityIndex in the ToneIndex table parameter.
	▪ Playing the call waiting tone according to the parameter "CallWaitingTone#' of a SIP INFO message.
	The device plays the tone received in the 'play tone CallWaitingTone#' parameter of an INFO message plus the value of this parameter minus 1.
	The valid range is -1 to 1,000. The default is -1 (i.e., not used).
	Notes:
	▪ It is assumed that all Call Waiting Tones are defined in sequence in the CPT file.
	▪ SIP Alert-Info header examples:
	  ✓ Alert-Info:<Bellcore-dr2>
	  ✓ Alert-Info:<http://…/Bellcore-dr2> (where "dr2" defines call waiting tone #2)
	▪ The SIP INFO message is according to Broadsoft's application server definition. Below is an example of such an INFO message:

Parameter	Description
	INFO sip:06@192.168.13.2:5060 SIP/2.0 Via:SIP/2.0/UDP 192.168.13.40:5060;branch=z9hG4bK040066422630 From: <sip:4505656002@192.168.13.40:5060>;tag=1455352915 To: <sip:06@192.168.13.2:5060> Call-ID:0010-0008@192.168.13.2 CSeq:342168303 INFO Content-Length:28 Content-Type:application/broadsoft play tone CallWaitingTone1

43.11.5.3   Call Forwarding Parameters

The call forwarding parameters are described in the table below.

Table 43-43: Call Forwarding Parameters

Parameter	Description
Web: Enable Call Forward **[EnableForward]**	Enables the Call Forwarding feature. ▪ **[0]** Disable ▪ **[1]** Enable (Default) For FXS interfaces, the Call Forward table (FwdInfo parameter) must be defined to use the Call Forward service. The device uses SIP REFER messages for call forwarding. **Notes:** ▪ To use this service, the devices at both ends must support this option. ▪ For the device to respond to SIP 3xx responses with a new SIP request (forwarding the original request), set this parameter to Enable.
Call Forwarding Table	
Web: Call Forwarding Table OVOC: Analog Gateway Provisioning > Tab: Call Forward **[FwdInfo]**	This table parameter configures call forwarding of IP-to-Tel calls (using SIP 302 response) to other device ports or an IP destination, based on the device's port to which the call was originally routed. The format of this parameter is as follows: [FwdInfo] FORMAT FwdInfo_Index = FwdInfo_Type, FwdInfo_Destination, FwdInfo_NoReplyTime; [\FwdInfo] Where, ▪ Index = Port number, where 0 denotes Port 1. For example: ▪ Below configuration forwards calls originally destined to Port 1 to "1001" upon On Busy: FwdInfo 0 = 1,1001,30; ▪ Below configuration forwards calls originally destined to Port 2 to an IP address upon On Busy: FwdInfo 1 = 1,2003@10.5.1.1,30; **Note:** For a detailed description of this table, see Configuring Call Forward on page 307.

Parameter	Description
Call Forward Reminder Ring Parameters **Notes:** ▪ These parameters are applicable only to FXS interfaces. ▪ For a description of this feature, see Call Forward Reminder Ring on page 279.	
Web/OVOC: Enable NRT Subscription **[EnableNRTSubscription]**	Enables endpoint subscription for Ring reminder event notification feature. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: AS Subscribe IPGroupID **[ASSubscribeIPGroupID]**	Defines the IP Group ID that contains the Application server for Subscription. The valid value range is 1 to 8. The default is -1 (i.e., not configured).
Web: NRT Retry Subscription Time OVOC: NRT Subscription Retry Time **[NRTSubscribeRetryTime]**	Defines the Retry period (in seconds) for Dialog subscription if a previous request failed. The valid value range is 10 to 7200. The default is 120.
Web/OVOC: Call Forward Ring Tone ID **[CallForwardRingToneID]**	Defines the ringing tone type played when call forward notification is accepted. The valid value range is 1 to 5. The default is 1.

43.11.5.4   Message Waiting Indication Parameters

The message waiting indication (MWI) parameters are described in the table below.

Table 43-44: MWI Parameters

Parameter	Description
Web: Enable MWI OVOC: MWI Enable **[EnableMWI]**	Enables Message Waiting Indication (MWI). ▪ **[0]** Disable (default). ▪ **[1]** Enable **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ The device supports only the receipt of SIP MWI NOTIFY messages (the device doesn't generate these messages). ▪ For more information on MWI, see 'Message Waiting Indication' on page 282.
Web/OVOC: MWI Analog Lamp **[MWIAnalogLamp]**	Enables the visual display of MWI. ▪ **[0]** Disable (default). ▪ **[1]** Enable = Enables visual MWI by supplying line voltage of approximately 100 VDC to activate the phone's lamp. **Notes:** ▪ This parameter is applicable only for FXS interfaces. ▪ This parameter can also be configured in a Tel Profile.

Parameter	Description
Web/OVOC: MWI Display **[MWIDisplay]**	Enables sending MWI information to the phone display. ▪ **[0]** Disable = (Default) MWI information isn't sent to display. ▪ **[1]** Enable = The device generates an MWI message (determined by the parameter CallerIDType), which is displayed on the MWI display. **Note:** ▪ This parameter is applicable only to FXS interfaces. ▪ This parameter can also be configured in a Tel Profile.
Web: Subscribe to MWI OVOC: Enable MWI Subscription **[EnableMWISubscription]**	Enables subscription to an MWI server. ▪ **[0]** No (default) ▪ **[1]** Yes **Notes:** ▪ To configure the MWI server address, use the MWIServerIP parameter. ▪ To configure whether the device subscribes per endpoint or per the entire device, use the parameter SubscriptionMode.
Web: MWI Server IP Address OVOC: MWI Server IP **[MWIServerIP]**	Defines the MWI server's IP address. If provided, the device subscribes to this IP address. The MWI server address can be configured as a numerical IP address or as a domain name. If not configured, the Proxy IP address is used instead.
Web/OVOC: MWI Server Transport Type **[MWIServerTransportType]**	Determines the transport layer used for outgoing SIP dialogs initiated by the device to the MWI server. ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When set to 'Not Configured', the value of the parameter SIPTransportType is used.
Web: MWI Subscribe Expiration Time OVOC: MWI Expiration Time **[MWIExpirationTime]**	Defines the MWI subscription expiration time in seconds. The default is 7200 seconds. The range is 10 to 2,000,000.
Web: MWI Subscribe Retry Time OVOC: Subscribe Retry Time **[SubscribeRetryTime]**	Defines the subscription retry time (in seconds) after last subscription failure. The default is 120 seconds. The range is 10 to 2,000,000.
Web: Subscription Mode **[SubscriptionMode]**	Determines the method the device uses to subscribe to an MWI server. ▪ **[0]** Per Endpoint = (Default) Each endpoint subscribes separately - typically used for FXS interfaces. ▪ **[1]** Per Gateway = Single subscription for the entire device - typically used for FXO interfaces.
OVOC: ETSI VMWI Type One Standard **[ETSIVMWITypeOneStandard]**	Determines the ETSI Visual Message Waiting Indication (VMWI) Type 1 sub-standard. ▪ **[0]** = (Default) ETSI VMWI between rings ▪ **[1]** = ETSI VMWI before ring DT_AS

Parameter	Description
	▪ **[2]** = ETSI VMWI before ring RP_AS ▪ **[3]** = ETSI VMWI before ring LR_DT_AS ▪ **[4]** = ETSI VMWI not ring related DT_AS ▪ **[5]** = ETSI VMWI not ring related RP_AS ▪ **[6]** = ETSI VMWI not ring related LR_DT_AS **Note:** For this parameter to take effect, a device reset is required.
OVOC: Bellcore VMWI Type One Standard **[BellcoreVMWITypeOneStandard]**	Determines the Bellcore VMWI sub-standard. ▪ **[0]** = (Default) Between rings. ▪ **[1]** = Not ring related. **Note:** For this parameter to take effect, a device reset is required.

[EnableLowVoltageMwiGeneration]	Defines the Message Waiting Indication (MWI) voltage level mode (low or high) that the FXS port generates to light a lamp on an FXS phone to indicate a message in waiting.
	▪ [0] = (Default) The FXS port generates a high DC voltage (90 VDC) for the MWI signal:
	  ✓ **Constant MWI Light Mode:** The FXS generates a constant high DC voltage to light up the MWI indicator on the phone. For this setup, make sure that the parameters NeonMwiOnDurationTime and NeonMwiOffDurationTime are configured to 1234 (default). (The parameters LedMwiOnDurationTime and LedMwiOffDurationTime are not applicable.)
	  ✓ **Blinking MWI Light Mode:** The FXS generates a high DC voltage to light up the MWI indicator and a normal on-hook voltage to turn it off. To configure the on-off light duration (blinking duty cycle), use the parameters NeonMwiOnDurationTime and NeonMwiOffDurationTime. This blinking mode is less recommended since the voltage transitions might "ding" the phone's ringer. (The parameters LedMwiOnDurationTime and LedMwiOffDurationTime are not applicable.)
	▪ [1] = The FXS port generates low on-hook voltage transitions (at 50 Hz) for the MWI signal:
	  ✓ **Constant MWI Light Mode:** The FXS constantly generates on-hook voltage transitions to constantly light up the MWI indicator on the phone. For this setup, make sure that the parameters LedMwiOnDurationTime and LedMwiOffDurationTime are configured to 1234 (default). (The parameters NeonMwiOnDurationTime and NeonMwiOffDurationTime are not applicable.)
	  ✓ **Blinking MWI Light Mode:** The FXS generates on-hook voltage transitions to light up the MWI indicator and stops transitioning the on-hook voltage to turn it off. To configure the on-off light duration (blinking duty cycle), use the parameters LedMwiOnDurationTime and LedMwiOffDurationTime. (The parameters NeonMwiOnDurationTime and NeonMwiOffDurationTime are not applicable.) .
	Note:

Parameter	Description
	▪ For the parameter to take effect, a device reset is required. ▪ The parameter is applicable only to FXS interfaces. ▪ The feature can be configured per port using the ini file parameter syntax: EnableLowVoltageMwiGeneration_x, where *x* is the port number.
[LedMwiOnDurationTime]	Defines the duration (in msec) that the visual message waiting indicator (lamp) on the phone is lit when using the on-hook low-voltage transitions mode (i.e., EnableLowVoltageMwiGeneration is configured to 1). If you want the MWI light indication to be lit constantly (instead of blinking), configure this parameter and the LedMwiOffDurationTime parameter to 1234. If you want the MWI light indication to blink, configure this parameter to any desired on-duration and the LedMwiOffDurationTime parameter to any off-duration. The valid value is 30 to 2010. The default is 1020. **Note:** ▪ For the parameter to take effect, a device reset is required. ▪ The parameter is applicable only to FXS interfaces. ▪ The parameter applies to all FXS ports.
[LedMwiOffDurationTime]	Defines the duration (in msec) that the visual message waiting indicator (lamp) on the phone is off when using the on-hook low-voltage transitions mode (i.e., EnableLowVoltageMwiGeneration is configured to 1). If you want the MWI light indication to be lit constantly (instead of blinking), configure this parameter and the LedMwiOnDurationTime parameter to 1234. If you want the MWI light indication to blink, configure this parameter to any desired off-duration and the LedMwiOnDurationTime parameter to any on-duration. The valid value is 30 to 2010. The default is 60. **Note:** ▪ For the parameter to take effect, a device reset is required. ▪ The parameter is applicable only to FXS interfaces. ▪ The parameter applies to all FXS ports.
[NeonMwiOnDurationTime]	Defines the duration (in msec) that the visual message waiting indicator (lamp) on the phone is lit when using the high-voltage mode (i.e., EnableLowVoltageMwiGeneration is configured to 0). If you want the MWI light indication to be lit constantly (instead of blinking), configure this parameter and the NeonMwiOffDurationTime parameter to 1234 (default). If you want the MWI light indication to blink, configure this parameter to any desired on-duration and the NeonMwiOffDurationTime parameter to any off-duration. The valid value is 30 to 2010. The default is 1234. **Note:** ▪ For the parameter to take effect, a device reset is required. ▪ The parameter is applicable only to FXS interfaces.

Parameter	Description
	▪ The parameter applies to all FXS ports.
[NeonMwiOffDurationTime]	Defines the duration (in msec) that the visual message waiting indicator (lamp) on the phone is off when using the high-voltage mode (i.e., EnableLowVoltageMwiGeneration is configured to 0).
	If you want the MWI light indication to be lit constantly (instead of blinking), configure this parameter and the NeonMwiOnDurationTime parameter to 1234 (default).
	If you want the MWI light indication to blink, configure this parameter to any desired off-duration and the NeonMwiOnDurationTime parameter to any on-duration.
	The valid value is 30 to 2010. The default is 1234.
	Note:
	▪ For the parameter to take effect, a device reset is required.
	▪ The parameter is applicable only to FXS interfaces.
	▪ The parameter applies to all FXS ports.

43.11.5.5   Call Hold Parameters

The call hold parameters are described in the table below.

Table 43-45: Call Hold Parameters

Parameter	Description
Web/OVOC: Enable Hold **[EnableHold]**	Enables the Call Hold feature that allows users, connected to the device, to place a call on hold (or remove from hold). This is done using the phone's Hook Flash button. On receiving a hold request, the remote party is placed on hold and hears the hold tone. ▪ **[0]** Disable ▪ **[1]** Enable (default) **Notes:** ▪ To use this service, the devices at both ends must support this option. ▪ This parameter can also be configured in an IP Profile.
Web/OVOC: Hold Format **[HoldFormat]**	Determines the format of the SDP in the Re-INVITE hold request. ▪ **[0]** 0.0.0.0 = (Default) The SDP "c=" field contains the IP address "0.0.0.0" and the "a=inactive" attribute. ▪ **[1]** Send Only = The SDP "c=" field contains the device's IP address and the "a=sendonly" attribute. ▪ **[2]** x.y.z.t = The SDP "c=" field contains the device's IP address and the "a=inactive" attribute. **Note:** The device does not send any RTP packets when it is in hold state.
Web/OVOC:Held Timeout **[HeldTimeout]**	Defines the time interval that the device allows for a call to remain on hold. If a Resume (un-hold Re-INVITE) message is received before the timer expires, the call is renewed. If this timer expires, the call is released (terminated). ▪ **[-1]** = (Default) The call is placed on hold indefinitely until the initiator of the on hold retrieves the call again.

Parameter	Description
	▪ **[0 - 2400]** = Time to wait (in seconds) after which the call is released.
Web: Call Hold Reminder Ring Timeout OVOC: CHRR Timeout **[CHRRTimeout]**	Defines the duration (in seconds) that the Call Hold Reminder Ring is played. If a user hangs up while a call is still on hold or there is a call waiting, then the FXS interface immediately rings the extension for the duration specified by this parameter. If the user off-hooks the phone, the call becomes active. The valid range is 0 to 600. The default is 30. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ This Reminder Ring feature can be disabled using the DisableReminderRing parameter.
[DisableReminderRing]	Disables the reminder ring, which notifies the FXS user of a call on hold or a waiting call when the phone is returned to on-hook position. ▪ **[0]** = (Default) The reminder ring feature is active. In other words, if a call is on hold or there is a call waiting and the phone is changed from offhook to onhook, the phone rings (for a duration defined by the CHRRTimeout parameter) to "remind" you of the call hold or call waiting. ▪ **[1]** = Disables the reminder ring. If a call is on hold or there is a call waiting and the phone is changed from offhook to onhook, the call is released (and the device sends a SIP BYE to the IP). **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ This parameter is typically used for MLPP, allowing preemption to clear held calls.
[PlayDTMFduringHold]	Determines whether the device sends DTMF signals (or DTMF SIP INFO message) when a call is on hold. ▪ **[0]** = (Default) Disable. ▪ **[1]** = Enable - If the call is on hold, the device stops playing the Held tone (if it is played) and sends DTMF: ✓ To Tel side: plays DTMF digits according to the received SIP INFO message(s). (The stopped held tone is not played again.) ✓ To IP side: sends DTMF SIP INFO messages to an IP destination if it detects DTMF digits from the Tel side.

43.11.5.6   Call Transfer Parameters

The call transfer parameters are described in the table below.

Table 43-46: Call Transfer Parameters

Parameter	Description
Web/OVOC: Enable Transfer **[EnableTransfer]**	Enables the Call Transfer feature. ▪ **[0]** Disable ▪ **[1]** Enable = (Default) Enable the call transfer service (using REFER). If the transfer service is enabled, the

Parameter	Description
	user can activate Transfer using hook-flash signaling. If this service is enabled, the remote party performs the call transfer. **Notes:** ▪ To use call transfer, the devices at both ends must support this option. ▪ To use call transfer, set the parameter EnableHold to 1.
Web: Transfer Prefix OVOC: Logical Prefix For Transferred Call **[xferPrefix]**	Defines the string that is added as a prefix to the transferred/forwarded called number when the REFER/3xx message is received. **Notes:** ▪ The number manipulation rules apply to the user part of the Refer-To and/or Contact URI before it is sent in the INVITE message. ▪ This parameter can be used to apply different manipulation rules to differentiate transferred/forwarded number from the originally dialed number.
Web: Transfer Prefix IP 2 Tel **[XferPrefixIP2Tel]**	Defines the prefix that is added to the destination number received in the SIP Refer-To header (for IP-to-Tel calls). This parameter is applicable to FXO blind transfer modes, i.e., LineTransferMode = 1, 2 or 3,. The valid range is a string of up to 9 characters. The default is an empty string.
Web/OVOC: Enable Semi-Attended Transfer **[EnableSemiAttendedTransfer]**	Determines the device behavior when Transfer is initiated while in Alerting state. ▪ **[0]** Disable = (Default) Send REFER with the Replaces header. ▪ **[1]** Enable = Send CANCEL, and after a 487 response is received, send REFER without the Replaces header.
Web: Blind OVOC: Blind Transfer **[KeyBlindTransfer]**	Defines the keypad sequence to activate blind transfer for established Tel-to-IP calls. The Tel user can perform blind transfer by dialing the KeyBlindTransfer digits, followed by a transferee destination number. After the KeyBlindTransfer DTMF digits sequence is dialed, the current call is put on hold (using a Re-INVITE message), a dial tone is played to the channel, and then the phone number collection starts. After the destination phone number is collected, it is sent to the transferee in a SIP REFER request in a Refer-To header. The call is then terminated and a confirmation tone is played to the channel. If the phone number collection fails due to a mismatch, a reorder tone is played to the channel. **Note:** For FXS/FXO interfaces, it is possible to configure whether the KeyBlindTransfer code is added as a prefix to the dialed destination number, by using the parameter KeyBlindTransferAddPrefix.
[BlindTransferOnOnHook]	Enables blind transfer by pressing the phone's hook-flash button, dialing the transferee's number without a "#" at the end, and then on-hooking the phone within a user-defined interval (configured by TimeBetweenDigits) or before a match is found by DigitMapping.

Parameter	Description
	▪ **[0]** Disable = (Default) Blind transfer is done according to the KeyBlindTransfer parameter. ▪ **[1]** Enable
OVOC: Blind Transfer Add Prefix **[KeyBlindTransferAddPrefix]**	Determines whether the device adds the Blind Transfer code (defined by the KeyBlindTransfer parameter) to the dialed destination number. ▪ **[0]** Disable (default) ▪ **[1]** Enable
OVOC: Blind Transfer Disconnect Timeout **[BlindTransferDisconnectTimeout]**	Defines the duration (in milliseconds) for which the device waits for a disconnection from the Tel side after the Blind Transfer Code (KeyBlindTransfer) has been identified. When this timer expires, a SIP REFER message is sent toward the IP side. If this parameter is set to 0, the REFER message is immediately sent. The valid value range is 0 to 1,000,000. The default is 0.

43.11.5.7   Three-Way Conferencing Parameters

The three-way conferencing parameters are described in the table below.

Table 43-47: Three-Way Conferencing Parameters

Parameter	Description
Web: Enable 3-Way Conference OVOC: Enable 3 Way **[Enable3WayConference]**	Enables the 3-Way Conference feature. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Three Way Conference Mode OVOC: 3 Way Mode **[3WayConferenceMode]**	Determines the mode of operation when the 3-Way Conference feature is used. ▪ **[0]** AudioCodes Media Server = (Default) The Conference-initiating INVITE (sent by the device) uses the ConferenceID concatenated with a unique identifier as the Request-URI. This same Request-URI is set as the Refer-To header value in the REFER messages that are sent to the two remote parties. This conference mode is used when operating with AudioCodes IPMedia conferencing server. ▪ **[1]** Non-AudioCodes Media Server = The Conference-initiating INVITE (sent by the device) uses only the ConferenceID as the Request-URI. The conference server sets the Contact header of the 200 OK response to the actual unique identifier (Conference URI) to be used by the participants. This Conference URI is then included (by the device) in the Refer-To header value in the REFER messages sent by the device to the remote parties. The remote parties join the conference by sending INVITE messages to the conference using this conference URI. ▪ **[2]** On Board = On-board, three-way conference. The conference is established on the device without the need

Parameter	Description

	of an external Conference server. You can limit the number of simultaneous, on-board 3-way conference calls, by using the MaxInBoardConferenceCalls parameter. The device utilizes resources from idle ports to establish the conference call. You can designate ports that you do not want to use as resources for on-board, conference calls initiated by other ports. This is configured using the 3WayConfNoneAllocateablePorts parameter.
	▪ **[3]** Huawei Media Server = The conference is managed by an external, third-party Conferencing server. The conference-initiating INVITE sent by the device, uses only the ConferenceID as the Request-URI. The Conferencing server sets the Contact header of the 200 OK response to the actual unique identifier (Conference URI) to be used by the participants. The Conference URI is included in the URI of the REFER with a Replaces header sent by the device to the Conferencing server. The Conferencing server then sends an INVITE with a Replaces header to the remote participants.
	Notes:
	▪ This parameter is applicable only to FXS interfaces.
	▪ When using an external conference server (options **[0]** or **[1])**, a conference call with up to six participants can be established.
	▪ For local, on-board three-way conferencing (option **[2])** on MP-112, the following additional parameter settings must be made:
	`EnableIPMediaChannels = 1`
	`[ IPMediaChannels ]`
	`FORMAT IPMediaChannels_Index =`
	`IPMediaChannels_ModuleID,`
	`IPMediaChannels_DSPChannelsReserved;`
	`IPMediaChannels 0 = 1, 2;`
	`[ \IPMediaChannels ]`

Web: Max 3 Way Conference OVOC: Max In Board Calls **[MaxInBoardConferenceCalls]**	Defines the maximum number of simultaneous, on-board three-way conference calls. The valid range is 0 to 2. The default is 2. **Notes:** ▪ For enabling on-board, three-way conferencing, use the 3WayConferenceMode parameter. ▪ For the maximum number of supported on-board, three-way conference calls, see the note in Section 24.9, Three-Way Conferencing. ▪ This parameter is applicable only to FXS interfaces.
Web: Three Way Conference Non Allocatable Ports OVOC: Non Allocateable Port Number **[3WayConfNoneAllocateablePorts]**	Defines the ports that are not allocated as resources for on-board three-way conference calls that are initiated by other ports. Ports that are not configured with this parameter (and that are idle) are used by the device as a resource for establishing these type of conference calls. The valid range is up to 8 ports. To add a range of ports, use the comma separator. For example, for not allowing the use of ports 2, 4 and 8 as resources, enter the following value: 2,4,8. The order of the entered values is not relevant (i.e., the example above can be entered as 8,2,4). The default is 0.

Parameter	Description
	Notes: ▪ To enable on-board, three-way conferencing, use the 3WayConferenceMode and MaxInBoardConferenceCalls parameters. ▪ This parameter is applicable only to FXS interfaces.
Web: Establish Conference Code OVOC: Establish Code **[ConferenceCode]**	Defines the DTMF digit pattern, which upon detection generates the conference call when three-way conferencing is enabled (Enable3WayConference is set to 1). The valid range is a 25-character string. The default is "!" (Hook-Flash). **Note:** If the FlashKeysSequenceStyle parameter is set to 1 or 2, the setting of the ConferenceCode parameter is overridden.
Web/OVOC: Conference ID **[ConferenceID]**	Defines the Conference Identification string. The valid value is a string of up to 16 characters. The default is "conf". The device uses this identifier in the conference-initiating INVITE that is sent to the media server when the Enable3WayConference parameter is set to 1.
Web: Use Different RTP port After Hold **[UseDifferentRTPportAfterHold]**	Enables the use of different RTP ports for the two calls involved in a three-way conference call made by the FXS endpoint in the initial outgoing INVITE requests. ▪ **[0]** Disable = First and second calls use the same RTP port in the initial outgoing INVITE request. If a three-way conference is then made, the device sends a re-INVITE to the held call to retrieve it and to change the RTP port to a different port number. For example: The first call is made on port 6000 and placed on hold. The second call is made, also on port 6000. The device sends a re-INVITE to the held call to retrieve it and changes the port to 6010. ▪ **[1]** Enable = First and second calls use different RTP ports in the initial outgoing INVITE request. If a three-way conference is then made, the device sends a re-INVITE to the held call to retrieve it, without changing the port of the held call. **Notes:** ▪ When this feature is enabled and only one RTP port is available, only one call can be made by the FXS endpoint, as there is no free RTP port for a second call. ▪ When this feature is enabled and you are using the Call Forking feature, every forked call is sent with a different RTP port. As the device can fork a call to up to 10 destinations, the device requires at least 10 free RTP ports. ▪ This parameter is applicable only to FXS interfaces.

43.11.5.8   MLPP and Emergency Call Parameters

The Multilevel Precedence and Preemption (MLPP) and emergency E911 call parameters are described in the table below.

Table 43-48: MLPP and Emergency E911 Call Parameters

Parameter	Description

Web/OVOC: Call Priority Mode **[CallPriorityMode]**	Enables priority call handling for all calls.
	▪ **[0]** Disable (default).
	▪ **[1]** MLPP = MLPP Priority Call handling is enabled. MLPP prioritizes call handling whereby the relative importance of various kinds of communications is strictly defined, allowing higher precedence communication at the expense of lower precedence communications. Higher priority calls override less priority calls when, for example, congestion occurs in a network.
	▪ **[2]** Emergency = Preemption of IP-to-Tel E911 emergency calls. If the device receives an E911 call and there are unavailable channels to receive the call, the device terminates one of the channel calls and sends the E911 call to that channel. The preemption is done only on a channel pertaining to the same Hunt Group for which the E911 call was initially destined and if the channel select mode (configured by the ChannelSelectMode parameter) is set to other than "By Dest Number" (0). The preemption is done only if the incoming IP-to-Tel call is identified as an emergency call. The device identifies emergency calls by one of the following:
	  ✓ The destination number of the IP call matches one of the numbers defined by the EmergencyNumbers parameter. (For E911, you must define this parameter with the value "911".)
	  ✓ The incoming SIP INVITE message contains the "emergency" value in the Priority header.
	Notes:
	▪ This parameter is applicable to FXS/FXO.
	▪ For FXO interfaces, the preemption is done only on existing IP-to-Tel calls. In other words, if all the current FXO channels are busy with calls that were initiated by the FXO (i.e., Tel-to-IP calls), new incoming emergency IP-to-Tel calls are dropped.
	▪ MLPP and Emergency services can also be configured in a Tel Profile.
	▪ For more information, see 'Pre-empting Existing Call for E911 IP-to-Tel Call' on page 289.

Parameter	Description
Emergency E911 Parameters	
Web/OVOC: Emergency Numbers **[EmergencyNumbers]**	Defines a list of "emergency" numbers. For FXS: When one of these numbers is dialed, the outgoing INVITE message includes the SIP Priority and Resource-Priority headers. If the user places the phone on-hook, the call is not disconnected. Instead, a Hold Re-INVITE request is sent to the remote party. Only if the remote party disconnects the call (i.e., a BYE is received) or a timer expires (set by the EmergencyRegretTimeout parameter) is the call terminated. For FXO: These emergency numbers are used for the preemption of E911 IP-to-Tel calls when there are unavailable or busy channels. In this scenario, the device terminates one of the busy channels and sends the emergency call to this channel. This feature is enabled by setting the CallPriorityMode parameter to 2 ("Emergency"). For a description of this feature, see 'Pre-empting Existing Call for E911 IP-to-Tel Call' on page 289. The list can include up to four different numbers, where each number can be up to four digits long. Example: EmergencyNumbers = '100','911','112'
Web: Emergency Calls Regret Timeout OVOC: Emergency Regret Timeout **[EmergencyRegretTimeout]**	Defines the time (in minutes) that the device waits before tearing-down an emergency call (defined by the parameter EmergencyNumbers). Until this time expires, an emergency call can only be disconnected by the remote party, typically, by a Public Safety Answering Point (PSAP). The valid range is 1 to 30. The default is 10. **Note:** This parameter is applicable only to FXS interfaces.
Multilevel Precedence and Preemption (MLPP) Parameters	
Web: MLPP DiffServ OVOC: Diff Serv **[MLPPDiffserv]**	Defines the DiffServ value (differentiated services code point/DSCP) used in IP packets containing SIP messages that are related to MLPP calls. This parameter defines DiffServ for incoming MLPP calls with the Resource-Priority header. The valid range is 0 to 63. The default is 50. **Notes:** ▪ The same value must be configured for this parameter and the parameter PremiumServiceClassControlDiffServ. ▪ Outgoing calls are tagged according to the parameter PremiumServiceClassControlDiffServ.
Web/OVOC: Precedence Ringing Type **[PrecedenceRingingType]**	Defines the index of the Precedence Ringing tone in the Call Progress Tones (CPT) file. This tone is used when the parameter CallPriorityMode is set to 1 and a Precedence call is received from the IP side. The valid range is -1 to 16. The default is -1 (i.e., plays standard ringing tone).
OVOC: E911 MLPP Behavior **[E911MLPPBehavior]**	Defines the E911 (or Emergency Telecommunication Services/ETS) MLPP Preemption mode: ▪ **[0]** = (Default) Standard Mode - ETS calls have the highest priority and preempt any MLPP call.

Parameter	Description
	• **[1]** = Treat as routine mode - ETS calls are handled as routine calls.
[RPRequired]	Determines whether the SIP resource-priority tag is added in the SIP Require header of the INVITE message for Tel-to-IP calls. • **[0]** Disable = Excludes the SIP resource-priority tag from the SIP Require header. • **[1]** Enable = (Default) Adds the SIP resource-priority tag in the SIP Require header. **Note:** This parameter is applicable only to MLPP priority call handling (i.e., only when the CallPriorityMode parameter is set to 1).

Multiple Differentiated Services Code Points (DSCP) per MLPP Call Priority Level (Precedence) Parameters

The MLPP service allows placement of priority calls, where properly validated users can preempt (terminate) lower-priority phone calls with higher-priority calls. For each MLPP call priority level, the DSCP can be set to a value from 0 to 63. The Resource Priority value in the Resource-Priority SIP header can be one of the following:

MLPP Precedence Level	Precedence Level in Resource-Priority SIP Header
0 (lowest)	routine
2	priority
4	immediate
6	flash
8	flash-override
9 (highest)	flash-override-override

Web/OVOC: RTP DSCP for MLPP Routine **[MLPPRoutineRTPDSCP]**	Defines the RTP DSCP for MLPP Routine precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call.
Web/OVOC: RTP DSCP for MLPP Priority **[MLPPPriorityRTPDSCP]**	Defines the RTP DSCP for MLPP Priority precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call.
Web/OVOC: RTP DSCP for MLPP Immediate **[MLPPImmediateRTPDSCP]**	Defines the RTP DSCP for MLPP Immediate precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call.
Web/OVOC: RTP DSCP for MLPP Flash **[MLPPFlashRTPDSCP]**	Defines the RTP DSCP for MLPP Flash precedence call level. The valid range is -1 to 63. The default is -1.

Parameter	Description
	Note: If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call.
Web/OVOC: RTP DSCP for MLPP Flash Override **[MLPPFlashOverRTPDSCP]**	Defines the RTP DSCP for MLPP Flash-Override precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call.
Web/OVOC: RTP DSCP for MLPP Flash-Override-Override **[MLPPFlashOverOverRTPDSCP]**	Defines the RTP DSCP for MLPP Flash-Override-Override precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call.

43.11.5.9   Call Cut-Through Parameters

The call cut-through parameters are described in the table below.

Table 43-49: Call Cut-Through Parameters

Parameter	Description
Web: Enable Calls Cut Through OVOC: Cut Through **[CutThrough]**	Enables FXS endpoints to receive incoming IP calls while the port is in off-hook state. **[0]** Disable (default) **[1]** Enable If enabled, the FXS interface answers the call and 'cuts through' the voice channel if there is no other active call on the port, even if the port is in off-hook state. When the call is terminated (by the remote IP party), the device plays a reorder tone for a user-defined time (configured by the CutThroughTimeForReorderTone parameter) and is then ready to answer the next incoming call without on-hooking the phone. The waiting call is automatically answered by the device when the current call is terminated (configured by setting the parameter EnableCallWaiting to 1). **Note:** - This feature is applicable only to FXS interfaces. - You can also configure the feature using an IP Profile (TelProfile_IP2TelCutThroughCallBehavior): [0] NO cut through, no paging = Disabled [1] cutThrough = Channel Cut-Through enabled. When the IP side ends the call, the device can play a reorder tone to the Tel side for a user-defined duration (configured by the CutThroughTimeForReorderTone parameter). Once the tone stops playing, the FXS phone is ready to automatically answer another incoming IP call, while in off-hook state. [2] cutThrough + paging = Channel Cut-Through enabled and no tones are played.

43.11.5.10   Automatic Dialing Parameters

The automatic dialing upon off-hook parameters are described in the table below.

Table 43-50: Automatic Dialing Parameters

Parameter	Description
Automatic Dialing Table	
Web: Automatic Dialing Table OVOC: Analog Gateway Provisioning > Automatic dialing **[TargetOfChannel]**	This table parameter defines telephone numbers that are automatically dialed when a specific FXS or FXO port is off-hooked. The format of this parameter is as follows: [TargetOfChannel] FORMAT TargetOfChannel_Index = TargetOfChannel_Destination, TargetOfChannel_Type, TargetOfChannel_HotLineToneDuration; [\TargetOfChannel] For example, the below configuration defines automatic dialing of phone number 911 when the phone connected to Port 1 is off-hooked for over 10 seconds: TargetOfChannel 0 = 911, 1 ,10; **Notes:** ▪ The first index of this table ini file parameter is 0. ▪ TargetOfChannel_Index is the port number, where 0 denotes Port 1. ▪ This is parameter is applicable only to FXS and FXO interfaces. ▪ For a detailed description of this table, see 'Configuring Automatic Dialing' on page 303.

43.11.5.11   Direct Inward Dialing Parameters

The Direct Inward Dialing (DID) parameters are described in the table below.

Table 43-51: DID Parameters

Parameter	Description

Web/OVOC: DID Wink
[EnableDIDWink]
	Enables Direct Inward Dialing (DID) using Wink-Start signaling, typically used for signaling between an E-911 switch and the PSAP.
	▪ **[0]** Disable (default)
	▪ **[1]** Single = The device can be used for connection to EIA/TIA-464B DID Loop Start lines. Both FXO (detection) and FXS (generation) are supported:
	  ✔ The FXO interface dials DTMF (or MF) digits upon detection of a Wink signal, instead of a dial tone.
	  ✔ The FXS interface generates a Wink signal upon detection of an off-hook state, instead of playing a dial tone.
	Example: (Wink) KP I(I) xxx-xxxx ST (Off Hook)
	Where:
	  ✔ I = one or two information digits
	  ✔ x = ANI
	Note: The FXO interface generates such MF digits when the Enable911PSAP parameter is set to 1.
	▪ **[2]** Double Wink = Double-wink signaling. The FXS interface generates the first wink upon detection of an off-hook state in the line. The second wink is generated after a user-defined interval (configured by the TimeBetweenDIDWinks parameter), after which the DTMF/MF digits are collected by the device. Digits that arrive

Parameter	Description

	between the first and second wink are ignored as they contain the same number.
	
	  Example: (Wink) KP 911 ST (Wink) KP I(I) xxx-xxxx ST (Off Hook)
	
	 ▪ **[3]** Wink & Polarity= The FXS interface generates the first wink after it detects an off-hook state. A polarity change from normal to reversed is generated after a user-defined time (configured by the TimeBetweenDIDWinks parameter). DTMF/MF digits are collected only after this polarity change. Digits that arrive between the first wink and the polarity change are ignored as they always contain the same number. In this mode, the FXS interface does not generate a polarity change to normal if the Tel-to-IP call is answered by an IP party. Polarity reverts to normal when the call is released.
	
	  Example: (Wink) KP 911 ST (Polarity) KP I(I) xxx-xxxx ST (Off Hook)
	
	 Notes:
	 ▪ Options [2] and [3] are applicable only to FXS interfaces.
	 ▪ The EnableReversalPolarity and PolarityReversalType parameters must be set to [1] for FXS interfaces.
	 ▪ See also the Enable911PSAP parameter.
	 ▪ This parameter can also be configured in a Tel Profile.

[TimeBetweenDIDWinks]	Defines the interval (in msec) for wink signaling: ▪ Double-wink signaling [2]: interval between the first and second wink. ▪ Wink and Polarity signaling [3]: interval between wink and polarity change. The valid range is 100 to 2000. The default is 1000. **Note:** See the EnableDIDWink parameter for configuring the wink signaling type.
Web/OVOC: Delay Before DID Wink **[DelayBeforeDIDWink]**	Defines the time interval (in msec) between the detection of the off-hook and the generation of the DID Wink. The valid range is 0 to 1,000. The default is 0. **Note:** This parameter is applicable only to FXS interfaces.
OVOC: NTT DID Signalling Form **[NTTDIDSignallingForm]**	Determines the type of DID signaling support for NTT (Japan) modem: DTMF- or Frequency Shift Keying (FSK)-based signaling. The devices can be connected to Japan's NTT PBX using 'Modem' DID lines. These DID lines are used to deliver a called number to the PBX. ▪ **[0]** = (Default) FSK-based signaling ▪ **[1]** = DTMF-based signaling **Note:** This parameter is applicable only to FXS interfaces.
OVOC: Enable DID **[EnableDID]**	This table parameter enables support for Japan NTT 'Modem' DID. FXS interfaces can be connected to Japan's NTT PBX using 'Modem' DID lines. These DID lines are used to deliver a called number to the PBX. The DID signal can be sent alone or combined with an NTT Caller ID signal. The format of this parameter is as follows: **[EnableDID]** FORMAT EnableDID_**Index** = EnableDID_**IsEnable**; **[\EnableDID]** Where,

Parameter	Description
	• Index = Port number (where 0 denotes Port 1). • IsEnable = Enables **[1]** or disables **[0]** (default) Japan NTT Modem DID support. For example: EnableDID 0 = 1; (DID is enabled on Port 1) **Note:** This parameter is applicable only to FXS interfaces.
[WinkTime]	Defines the time (in msec) elapsed between two consecutive polarity reversals. This parameter can be used for DID signaling. The valid range is 0 to 4,294,967,295. The default is 200. **Notes:** • This parameter is applicable to FXS and FXO interfaces. • For this parameter to take effect, a device reset is required.

43.11.6 Answer and Disconnect Supervision Parameters

The answer and disconnect supervision parameters are described in the table below.

Table 43-52: Answer and Disconnect Parameters

Parameter	Description
Web: Answer Supervision OVOC: Enable Voice Detection **[EnableVoiceDetection]**	Enables the sending of SIP 200 OK upon detection of speech, fax, or modem. • **[1]** Yes = The device sends a SIP 200 OK (in response to an INVITE message) when speech, fax, or modem is detected. • **[0]** No = (Default) The device sends a SIP 200 OK only after it completes dialing. Typically, this feature is used only when early media (enabled using the EnableEarlyMedia parameter) is used to establish the voice path before the call is answered. **Note:** This feature is applicable only to one-stage dialing (FXO).
Web/OVOC: Max Call Duration (min) **[MaxCallDuration]**	Defines the maximum duration (in minutes) of a call. If this duration is reached, the device terminates the call. This feature is useful for ensuring available resources for new calls, by ensuring calls are properly terminated. The valid range is 0 to 35,791. The default is 0 (i.e., no limitation).
Web/OVOC: Disconnect on Dial Tone **[DisconnectOnDialTone]**	Determines whether the device disconnects a call when a dial tone is detected from the PBX. • **[0]** Disable = (Default) Call is not released. • **[1]** Enable = Call is released if a dial tone is detected on the device's FXO port. **Notes:** • This parameter is applicable only to FXO interfaces. • This option is in addition to the mechanism that disconnects a call when either busy or reorder tones are detected.

Parameter	Description
Web: Send Digit Pattern on Connect OVOC: Connect Code **[TelConnectCode]**	Defines a digit pattern to send to the Tel side after a SIP 200 OK is received from the IP side. The digit pattern is a user-defined DTMF sequence that is used to indicate an answer signal (e.g., for billing). The valid range is 1 to 8 characters. **Note:** This parameter is applicable to FXO.
Web: Disconnect on Broken Connection OVOC: Disconnect Calls on Broken Connection **[DisconnectOnBrokenConnection]**	Determines whether the device releases the call if RTP packets are not received within a user-defined timeout. ▪ **[0]** No ▪ **[1]** Yes (default) **Notes:** ▪ The timeout is configured by the BrokenConnectionEventTimeout parameter. ▪ This feature is applicable only if the RTP session is used without Silence Compression. If Silence Compression is enabled, the device doesn't detect a broken RTP connection. ▪ During a call, if the source IP address (from where the RTP packets are received) is changed without notifying the device, the device filters these RTP packets. To overcome this, set the DisconnectOnBrokenConnection parameter to 0; the device doesn't detect RTP packets arriving from the original source IP address and switches (after 300 msec) to the RTP packets arriving from the new source IP address. ▪ This parameter can also be configured in an IP Profile.
Web: Broken Connection Timeout OVOC: Broken Connection Event Timeout **[BrokenConnectionEventTimeout]**	Defines the time period (in 100-msec units) after which a call is disconnected if an RTP packet is not received. The valid range is from 3 (i.e., 300 msec) to an unlimited value (e.g., 20 hours). The default is 100 (i.e., 10000 msec or 10 seconds). **Notes:** ▪ This parameter is applicable only if the parameter DisconnectOnBrokenConnection is set to 1. ▪ Currently, this feature functions only if Silence Suppression is disabled.
Web: Disconnect Call on Silence Detection OVOC: Disconnect On Detection Of Silence **[EnableSilenceDisconnect]**	Determines whether calls are disconnected after detection of silence. ▪ **[1]** Yes = The device disconnects calls in which silence occurs (in both call directions) for more than a user-defined time. ▪ **[0]** No = (Default) Call is not disconnected when silence is detected. The silence duration can be configured by the FarEndDisconnectSilencePeriod parameter (default 120). **Note:** ▪ The parameter is not supported by MP-124 Rev. E.

Parameter	Description
	▪ To activate this feature, enable Silence Suppression for the used coder and configure the FarEndDisconnectSilenceMethod parameter to 1.
Web: Silence Detection Period **[sec]** OVOC: Silence Detection Time Out **[FarEndDisconnectSilencePeriod]**	Defines the duration of the silence period (in seconds) after which the call is disconnected. The range is 10 to 28,800 (i.e., 8 hours). The default is 120 seconds. **Note:** ▪ For this parameter to take effect, a device reset is required. ▪ The parameter is not supported by MP-124 Rev. E.
Web: Silence Detection Method **[FarEndDisconnectSilenceMethod]**	Determines the silence detection method. ▪ **[0]** None = Silence detection option is disabled. ▪ **[1]** Packets Count = According to packet count. ▪ **[2]** Voice/Energy Detectors = (Default) According to energy and voice detectors. ▪ **[3]** All = According to packet count, and energy and voice detectors. **Note:** ▪ For this parameter to take effect, a device reset is required. ▪ The parameter is not supported by MP-124 Rev. E.
[FarEndDisconnectSilenceThreshold]	Defines the threshold of the packet count (in percentages) below which is considered silence by the device. The valid range is 1 to 100%. The default is 8%. **Notes:** ▪ This parameter is applicable only if silence is detected according to packet count (FarEndDisconnectSilenceMethod is set to 1). ▪ For this parameter to take effect, a device reset is required. ▪ The parameter is not supported by MP-124 Rev. E.
[BrokenConnectionDuringSilence]	Enables the generation of the BrokenConnection event during a silence period if the channel's NoOp feature is enabled (using the parameter NoOpEnable) and if the channel stops receiving NoOp RTP packets. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: Disconnect Call on Busy Tone Detection OVOC: Disconnect On Detection End Tones **[DisconnectOnBusyTone]**	Determines whether a call is disconnected upon detection of a busy tone. ▪ **[0]** Disable = Call is not disconnected upon detection of a busy tone. ▪ **[1]** Enable = (Default) Call is released upon detection of busy or reorder (fast busy) tone. **Notes:** ▪ This parameter is applicable only to FXO interfaces. ▪ This parameter can also be configured in a Tel Profile.

Parameter	Description
Polarity (Current) Reversal for Call Release (Analog Interfaces) Parameters	
[SetDefaultLinePolarityState]	Defines the FXO line polarity, required for DID signaling. ▪ **[0]** = Positive line polarity ▪ **[1]** = Negative line polarity ▪ **[2]** = (Default) Auto - The device detects the polarity upon power-up or upon insertion of the RJ-11 cable, and uses it as a reference polarity. Typically, if the RJ-11 cabling is connected correctly (without crossing, Tip to Tip, Ring to Ring), the Tip line is positive compared to the Ring line. In this case, set this parameter to 0. With this configuration, the device assumes that the idle line polarity is Tip line positive. When the device receives a SIP INVITE, it checks the FXO line polarity. If the polarity is "Reversed", it skips this FXO line and goes to the next line. **Notes:** ▪ A device reset is required for the settings of the parameter to take effect. ▪ To take advantage of this new feature, configure all FXO lines as a single Trunk Group with ascending or descending channel select mode, and configure routing rules to route incoming INVITE messages to this Trunk Group. ▪ This parameter is applicable only to FXO interfaces.
Web: Enable Polarity Reversal OVOC: Enable Reversal Polarity **[EnableReversalPolarity]**	Enables the polarity reversal feature for call release. ▪ **[0]** Disable = (Default) Disable the polarity reversal service. ▪ **[1]** Enable = Enable the polarity reversal service. If the polarity reversal service is enabled, the FXS interface changes the line polarity on call answer and then changes it back on call release. The FXO interface sends a 200 OK response when polarity reversal signal is detected (applicable only to one-stage dialing) and releases a call when a second polarity reversal signal is detected. **Note:** This parameter can also be configured in a Tel Profile.
Web/OVOC: Enable Current Disconnect **[EnableCurrentDisconnect]**	Enables call release upon detection of a Current Disconnect signal. ▪ **[0]** Disable = (Default) Disable the current disconnect service. ▪ **[1]** Enable = Enable the current disconnect service. If the current disconnect service is enabled: ▪ The FXO releases a call when a current disconnect signal is detected on its port. ▪ The FXS interface generates a 'Current Disconnect Pulse' after a call is released from IP. The current disconnect duration is configured by the CurrentDisconnectDuration parameter. The current disconnect threshold (FXO only) is configured by the

Parameter	Description
	CurrentDisconnectDefaultThreshold parameter. The frequency at which the analog line voltage is sampled is configured by the TimeToSampleAnalogLineVoltage parameter. **Note:** This parameter can also be configured in a Tel Profile.
OVOC: Polarity Reversal Type **[PolarityReversalType]**	Defines the voltage change slope during polarity reversal or wink. ▪ **[0]** = (Default) Soft reverse polarity. ▪ **[1]** = Hard reverse polarity. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ Some Caller ID signals use reversal polarity and/or Wink signals. In these cases, it is recommended to set the parameter PolarityReversalType to 1 (Hard). ▪ For this parameter to take effect, a device reset is required.
OVOC: Current Disconnect Duration **[CurrentDisconnectDuration]**	Defines the duration (in msec) of the current disconnect pulse. The range is 200 to 1500. The default is 900. **Notes:** ▪ This parameter is applicable for FXS and FXO interfaces. ▪ The FXO interface detection window is 100 msec below the parameter's value and 350 msec above the parameter's value. For example, if this parameter is set to 400 msec, then the detection window is 300 to 750 msec. ▪ For this parameter to take effect, a device reset is required.
[CurrentDisconnectDefaultThreshold]	Defines the line voltage threshold at which a current disconnect detection is considered. The valid range is 0 to 20 Volts. The default is 4 Volts. **Notes:** ▪ This parameter is applicable only to FXO interfaces. ▪ For this parameter to take effect, a device reset is required.
[TimeToSampleAnalogLineVoltage]	Defines the frequency at which the analog line voltage is sampled (after offhook), for detection of the current disconnect threshold. The valid range is 100 to 2500 msec. The default is 1000 msec. **Notes:** ▪ This parameter is applicable only to FXO interfaces. ▪ For this parameter to take effect, a device reset is required.

43.11.7 Tone Parameters

This subsection describes the device's tone parameters.

43.11.7.1   Telephony Tone Parameters

The telephony tone parameters are described in the table below.

Table 43-53: Tone Parameters

Parameter	Description
Web: SIP Hold Behavior **[SIPHoldBehavior]**	Enables the device to handle incoming re-INVITE messages with the "a=sendonly" attribute in the SDP, in the same way as if an "a=inactive" is received in the SDP. When enabled, the device plays a held tone to the Tel phone and responds with a SIP 200 OK containing the "a=recvonly" attribute in the SDP. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web/OVOC: Dial Tone Duration **[sec]** **[TimeForDialTone]**	Defines the duration (in seconds) that the dial tone is played. FXS interfaces play the dial tone after the phone is picked up (off-hook). FXO interfaces play the dial tone after the port is seized in response to ringing (from PBX/PSTN). The valid range is 0 to 60. The default time is 16. **Notes:** ▪ During play of dial tone, the device waits for DTMF digits. ▪ This parameter is not applicable when Automatic Dialing is enabled.
Web/OVOC: Stutter Tone Duration **[StutterToneDuration]**	Defines the duration (in msec) of the confirmation tone. A stutter tone is played (instead of a regular dial tone) when a Message Waiting Indication (MWI) is received. The stutter tone is composed of a confirmation tone (Tone Type #8), which is played for the defined duration (StutterToneDuration) followed by a stutter dial tone (Tone Type #15). Both these tones are defined in the CPT file. The range is 1,000 to 60,000. The default is 2,000 (i.e., 2 seconds). **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ If you want to configure the duration of the confirmation tone to longer than 16 seconds, you must increase the value of the parameter TimeForDialTone accordingly. ▪ The MWI tone takes precedence over the call forwarding reminder tone. For more information on MWI, see Message Waiting Indication on page 282.

Parameter	Description
Web: FXO AutoDial Play BusyTone OVOC: Auto Dial Play Busy Tone **[FXOAutoDialPlayBusyTone]**	Determines whether the device plays a busy / reorder tone to the PSTN side if a Tel-to-IP call is rejected by a SIP error response (4xx, 5xx or 6xx). If a SIP error response is received, the device seizes the line (off-hook), and then plays a busy / reorder tone to the PSTN side (for the duration defined by the parameter TimeForReorderTone). After playing the tone, the line is released (on-hook). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Note:** This parameter is applicable only to FXO interfaces.
Web: Hotline Dial Tone Duration OVOC: Hot Line Tone Duration **[HotLineToneDuration]**	Defines the duration (in seconds) of the hotline dial tone. If no digits are received during this duration, the device initiates a call to a user-defined number (configured in the Automatic Dialing table - TargetOfChannel - see Configuring Automatic Dialing on page 303). The valid range is 0 to 60. The default is 16. **Notes:** ▪ This parameter is applicable to FXS and FXO interfaces. ▪ You can define the Hotline duration per FXS/FXO port using the Automatic Dialing table.
Web/OVOC: Reorder Tone Duration **[sec]** **[TimeForReorderTone]**	Defines the duration (in seconds) that the device plays a busy or reorder tone before releasing the line. Typically, after playing the busy or reorder tone for this duration, the device starts playing an offhook warning tone. The valid range is 0 to 254. The default is 0 seconds. Note that the Web interface denotes the default value as a string value of "255". **Notes:** ▪ The selected busy or reorder tone is according to the SIP release cause code received from IP. ▪ This parameter can also be configured in a Tel Profile.
Web: Time Before Reorder Tone **[sec]** OVOC: Time For Reorder Tone **[TimeBeforeReorderTone]**	Defines the delay interval (in seconds) from when the device receives a SIP BYE message (i.e., remote party terminates call) until the device starts playing a reorder tone to the FXS phone. The valid range is 0 to 60. The default is 0. **Note:** This parameter is applicable only to FXS interfaces.
Web: Cut Through Reorder Tone Duration **[sec]** **[CutThroughTimeForReOrderTone]**	Defines the duration (in seconds) of the reorder tone played to the Tel side after the IP call party releases the call, for the Cut-Through feature. After the tone stops playing, an incoming call is immediately answered if the FXS is off-hooked. The valid values are 0 to 30. The default is 0 (i.e., no reorder tone is played). **Note:** To enable the Cut-Through feature, use the CutThrough (for FXS channels) parameter.

Parameter	Description
Web/OVOC: Enable Comfort Tone **[EnableComfortTone]**	Determines whether the device plays a comfort tone (Tone Type #18) to the FXS/FXO endpoint after a SIP INVITE is sent and before a SIP 18x response is received. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** This parameter is applicable to FXS and FXO interfaces.
[WarningToneDuration]	Defines the duration (in seconds) for which the offhook warning tone is played to the user. The valid range is -1 to 2,147,483,647. The default is 600. **Note:** A negative value indicates that the tone is played infinitely.

Web: Play Ringback Tone to Tel OVOC: Play Ring Back Tone To Tel **[PlayRBTone2Tel]**	Determines the playing method of the ringback tone to the Tel (for analog interfaces) side.
	▪ **[0]** Don't Play =
	  ✓ Ringback tone is not played.
	▪ **[1]** Play on Local =
	  ✓ Plays a ringback tone to the Tel side of the call when a SIP 180/183 response is received.
	  ✓ **[2]** Prefer IP = (Default):
	  ✓ Plays a ringback tone to the Tel side only if a 180/183 response without SDP is received. If 180/183 with SDP message is received, the device cuts through the voice channel and doesn't play the ringback tone.
	▪ **[3]** Play Local Until Remote Media Arrive = Plays a ringback tone according to received media. The behaviour is similar to **[2].** If a SIP 180 response is received and the voice channel is already open (due to a previous 183 early media response or due to an SDP in the current 180 response), the device plays a local ringback tone if there are no prior received RTP packets. The device stops playing the local ringback tone as soon as it starts receiving RTP packets. At this stage, if the device receives additional 18x responses, it does not resume playing the local ringback tone.
	Note: This parameter is applicable to the Gateway and IP-to-IP applications.

Web: Play Ringback Tone to IP OVOC: Play Ring Back Tone To IP **[PlayRBTone2IP]**	Determines whether the device plays a ringback tone to the IP side for IP-to-Tel calls. ▪ **[0]** Don't Play = (Default) Ringback tone isn't played. ▪ **[1]** Play = Ringback tone is played after SIP 183 session progress response is sent. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ To enable the device to send a 183/180+SDP responses, set the EnableEarlyMedia parameter to 1. ▪ If the EnableDigitDelivery parameter is set to 1, the device doesn't play a ringback tone to IP and doesn't send 183 or 180+SDP responses.

Parameter	Description
	▪ This parameter can also be configured in an IP Profile.
Tone Index Table	
Web: Tone Index Table OVOC: Analog Gateway Provisioning > Tone Index **[ToneIndex]**	This table parameter configures the Tone Index table, which allows you to define distinctive ringing and call waiting tones per FXS endpoint (or for a range of FXS endpoints). The format of this parameter is as follows: [ToneIndex] FORMAT ToneIndex_Index = ToneIndex_FXSPort_First, ToneIndex_FXSPort_Last, ToneIndex_SourcePrefix, ToneIndex_DestinationPrefix, ToneIndex_PriorityIndex; [\ToneIndex] For example, the configuration below plays the tone Index #3 to FXS ports 1 and 2 if the source number prefix of the received call is 20. ToneIndex 1 = 1, 2, 20*, , 3; **Note:** For a detailed description of this table, see Configuring FXS Distinctive Ringing and Call Waiting Tones per Source/Destination Number.

43.11.7.2   Tone Detection Parameters

The signal tone detection parameters are described in the table below.

Table 43-54: Tone Detection Parameters

Parameter	Description
OVOC: DTMF Enable **[DTMFDetectorEnable]**	Enables the detection of DTMF signaling. ▪ **[0]** = Disable ▪ **[1]** = Enable (default)
OVOC: MF R1 Enable **[MFR1DetectorEnable]**	Enables the detection of MF-R1 signaling. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
OVOC: User Defined Tone Enable **[UserDefinedToneDetectorEnable]**	Enables the detection of User Defined Tones signaling, applicable for Special Information Tone (SIT) detection. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
OVOC: SIT Enable **[SITDetectorEnable]**	Enables SIT detection according to the ITU-T recommendation E.180/Q.35. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable (applicable to FXO interfaces): ▪ SITDetectorEnable = 1 ▪ UserDefinedToneDetectorEnable = 1 ▪ DisconnectOnBusyTone = 1 (applicable for busy, reorder, and SIT tones) **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
OVOC: UDT Detector Frequency Deviation **[UDTDetectorFrequencyDeviation]**	Defines the deviation (in Hz) allowed for the detection of each signal frequency. The valid range is 1 to 50. The default is 50. **Note:** For this parameter to take effect, a device reset is required.
OVOC: CPT Detector Frequency Deviation **[CPTDetectorFrequencyDeviation]**	Defines the deviation (in Hz) allowed for the detection of each CPT signal frequency. The valid range is 1 to 30. The default is 10. **Note:** For this parameter to take effect, a device reset is required.

43.11.7.3   Metering Tone Parameters

The metering tone parameters are described in the table below.

Table 43-55: Metering Tone Parameters

Parameter	Description
Web: Generate Metering Tones OVOC: Metering Mode **[PayPhoneMeteringMode]**	Determines the method used to configure the metering tones that are generated to the Tel side. ▪ **[0]** Disable = (Default) Metering tones aren't generated. ▪ **[1]** Internal Table = Metering tones are generated according to the device's Charge Code table (using the ChargeCode parameter). **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ If you select 'Internal Table', you must configure the Charge Codes table, using the ChargeCode parameter (see Configuring Charge Codes Table on page 300).
Web: Analog Metering Type OVOC: Metering Type **[MeteringType]**	Determines the metering method for generating pulses (sinusoidal metering burst frequency) by the FXS port. ▪ **[0]** 12 KHz = (Default) 12 kHz sinusoidal bursts. ▪ **[1]** 16 KHz = 16 kHz sinusoidal bursts. ▪ **[2]** = Polarity Reversal pulses. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to FXS interfaces.
Web: Analog TTX Voltage Level OVOC: TTX Voltage Level **[AnalogTTXVoltageLevel]**	Determines the metering signal/pulse voltage level (TTX). ▪ **[0]** 0V = 0 Vrms sinusoidal bursts. ▪ **[1]** 0.5V = (Default) 0.5 Vrms sinusoidal bursts. ▪ **[2]** 1V = 1 Vrms sinusoidal bursts **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to FXS interfaces.

Parameter	Description
Charge Codes Table	
Web: Charge Codes Table OVOC: Charge Codes **[ChargeCode]**	This table parameter configures metering tones and their time intervals that the FXS interface generates to the Tel side. The format of this parameter is as follows: [ChargeCode] FORMAT ChargeCode_Index = ChargeCode_EndTime1, ChargeCode_PulseInterval1, ChargeCode_PulsesOnAnswer1, ChargeCode_EndTime2, ChargeCode_PulseInterval2, ChargeCode_PulsesOnAnswer2, ChargeCode_EndTime3, ChargeCode_PulseInterval3, ChargeCode_PulsesOnAnswer3, ChargeCode_EndTime4, ChargeCode_PulseInterval4, ChargeCode_PulsesOnAnswer4; [\ChargeCode] Where, ▪ EndTime = Period (1 - 4) end time. ▪ PulseInterval = Period (1 - 4) pulse interval. ▪ PulsesOnAnswer = Period (1 - 4) pulses on answer. For example: ChargeCode 1 = 7,30,1,14,20,2,20,15,1,0,60,1; ChargeCode 2 = 5,60,1,14,20,1,0,60,1; ChargeCode 3 = 0,60,1; ChargeCode 0 = 6, 3, 1, 12, 2, 1, 18, 5, 2, 0, 2, 1; **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ To associate a configured Charge Code to an outgoing Tel-to-IP call, use the Tel to IP Routing. ▪ To configure the Charge Codes table using the Web interface, see Configuring Charge Codes Table on page 300.

43.11.8 Telephone Keypad Sequence Parameters

The telephony keypad sequence parameters are described in the table below.

Table 43-56: Telephone Keypad Sequence Parameters

Parameter	Description
Web/OVOC: Call Pickup Key **[KeyCallPickup]**	Defines the keying sequence for performing a call pick-up. Call pick-up allows the FXS endpoint to answer another telephone's incoming call by pressing this user-defined sequence of digits. When the user dials these digits (e.g., #77), the incoming call from another phone is forwarded to the user's phone. The valid value is a string of up to 15 characters (0-9, #, and *). The default is undefined. **Notes:** ▪ Call pick-up is configured only for FXS endpoints pertaining to the same Hunt Group.

Parameter	Description
	▪ This parameter is applicable only to FXS interfaces.
Prefix for External Line	
[Prefix2ExtLine]	Defines a string prefix (e.g., '9' dialed for an external line) that when dialed, the device plays a secondary dial tone (i.e., stutter tone) to the FXS line and then starts collecting the subsequently dialed digits from the FXS line.
	The valid value is a string of up to three characters, which can include digits (0-9), #, and * (e.g., *72). The default is an empty string.
	Notes:
	▪ You can enable the device to add this string as the prefix to the collected (and sent) digits, using the parameter AddPrefix2ExtLine.
	▪ This parameter is applicable only to FXS interfaces.
[AddPrefix2ExtLine]	Determines whether the prefix string for accessing an external line (defined by the parameter Prefix2ExtLine) is added to the dialed number as the prefix and together sent to the IP destination (Tel-to-IP calls).
	▪ **[0]** = Disable (default)
	▪ **[1]** = Enable
	For example, if this parameter is enabled and the prefix string for the external line is defined as "9" (using the parameter Prefix2ExtLine) and the FXS user wants to make a call to destination "123", the device collects and sends all the dialed digits, including the prefix string, as "9123" to the IP destination number.
	Note: This parameter is applicable only to FXS interfaces.
Hook Flash Parameters	
Web: Flash Keys Sequence Style **[FlashKeysSequenceStyle]**	Determines the hook-flash key sequence for FXS interfaces.
	▪ **[0]** Flash hook = (Default) Only the phone's flash button is used, according to the following scenarios:
	✓ During an existing call, if the user presses the flash button, the call is put on hold; a dial tone is heard and the user is able to initiate a second call. Once the second call is established, on-hooking transfers the first (held) call to the second call.
	✓ During an existing call, if a call comes in (call waiting), pressing the flash button places the active call on hold and answers the waiting call; pressing flash again toggles between these two calls.
	▪ **[1]** Sequence 1 = Sequence of flash and digit:
	✓ Flash + 1: holds a call or toggles between two existing calls
	✓ Flash + 2: makes a call transfer.
	✓ Flash + 3: makes a three-way conference call (if the Three-Way Conference feature is enabled, i.e., the parameter Enable3WayConference is set to 1 and the parameter 3WayConferenceMode is set to 2).
	▪ **[2]** Sequence 2 = Sequence of flash and digit:
	✓ Flash only: Places a call on hold.

Parameter	Description
	✓ Flash + 1: 1) When the device handles two calls (an active and a held call) and this key sequence is dialed, it sends a SIP BYE message to the active call and the previously held call becomes the active call. 2) When there is an active call and an incoming waiting call, if this key sequence is dialed, the device disconnects the active call and the waiting call becomes an active call. ✓ Flash + 2: Places a call on hold and answers a call-waiting call, or toggles between active and on-hold calls. ✓ Flash + 3: Makes a three-way conference call. This is applicable only if the Enable3WayConference parameter is set to 1 and the 3WayConferenceMode parameter is set to 2. Note that the settings of the ConferenceCode parameter is ignored. ✓ Flash + 4: Makes a call transfer. **Note:** This parameter is applicable only to FXS interfaces.
Web: Flash Keys Sequence Timeout **[FlashKeysSequenceTimeout]**	Defines the Flash keys sequence timeout - the time (in msec) that the device waits for digits after the user presses the Flash button (Flash Hook + Digit mode - when the parameter FlashKeysSequenceStyle is set to 1 or 2). The valid range is 100 to 5,000. The default is 2,000.
Keypad Feature - Call Forward Parameters	
Web: Forward Unconditional OVOC: Call Forward Unconditional **[KeyCFUnCond]**	Defines the keypad sequence to activate the immediate call forward option.
Web: Forward No Answer OVOC: Call Forward No Answer **[KeyCFNoAnswer]**	Defines the keypad sequence to activate the forward on no answer option.
Web: Forward On Busy OVOC: Call Forward Busy **[KeyCFBusy]**	Defines the keypad sequence to activate the forward on busy option.
Web: Forward On Busy or No Answer OVOC: CF Busy Or No Answer **[KeyCFBusyOrNoAnswer]**	Defines the keypad sequence to activate the forward on 'busy or no answer' option.
Web: Do Not Disturb OVOC: CF Do Not Disturb **[KeyCFDoNotDisturb]**	Defines the keypad sequence to activate the Do Not Disturb option (immediately reject incoming calls).
To activate the required forward method from the telephone: 1 Dial the user-defined sequence number on the keypad; a dial tone is heard. 2 Dial the telephone number to which the call is forwarded (terminate the number with #); a confirmation tone is heard.	
Web: Forward Deactivate OVOC: Call Forward Deactivation **[KeyCFDeact]**	Defines the keypad sequence to deactivate any of the call forward options. After the sequence is pressed, a confirmation tone is heard.
Keypad Feature - Caller ID Restriction Parameters	

Parameter	Description
Web: Restricted Caller ID Activate OVOC: CLIR **[KeyCLIR]**	Defines the keypad sequence to activate the restricted Caller ID option. After the sequence is pressed, a confirmation tone is heard.
Web: Restricted Caller ID Deactivate OVOC: CLIR Deactivation **[KeyCLIRDeact]**	Defines the keypad sequence to deactivate the restricted Caller ID option. After the sequence is pressed, a confirmation tone is heard.
Keypad Feature - Hotline Parameters	
Web: Hot-line Activate OVOC: Hot Line **[KeyHotLine]**	Defines the keypad sequence to activate the delayed hotline option. To activate the delayed hotline option from the telephone, perform the following: **1** Dial the user-defined sequence number on the keypad; a dial tone is heard. **2** Dial the telephone number to which the phone automatically dials after a configurable delay (terminate the number with #); a confirmation tone is heard.
Web: Hot-line Deactivate OVOC: Hot Line Deactivation **[KeyHotLineDeact]**	Defines the keypad sequence to deactivate the delayed hotline option. After the sequence is pressed, a confirmation tone is heard.
Keypad Feature - Transfer Parameters **Note:** See the description of the KeyBlindTransfer parameter for this feature.	
Keypad Feature - Call Waiting Parameters	
Web: Call Waiting Activate OVOC: Keypad Features CW **[KeyCallWaiting]**	Defines the keypad sequence to activate the Call Waiting option. After the sequence is pressed, a confirmation tone is heard.
Web: Call Waiting Deactivate OVOC: Keypad Features CW Deact **[KeyCallWaitingDeact]**	Defines the keypad sequence to deactivate the Call Waiting option. After the sequence is pressed, a confirmation tone is heard.
Keypad Feature - Reject Anonymous Call Parameters	
Web: Reject Anonymous Call Activate OVOC: Reject Anonymous Call **[KeyRejectAnonymousCall]**	Defines the keypad sequence to activate the reject anonymous call option, whereby the device rejects incoming anonymous calls. After the sequence is pressed, a confirmation tone is heard.
Web: Reject Anonymous Call Deactivate OVOC: Reject Anonymous Call Deact **[KeyRejectAnonymousCallDeact]**	Defines the keypad sequence that de-activates the reject anonymous call option. After the sequence is pressed, a confirmation tone is heard.

43.11.9 General FXO Parameters

The general FXO and FXS parameters are described in the table below.

Table 43-57: General FXO and FXS Parameters

Parameter	Description
FXS Parameters	
Web: FXS Coefficient Type OVOC: Country Coefficients **[FXSCountryCoefficients]**	Determines the FXS line characteristics (AC and DC) according to USA or Europe (TBR21) standards. ▪ **[66]** Europe = TBR21 ▪ **[70]** USA = (Default) United States **Note:** For this parameter to take effect, a device reset is required.
[EnhancedFXSLineCurrent]	Defines the FXS off-hook current, which is the current that the device supplies to the analog line when it is in off-hook state. ▪ [0] 20 mA (Default) ▪ [1] 25 mA ▪ [2] 32 mA **Notes:** ▪ MP-11x / MP-124 Rev. D: The parameter is applicable only to the first four FXS ports; the other ports have a fixed current of 20 mA. ▪ MP-124 Rev. E: The device can support 32 mA on all ports (impedance is limited to 1,000 ohm). ▪ For the parameter to take effect, a device reset is required.
FXO Parameters	
Web: FXO Coefficient Type OVOC: Country Coefficients **[CountryCoefficients]**	Determines the FXO line characteristics (AC and DC) according to USA or TBR21 standard. ▪ **[66]** Europe = TBR21 ▪ **[70]** USA = (Default) United States **Note:** For this parameter to take effect, a device reset is required.
[FXODCTermination]	Defines the FXO line DC termination (i.e., resistance). ▪ **[0]** = (Default) DC termination is set to 50 Ohms. ▪ **[1]** = DC termination set to 800 Ohms. The termination changes from 50 to 800 Ohms only when moving from onhook to offhook. **Note:** For this parameter to take effect, a device reset is required.
[EnableFXOCurrentLimit]	Enables limiting the FXO loop current to a maximum of 60 mA (according to the TBR21 standard). ▪ **[0]** = (Default) FXO line current limit is disabled. ▪ **[1]** = FXO loop current is limited to a maximum of 60 mA. **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
[FXONumberOfRings]	Defines the number of rings before the device's FXO interface answers a call by seizing the line. The valid range is 0 to 10. The default is 0. When set to 0, the FXO seizes the line after one ring. When set to 1, the FXO seizes the line after two rings. **Notes:** ▪ This parameter is applicable only if automatic dialing is not used. ▪ If caller ID is enabled and if the number of rings defined by the parameter RingsBeforeCallerID is greater than the number of rings defined by this parameter, the greater value is used.
Web/OVOC: Dialing Mode **[IsTwoStageDial]**	Determines the dialing mode for IP-to-Tel (FXO) calls. ▪ **[0]** One Stage = One-stage dialing. In this mode, the device seizes one of the available lines (according to the ChannelSelectMode parameter), and then dials the destination phone number received in the INVITE message. To specify whether the dialing must start after detection of the dial tone or immediately after seizing the line, use the IsWaitForDialTone parameter. ▪ **[1]** Two Stages = (Default) Two-stage dialing. In this mode, the device seizes one of the PSTN/PBX lines without performing any dialing, connects the remote IP user to the PSTN/PBX, and all further signaling (dialing and Call Progress Tones) is performed directly with the PBX without the device's intervention. **Note:** This parameter can also be configured in a Tel Profile.
Web/OVOC: Waiting For Dial Tone **[IsWaitForDialTone]**	Determines whether or not the device waits for a dial tone before dialing the phone number for IP-to-Tel (FXO) calls. ▪ **[0]** No ▪ **[1]** Yes (default) When one-stage dialing and this parameter are enabled, the device dials the phone number (to the PSTN/PBX line) only after it detects a dial tone. If this parameter is disabled, the device immediately dials the phone number after seizing the PSTN/PBX line without 'listening' for a dial tone. **Notes:** ▪ The correct dial tone parameters must be configured in the CPT file. ▪ The device may take 1 to 3 seconds to detect a dial tone (according to the dial tone configuration in the CPT file). If the dial tone is not detected within 6 seconds, the device releases the call and sends a SIP 500 "Server Internal Error" response.
Web: Time to Wait before Dialing **[msec]** OVOC: Time Before Dial **[WaitForDialTime]**	Defines the delay before the device starts dialing on the FXO line in the following scenarios: ▪ The delay between the time the line is seized and dialing begins during the establishment of an IP-to-Tel call.

Parameter	Description
	Note: Applicable only for one-stage dialing when the parameter IsWaitForDialTone is disabled. ▪ The delay between detection of a Wink and the start of dialing during the establishment of an IP-to-Tel call (for DID lines, EnableDIDWink is set to 1). ▪ For call transfer - the delay after hook-flash is generated and dialing begins. The valid range (in milliseconds) is 0 to 20,000 (i.e., 20 seconds). The default is 1,000 (i.e., 1 second).
Web: Ring Detection Timeout **[sec]** OVOC: Timeout Between Rings **[FXOBetweenRingTime]**	Defines the timeout (in seconds) for detecting the second ring after the first detected ring. If automatic dialing is not used and Caller ID is enabled, the device seizes the line after detection of the second ring signal (allowing detection of caller ID sent between the first and the second rings). If the second ring signal is not received within this timeout, the device doesn't initiate a call to IP. If automatic dialing is used, the device initiates a call to IP when the ringing signal is detected. The FXO line is seized only if the remote IP party answers the call. If the remote party doesn't answer the call and the second ring signal is not received within this timeout, the device releases the IP call. This parameter is typically set to between 5 and 8. The default is 8. **Notes:** ▪ This parameter is applicable only for Tel-to-IP calls. ▪ This timeout is calculated from the end of the ring until the start of the next ring. For example, if the ring cycle is two seconds on and four seconds off, the timeout value should be configured to five seconds (i.e., greater than the off time, e.g., four).
Web: Rings before Detecting Caller ID OVOC: Rings Before Caller ID **[RingsBeforeCallerID]**	Determines the number of rings before the device starts detecting Caller ID. ▪ **[0]** 0 = Before first ring. ▪ **[1]** 1 = (Default) After first ring. ▪ **[2]** 2 = After second ring.
Web/OVOC: Guard Time Between Calls **[GuardTimeBetweenCalls]**	Defines the time interval (in seconds) after a call has ended and a new call can be accepted for IP-to-Tel (FXO) calls. The valid range is 0 to 10. The default is 1. **Note:** Occasionally, after a call ends and on-hook is applied, a delay is required before placing a new call (and performing off-hook). This is necessary to prevent incorrect hook-flash detection or other glare phenomena.
Web: FXO Double Answer **[EnableFXODoubleAnswer]**	Enables the FXO Double Answer feature, which rejects (disconnects) incoming Tel (FXO)-to-IP collect calls and signals (informs) this call denial to the PSTN. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** This feature can also be configured in a Tel Profile.

Parameter	Description
FXO Ring Timeout fxo-ring-timeout [FXORingTimeout]	Defines the delay (in msec) before the device generates a SIP INVITE (call) to the IP side upon detection of a RING_START event from the Tel (FXO) side. This occurs instead of waiting for a RING_END event. This feature is useful for telephony services that employ constant ringing (i.e., no RING_END is sent). For example, Ringdown circuit is a service that sends a constant ringing current over the line, instead of cadence-based 2 second on, 4 second off. For example, when a telephone goes off-hook, a phone at the other end instantly rings. If a RING_END event is received before the timeout expires, the device does not initiate a call and ignores the detected ring. The device ignores RING_END events detected after the timeout expires. The valid value range is 0 to 50 (msec), in steps of 100-msec. For example, a value of 50 represents 5 sec. The default value is 0 (i.e., standard ring operation - the FXO interface sends an INVITE upon receipt of the RING_END event). **Note:** The parameter can be configured for a Tel Profile.
[EnablePulseDialDetection]	Enables the device to detect pulse (rotary) dialing from analog equipment (e.g., telephones) connected to the device's FXS port interfaces. ▪ [0] Disable (default) ▪ [1] Enable **Note:** For the parameter to take effect, a device reset is required.
[EnablePulseDialGeneration]	Enables pulse dialing generation to the analog side when dialing is received from the FXO side. ▪ **[0]** Disable = (Default) Device generates DTMF signals. ▪ **[1]** Enable = Generates pulse dialing. **Note:** For the parameter to take effect, a device reset is required.
[PulseDialGenerationBreakTime]	Defines the duration of the Break connection (off-hook) for FXO pulse dial generation. The valid value range is 20 to 120 (in msec). The default is 60. **Note:** For the parameter to take effect, a device reset is required.
[PulseDialGenerationMakeTime]	Defines the duration of the Make connection (on-hook) for FXO pulse dial generation. The valid value range is 20 to 120 (in msec). The default is 40. **Note:** For the parameter to take effect, a device reset is required.
[PulseDialGenerationInterDigitTime]	Defines the inter-digit duration (time between consecutively dialed digits) for FXO pulse dial generation. The valid value range is 300 to 1500 (in msec). The default is 700.

Parameter	Description
	Note: For the parameter to take effect, a device reset is required.

43.11.10   Hunt Groups and Routing Parameters

The routing parameters are described in the table below.

Table 43-58: Routing Parameters

Parameter	Description
Endpoint Phone Number Table	
Web: Endpoint Phone Number Table OVOC: SIP Endpoints > Phones **[TrunkGroup]**	This table parameter configures and activates the device's endpoints. This is done by defining telephone numbers and assigning them to Hunt Groups. The format of this parameter is shown below: **[TrunkGroup]** FORMAT TrunkGroup_Index = TrunkGroup_TrunkGroupNum, TrunkGroup_FirstTrunkId, TrunkGroup_FirstBChannel, TrunkGroup_LastBChannel, TrunkGroup_FirstPhoneNumber, TrunkGroup_ProfileId, TrunkGroup_LastTrunkId, TrunkGroup_Module; **[\TrunkGroup]** For example: The configuration below assigns channels 1 through 4 to Hunt Group 1 and assigns phone numbers 101 to Channel 1, 102 to Channel 2, and so on: TrunkGroup 0 = 1, 255, 1, 4, 101, 0, 255, 255; **Note:** For a description of this table, see Configuring Endpoint Phone Numbers on page 227.
Hunt Group Settings	
Web: Hunt Group Settings OVOC: SIP Routing > Hunt Group **[TrunkGroupSettings]**	This table parameter configures the rules for channel allocation per Hunt Group. The format of this parameter is as follows: [TrunkGroupSettings] FORMAT TrunkGroupSettings_Index = TrunkGroupSettings_TrunkGroupId, TrunkGroupSettings_ChannelSelectMode, TrunkGroupSettings_RegistrationMode, TrunkGroupSettings_GatewayName, TrunkGroupSettings_ContactUser, TrunkGroupSettings_ServingIPGroup, TrunkGroupSettings_DedicatedConnectionMode, TrunkGroupSettings_MWIInterrogationType, TrunkGroupSettings_TrunkGroupName; [\TrunkGroupSettings] **Note:** For a description of this table, see 'Configuring Hunt Group Settings' on page 229.

Parameter	Description
Web: Channel Select Mode OVOC: Channel Selection Mode **[ChannelSelectMode]**	Defines the method for allocating incoming IP-to-Tel calls to a channel (port) for all Hunt Groups. ▪ **[0]** By Dest Phone Number (default) ▪ **[1]** Cyclic Ascending ▪ **[2]** Ascending ▪ **[3]** Cyclic Descending ▪ **[4]** Descending ▪ **[5]** Dest Number + Cyclic Ascending. ▪ **[6]** By Source Phone Number ▪ **[9]** Ring to Hunt Group ▪ **[11]** Dest Number + Ascending **Notes:** ▪ For a detailed description of the parameter's options, see 'Configuring Hunt Group Settings' on page 229. ▪ Channel select mode per Hunt Group can be configured in the Hunt Group Settings (see 'Configuring Hunt Group Settings' on page 229).
Web: Default Destination Number **[DefaultNumber]**	Defines the default destination phone number, which is used if the received message doesn't contain a called party number and no phone number is configured in the <Endpoint Phone Number Table' (see to Configuring Endpoint Phone Numbers on page 227). This parameter is used as a starting number for the list of channels comprising all the device's Hunt Groups. The default is 1000.
Web: Source IP Address Input **[SourceIPAddressInput]**	Determines which IP address the device uses to determine the source of incoming INVITE messages for IP-to-Tel routing. ▪ **[-1]** = (Default) Not configured. ▪ **[0]** SIP Contact Header = The IP address in the Contact header of the incoming INVITE message is used. ▪ **[1]** Layer 3 Source IP = The actual IP address (Layer 3) from where the SIP packet was received is used.

Parameter	Description
Web: Use Source Number As Display Name OVOC: Display Name **[UseSourceNumberAsDisplayName]**	Determines the use of Tel Source Number and Display Name for Tel-to-IP calls. ▪ **[0]** No = (Default) If a Tel Display Name is received, the Tel Source Number is used as the IP Source Number and the Tel Display Name is used as the IP Display Name. If no Display Name is received from the Tel side, the IP Display Name remains empty. ▪ **[1]** Yes = If a Tel Display Name is received, the Tel Source Number is used as the IP Source Number and the Tel Display Name is used as the IP Display Name. If no Display Name is received from the Tel side, the Tel Source Number is used as the IP Source Number and also as the IP Display Name. ▪ **[2]** Overwrite = The Tel Source Number is used as the IP Source Number and also as the IP Display Name (even if the received Tel Display Name is not empty). ▪ **[3]** Original = Similar to option **[2],** except that the operation is done before regular calling number manipulation.
Web/OVOC: Use Display Name as Source Number **[UseDisplayNameAsSourceNumber]**	Defines how the display name (caller ID) received from the IP side (in the SIP From header) effects the source number sent to the Tel side, for IP-to-Tel calls. ▪ [0] No = (Default) If a display name is received from the IP side, the source number of the IP side is used as the Tel source number. ▪ [1] Yes = If a display name is received from the IP side, the display name of the IP side is used as the Tel source number and Presentation is set to Allowed (0). If no display name is received from the IP side, the source number of the IP side is used as the Tel source number and Presentation is set to Restricted (1). For example: ✓ If 'From: 100 <sip:200@201.202.203.204>' is received from the IP side, the outgoing source number (and display name) are set to "100" and Presentation is set to Allowed (0). ✓ If 'From: <sip:400@101.102.103.104>' is received from the IP side, the outgoing source number is set to "400" and Presentation is set to Restricted (1). ▪ [2] Preferred = If a display name is received from the IP side, the display name of the IP side is used as the Tel source number. If no display name is received from the IP side, this setting does not affect the Tel source number.
Web: Use Routing Table for Host Names and Profiles OVOC: Use Routing Table For Host Names **[AlwaysUseRouteTable]**	Determines whether to use the device's routing table to obtain the URI host name and optionally, an IP profile (per call) even if a Proxy server is used. ▪ **[0]** Disable = (Default) Don't use internal routing table. ▪ **[1]** Enable = Use the Tel to IP Routing. **Notes:** ▪ This parameter appears only if the 'Use Default Proxy' parameter is enabled.

Parameter	Description
	▪ The domain name is used instead of a Proxy name or IP address in the INVITE SIP URI.
Web/OVOC: Tel to IP Routing Mode **[RouteModeTel2IP]**	For a description of this parameter, see 'Configuring Tel to IP Routing' on page 250.
Tel to IP Routing	
Web: Tel to IP Routing OVOC: SIP Routing > Tel to IP **[Prefix]**	This table parameter configures the Tel to IP Routing for routing Tel-to-IP calls. The format of this parameter is as follows: [PREFIX] FORMAT PREFIX_Index = PREFIX_DestinationPrefix, PREFIX_DestAddress, PREFIX_SourcePrefix, PREFIX_ProfileId, PREFIX_MeteringCode, PREFIX_DestPort, PREFIX_SrcIPGroupID, PREFIX_DestHostPrefix, PREFIX_DestIPGroupID, PREFIX_SrcHostPrefix, PREFIX_TransportType, PREFIX_SrcTrunkGroupID, PREFIX_DestSRD, PREFIX_CostGroup, PREFIX_ForkingGroup; [\PREFIX] For example: PREFIX 0 = *, domain.com, *, 0, 255, $$, -1, , 1, , -1, -1, -1,,; PREFIX 1 = 20, 10.33.37.77, *, 0, 255, $$, -1, , 2, , 0, -1,,; **Note:** For a detailed description of this table, see 'Configuring Tel to IP Routing' on page 250.
IP to Hunt Group Routing Table	
Web: IP to Hunt Group Routing Table OVOC: SIP Routing > IP to Hunt **[PSTNPrefix]**	This table parameter configures the routing of IP-to-Hunt Groups. The format of this parameter is as follows: [PSTNPrefix] ORMAT PstnPrefix_Index = PstnPrefix_DestPrefix, PstnPrefix_TrunkGroupId, PstnPrefix_SourcePrefix, PstnPrefix_SourceAddress, PstnPrefix_ProfileId, PstnPrefix_SrcIPGroupID, PstnPrefix_DestHostPrefix, PstnPrefix_SrcHostPrefix, PstnPrefix_SrcSRDID, PstnPrefix_TrunkId; [\PSTNPrefix] For example: PstnPrefix 0 = 100, 1, 200, *, 0, 2, , , ,; PstnPrefix 1 = *, 2, *, , 1, 3, acl, joe, , ,; **Note:** For a detailed description of this table, see 'Configuring IP to Hunt Group Routing Table' on page 257.
Web/OVOC: IP to Tel Routing Mode **[RouteModeIP2Tel]**	Determines whether to route IP calls to the Hunt Group before or after manipulation of the destination number (configured in 'Configuring Source/Destination Number Manipulation Rules' on page 235).

Parameter	Description
	▪ **[0]** Route calls before manipulation = (Default) Calls are routed before the number manipulation rules are applied. ▪ **[1]** Route calls after manipulation = Calls are routed after the number manipulation rules are applied.
Web: IP Security OVOC: Secure Call From IP **[SecureCallsFromIP]**	Determines the device's policy on accepting or blocking SIP calls (IP-to-Tel calls). This is useful in preventing unwanted SIP calls, SIP messages, and/or VoIP spam. ▪ **[0]** Disable = (Default) The device accepts all SIP calls. ▪ **[1]** Secure Incoming calls = The device accepts SIP calls (i.e., calls from the IP side) only from IP addresses that are defined in the Tel to IP Routing or Proxy Set table, or IP addresses resolved from DNS servers from FQDN values defined in the Proxy Set table. All other incoming calls are rejected. ▪ **[2]** Secure All calls = The device accepts SIP calls only from IP addresses (in dotted-decimal notation format) that are defined in the Tel to IP Routing table or Proxy Set table, and rejects all other incoming calls. In addition, if an FQDN is defined in the routing table or Proxy Set table, the call is allowed to be sent only if the resolved DNS IP address appears in one of these tables; otherwise, the call is rejected. Therefore, the difference between this option and option **[1]** is that this option is concerned only about numerical IP addresses that are defined in the tables. **Note:** If this parameter is set to **[0]** or **[1],** when using Proxies or Proxy Sets, it is unnecessary to configure the Proxy IP addresses in the routing table. The device allows SIP calls received from the Proxy IP addresses even if these addresses are not configured in the routing table.
Web/OVOC: Filter Calls to IP **[FilterCalls2IP]**	Enables filtering of Tel-to-IP calls when a Proxy is used (i.e., IsProxyUsed parameter is set to 1 - see 'Configuring Proxy and Registration Parameters' on page 208). ▪ **[0]** Don't Filter = (Default) The device doesn't filter calls when using a Proxy. ▪ **[1]** Filter = Filtering is enabled. When this parameter is enabled and a Proxy is used, the device first checks the Tel to IP Routingbefore making a call through the Proxy. If the number is not allowed (i.e., number isn't listed in the table or a call restriction routing rule of IP address 0.0.0.0 is applied), the call is released. **Note:** When no Proxy is used, this parameter must be disabled and filtering is according to the Tel to IP Routing.

Parameter	Description
Web: Add CIC **[AddCicAsPrefix]**	Determines whether to add the Carrier Identification Code (CIC) as a prefix to the destination phone number for IP-to-Tel calls. When this parameter is enabled, the 'cic' parameter in the incoming SIP INVITE can be used for IP-to-Tel routing decisions. It routes the call to the appropriate Hunt Group based on this parameter's value. ▪ **[0]** No (default) ▪ **[1]** Yes For example, as a result of receiving the below INVITE, the destination number after number manipulation is cic+167895550001: INVITE sip:5550001;cic=+16789@172.18.202.60:5060;user=phone SIP/2.0 **Note:** After the cic prefix is added, the IP to Hunt Group Routing Table can be used to route this call to a specific Hunt Group. The Destination Number IP to Tel Manipulation table must be used to remove this prefix before placing the call to the Tel.
Web: ENUM Resolution CLI: enum-service-domain **[EnumService]**	Defines the ENUM service for translating telephone numbers to IP addresses or domain names (FQDN). For example, e164.arpa, e164.customer.net, or NRENum.net. The valid value is a string of up to 50 characters. The default is "e164.arpa". **Note:** ENUM-based routing is configured in the Outbound IP Routing table using the "ENUM" string value as the destination address to denote this parameter's value.

43.11.11   IP Connectivity Parameters

The IP connectivity parameters are described in the table below.

Table 43-59: IP Connectivity Parameters

Parameter	Description
Web: Enable Alt Routing Tel to IP OVOC: Enable Alternative Routing **[AltRoutingTel2IPEnable]**	Enables the Alternative Routing feature for Tel-to-IP calls. ▪ **[0]** Disable = (Default) Disables the Alternative Routing feature. ▪ **[1]** Enable = Enables the Alternative Routing feature. ▪ **[2]** Status Only = The Alternative Routing feature is disabled, but read-only information on the QoS of the destination IP addresses is provided.
Web: Alt Routing Tel to IP Mode OVOC: Alternative Routing Mode **[AltRoutingTel2IPMode]**	Determines the IP Connectivity event(s) reason for triggering Alternative Routing. ▪ **[0]** None = Alternative routing is not used. ▪ **[1]** Connectivity = Alternative routing is performed if a ping or SIP OPTIONS message to the initial destination fails (determined according to the AltRoutingTel2IPConnMethod parameter).

Parameter	Description
	▪ **[2]** QoS = Alternative routing is performed if poor QoS is detected. ▪ **[3]** Both = (Default) Alternative routing is performed if either ping or SIP OPTIONS to initial destination fails, poor QoS is detected, or the DNS host name is not resolved. **Notes:** ▪ QoS is quantified according to delay and packet loss calculated according to previous calls. QoS statistics are reset if no new data is received within two minutes. ▪ To receive quality information (displayed in the 'Quality Status' and 'Quality Info.' fields in 'Viewing IP Connectivity' on page 417) per destination, this parameter must be set to 2 or 3.
Web: Alt Routing Tel to IP Connectivity Method OVOC: Alternative Routing Telephone to IP Connection Method **[AltRoutingTel2IPConnMethod]**	Determines the method used by the device for periodically querying the connectivity status of a destination IP address. ▪ **[0]** ICMP Ping = (Default) Internet Control Message Protocol (ICMP) ping messages. ▪ **[1]** SIP OPTIONS = The remote destination is considered offline if the latest OPTIONS transaction timed out. Any response to an OPTIONS request, even if indicating an error, brings the connectivity status to online.
Web: Alt Routing Tel to IP Keep Alive Time OVOC: Alternative Routing Keep Alive Time **[AltRoutingTel2IPKeepAliveTime]**	Defines the time interval (in seconds) between SIP OPTIONS Keep-Alive messages used for the IP Connectivity application. The valid range is 5 to 2,000,000. The default is 60.
Web: Max Allowed Packet Loss for Alt Routing **[%]** **[IPConnQoSMaxAllowedPL]**	Defines the packet loss (in percentage) at which the IP connection is considered a failure and Alternative Routing mechanism is activated. The default is 20%.
Web: Max Allowed Delay for Alt Routing **[msec]** **[IPConnQoSMaxAllowedDelay]**	Defines the transmission delay (in msec) at which the IP connection is considered a failure and the Alternative Routing mechanism is activated. The range is 100 to 10,000. The default is 250.

43.11.12   Alternative Routing Parameters

The alternative routing parameters are described in the table below.

Table 43-60: Alternative Routing Parameters

Parameter	Description
Web/OVOC: Redundant Routing Mode **[RedundantRoutingMode]**	Determines the type of redundant routing mechanism when a call can't be completed using the main route. ▪ **[0]** Disable = No redundant routing is used. If the call can't be completed using the main route (using the active Proxy or the first matching rule in the Routing table), the call is disconnected. ▪ **[1]** Routing Table = (Default) Internal routing table is used to locate a redundant route. ▪ **[2]** Proxy = Proxy list is used to locate a redundant route.

Parameter	Description
	Note: To implement the Redundant Routing Mode mechanism, you first need to configure the parameter AltRouteCauseTEL2IP (Reasons for Alternative Routing table).
[EnableAltMapTel2IP]	Enables different Tel-to-IP destination number manipulation rules per routing rule when several (up to three) Tel-to-IP routing rules are defined and if alternative routing using release causes is used. For example, if an INVITE message for a Tel-to-IP call is returned with a SIP 404 Not Found response, the call can be re-sent to a different destination number (as defined using the parameter NumberMapTel2IP). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
Web/OVOC: Alternative Routing Tone Duration **[ms]** **[AltRoutingToneDuration]**	Defines the duration (in milliseconds) for which the device plays a tone to the endpoint on each attempt for Tel-to-IP alternative routing. When the device finishes playing the tone, a new SIP INVITE message is sent to the new IP destination. The tone played is the call forward tone (Tone Type #25 in the CPT file). The valid range is 0 to 20,000. The default is 0 (i.e., no tone is played). **Note:** This parameter is applicable only to Tel-to-IP alternative routing.
Reasons for Alternative Tel-to-IP Routing Table	
Web: Reasons for Alternative Routing OVOC: Alt Route Cause Tel to IP **[AltRouteCauseTel2IP]**	This table parameter configures SIP call failure reason values received from the IP side. If an IP call is released as a result of one of these reasons, the device attempts to locate an alternative IP route for the call in the Tel to IP Routing (if a Proxy is not used) or used as a redundant Proxy (you need to set the parameter RedundantRoutingMode to 2). The release reason for Tel-to-IP calls is provided in SIP 4xx, 5xx, and 6xx response codes. The format of this parameter is as follows: [AltRouteCauseTel2IP] FORMAT AltRouteCauseTel2IP_Index = AltRouteCauseTel2IP_ReleaseCause; [\AltRouteCauseTel2IP] For example: AltRouteCauseTel2IP 0 = 486; (Busy Here) AltRouteCauseTel2IP 1 = 480; (Temporarily Unavailable) AltRouteCauseTel2IP 2 = 408; (No Response) **Note:** For a detailed description of this table, see 'Alternative Routing Based on SIP Responses' on page 263.
Reasons for Alternative IP-to-Tel Routing Table	
Web: Reasons for Alternative IP-to-Tel Routing OVOC: Alt Route Cause IP to Tel **[AltRouteCauseIP2Tel]**	This table parameter configures call failure reason values received from the Tel side . If a call is released as a result of one of these reasons, the device attempts to locate an alternative Hunt Group for the call in the IP to Hunt Group Routing Table. The format of this parameter is as follows: [AltRouteCauseIP2Tel] FORMAT AltRouteCauseIP2Tel_Index = AltRouteCauseIP2Tel_ReleaseCause; [\AltRouteCauseIP2Tel]

Parameter	Description
	For example: AltRouteCauseIP2Tel 0 = 3 (No Route to Destination) AltRouteCauseIP2Tel 1 = 1 (Unallocated Number) AltRouteCauseIP2Tel 2 = 17 (Busy Here) AltRouteCauseIP2Tel 2 = 27 (Destination Out of Order) **Note:** For a detailed description of this table, see 'Alternative Routing to Trunk upon Q.931 Call Release Cause Code' on page 265.
Forward On Busy Trunk Destination Table	
Web/OVOC: Forward On Busy Trunk Destination **[ForwardOnBusyTrunkDest]**	This table parameter configures the Forward On Busy Trunk Destination table. This table allows you to define an alternative IP destination if a trunk is busy for IP-to-Tel calls. The format of this parameter is as follows: [ForwardOnBusyTrunkDest] FORMAT ForwardOnBusyTrunkDest_Index = ForwardOnBusyTrunkDest_TrunkGroupId, ForwardOnBusyTrunkDest_ForwardDestination; [\ForwardOnBusyTrunkDest] For example, the below configuration forwards IP-to-Tel calls to destination user "112" at host IP address 10.13.4.12, port 5060, using transport protocol TCP, if Trunk Group ID 2 is unavailable: ForwardOnBusyTrunkDest 1 = 2, 112@10.13.4.12:5060;transport=tcp; **Note:** For a detailed description of this table, see 'Alternative Routing to IP Destination upon Busy Trunk' on page 267.

43.11.13   Number Manipulation Parameters

The number manipulation parameters are described in the table below.

Table 43-61: Number Manipulation Parameters

Parameter	Description
Web: Copy Destination Number to Redirect Number OVOC: Copy Dest to Redirect Number **[CopyDest2RedirectNumber]**	Determines whether the device copies the called number to the outgoing SIP Diversion header for Tel-to-IP calls. Therefore, the called number is used as a redirect number. Call redirection information is typically used for Unified Messaging and voice mail services to identify the recipient of a message. ▪ **[0]** Don't copy = (Default) Disable. ▪ **[1]** Copy after phone number manipulation = Copies the called number after manipulation. The device first performs Tel-to-IP destination phone number manipulation (i.e., on the SIP To header), and only then copies the manipulated called number to the SIP Diversion header for the Tel-to-IP call. Therefore, with this option, the called and redirect numbers are identical. ▪ **[2]** Copy before phone number manipulation = Copies the called number before manipulation. The device first copies the original called number to the SIP Diversion header, and then performs Tel-to-IP destination phone number manipulation. Therefore, this allows you to have different numbers for the

Parameter	Description
	called (i.e., SIP To header) and redirect (i.e., SIP Diversion header) numbers. **Note:** This parameter can also be configured in an IP Profile.
Web/OVOC: Add Hunt Group ID as Prefix **[AddTrunkGroupAsPrefix]**	Determines whether the Hunt Group ID is added as a prefix to the destination phone number (i.e., called number) for Tel-to-IP calls. ▪ **[0]** No = (Default) Don't add Hunt Group ID as prefix. ▪ **[1]** Yes = Add Hunt Group ID as prefix to called number. **Notes:** ▪ This option can be used to define various routing rules. ▪ To use this feature, you must configure the Hunt Group IDs (see Configuring Endpoint Phone Numbers on page 227).
Web: Add Trunk ID as Prefix OVOC: Add Port ID As Prefix **[AddPortAsPrefix]**	Determines whether or not the port number is added as a prefix to the called (destination) number for Tel-to-IP calls. ▪ **[0]** No (Default) ▪ **[1]** Yes If enabled, the device adds the following prefix to the called phone number: port number (single digit in the range 1 to 8for 8-port devices, two digits in the range 01 to 24 for MP-124). This option can be used to define various routing rules.
Web/OVOC: Add Trunk Group ID as Prefix to Source **[AddTrunkGroupAsPrefixToSource]**	Determines whether the device adds the Hunt Group ID (from where the call originated) as the prefix to the calling number (i.e. source number). ▪ **[0]** No (default) ▪ **[1]** Yes
Web: IP to Tel Remove Routing Table Prefix OVOC: Remove Prefix **[RemovePrefix]**	Determines whether or not the device removes the prefix (as configured in the IP to Hunt Group Routing Table - see 'Configuring IP to Hunt Group Routing Table' on page 257) from the destination number for IP-to-Tel calls, before sending it to the Tel. ▪ **[0]** No (default) ▪ **[1]** Yes For example: To route an incoming IP-to-Tel call with destination number "21100", the IP to Hunt Group Routing Table is scanned for a matching prefix. If such a prefix is found (e.g., "21"), then before the call is routed to the corresponding Hunt Group, the prefix "21" is removed from the original number, and therefore, only "100" remains. **Notes:** ▪ This parameter is applicable only if number manipulation is performed after call routing for IP-to-Tel calls (i.e., RouteModeIP2Tel parameter is set to 0). ▪ Similar operation (of removing the prefix) is also achieved by using the usual number manipulation rules.
[SwapTel2IPCalled&CallingNumbers]	Determines whether the device swaps the calling and called numbers received from the Tel side (for Tel-to-IP calls). The SIP INVITE message contains the swapped numbers. ▪ **[0]** = (Default) Disabled ▪ **[1]** = Swap calling and called numbers

Parameter	Description
	Note: This parameter can also be configured in a Tel Profile.
Web: Add Number Plan and Type to RPI Header OVOC: Add Ton 2 RPI **[AddTON2RPI]**	Determines whether the TON/PLAN parameters are included in the Remote-Party-ID (RPID) header. ▪ **[0]** No ▪ **[1]** Yes (default) If the Remote-Party-ID header is enabled (EnableRPIHeader = 1) and AddTON2RPI = 1, it's possible to configure the calling and called number type and number plan using the Number Manipulation tables for Tel-to-IP calls.
Web/OVOC: Source Manipulation Mode **[SourceManipulationMode]**	Determines the SIP headers containing the source number after manipulation: ▪ **[0]** = (Default) The SIP From and P-Asserted-Identity headers contain the source number after manipulation. ▪ **[1]** = Only SIP From header contains the source number after manipulation, while the P-Asserted-Identity header contains the source number before manipulation.
Calling Name Manipulations IP-to-Tel Table	
[CallingNameMapIp2Tel]	Configures rules for manipulating the calling name (caller ID) in the received SIP message for IP-to-Tel calls. This can include modifying or removing the calling name. The format of this table ini file parameter is as follows: [CallingNameMapIp2Tel] FORMAT CallingNameMapIp2Tel_Index = CallingNameMapIp2Tel_DestinationPrefix, CallingNameMapIp2Tel_SourcePrefix, CallingNameMapIp2Tel_CallingNamePrefix, CallingNameMapIp2Tel_SourceAddress, CallingNameMapIp2Tel_RemoveFromLeft, CallingNameMapIp2Tel_RemoveFromRight, CallingNameMapIp2Tel_LeaveFromRight, CallingNameMapIp2Tel_Prefix2Add, CallingNameMapIp2Tel_Suffix2Add; [\CallingNameMapIp2Tel] **Note:** For a detailed description of this table, see 'Configuring SIP Calling Name Manipulation' on page 242.
Calling Name Manipulations Tel-to-IP Table	
[CallingNameMapTel2Ip]	This table parameter configures rules for manipulating the calling name (caller ID) for Tel-to-IP calls. This can include modifying or removing the calling name. [CallingNameMapTel2Ip] FORMAT CallingNameMapTel2Ip_Index = CallingNameMapTel2Ip_DestinationPrefix, CallingNameMapTel2Ip_SourcePrefix, CallingNameMapTel2Ip_CallingNamePrefix, CallingNameMapTel2Ip_SrcTrunkGroupID, CallingNameMapTel2Ip_SrcIPGroupID, CallingNameMapTel2Ip_RemoveFromLeft, CallingNameMapTel2Ip_RemoveFromRight, CallingNameMapTel2Ip_LeaveFromRight, CallingNameMapTel2Ip_Prefix2Add, CallingNameMapTel2Ip_Suffix2Add; [\CallingNameMapTel2Ip]

Parameter	Description
	Note: For a detailed description of this table, see 'Configuring SIP Calling Name Manipulation' on page 242.
Destination Phone Number Manipulation for IP-to-Tel Calls Table	
Web: Destination Phone Number Manipulation Table for IP > Tel Calls OVOC: SIP Manipulations > Destination IP to Telcom **[NumberMapIP2Tel]**	This table parameter manipulates the destination number of IP-to-Tel calls. The format of this parameter is as follows: [NumberMapIp2Tel] FORMAT NumberMapIp2Tel_Index = NumberMapIp2Tel_DestinationPrefix, NumberMapIp2Tel_SourcePrefix, NumberMapIp2Tel_SourceAddress, NumberMapIp2Tel_NumberType, NumberMapIp2Tel_NumberPlan, NumberMapIp2Tel_RemoveFromLeft, NumberMapIp2Tel_RemoveFromRight, NumberMapIp2Tel_LeaveFromRight, NumberMapIp2Tel_Prefix2Add, NumberMapIp2Tel_Suffix2Add, NumberMapIp2Tel_IsPresentationRestricted; [\NumberMapIp2Tel] For example: NumberMapIp2Tel 0 = 03,22,$$,$$,$$,2,667,$$,$$; **Note:** For a detailed description of this table, see 'Configuring Source/Destination Number Manipulation' on page 235.
OVOC: Perform Additional IP2TEL Destination Manipulation **[PerformAdditionalIP2TELDe stinationManipulation]**	Enables additional destination number manipulation for IP-to-Tel calls. The additional manipulation is done on the initially manipulated destination number, and this additional rule is also configured in the manipulation table (NumberMapIP2Tel parameter). This enables you to configure only a few manipulation rules for complex number manipulation requirements (that generally require many rules). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
Destination Phone Number Manipulation for Tel-to-IP Calls Table	
Web: Destination Phone Number Manipulation Table for Tel > IP Calls OVOC: SIP Manipulations > Destination Telcom to IPs **[NumberMapTel2IP]**	This table parameter manipulates the destination number of Tel-to-IP calls. The format of this parameter is as follows: [NumberMapTel2Ip] FORMAT NumberMapTel2Ip_Index = NumberMapTel2Ip_DestinationPrefix, NumberMapTel2Ip_SourcePrefix, NumberMapTel2Ip_SourceAddress, NumberMapTel2Ip_NumberType, NumberMapTel2Ip_NumberPlan, NumberMapTel2Ip_RemoveFromLeft, NumberMapTel2Ip_RemoveFromRight, NumberMapTel2Ip_LeaveFromRight, NumberMapTel2Ip_Prefix2Add, NumberMapTel2Ip_Suffix2Add, NumberMapTel2Ip_IsPresentationRestricted, NumberMapTel2Ip_SrcTrunkGroupID, NumberMapTel2Ip_ SrcIPGroupID; [\NumberMapTel2Ip] For example: NumberMapTel2Ip 0 = 01,$$,*,0,0,2,$$,$$,971,$$,$$,$$,$$; NumberMapTel2Ip 1 = 10,10,*,255,255,3,0,5,100,$$,255,$$,$$;

Parameter	Description
	Note: For a detailed description of this table, see 'Configuring Source/Destination Number Manipulation' on page 235.
Source Phone Number Manipulation for IP-to-Tel Calls Table	
Web: Source Phone Number Manipulation Table for IP > Tel Calls OVOC: SIP Manipulations > Source IP to Telcom **[SourceNumberMapIP2Tel]**	This *para*meter table manipulates the source number for IP-to-Tel calls. The format of this parameter is as follows: [SourceNumberMapIp2Tel] FORMAT SourceNumberMapIp2Tel_Index = SourceNumberMapIp2Tel_DestinationPrefix, SourceNumberMapIp2Tel_SourcePrefix, SourceNumberMapIp2Tel_SourceAddress, SourceNumberMapIp2Tel_NumberType, SourceNumberMapIp2Tel_NumberPlan, SourceNumberMapIp2Tel_RemoveFromLeft, SourceNumberMapIp2Tel_RemoveFromRight, SourceNumberMapIp2Tel_LeaveFromRight, SourceNumberMapIp2Tel_Prefix2Add, SourceNumberMapIp2Tel_Suffix2Add, SourceNumberMapIp2Tel_IsPresentationRestricted; [\SourceNumberMapIp2Tel] For example: SourceNumberMapIp2Tel 0 = 22,03,$$,$$,$$,$$,2,667,$$,$$; SourceNumberMapIp2Tel 1 = 034,01,1.1.1.1,$$,0,2,$$,$$,972,$$,10; **Note:** For a detailed description of this table, see 'Configuring Source/Destination Number Manipulation' on page 235.
OVOC: Perform Additional IP2TEL Source Manipulation **[PerformAdditionalIP2TELSourceManipulation]**	Enables additional source number manipulation for IP-to-Tel calls. The additional manipulation is done on the initially manipulated source number, and this additional rule is also configured in the manipulation table (SourceNumberMapIP2Tel parameter). This enables you to configure only a few manipulation rules for complex number manipulation requirements (that generally require many rules). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
Source Phone Number Manipulation for Tel-to-IP Calls Table	
Web: Source Phone Number Manipulation Table for Tel > IP Calls OVOC: SIP Manipulations > Source Telcom to IP **[SourceNumberMapTel2IP]**	This table parameter manipulates the source phone number for Tel-to-IP calls. The format of this parameter is as follows: [SourceNumberMapTel2Ip] FORMAT SourceNumberMapTel2Ip_Index = SourceNumberMapTel2Ip_DestinationPrefix, SourceNumberMapTel2Ip_SourcePrefix, SourceNumberMapTel2Ip_SourceAddress, SourceNumberMapTel2Ip_NumberType, SourceNumberMapTel2Ip_NumberPlan, SourceNumberMapTel2Ip_RemoveFromLeft, SourceNumberMapTel2Ip_RemoveFromRight, SourceNumberMapTel2Ip_LeaveFromRight, SourceNumberMapTel2Ip_Prefix2Add, SourceNumberMapTel2Ip_Suffix2Add, SourceNumberMapTel2Ip_IsPresentationRestricted, NumberMapTel2Ip_SrcTrunkGroupID, NumberMapTel2Ip_SrcIPGroupID; [\SourceNumberMapTel2Ip]

Parameter	Description
	For example: SourceNumberMapTel2Ip 0 = 22,03,$$,0,0,$$,2,$$,667,$$,0,$$,$$; SourceNumberMapTel2Ip 0 = 10,10,*,255,255,3,0,5,100,$$,255,$$,$$; **Note:** For a detailed description of this table, see 'Configuring Source/Destination Number Manipulation' on page 235.
Redirect Number Tel-to-IP Table	
Web: Redirect Number Tel -> IP OVOC: Redirect Number Map Tel to IP **[RedirectNumberMapTel2IP]**	This table parameter manipulates the Redirect Number for Tel-to-IP calls. The format of this parameter is as follows: [RedirectNumberMapTel2Ip] FORMAT RedirectNumberMapTel2Ip_Index = RedirectNumberMapTel2Ip_DestinationPrefix, RedirectNumberMapTel2Ip_RedirectPrefix, RedirectNumberMapTel2Ip_RemoveFromLeft, RedirectNumberMapTel2Ip_RemoveFromRight, RedirectNumberMapTel2Ip_LeaveFromRight, RedirectNumberMapTel2Ip_Prefix2Add, RedirectNumberMapTel2Ip_Suffix2Add, RedirectNumberMapTel2Ip_IsPresentationRestricted, RedirectNumberMapTel2Ip_SrcTrunkGroupID, RedirectNumberMapTel2Ip_SrcIPGroupID; [\RedirectNumberMapTel2Ip] For example: RedirectNumberMapTel2Ip 1 = *, *, 4, 0, 255, , , 255, -1, -1; **Note:** For a description of this table, see 'Configuring Redirect Number Manipulation' on page 245.
Phone Context Table	
Web: Phone Context Table OVOC: SIP Manipulations > Phone Context **[PhoneContext]**	This table parameter configures the Phone Context table. This parameter maps NPI and TON to the SIP 'phone-context' parameter, and vice versa. The format for this parameter is as follows: [PhoneContext] FORMAT PhoneContext_Index = PhoneContext_Npi, PhoneContext_Ton, PhoneContext_Context; [\PhoneContext] For example: PhoneContext 0 = 0,0,unknown.com PhoneContext 1 = 1,1,host.com PhoneContext 2 = 9,1,na.e164.host.com **Note:** For a detailed description of this table, see 'Mapping NPI/TON to SIP Phone-Context' on page 247.
Web/OVOC: Add Phone Context As Prefix **[AddPhoneContextAsPrefix]**	Determines whether the received Phone-Context parameter is added as a prefix to the outgoing Called and Calling numbers. ▪ **[0]** Disable (default) ▪ **[1]** Enable

43.12 Least Cost Routing Parameters

The Least Cost Routing (LCR) parameters are described in the table below.

Table 43-62: LCR Parameters

Parameter	Description
Web: Routing Rule Groups Table **[RoutingRuleGroups]**	This table parameter enables the LCR feature and configures the average call duration and default call cost. The default call cost determines whether routing rules that are not configured with a Cost Group are considered as a higher or lower cost route compared to other matching routing rules that are assigned Cost Groups. [RoutingRuleGroups] FORMAT RoutingRuleGroups_Index = RoutingRuleGroups_LCREnable, RoutingRuleGroups_LCRAverageCallLength, RoutingRuleGroups_LCRDefaultCost; [\RoutingRuleGroups] **Note:** For a detailed description of this table, see 'Enabling LCR and Configuring Default LCR' on page 189.
Web: Cost Group Table OVOC: Cost Group Provisioning > Cost Group **[CostGroupTable]**	This table parameter configures the Cost Groups for LCR, where each Cost Group is configured with a name, fixed call connection charge, and a call rate (charge per minute). [CostGroupTable] FORMAT CostGroupTable_Index = CostGroupTable_CostGroupName, CostGroupTable_DefaultConnectionCost, CostGroupTable_DefaultMinuteCost; [\CostGroupTable] For example: CostGroupTable 2 = "Local Calls", 2, 1; **Note:** For a detailed description of this table, see 'Configuring Cost Groups' on page 191.
Web: Cost Group > Time Band Table OVOC: Time Band Provisioning > Time Band **[CostGroupTimebands]**	This table parameter configures time bands and associates them with Cost Groups. [CostGroupTimebands] FORMAT CostGroupTimebands_TimebandIndex = CostGroupTimebands_StartTime, CostGroupTimebands_EndTime, CostGroupTimebands_ConnectionCost, CostGroupTimebands_MinuteCost; [\CostGroupTimebands] **Note:** For a detailed description of this table, see 'Configuring Time Bands for Cost Groups' on page 192.

43.13 Standalone Survivability Parameters

The Stand-alone Survivability (SAS) parameters are described in the table below.

Table 43-63: SAS Parameters

Parameter	Description
Web: Enable SAS OVOC: Enable **[EnableSAS]**	Enables the Stand-Alone Survivability (SAS) feature. ▪ **[0]** Disable (default)

Parameter	Description
	▪ **[1]** Enable When enabled, the device receives the registration requests from different SIP entities in the local network and then forwards them to the defined proxy. If the connection to the proxy fails ('Emergency Mode'), the device serves as a proxy by allowing calls internal to the local network or outgoing to PSTN. **Note:** For this parameter to take effect, a device reset is required.
Web: SAS Local SIP UDP Port OVOC: Local SIP UDP **[SASLocalSIPUDPPort]**	Defines the local UDP port for sending and receiving SIP messages for SAS. The SIP entities in the local network need to send the registration requests to this port. When forwarding the requests to the proxy ('Normal Mode'), this port serves as the source port. The valid range is 1 to 65,534. The default is 5080.
Web: SAS Default Gateway IP OVOC: Default Gateway IP **[SASDefaultGatewayIP]**	Defines the Default Gateway used in SAS 'Emergency Mode'. When an incoming SIP INVITE is received and the destination Address-Of-Record is not included in the SAS database, the request is immediately sent to this default gateway. The address can be configured as an IP address (dotted-decimal notation) or as a domain name (up to 49 characters). You can also configure the IP address with a destination port, e.g., "10.1.2.3:5060". The default is a null string, i.e., the local IP address of the gateway.
Web: SAS Registration Time OVOC: Registration Time **[SASRegistrationTime]**	Defines the value of the SIP Expires header that is sent in a 200 OK response to an incoming REGISTER message when in SAS 'Emergency Mode'. The valid range is 0 (Analog) to 2,000,000. The default is 20.
Web: SAS Local SIP TCP Port OVOC: Local SIP TCP Port **[SASLocalSIPTCPPort]**	Defines the local TCP port used to send/receive SIP messages for the SAS application. The SIP entities in the local network need to send the registration requests to this port. When forwarding the requests to the proxy ('Normal Mode'), this port serves as the source port. The valid range is 1 to 65,534. The default is 5080.
Web: SAS Local SIP TLS Port OVOC: Local SIP TLS Port **[SASLocalSIPTLSPort]**	Defines the local TLS port used to send/receive SIP messages for the SAS application. The SIP entities in the local network need to send the registration requests to this port. When forwarding the requests to the proxy ('Normal Mode'), this port serves as the source port. The valid range is 1 to 65,534. The default is 5081.

Parameter	Description
Web: SAS Connection Reuse **[SASConnectionReuse]**	Enables the re-use of the same TCP connection for sessions with the same user in the SAS application. ▪ **[0]** Disable ▪ **[1]** Enable (default) The device can use the same TCP connection for multiple SIP requests / responses for a specific SIP UA. The benefits of this feature include less CPU and memory usage because fewer TCP connections are open and reduced network congestion. For example, assume the following: ▪ User A sends a REGISTER message to SAS with transport=TCP. ▪ User B sends an INVITE message to A using SAS. In this scenario, the SAS application forwards the INVITE request using the TCP connection that User A initially opened with the REGISTER message.
Web/OVOC: Enable Record-Route **[SASEnableRecordRoute]**	Determines whether the device's SAS application adds the SIP Record-Route header to SIP requests. This ensures that SIP messages traverse the device's SAS agent by including the SAS IP address in the Record-Route header. ▪ **[0]** Disable (default) ▪ **[1]** Enable The Record-Route header is inserted in a request by a SAS proxy to force future requests in the dialog session to be routed through the SAS agent. Each traversed proxy in the path can insert this header, causing all future dialogs in the session to pass through it as well. When this feature is enabled, the SIP Record-Route header includes the URI "lr" parameter, indicating loose routing, for example: `Record-Route: <sip:server10.biloxi.com;lr>`
Web: SAS Proxy Set OVOC: Proxy Set **[SASProxySet]**	Defines the Proxy Set (index number) used in SAS Normal mode to forward REGISTER and INVITE requests from users that are served by the SAS application. The valid range is 0 to 5. The default is 0 (i.e., default Proxy Set).
Web: Redundant SAS Proxy Set OVOC: Redundant Proxy Set **[RedundantSASProxySet]**	Defines the Proxy Set (index number) used in SAS Emergency mode for fallback when the user is not found in the Registered Users database. Each time a new SIP request arrives, the SAS application checks whether the user is listed in the registration database. If the user is located in the database, the request is sent to the user. If the user is not found, the request is forwarded to the next redundant SAS defined in the Redundant SAS Proxy Set. If that SAS Proxy IP appears in the Via header of the request, it is not forwarded (thereby, preventing loops in the request's course). If no such redundant SAS exists, the SAS sends the request to its default gateway (configured by the parameter SASDefaultGatewayIP). The valid range is -1 to 5. The default is -1 (i.e., no redundant Proxy Set).

Parameter	Description
Web/OVOC: SAS Block Unregistered Users **[SASBlockUnRegUsers]**	Determines whether the device rejects SIP INVITE requests received from unregistered SAS users. This applies to SAS Normal and Emergency modes. ▪ **[0]** Un-Block = (Default) Allow INVITE from unregistered SAS users. ▪ **[1]** Block = Reject dialog-establishment requests from un-registered SAS users.
[SASEnableContactReplace]	Enables the device to change the SIP Contact header so that it points to the SAS host and therefore, the top-most SIP Via header and the Contact header point to the same host. ▪ **[0]** (default) = Disable - when relaying requests, the SAS agent adds a new Via header (with the SAS IP address) as the top-most Via header and retains the original Contact header. Thus, the top-most Via header and the Contact header point to different hosts. ▪ **[1]** = Enable - the device changes the Contact header so that it points to the SAS host and therefore, the top-most Via header and the Contact header point to the same host. **Note:** Operating in this mode causes all incoming dialog requests to traverse the SAS, which may cause load problems.
Web: SAS Survivability Mode OVOC: Survivability Mode **[SASSurvivabilityMode]**	Determines the Survivability mode used by the SAS application. ▪ **[0]** Standard = (Default) Incoming INVITE and REGISTER requests are forwarded to the defined Proxy list of SASProxySet in Normal mode and handled by the SAS application in Emergency mode. ▪ **[1]** Always Emergency = The SAS application does not use Keep-Alive messages towards the SASProxySet, instead it always operates in Emergency mode (as if no Proxy in the SASProxySet is available). ▪ **[2]** Ignore Register = Use regular SAS Normal/Emergency logic (same as option **[0])**, but when in Normal mode incoming REGISTER requests are ignored. ▪ **[3]** Auto-answer REGISTER = When in Normal mode, the device responds to received REGISTER requests by sending a SIP 200 OK (instead of relaying the registration requests to a Proxy), and enters the registrations in its SAS database. ▪ **[4]** Use Routing Table only in Normal mode = The device uses the IP-to-IP Routing table to route IP-to-IP SAS calls only when in SAS Normal mode (and is unavailable when SAS is in Emergency mode). This allows routing of SAS IP-to-IP calls to different destinations (and not only to the SAS Proxy Set).
[SASEmergencyModeRouteRegister]	Enables the device to use the SAS IP-to-IP Routing table to route incoming SIP REGISTER messages from SIP User Agents (UA) located within the Enterprise, during SAS Emergency state.

Parameter	Description
	▪ **[0]** = (Default) Disabled. The device accepts REGISTER messages, adds the UA (Contact and AOR) to its internal SAS registration database, and sends a SIP 200 OK to the UA. ▪ **[1]** = Enabled. During Emergency state, the device uses the SAS IP-to-IP Routing table to route incoming REGISTER messages to, for example, a different proxy server. This option is applicable if the 'SAS Survivability Mode' parameter is configured to any value other than 'Use Routing Table only in Normal mode'. If no matching routing rule is found in the IP-to-IP Routing table, the device accepts the REGISTER message (as described when the SASEmergencyModeRouteRegister parameter is configured to 0).
Web: SAS Subscribe Response **[SASSubscribeResponse]**	Defines the SIP response upon receipt of a SUBSCRIBE message when SAS is in Emergency mode. For example, if this parameter is set to "200", then SAS sends a SIP 200 OK in response to a SUBSCRIBE message, when in Emergency mode. The valid value is 200 to 699. The default is 489.
Web: Enable ENUM **[SASEnableENUM]**	Enables SAS to perform ENUM (E.164 number to URI mapping) queries when receiving INVITE messages in SAS emergency mode. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: SAS Binding Mode OVOC: Binding Mode **[SASBindingMode]**	Determines the SAS application database binding mode. ▪ **[0]** URI = (Default) If the incoming AoR in the INVITE requests is using a 'tel:' URI or 'user=phone' is defined, the binding is performed according to the user part of the URI only. Otherwise, the binding is according to the entire URI, i.e., User@Host. ▪ **[1]** User Part only = The binding is always performed according to the User Part only.
Web: SAS Emergency Numbers **[SASEmergencyNumbers]**	Defines emergency numbers for the device's SAS application. When the device's SAS agent receives a SIP INVITE (from an IP phone) that includes one of the emergency numbers (in the SIP user part), it forwards the INVITE to the default gateway (configured by the parameter SASDefaultGatewayIP), i.e., the device itself, which sends the call directly to the PSTN. This is important for routing emergency numbers such as 911 (in North America) directly to the PSTN. This is applicable to SAS operating in Normal and Emergency modes. Up to four emergency numbers can be defined, where each number can be up to four digits.
[SASEmergencyPrefix]	Defines a prefix that is added to the Request-URI user part of the INVITE message that is sent by the device's SAS agent when in Emergency mode to the default gateway or to any other destination (using the IP-to-IP Routing table). This parameter is required to differentiate between normal SAS calls routed to the default gateway and emergency

Parameter	Description
	SAS calls. Therefore, this allows you to define different manipulation rules for normal and emergency calls.
	This valid value is a character string. The default is an empty "" string.
Web: SAS Entering Emergency Mode **[SASEnteringEmergencyMode]**	Determines for which sent SIP message types the device enters SAS Emergency mode if no response is received for them from the proxy server.
	▪ **[0]** = (Default) SAS enters Emergency mode only if no response is received from sent SIP OPTIONS messages.
	▪ **[1]** = SAS enters Emergency mode if no response is received from sent SIP OPTIONS, INVITE, or REGISTER messages.
	Note: If the keep-alive mechanism is disabled for the Proxy Set (in the Proxy Set table) and this parameter is set to [1], SAS enters Emergency mode only if no response is received from sent INVITE or REGISTER messages.
sas-indialog-mode [SASInDialogRequestMode]	Defines how the device sends incoming SIP dialog requests received from users when not in SAS Emergency mode.
	▪ **[0]** = (Default) Send according to the SIP Request-URI.
	▪ **[1]** = Send to Proxy server.
Web: SAS Inbound Manipulation Mode **[SASInboundManipulationMode]**	Enables destination number manipulation of incoming INVITE messages when SAS is in Emergency mode. The manipulation rule is done in the IP to IP Inbound Manipulation table.
	▪ **[0]** None (default)
	▪ **[1]** Emergency Only
	Notes:
	▪ Inbound manipulation applies only to INVITE requests.
	▪ For more information on SAS inbound manipulation, see 'Manipulating Destination Number of Incoming INVITE' on page 343.
SAS Registration Manipulation Table	
Web: SAS Registration Manipulation OVOC: Stand-Alone Survivability **[SASRegistrationManipulation]**	This table parameter configures the SAS Registration Manipulation table. This table is used by the SAS application to manipulate the SIP Request-URI user part of incoming INVITE messages and of incoming REGISTER request AoR (To header), before saving it to the registered users database. The format of this table parameter is as follows:
	[SASRegistrationManipulation] FORMAT SASRegistrationManipulation_Index = SASRegistrationManipulation_RemoveFromRight, SASRegistrationManipulation_LeaveFromRight, SASRegistrationManipulation_RuleApplyTo; [\SASRegistrationManipulation]
	For example, the manipulation rule below routes an INVITE with Request-URI header

Parameter	Description
	"sip:7184002@10.33.4.226" to user "4002@10.33.4.226" (i.e., keep only four digits from right of user part): `SASRegistrationManipulation 0 = 0, 4, 2;` **Note:** For a detailed description of this table, see 'Manipulating URI user part of Incoming REGISTER' on page 342.
Web: SAS IP-to-IP Routing Table	
[IP2IPRouting]	This table parameter configures the IP-to-IP Routing table for SAS routing rules. The format of this parameter is as follows: [IP2IPRouting] FORMAT IP2IPRouting_Index = IP2IPRouting_SrcIPGroupID, IP2IPRouting_SrcUsernamePrefix, IP2IPRouting_SrcHost, IP2IPRouting_DestUsernamePrefix, IP2IPRouting_DestHost, IP2IPRouting_DestType, IP2IPRouting_DestIPGroupID, IP2IPRouting_DestSRDID, IP2IPRouting_DestAddress, IP2IPRouting_DestPort, IP2IPRouting_DestTransportType, IP2IPRouting_AltRouteOptions; [\IP2IPRouting] For example: IP2IPRouting 1 = -1, *, *, *, *, 0, -1, -1, , 0, -1, 0; **Note:** For a detailed description of this table parameter, see 'SAS Routing Based on IP-to-IP Routing Table' on page 346.

43.14 Auxiliary and Configuration File Name Parameters

The configuration files (i.e., auxiliary files) can be loaded to the device using the Web interface or a TFTP session. For loading these files using the *ini* file, you need to configure these files in the *ini* file and configured whether they must be stored in the non-volatile memory. The table below lists the *ini* file parameters associated with these auxiliary files. For more information on the auxiliary files, see 'Loading Auxiliary Files' on page 365.

Table 43-64: Auxiliary and Configuration File Parameters

Parameter	Description
General Parameters	
[SetDefaultOnIniFileProcess]	Determines if all the device's parameters are set to their defaults before processing the updated *ini* file. ▪ **[0]** = Disable - parameters not included in the downloaded *ini* file are not returned to default settings (i.e., retain their current settings). ▪ **[1]** = Enable (default). **Note:** This parameter is applicable only for automatic HTTP update or Web *ini* file upload (not applicable if the *ini* file is loaded using BootP).
[SaveConfiguration]	Determines if the device's configuration (parameters and files) is saved to flash (non-volatile memory).

Parameter	Description
	▪ **[0]** = Configuration isn't saved to flash memory. ▪ **[1]** = (Default) Configuration is saved to flash memory.
Auxiliary and Configuration File Name Parameters	
Web/OVOC: Call Progress Tones File **[CallProgressTonesFilename]**	Defines the name of the file containing the Call Progress Tones definitions. For more information on how to create and load this file, refer to *DConvert Utility User's Guide*. **Note:** For this parameter to take effect, a device reset is required.
Web/OVOC: Prerecorded Tones File **[PrerecordedTonesFileName]**	Defines the name (and path) of the file containing the Prerecorded Tones. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ PRT is not supported by MP-124 Rev. E.
Web: Dial Plan File OVOC: Dial Plan File Name **[DialPlanFileName]**	Defines the name (and path) of the Dial Plan file. This file should be created using AudioCodes DConvert utility (refer to *DConvert Utility User's Guide*).
[UserInfoFileName]	Defines the name (and path) of the file containing the User Information data.

43.15 Automatic Update Parameters

The automatic update of software and configuration files parameters are described in the table below.

Table 43-65: Automatic Update of Software and Configuration Files Parameters

Parameter	Description
General Automatic Update Parameters	
[AutoUpdateCmpFile]	Enables the Automatic Update mechanism for the cmp file. ▪ **[0]** = (Default) The Automatic Update mechanism doesn't apply to the cmp file. ▪ **[1]** = The Automatic Update mechanism includes the cmp file. **Note:** For this parameter to take effect, a device reset is required.
[AutoUpdateFrequency]	Defines the number of minutes that the device waits between automatic updates. The default is 0 (i.e., the update at fixed intervals mechanism is disabled). **Note:** For this parameter to take effect, a device reset is required.
[AutoUpdatePredefinedTime]	Defines schedules (time of day) for automatic updates. The format of this parameter is: 'HH:MM', where *HH* denotes the hour and *MM* the minutes, for example, 20:18. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The actual update time is randomized by five minutes to reduce the load on the Web servers.

Parameter	Description
[AupdHttpUserAgent]	Defines the information sent in the HTTP User-Agent header in the HTTP Get requests sent by the device to the provisioning server for the Automatic Update mechanism.
	The valid value is a string of up to 511 characters. The information can include any user-defined string or the following string variable tags (case-sensitive):
	▪ <NAME>: product name, according to the installed Software License Key
	▪ <MAC>: device's MAC address
	▪ <VER>: software version currently installed on the device, e.g., "7.00.200.001"
	▪ <CONF>: configuration version, as configured by the ini file parameter, INIFileVersion or CLI command, configuration-version
	The device automatically populates these tag variables with actual values in the sent header. By default, the device sends the following in the User-Agent header:
	```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)```
	Notes:
	▪ The variable tags are case-sensitive.
	▪ If you configure the parameter with the <CONF> variable tag, you must reset the device with a burn-to-flash for your settings to take effect.
	▪ The tags can be defined in any order.
	▪ The tags must be defined adjacent to one another (i.e., no spaces).
OVOC: AUPD Verify Certificates **[AUPDVerifyCertificates]**	Determines whether the Automatic Update mechanism verifies server certificates when using HTTPS.
	▪ **[0]** = Disable (default)
	▪ **[1]** = Enable
[AUPDCheckIfIniChanged]	Determines whether the Automatic Update mechanism performs CRC checking to determine if the *ini* file has changed prior to processing.
	▪ **[0]** = (Default) Do not check CRC. The *ini* file is loaded whenever the server provides it.
	▪ **[1]** = Check CRC for the entire file. Any change, including line order, causes the *ini* file to be re-processed.
	▪ **[2]** = Check CRC for individual lines. Use this option when the HTTP server scrambles the order of lines in the provided *ini* file.
[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.

Parameter	Description
	▪ **[1]** = The device immediately resets after an *ini* file with this parameter set to 1 is loaded.
Software/Configuration File URL Path for Automatic Update Parameters	
[CmpFileURL]	Defines the name of the *cmp* file and the path to the server (IP address or FQDN) from where the device can load the *cmp* file and update itself. The *cmp* file can be loaded using HTTP/HTTPS, FTP, FTPS, or NFS. For example: http://192.168.0.1/filename **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When this parameter is configured, the device always loads the *cmp* file after it is reset. ▪ The *cmp* file is validated before it's burned to flash. The checksum of the *cmp* file is also compared to the previously burnt checksum to avoid unnecessary resets. ▪ The maximum length of the URL address is 255 characters.
[IniFileURL]	Defines the name of the *ini* file and the path to the server (IP address or FQDN) on which it is located. The *ini* file can be loaded using HTTP/HTTPS, FTP, FTPS, or NFS. For example: http://192.168.0.1/filename http://192.8.77.13/config<MAC> https://<username>:<password>@<IP address>/<file name> **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When using HTTP or HTTPS, the date and time of the *ini* file are validated. Only more recently dated *ini* files are loaded. ▪ The optional string <MAC> is replaced with the device's MAC address. Therefore, the device requests an *ini* file name that contains its MAC address. This option allows the loading of specific configurations for specific devices. ▪ The maximum length of the URL address is 99 characters.
[AUPDResetURLOnWebConfig]	Defines if the URLs configured for the **CmpFileURL** and **IniFileURL** parameters are deleted when you reset the device with burn to flash, through Web interface. ▪ **[0]** = URLs are maintained. ▪ **[1]** = (Default) URLs are deleted (as device assumes that you want to configure the device manually instead of using auto update). **Note:** If **IniFileURL** is configured, the "Burn to Flash" field is set to **No** by default (instead of **Yes**) .
[PrtFileURL]	Defines the name of the Prerecorded Tones (PRT) file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file. **Notes:** ▪ The maximum length of the URL address is 99 characters. ▪ PRT is not supported by MP-124 Rev. E.

Parameter	Description
[CptFileURL]	Defines the name of the CPT file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file. **Note:** The maximum length of the URL address is 99 characters.
[TLSRootFileUrl]	Defines the name of the TLS trusted root certificate file and the URL from where it can be downloaded. **Note:** For this parameter to take effect, a device reset is required.
[TLSCertFileUrl]	Defines the name of the TLS certificate file and the URL from where it can be downloaded. **Note:** For this parameter to take effect, a device reset is required.
[TLSPkeyFileUrl]	Defines the URL for downloading a TLS private key file using the Automatic Update facility.
[UserInfoFileURL]	Defines the name of the User Information file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file **Note:** The maximum length of the URL address is 99 characters.

44   DSP Templates

The tables below list the maximum supported channel capacity.

> **Notes:**
>
> - To select the DSP Template that you want to use on the device, see Configuring DSP Templates on page 177.
> - Installation and use of voice coders is subject to obtaining the appropriate license and royalty payments.
> - The number of channels refers to the maximum channel capacity of the device.
> - The G.727 coder is not supported by MP-124 Rev. E.
> - For additional DSP templates, contact your AudioCodes representative.

Table 44-1: Maximum Channel Capacity for MP-11x and MP-124 Rev. D

	DSP Template			
	0		1	
	Maximum Channels			
Model	Default (no SRTP	SRTP Enabled	Default (no SRTP	SRTP Enabled
MP-112 FXS/FXO	2	2	2	2
MP-114 FXS/FXO	4	3	3	3
MP-118 FXS/FXO	8	6	6	6
MP-124 Rev. D	24	18	18	18
Voice Coder				
G.711 A/Mu-law PCM	√	√	√	√
G.726 ADPCM	√	√	√	√
G.727 ADPCM	√	√	√	√
G.723.1	√	√	√	√
G.729 A, B	√	√	√	√
EG.711	√	√	-	-
G.722	-	-	√	√

Table 44-2: Maximum Channel Capacity for MP-124 Rev. E

Voice Coder	Maximum Channels	
	Default (no SRTP)	SRTP Enabled
G.711 A/Mu-law PCM	24	17
G.726 ADPCM	24	17
G.723.1	24	17
G.729 A, B	24	17
G.722	21	16

45   Selected Technical Specifications

The main technical specifications of the MP-11x and MP-124 devices are listed in the table below.

> **Notes:**
>
> - All specifications in this document are subject to change without prior notice.
> - The compliance and regulatory information can be downloaded from AudioCodes website at https://www.audiocodes.com/library?categoryGroup=Compliance+and+Regulatory+Information.

Table 45-1: MediaPack Technical Specifications

Function	Specification
Interfaces	
Voice Ports	▪ MP-112: 2 ports ▪ MP-114: 4 ports ▪ MP-118: 8 ports ▪ MP-124: 24 ports
Telephone Interfaces	▪ MP-112: FXS, RJ-11 ▪ MP-114 & MP-118: FXS, FXO or mixed FXS/FXO, RJ-11 ▪ MP-124: FXS, 50-pin Telco
Lifeline	Automatic cut through of a single analog line (FXS version only, refers only for the middle column – 4/8 ports)
Network Interface	10/100Base-TX, RJ-45
Indicators Channel	Status and activity LEDs
Voice, Fax, Modem	
Voice over Packet Capabilities	G.168-2004 compliant Echo Cancellation, VAD, CNG, Dynamic programmable Jitter Buffer, modem detection and auto switch to PCM
Voice Compression	G.711, G.723.1, G.726 ADPCM, G.727 ADPCM, G.729A/B, G.722 **Note:** The G.727 is currently not supported by MP-124 Rev. E.
Fax over IP	T.38 compliant Group 3 fax relay up to 14.4 kbps with automatic switching to PCM or ADPCM
3-Way Conference	3-Way conference with local mixing
Quality Enhancement	DiffServ, TOS, 802.1 P/Q VLAN tagging, RTCP XR
IP Transport	RTP/RTCP per IETF RFC 3550 and 3551, Multiplexing (aggregated RTP streams of several channels for saving network bandwidth)
Stand Alone Survivability (SAS) Application	
Max. Registered Users	SAS ensures call continuity between LAN SIP clients upon connectivity failure with IP Centrex services (e.g., WAN IP PBX).

Function	Specification
Capacity	
Registered Users	25
Signaling	
Signaling	▪ MP-112: FXS Loop-start ▪ MP-114 & MP-118: FXS, FXO Loop-start ▪ MP-124: FXS Loop-start
In-band Signaling	DTMF (TIA 464B) User-defined and call progress tones
Out-of-Band Signaling	DTMF Relay (RFC 2833), DTMF via SIP INFO
Control	SIP (RFC 3261)
Provisioning	
Protocols	▪ BootP, DHCP, TFTP and HTTP for Automatic Installation ▪ DHCP options 66,67 in auto update mode ▪ Remote management using Web browser ▪ OVOC / SNMP V3 ▪ Syslog support ▪ RS-232 for basic configuration (via CLI) ▪ Voice Menu using touch-tone phone (FXS interface) for basic configuration
Security	
Media	SRTP
Control	H.235, IPSec, TLS/SIPS
Management	HTTPS, Access List, IPSec
Physical	
Power	Single universal power supply 100-240V 0.3A max. 50-60 Hz or -48V DC **Note:** -48V DC is supported only on MP-124D.
Environmental	▪ Operational: 5 to 40°C (41 to 104°F) ▪ Storage: -25 to 85°C (-13 to 185°F) ▪ Humidity: 10 to 90% non-condensing
Dimensions	▪ MP-112: 42 x 172 x 220 mm ▪ MP-114 & MP-118: 42 x 172 x 220 mm ▪ MP-124: 44 x 445 x 269 mm
Weight	▪ MP-1xx: 0.5 kg (1.1 lbs.) approx. ▪ MP-124: 1.8 kg (4 lbs.)
Mounting	Rack mount, Table top, Wall mount

Function	Specification
Additional Features	
Message Waiting Indication	High (Neon) and Low (LED) Voltage, FSK, Stutter Dial Tone
High Availability	**PSTN Fallback:** Support of PSTN fallback due to Power failure, if the IP connection is down or due to customer defined IP QoS thresholds **Stand Alone Survivability (SAS):** Supports SAS of up to 25 SIP users (UA)
Ring voltage	Sine: 54 V RMS typical (balanced ringing only)
Ring Frequency	25-100Hz
Maximum Ringer Load	Ringer Equivalency Number (REN) 3
Loop Impedance (including phone impedance)	Up to 1500 ohm for the MP-11x, Up to 1600 ohm for the MP-124
Off-hook Current	▪ MP-11x: 20/25/32 mA (Note: For MP-118, 25/32 mA is supported only on the first 4 ports) ▪ MP-124: 20 mA (minimum), 20/25/32 mA on first 4 ports
Lifeline	Supported in all ports of Mixed FXS/FXO and in first port of MP-114/FXS and MP-118/FXS using special Lifeline cable
Caller ID	Bellcore GR-30-CORE Type 1 using Bell 202 FSK modulation, ETSI Type 1, NTT, Denmark, India, Brazil, British and DTMF ETSI CID (ETS 300-659-1)
Polarity Reversal / Wink	Immediate or smooth to prevent erroneous ringing
Metering Tones	12/16 KHz sinusoidal bursts, Generation on FXS
Distinctive Ringing	By frequency (15-100 Hz) and cadence patterns
Over-voltage protection and surge immunity	Routing of FXS telephony cables outdoors can be done only in conjunction with AudioCodes' approved primary surge protector and proper installation and grounding. When done correctly, the installation will meet ITU-T K.21 (basic) standards.

International Headquarters
1 Hayarden Street,
Airport City
Lod 7019900, Israel
Tel: +972-3-976-4000
Fax: +972-3-976-4040

AudioCodes Inc.
80 Kingsbridge Rd
Piscataway, NJ 08854, USA
Tel: +1-732-469-0880
Fax: +1-732-469-2298

Contact us: https://www.audiocodes.com/corporate/offices-worldwide
Website: https://www.audiocodes.com

Document #: LTRT-65456