User's Manual

*AudioCodes Family of MediaPack™ Analog Media Gateways*

# MediaPack 1288

## High-Density Analog Media Gateway

Version 7.0


HD VoIP Sounds Better

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: May-16-2018

## WEEE EU Directive

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

## Customer Support

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

## Abbreviations and Terminology

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

## General Notes, Warnings, and Safety Information

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

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

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

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

**Notes:**

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

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

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

# Document Revision Record

<table>
<tr><th>LTRT</th><th>Description</th></tr>
<tr><td>28040</td><td>Initial document release for Ver. 7.0.</td></tr>
<tr><td>28041</td><td>Front page photo<br>Updated sections: Default OAMP IP Address (IP addresses); Accessing the Web Interface (autocomplete login); Advanced User Accounts Configuration (password); CLI-Based Management (user access); Understanding Configuration Modes (note); Configuring Underlying Ethernet Devices (max.); First Incoming Packet Mechanism (NAT by Signaling); Configuring the Device's LDAP Cache; Centralized Third-Party Routing Server (aupported SIP messages, credentials for authentication); Configuring the SEM Server (port removed); MAC Address Placeholder in Configuration File Name (note); Configuring RTCP XR (PublicationIPGroupID); Maintaining Same Syslog SID/BID over Multiple Devices (removed); Configuring Test Call Endpoints (play DTMF removed).<br>New sections: Refreshing the LDAP Cache; Clearing the LDAP Cache<br>Updated parameters: CallSetupRules_ActionType (enums); IpProfile_MediaIPVersionPreference; NATMode (Nat by Signaling); SendAcSessionIDHeader (removed); QOEPort (removed); MaxGeneratedRegistersRate; RTCPXRESCTransportType (removed); RTCPXREscIP (removed); RTCPXRReportMode; PublicationIPGroupID<br>New parameters: WebLoginBlockAutoComplete; EnforcePasswordComplexity; GeneratedRegistersInterval</td></tr>
<tr><td>28042</td><td>REN 3 update.<ul><li>Updated: Product Overview; CLI (removed telnet); Configuring TLS Certificate Contexts Configuring Proxy Sets (keep-alive); Viewing and Deleting DHCP Clients; Configuring Dial Plans; FXS Coefficient Types; WebRTC (RFCs); Pre-Configured IP Groups; Normal Mode; Emergency Mode; Auto Answer to Registrations; Software License Key; Software License Key; Viewing Port Status; MAC Address Placeholder in Configuration File Name (note); Viewing Device Information; Configuring CDR Reporting (note); Configuring RADIUS Accounting; Services; Technical Specifications;</li><li>New sections: Customizing the Web Interface; Viewing the License Key; Viewing the Device's Product Key; Viewing Proxy Set Status</li><li>Updated parameters: TLSContexts_ServerCipherString; TLSContexts_ClientCipherString; NATTranslation_SourceStartPort; NATTranslation_SourceEndPort; NATTranslation_TargetStartPort; NATTranslation_TargetEndPort; IPOutboundManipulation_PrivacyRestrictionMode; DialPlans_Name; DialPlanRule_Name; EnableCoreDump; PrackMode (removed); SessionExpiresDisconnectTime; RADIUSRetransmission; RadiusTO: WelcomeMessage; UserInfoFileURL; ProxySet; EnableWebAccessFromAllInterfaces; EnableLanWatchDog; EnableCoreDump; EnableBusyOut; BrokenConnectionEventTimeout;</li><li>New parameters: ProxySet_SuccessDetectionRetries; ProxySet_SuccessDetectionInterval; ProxySet_FailureDetectionRetransmissions; ProxySet_MinActiveServersLB; UseProductName; UserProductName; UseWebLogo; WebLogoText; LogoWidth; LogoFileName; WebFaviconFileUrl; EnableNonCallCdr; UseProductName; UserProductName; UseWebLogo</li><li>Deleted parameters: AMR-WB; LifeLineType</li></ul></td></tr>
</table>

| LTRT | Description |
|---|---|
| 28045 | ▪ Updates sections: Configuring VoIP LAN Interface for OAMP (ports); CLI; Configuring Web User Accounts (typo); Disabling SNMP; Configuring TLS Certificate Contexts; Assigning CSR-based Certificates to TLS Contexts (SHAI); Generating Private Keys for TLS Contexts (4096); Configuring Physical Ethernet Ports; Configuring IP Network Interfaces (note); Configuring Firewall Settings; Assigning IDS Policies; Configuring LDAP Servers (max); Configuring Call Setup Rules; Alternative Routing Based on IP Connectivity; Alternative Routing Based on SIP Responses; Configuring SIP Response Codes for Alternative Routing Reasons; Configuring RADIUS Accounting; Configuring SBC IP-to-IP Routing; Configuring Dial Plans (priority); Creating Core Dump and Debug Files upon Device Crash (reset); Configuring DTMF Tones for Test Calls; Configuring Basic Test Calls; Configuring SBC Test Call with External Proxy (removed)<br>▪ Updated parameters: TLSContexts_ServerCipherString; TLSContexts_ClientCipherString; AccessList_Start_Port; AccessList_End_Port; SIPInterface_InterfaceName; ProxySet_ProxyName; MessageManipulations_ManipulationName; MessagePolicy_Name; TelProfile_ProfileName; IpProfile_SBCUseSilenceSupp; [_ManipulationName; PREFIX_RouteName; PstnPrefix_RouteName; GWRoutingPolicy_Name; SBCAdmissionControl_AdmissionControlName; SBCAdmissionControl_Rate; Classification_ClassificationName; IP2IPRouting_RouteName; SBCRoutingPolicy_Name; IPInboundManipulation_ManipulationName; IPOutboundManipulation_ManipulationName; Test_Call_Play (tone type); EnableWebAccessFromAllInterfaces; ResetWebPassword; DisableSNMP; KeepAliveTrapPort (default); SBCtestID (removed); EnableCoreDump; SSHMaxLoginAttempts; IgnoreAlertAfterEarlyMedia; EnableBusyOut; ECNLPMode; SecureCallsFromIP; AltRoutingTel2IPEnable; IpProfile_SBCPlayHeldTone; ProxySet_IsProxyHotSwap; EnablePChargingVector<br>▪ New parameters: TLSContexts_DTLSVersion; TLSContexts_DHKeySize; CustomerSN; CallSetupRules_QueryTarget |
| 28052 | ▪ Updated sections: Gateway Application; Viewing the Home Page (FXS/FXO LEDs); Disabling Enabling SNMP; Configuring Voice Settings; Configuring NAT Translation per IP Interface; Silence Suppression (removed); Configuring IP-to-Hunt Group Routing Rules; Fax / Modem Transparent Mode; Comfort Noise Generation; Configuring Media (SRTP) Security; SIP-based Media Recording (France URL); Configuring SIP Recording Rules (timestamp); Enabling LDAP Searches for Numbers with Characters; Configuring Call Setup Rules; Configuring Media Realm Extensions; Configuring SIP Message Manipulation (max.); Calls Termination by PBX (silence det. removed); Interworking SIP Early Media; Configuring Call Preemption for SBC Emergency Calls (note); DHCP-based Provisioning (note); Automatic Update from Remote Servers; Viewing Active Alarms (note); Configuring CDR Reporting (note); Configuring RADIUS Accounting (figure)<br>▪ New sections: Configuring Dual Registration<br>▪ Updated parameters: MediaRealmExtension_IPv4IF; MediaRealmExtension_IPv6IF; SRD_SBCOperationMode; ProxySet_ProxyName; ProxySet_EnableProxyKeepAlive; CodersGroup0_Sce; CodersGroupX_Sce; IpProfile_SCE (removed); Configure IpProfile_SBCPlayHeldTone; IpProfile_SBCSDPPtimeAnswer (Preferred Value); IpProfile_SBCPreferredPTime; PstnPrefix_SourceAddress; TelnetServerEnable; DisableSNMP; EnableLanWatchDog (removed); SetupTime; ConnectTime; ReleaseTime; LifeLineType; SyslogOptimization (default): EnableSIPRemoteReset; IsCiscoSCEMode; EnableBusyOut; EnableSilenceCompression (removed); ModemBypasspayloadType; FaxBypassPayloadType (range); removed – EnableSilenceDisconnect / FarEndDisconnectSilencePeriod / FarEndDisconnectSilenceMethod / FarEndDisconnectSilenceThreshold / |

| LTRT | Description |
|---|---|
| | BrokenConnectionDuringSilence; TrunkLifeLineType; BriTEIAssignTrigger; BriTEIRemoveTrigger; UseDisplayNameAsSourceNumber; SBCKeepContactUserinRegister<br>▪ New parameters: CallSetupRules_QueryTarget; IpProfile_SBCAdaptRFC2833BWToVoiceCoderBW; ActiveAlarmTableMaxSize; NoAlarmForDisabledPort; SBCRemoveSIPSFromNonSecuredTransport; TimeZoneFormat; SRTPTunnelingValidateRTPRxAuthentication; SRTPTunnelingValidateRTCPRxAuthentication; HookFlashFromMediaIP; SBCRemoveSIPSFromNonSecuredTransport; SIPRecTimeStamp |

## Documentation Feedback

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

**This page is intentionally left blank.**

# 1 Introduction

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

## 1.1 Product Overview

AudioCodes MediaPack 1288 (MP-1288), hereafter referred to as *device*, is a cost-effective best-of-breed, high density analog media voice-over-IP (VoIP) gateway. The device provides superior voice technology for connecting legacy telephones, fax machines and modems with IP-based telephony networks, as well as for integration with IP PBX systems. It is designed and tested to be fully interoperable with leading softswitches, unified communications (UC) servers and SIP proxies.

The device is designed for carrier environments including 1+1 power supplies and 1+1 Ethernet redundancy, maintaining high voice quality to deliver reliable enterprise VoIP communications. Advanced call routing mechanisms, network voice quality monitoring and survivability capabilities (including PSTN fallback) result in minimum communications downtime.

The device can be deployed for the following applications:

- Enterprise campus deployments
- PSTN emulation for service providers
- Large-scale analog integration with Microsoft Skype for Business environments or other cloud-based or hybrid PBX deployments

The device offers the following main benefits:

- High density analog media gateway supporting up to 288 FXS Ports
- Ideal for large analog deployments for converting voice, fax and modem calls to IP
- Scalable solution with three capacity options: 288, 216 and 144 ports
- Cost-effective - single management interface, single IP, no need to stack and cable multiple small analog gateways
- Reduced footprint - 3U chassis
- Designed for carrier environments, providing high availability with dual Power Supply modules and Ethernet redundancy
- Rich interoperability and partnerships that extend across multiple vendor devices and protocol implementations
- Delivers high service performance and voice quality
- Remote management through HTTP/S-based embedded Web server, CLI, and ini file.

The device offers the following key features:

- Support for advanced voice coders such as NB-AMR and NB-Opus
- Support for SRTP on all channels without capacity hit
- Automatic switching to PSTN via lifeline interfaces upon power failure
- Integrated protection against surge damage on FXS ports (ITU-T K.21 - basic level compliance)
- Supports short and long haul up to 7.5 Km
- Support for emergency / elevator phones that require higher loop current and increased ring voltage
- Rich and Powerful SIP normalization and routing mechanisms for seamless interoperability

- SIP header manipulation
- Extensive fax support including T.38 version 3

## 1.2 Typographical Conventions

This document uses the following typographical conventions to convey information:

**Table 1-1: Typographical Conventions**

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

## 1.3 Getting Familiar with Configuration Concepts and Terminology

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

### 1.3.1 Gateway Application

The objective of your configuration is to enable the device to forward calls between the IP-based endpoints and PSTN-based endpoints. The PSTN-based endpoints are analog FXS endpoints (plain old telephone service or POTS). The IP-based endpoints (SIP entities) can be servers such as SIP proxy servers and IP PBXs, or end users such as LAN IP phones. In the SIP world, the endpoints are referred to as SIP user agents (UA). The UA that initiates the call is referred to as the user agent client (UAC); the UA that accepts the call is referred to as the user-agent server (UAS).

The following table describes the main configuration concepts and terminology.

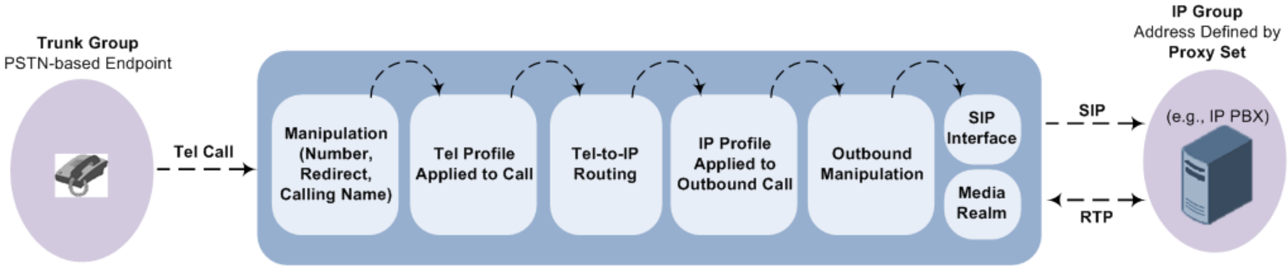
**Table 1-2: Configuration Concepts and Terminology**

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

| Configuration Terms | Description |
|---|---|
| SRDs | The SRD is a logical representation of your entire VoIP network. The SRD is in effect, the foundation of your configuration to which all other previously mentioned configuration entities are associated.<br>Typically, only a single SRD is required and this is the recommended configuration topology. As the device provides a default SRD, in a single SRD topology, the device automatically assigns the SRD to newly created configuration entities. Thus, in such scenarios, there is no need to get involved with SRD configuration.<br>Multiple SRDs are required only for multi-tenant deployments. |
| IP Profiles | The IP Profile is an optional configuration entity that defines a wide range of call settings for a specific SIP entity (IP Group). The IP Profile includes signaling and media related settings, for example, jitter buffer, voice coders, fax signaling method, SIP header support (local termination if not supported), and media security method. The IP Profile is in effect, the interoperability "machine" of the device, enabling communication with SIP endpoints supporting different call "languages".<br>The IP Profile is associated with the SIP entity, by assigning the IP Profile to the IP Group of the SIP entity. |
| Tel Profiles | The Tel Profile is an optional configuration entity that defines a wide range of call settings for a specific PSTN-based endpoint. The IP Profile includes settings such as message waiting indication (MWI), input gain, voice volume and fax signaling method.<br>The Tel Profile is associated with the PSTN-based endpoint, by assigning it to the Hunt Group belonging to the endpoint. |
| Tel-to-IP Routing Rules | Tel-to-IP routing rules are used to route calls from PSTN-based endpoints to an IP destination (SIP entity). The PSTN side can be denoted by a specific Hunt Group, or calling or called telephone number prefix and suffix. The SIP entity can be denoted by an IP Group or other IP destinations such as IP address, FQDN, E.164 Telephone Number Mapping (ENUM service), and Lightweight Directory Access Protocol (LDAP). |
| IP-to-Tel (Hunt Group) Routing Rules | IP-to-Tel routing rules are used to route incoming IP calls to Hunt Groups. The specific channel pertaining to the Hunt Group to which the call is routed can also be configured. |
| Accounts | Accounts are used to register or authenticate PSTN-based endpoints with a SIP entity (e.g., a registrar or proxy server). The device does this on behalf of the PSTN-based endpoint. Authentication (SIP 401) is typically relevant for INVITE messages forwarded by the device to a SIP entity. Registration is for REGISTER messages, which are initiated by the device on behalf of the PSTN-based endpoint. |

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

**Figure 1-1: Gateway Configuration Terminology for Call Processing**



**This page is intentionally left blank.**

# Part I

## Getting Started with Initial Connectivity

# 2 Introduction

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

**This page is intentionally left blank.**

# 3 Default OAMP IP Address

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

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

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

| IP Address | Value |
|---|---|
| Application Type | OAMP + Media + Control |
| IP Address | 192.168.0.2 |
| Prefix Length | 24 (255.255.255.0) |
| Default Gateway | 192.168.0.1 |
| Underlying Device | vlan 1 |
| Interface Name | O+M+C |

**This page is intentionally left blank.**

# 4 Configuring VoIP LAN Interface for OAMP

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

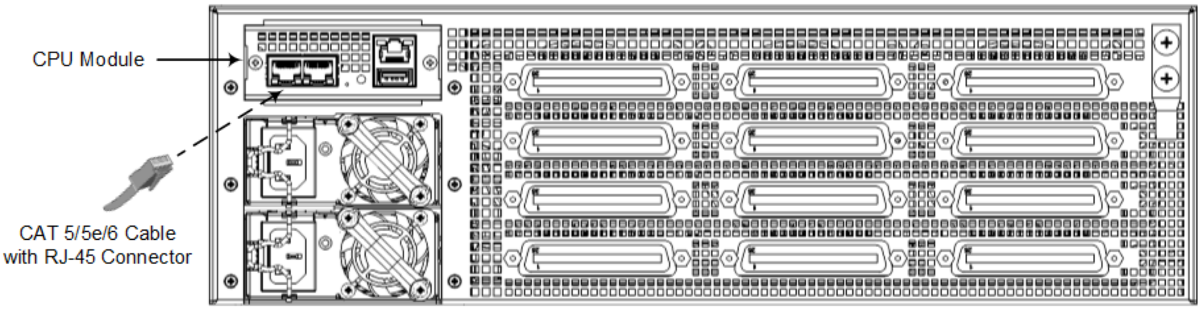
- Embedded HTTP/S-based Web server - see "Web Interface" on page 33
- Embedded command line interface (CLI) - see "CLI" on page 35

## 4.1 Web Interface

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

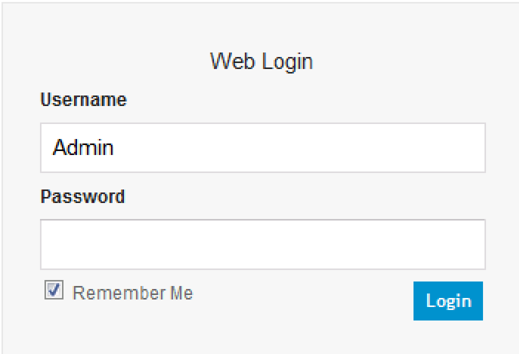
➢ **To configure the VoIP-LAN IP Address for OAMP, using the Web interface:**

1. Connect one of the Ethernet ports on the CPU module (rear panel) directly to the network interface of your computer, using a straight-through Ethernet cable.




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

**Figure 4-1: Web Login Screen**




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

4. Open the Physical Ports Settings page (Configuration tab > VoIP menu > Network > Physical Ports Table) and then configure the device's physical Ethernet port-pair (group) that you want to later assign to the OAMP interface. For more information, see Configuring Physical Ethernet Ports on page 127.
5. Open the Interface table (**Configuration** tab > **VoIP** menu > **Network** > **IP Interfaces Table**).
6. Select the index row corresponding to the **OAMP + Media + Control** application type, and then click **Edit**.
7. Change the IP address to correspond with your network IP addressing scheme, for example:
   - IP Address: 10.8.6.86
   - Prefix Length: 24 (for 255.255.255.0)
   - Gateway: 10.8.6.85
   - Underlying Device: Select the Ethernet Device (VLAN and associated Ethernet port group) for OAMP
8. Click **Add**.
9. Save your settings by resetting the device with a flash burn (see "Resetting the Device" on page 487).
10. Disconnect the device from the PC and cable the device to your network. You can now access the management interface using the new OAMP IP address.

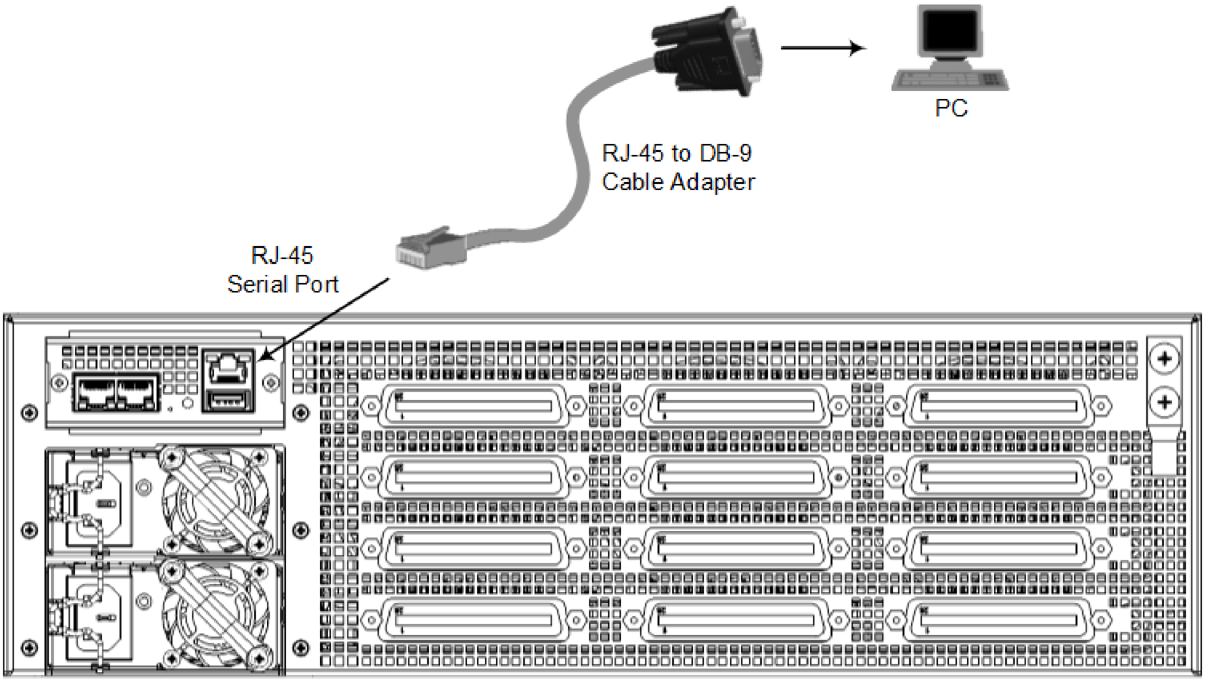
**Note:** When you complete the above procedure, change your PC's IP address to correspond with your network requirements.

## 4.2 CLI

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

➢ **To configure the OAMP IP address in the CLI:**

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




2. Establish serial communication with the device using a terminal emulator program such as HyperTerminal, with the following communication port settings:
   - Baud Rate: 115,200 bps
   - Data Bits: 8
   - Parity: None
   - Stop Bits: 1
   - Flow Control: None
3. At the CLI prompt, type the username (default is "Admin" - case sensitive):
```
Username: Admin
```
4. At the prompt, type the password (default is "Admin" - case sensitive):
```
Password: Admin
```
5. At the prompt, type the following:
```
enable
```
6. At the prompt, type the password again:
```
Password: Admin
```

7. Access the VoIP configuration mode:

```
# configure voip
```

8. Access the Interface table:

```
(config-voip)# interface network-if 0
```

9. Configure the IP address:

```
(network-if-0)# ip-address <IP address>
```

10. Configure the prefix length:

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

11. Configure the Default Gateway address:

```
(network-if-0)# gateway <IP address>
```

12. Apply your settings:

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

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

# Part II

## Management Tools

# 5 Introduction

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

The device provides the following management tools:

- Embedded HTTP/S-based Web server - see "Web-based Management" on page 41
- Command Line Interface (CLI) - see "CLI-Based Management" on page 77
- Simple Network Management Protocol (SNMP) - 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, whenever a parameter is mentioned, its corresponding Web, CLI, and ini file parameter is mentioned. The *ini* file parameters are enclosed in square brackets [...].
- For a list and description of all the configuration parameters, see "Configuration Parameters Reference" on page 643.

**This page is intentionally left blank.**

# 6 Web-Based Management

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

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

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

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

**Notes:**

- The Web interface allows you to configure most of the device's settings. However, additional configuration parameters may exist that are not available in the Web interface and which can only be configured using other management tools.
- Some Web interface pages and/or parameters are available only for certain hardware configurations or software features. The software features are determined by the installed Software License Key (see "Software License Key" on page 510).

## 6.1 Getting Acquainted with the Web Interface

This section provides a description of the Web interface.

### 6.1.1 Computer Requirements

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

- A network connection to the device
- One of the following Web browsers:
  - Microsoft™ Internet Explorer™ (Version 11.0.13 and later)
  - Mozilla Firefox® (Versions 5 through 9.0)
- Recommended screen resolutions: 1024 x 768 pixels, or 1280 x 1024 pixels

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

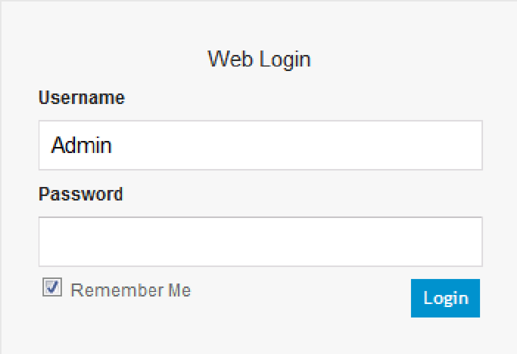
### 6.1.2 Accessing the Web Interface

The following procedure describes how to access the Web interface.

➢ **To access the Web interface:**

1. Open a standard Web browser (see "Computer Requirements" on page 41).
2. In the Web browser, specify the OAMP IP address of the device (e.g., http://10.1.10.10); the Web interface's Login window appears, as shown below**:**

**Figure 6-1: Web Login Screen**




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:**

- By default, Web access is only through the IP address of the OAMP interface. However, you can allow access from all of the device's IP network interfaces, by setting the EnableWebAccessFromAllInterfaces parameter to 1.
- The default login username and password is "Admin". To change the login credentials, see "Configuring the Web User Accounts" on page 65.
- By default, autocompletion of the login username is enabled whereby the 'Username' field offers previously entered usernames. To disable autocompletion, use the WebLoginBlockAutoComplete ini file parameter.
- If you want the Web browser to remember your password, select the 'Remember Me' check box and then agree to the browser's prompt (depending on your browser) to save the password for future logins. On your next login attempt, simply press the Tab or Enter keys to auto-fill the 'Username' and 'Password' fields, and then click **Login**.
- Depending on your Web browser's settings, a security warning box may be displayed. The reason for this is that the device's certificate is not trusted by your PC. The browser may allow you to install the certificate, thus skipping the warning box the next time you connect to the device. If you are using Windows Internet Explorer, click **View Certificate**, and then **Install Certificate**. The browser also warns you if the host name used in the URL is not identical to the one listed in the certificate. To resolve this, add the IP address and host name (ACL_nnnnnn, where *nnnnnn* is the serial number of the device) to your hosts file, located at /etc/hosts on UNIX or C:\Windows\System32\Drivers\ETC\hosts on Windows; then use the host name in the URL (e.g., https://ACL_280152).
  Below is an example of a host file:
  127.0.0.1    localhost
  10.31.4.47   ACL_280152

### 6.1.3 Areas of the GUI

The areas of the Web interface's GUI are shown in the figure below and described in the subsequent table.

**Figure 6-2: Main Areas of the Web Interface GUI**

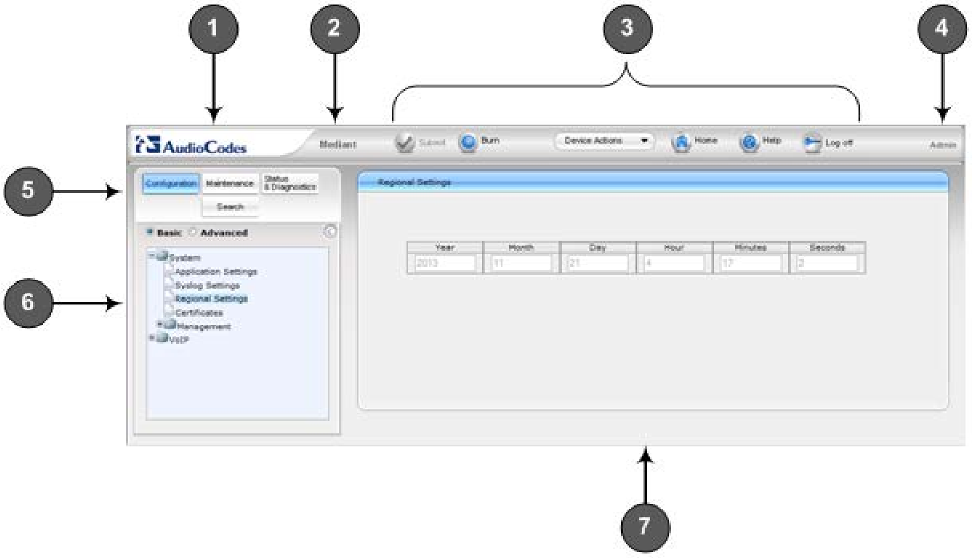


**Table 6-1: Description of the Web GUI Areas**

| Item # | Description |
|---|---|
| **1** | AudioCodes company logo. |
| **2** | Product name. |
| **3** | Toolbar, providing frequently required command buttons. For more information, see "Toolbar Description" on page 44. |
| **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:<br>▪ **Configuration**, **Maintenance**, and **Status & Diagnostics** tabs: Access the configuration menus (see "Working with Configuration Pages" on page 47)<br>▪ **Search** tab: Enables a search engine for searching configuration parameters (see "Searching for Configuration Parameters" on page 54) |
| **6** | Navigation tree, displaying a tree-like structure of elements (configuration menus or search engine) pertaining to the selected tab on the Navigation bar. For more information, see "Navigation Tree" on page 44. |
| **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 47. |

## 6.1.4 Toolbar Description

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

**Table 6-2: Description of Toolbar Buttons**

| Icon | Button Name | Description |
|---|---|---|
| | **Submit** | Applies parameter settings to the device (see "Saving Configuration" on page 490).<br>**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 490). |
| Device Actions | **Device Actions** | Opens a drop-down list with frequently needed commands:<br>▪ **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 517).<br>▪ **Save Configuration File:** Opens the Configuration File page for saving the *ini* file to a folder on your PC (see "Backing Up and Loading Configuration File" on page 517).<br>▪ **Reset:** Opens the Maintenance Actions page for performing various maintenance procedures such as resetting the device (see "Resetting the Device" on page 487).<br>▪ **Software Upgrade Wizard:** Starts the Software Upgrade Wizard for upgrading the device's software (see "Software Upgrade Wizard" on page 513). |
| | **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 57). |
| | **Log off** | Logs off a session with the Web interface (see "Logging Off the Web Interface" on page 58). |
| - | **Reset** | If you modify a parameter on a page that takes effect only after a device reset, after you click the **Submit** button, the toolbar displays "Reset". This is a reminder that you need to later save your settings to flash memory and reset the device. |

## 6.1.5 Navigation Tree

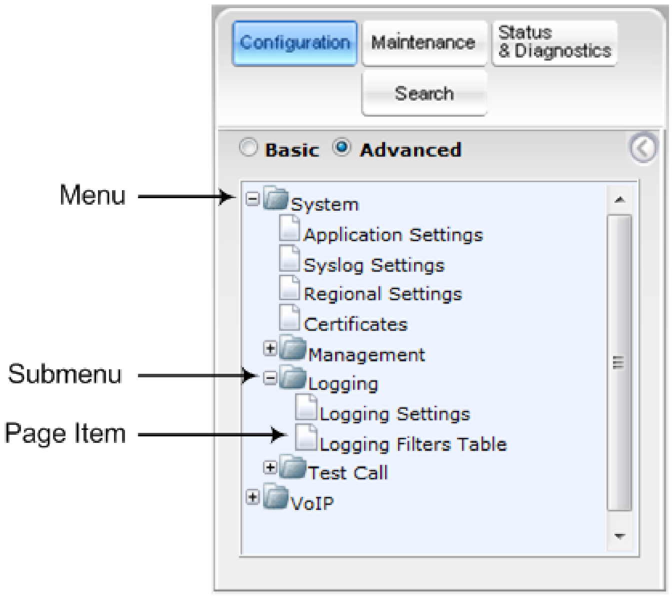
The Navigation tree is located in the Navigation pane and displays a tree-like structure of menus pertaining to the selected tab on the Navigation bar. You can drill-down to the required page item level to open its corresponding page in the Work pane.

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

- *Menu*: first level (highest level)
- *Submenu*: second level - contained within a menu

- *Page item*: last level (lowest level in a menu) - contained within a menu or submenu

**Figure 6-3: Navigating in Hierarchical Menu Tree (Example)**




**Note:** The figure above is used only as an example. The displayed menus depend on supported features based on the Software License Key installed on your device.

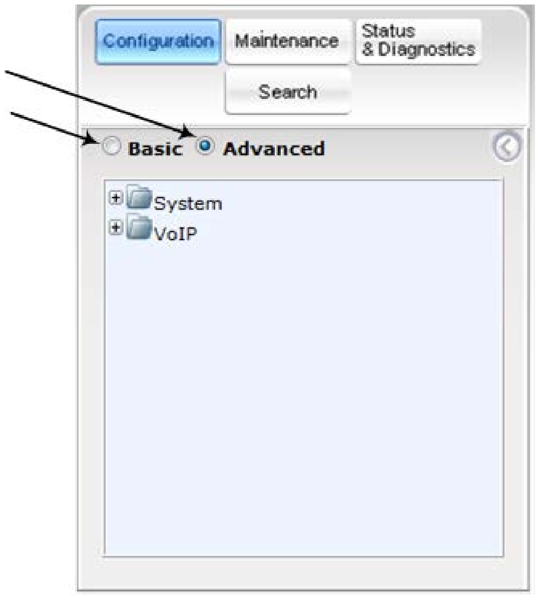
### 6.1.5.1 Displaying Navigation Tree in Basic and Full View

You can view an expanded or reduced display of the Navigation tree. This affects the number of displayed menus and submenus in the tree. The expanded view displays all the menus pertaining to the selected configuration tab; the reduced view displays only commonly used menus.

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

- To display all menus and submenus, select the **Advanced** option.

**Figure 6-4: Basic and Full View Options**



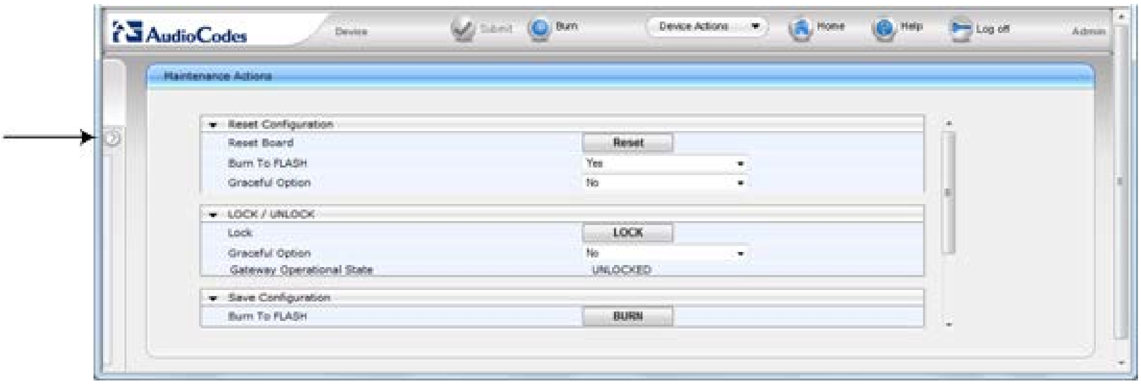

**Note:** After you reset the device, the Web GUI is displayed in **Basic** view.

### 6.1.5.2 Showing / Hiding the Navigation Pane

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

- To hide the Navigation pane, click the left-pointing arrow ; the pane is hidden and the button is replaced by the right-pointing arrow button.
- To show the Navigation pane, click the right-pointing arrow ; the pane is displayed and the button is replaced by the left-pointing arrow button.

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



## 6.1.6 Working with Configuration Pages

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

### 6.1.6.1 Accessing Pages

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

➢ **To open a configuration page:**

1. On the Navigation bar, click the required tab (**Configuration**, **Maintenance**, or **Status & Diagnostics**); the menus pertaining to the selected tab appear in the Navigation tree.
2. Navigate to the required page item, by performing the following:
   - Drill-down using the **plus** ⊞ sign to expand the menu and submenus.
   - Drill-up using the **minus** ⊟ sign to collapse the menu and submenus.
3. Click the required page item; the page opens in the Work pane.

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

> **Note:** Depending on the access level of your Web user account, certain pages may not be accessible or may be read-only (see "Configuring Web User Accounts" on page 65). If a page is read-only, "Read-Only Mode" is displayed at the bottom of the page.

### 6.1.6.2 Viewing Parameters

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

- Displaying "basic" and "advanced" parameters - see "Displaying Basic and Advanced Parameters" on page 47
- Displaying parameter groups - see "Showing / Hiding Parameter Groups" on page 48

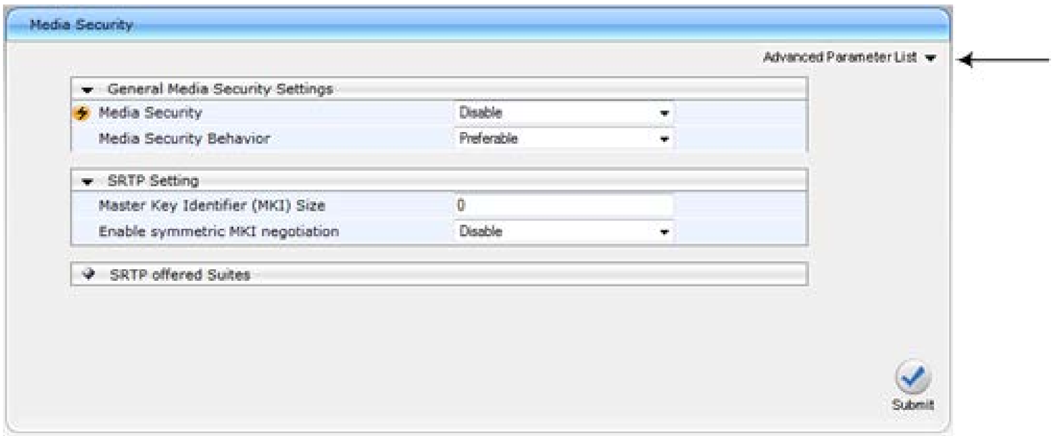
#### 6.1.6.2.1 Displaying Basic and Advanced Parameters

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

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

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

**Figure 6-6: Toggling between Basic and Advanced View**



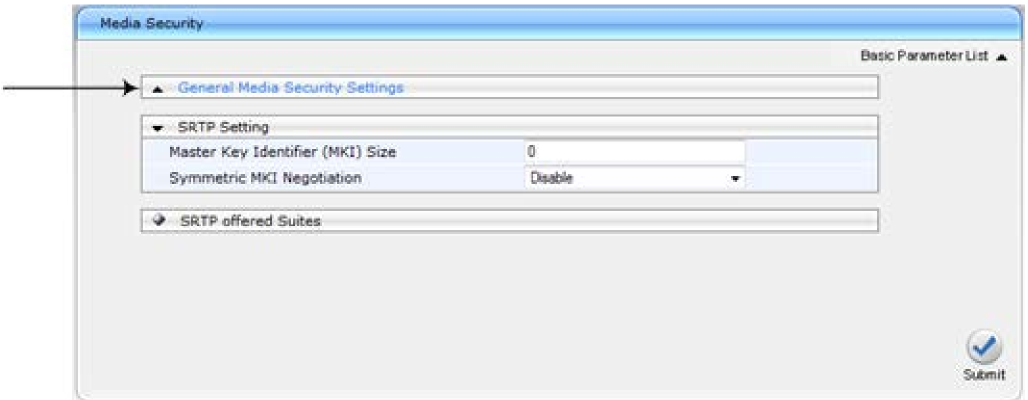

**Notes:**

- When the Navigation tree is in **Advanced** display mode (see "Navigation Tree" on page 44), configuration pages display all their parameters.
- If you reset the device, the Web pages display only the basic parameters.
- The basic parameters are displayed in a different background color to the advanced parameters.

### 6.1.6.2.2 Showing / Hiding Parameter Groups

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

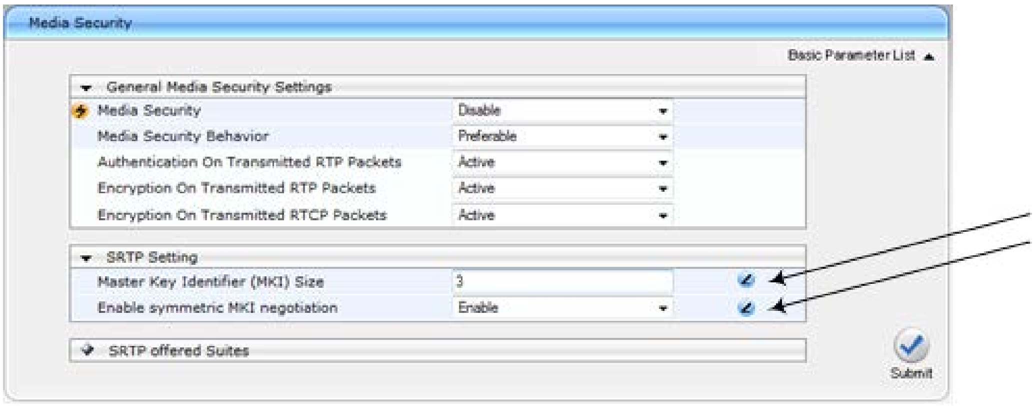
**Figure 6-7: Expanding and Collapsing Parameter Groups**



### 6.1.6.3 Modifying and Saving Parameters

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

**Figure 6-8: Edit Symbol after Modifying Parameter Value**




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

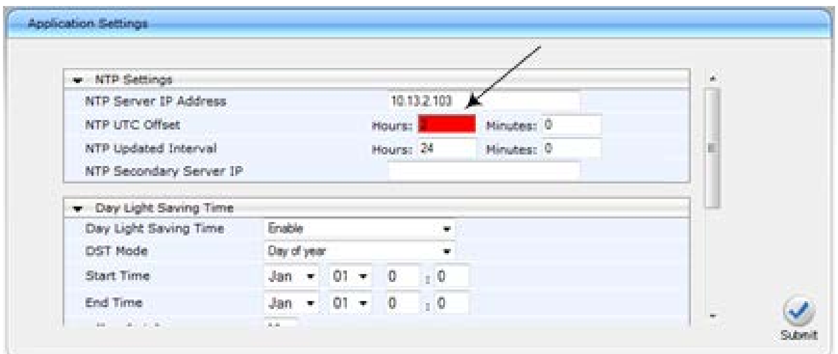
- On the toolbar, click the **Submit** button.
- At the bottom of the page, click the **Submit** button.

When you click **Submit**, modifications to parameters with on-the-fly capabilities are immediately applied to the device and take effect. Parameters displayed on the page with the lightning icon take effect only after a device reset. For resetting the device, see "Resetting the Device" on page 487.

**Note:** Parameters saved to the volatile memory (by clicking **Submit**), revert to their previous settings after a hardware or software reset, or if the device is powered down. Thus, to ensure parameter changes (whether on-the-fly or not) are retained, save ('burn') them to the device's non-volatile memory, i.e., flash (see "Saving Configuration" on page 490).

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

**Figure 6-9: Value Reverts to Previous Valid Value**



### 6.1.7 Working with Tables

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

**Figure 6-10: Displayed Details Pane**

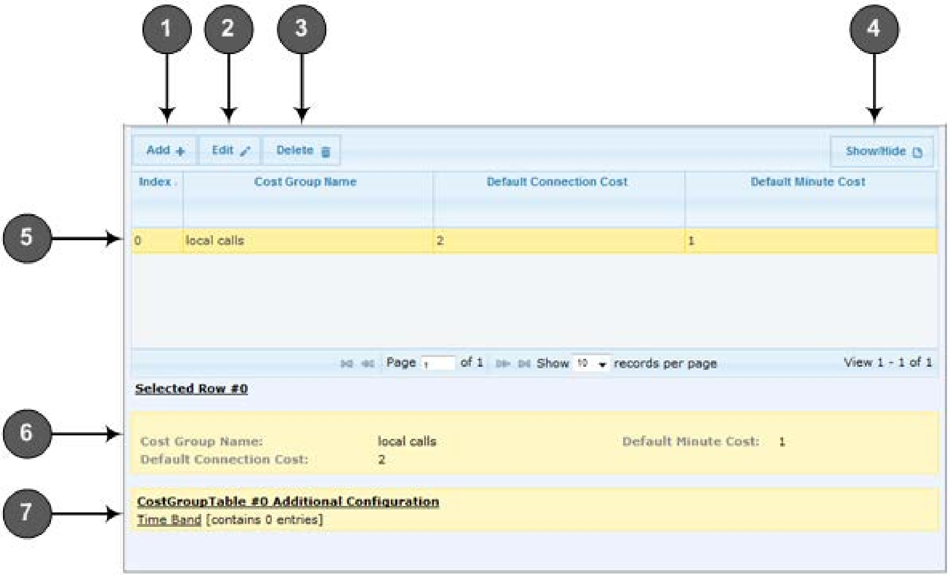



**Table 6-3: Enhanced Table Design Description**

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

#### 6.1.7.1 Table Toolbar Description

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

**Table 6-4: Table Toolbar Description**

| Button | Name | |
|---|---|---|
|   | **Add** | Adds a new index entry row to the table. When you click this button, the Add Row dialog box appears with parameters for configuring the new entry. When you have completed configuration, click the **Add** button in the dialog box to add it to the table. |
|   | **Edit** | Edits the selected row. When you click this button, the Edit Row dialog box appears to modify the row entry. When you have completed configuration, click the **Save** button in the dialog box. |

| Button | Name | |
|---|---|---|
| Delete | **Delete** | Removes the selected row from the table. When you click this button, the Delete Row confirmation box appears requesting you to confirm deletion. Click **Delete** to accept deletion. |
| Show / Hide | **Show / Hide** | Toggles between displaying and hiding the full configuration of a selected row. The configuration is displayed below the table and is useful for tables containing many parameters, which cannot all be displayed in the work pane. |
| Action | **Action** | Provides a drop-down list with commands (e.g., Register and Unregister) relevant to the specific table (e.g., Account table).<br>**Note:** The button only appears in certain tables. |
| Insert + | **Insert** | Adds a new table row at a selected index. You can add the row at any existing (configured) index number. If you select a row and then click the button, after configuring the parameters, the row is automatically added to the index of the selected row and the row that previously occupied the index row and all rows below it are moved one index down in the table. For example, if you select Index 2 and then click the button, the new row is assigned Index 2 and the row previously occupying Index 2 is moved down to Index 3 and so on.<br>**Notes:**<br>▪ The button is available only if the table is sorted according to 'Index' column; otherwise, the button is grayed out. For sorting tables, see "Sorting Tables by Column" on page 53.<br>▪ The button appears only in certain tables. |
| Down | **Down** | Moves a selected row one index down. The index number of the row changes according to its new position in the table. The row that previously occupied the index row and all rows below it are moved one index down in the table.<br>**Notes:**<br>▪ The button is available only if the table contains more than one row and is sorted according to 'Index' column; otherwise, the button is grayed out. For sorting tables, see "Sorting Tables by Column" on page 53.<br>▪ The button appears only in certain tables. |
| Up | **Up** | Moves a selected row one index up. The index number of the row changes according to its new position in the table. The row that previously occupied the index row and all rows below it are moved one index down in the table.<br>**Notes:**<br>▪ The button is available only if the table contains more than one row and is sorted according to 'Index' column; otherwise, the button is grayed out. For sorting tables, see "Sorting Tables by Column" on page 53.<br>▪ The button appears only in certain tables. |
| Clone | **Clone** | Adds a new row with identical settings as the selected row.<br>**Note:** The button only appears in certain tables. |

### 6.1.7.2 Toggling Display Mode of Table Dialog Boxes

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

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

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

### 6.1.7.3 Scrolling through Table Pages

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

**Figure 6-11: Viewing Table Rows per Page**

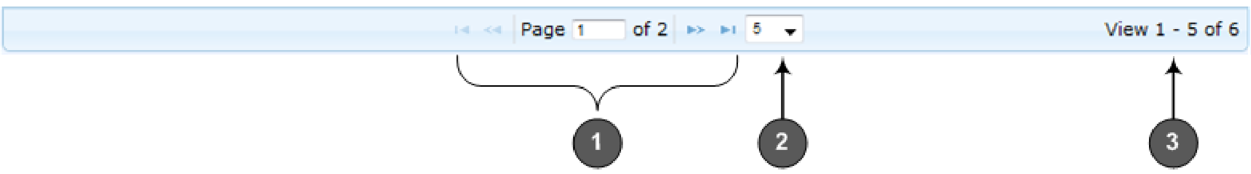


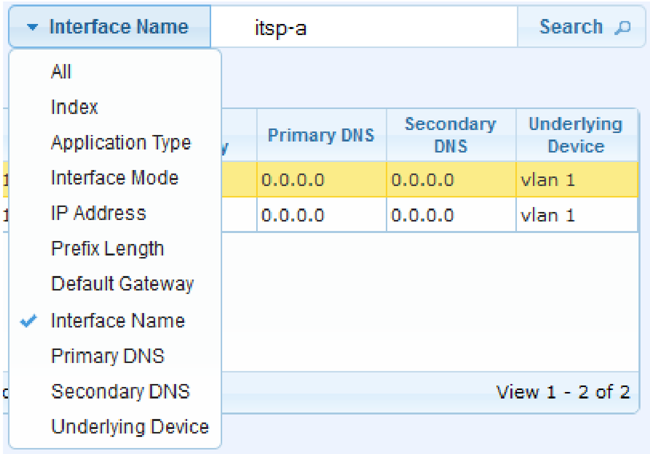

**Table 6-5: Row Display and Page Navigation**

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

### 6.1.7.4 Searching Table Entries

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

**Figure 6-12: Searching Table Entries**




### ➢ To search for a table entry:

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

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

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

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

## 6.1.7.5 Sorting Tables by Column

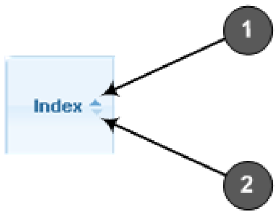
You can sort table rows by any table column and in ascending (e.g., 1, 2 and 3, or a, b, and c) or descending (3, 2, and 1, or c, b, and a) order. For example, instead of the rows being sorted by the Index column in ascending order (e.g., 1, 2, and 3), you can sort the rows by Index column in descending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

### ➢ To sort table rows by column:

1. Click the heading name of the column that you want to sort the table rows by; the up-down arrows appear alongside the heading name and the up button is bolded (see Item

1 in the figure below), indicating that the column is sorted in ascending order:

**Figure 6-13: Sorting Table Rows by Column**




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

### 6.1.8 Searching for Configuration Parameters

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

➢ **To search for a parameter:**

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

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

**Figure 6-14: Searched Result Screen**

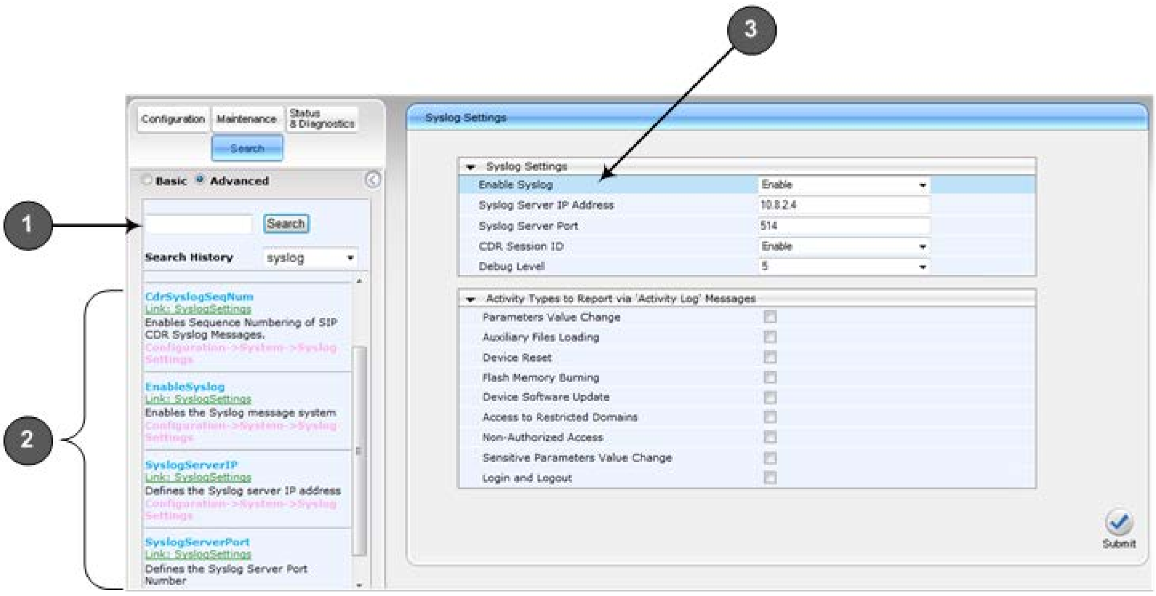


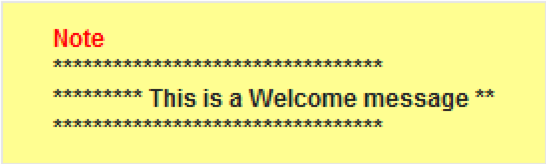

**Table 6-6: Search Description**

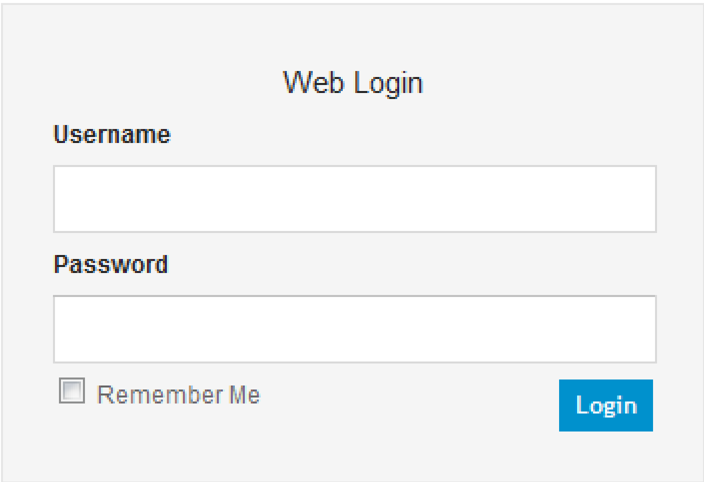
| Item # | Description |
|---|---|
| 1 | Search field for entering search key and **Search** button for activating the search process. |
| 2 | Search results listed in Navigation pane. |
| 3 | Found parameter, highlighted on relevant Web page |

### 6.1.9 Creating a Login Welcome Message

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

**Figure 6-15: User-Defined Web Welcome Message after Login**




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

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

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

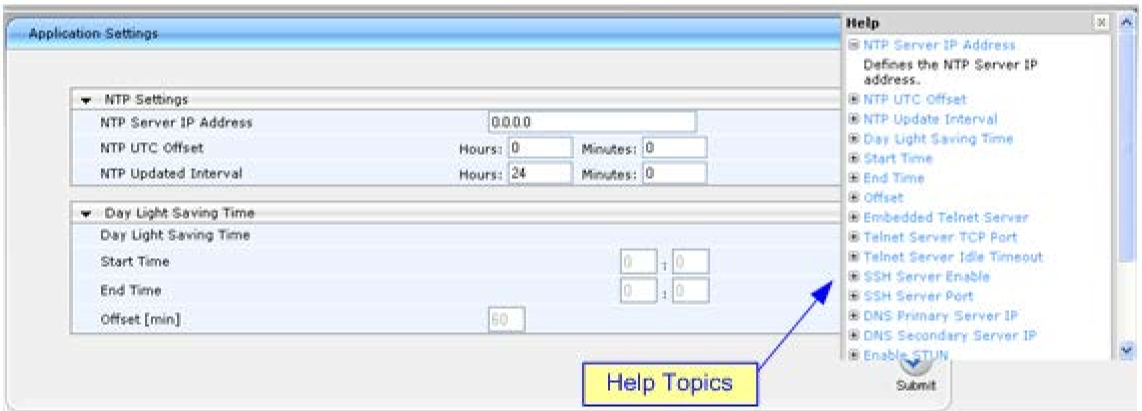
### 6.1.10 Getting Help

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

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

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

**Figure 6-16: Help Topic for Current Page**




2. To view a description of a parameter, click the **plus** ⊞ sign to expand the parameter. To collapse the description, click the **minus** ⊟ sign.
3. To close the Help topic, click the **close** button located on the top-right corner of the Help topic window or simply click the **Help** button.

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

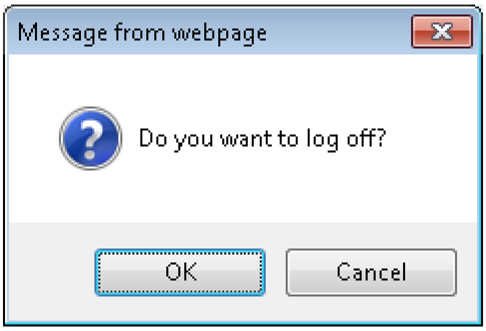
### 6.1.11 Logging Off the Web Interface

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

➢ **To log off the Web interface:**

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

**Figure 6-17: Log Off Confirmation Box**




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

## 6.2 Customizing the Web Interface

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

- Corporate logo (see Replacing the Corporate Logo on page 58)
- Device's (product) name (see Customizing the Product Name on page 60)
- Favicon (see Customizing the Favicon on page 60)
- Login welcome message (see Creating a Login Welcome Message on page 56)

**Note:**

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

### 6.2.1 Replacing the Corporate Logo

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

- Web Login screen
- Menu bar

You can replace the logo with one of the following:

- A different image (see Replacing the Corporate Logo with an Image on page 59)

- Text (see Replacing the Corporate Logo with Text on page 59)

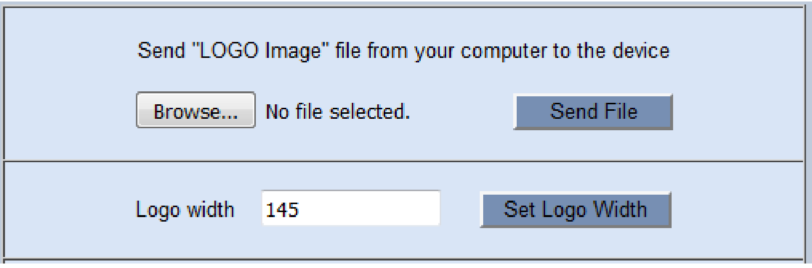
### 6.2.1.1 Replacing the Corporate Logo with an Image

You can replace the logo with a different image.

➢ **To customize the logo:**

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

**Figure 6-18: Customizing Web Logo**




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

**Note:**

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

### 6.2.1.2 Replacing the Corporate Logo with Text

You can replace the logo with text.

➢ **To replace the logo with text:**

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

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

2. Load the ini file using the Auxiliary Files page (see Loading Auxiliary Files on page 493).
3. Reset the device with a save-to-flash for your settings to take effect.

## 6.2.2 Customizing the Product Name

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

- **Web Login screen**
- **Ini file "Board" field:**

```
Board: My Product Name
```

- **CLI prompt:**

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

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

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

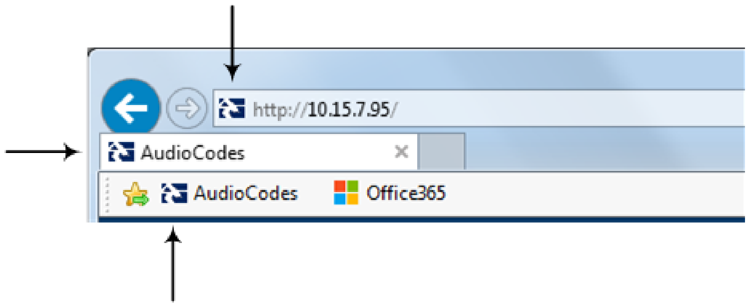
```
UseProductName = 1
UserProductName = < name >
```

2. Load the ini file using the Auxiliary Files page (see Loading Auxiliary Files on page 493).
3. Reset the device with a save-to-flash for your settings to take effect.

## 6.2.3 Customizing the Favicon

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

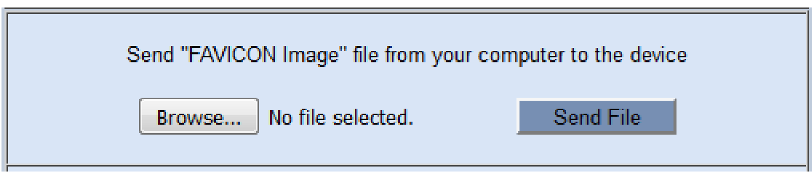
**Figure 6-19: Favicon Display in Browser**




➢ **To customize the favicon:**

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

**Figure 6-20: Customizing Favicon**




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

loads the image file.

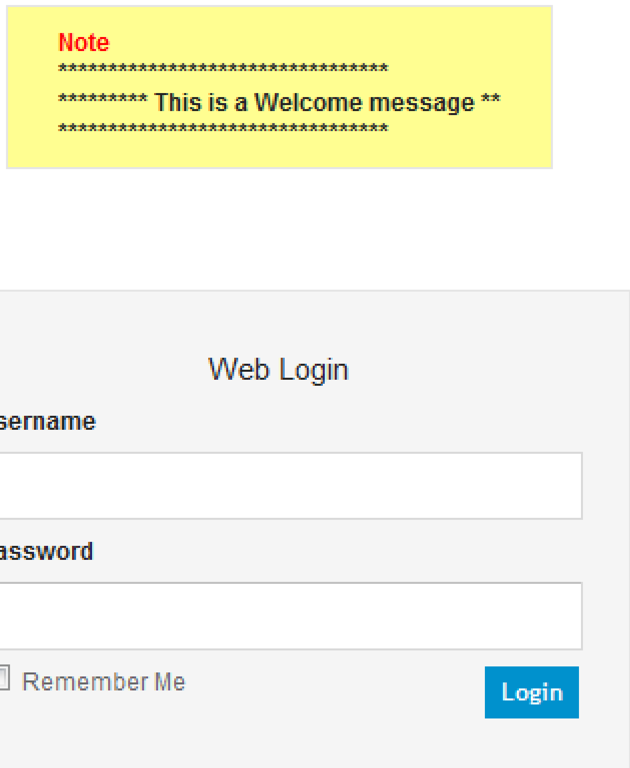
6. On the left pane, click **Back to Main** to exit the Admin page.
7. Reset the device with a save-to-flash for your settings to take effect.

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

### 6.2.4 Creating a Login Welcome Message

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

**Figure 6-21: Creating Login Welcome Message**




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

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

```
[WelcomeMessage ]
FORMAT WelcomeMessage_Index = WelcomeMessage_Text;
```

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

2. Load the ini file using the Auxiliary Files page (see Loading Auxiliary Files on page 493).
3. Reset the device with a save-to-flash for your settings to take effect.

➢ **To remove the welcome message:**

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

## 6.3 Viewing the Home Page

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

➢ **To access the Home page:**

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


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:** 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)
- **Number of FXS Ports:** Number of analog (FXS) ports (depending on ordered hardware configuration)
- **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 487.

The table below describes the areas of the Home page.

**Table 6-8: Home Page Description**

| Item # | Description |
|---|---|
| 1 | Displays the highest severity of an active alarm raised (if any) by the device:<br>▪ Green = No alarms |

| Item # | Description |
|---|---|
| | ▪ Red = Critical alarm<br>▪ Orange = Major alarm<br>▪ Yellow = Minor alarm<br>To view active alarms, click the Alarms area to open the Active Alarms page (see Viewing Active Alarms on page 551). |
| 2 | Displays the status of the Power Supply modules:<br>▪ Green: Power received by chassis<br>▪ Red: Power Supply module is faulty<br>If you click the area, the Components Status page opens, providing detailed status information. |
| 3 | Displays the status of the FXS blades:<br>▪ Green: FXS blade operating normally<br>▪ Red: FXS blade failure (for more information, see the Hardware Installation Manual)<br>If you click an FXS blade, the Port Status page opens, providing detailed status information of the FXS channels. |
| 4 | Displays the status of the Fan Tray module:<br>▪ Green: Fans operating normally<br>▪ Red: At least one fan is faulty<br>If you click the area, the Components Status page opens, providing detailed status information. |
| 5 | Displays the LEDs on the front panel:<br>▪ **Sys:**<br>✓ Green: Device operating normally<br>✓ Orange: Chassis temperature approaching critical level<br>✓ Red: Faulty CPU module or software version<br>▪ **Tel:**<br>✓ Green: FXS blade operating normally<br>✓ Orange: Chassis temperature approaching critical level<br>✓ Red: At least one faulty FXS blade<br>▪ **Power:**<br>✓ Green: At least one Power Supply module is operating normally<br>✓ Red: One of the Power Supply modules is faulty<br>▪ **Fan:**<br>✓ Green: Fans operating normally<br>✓ Red: At least one fan is faulty.<br>If you click the area, the Components Status page opens, providing detailed status information. |
| 6 | Displays the status of the Gigabit Ethernet ports on the CPU module:<br>▪ Green: Ethernet link is working<br>▪ Red: Ethernet link is not working properly |
| 7 | Displays the status of the USB port on the CPU module:<br>▪ Green: USB port connected ok<br>▪ Gray: No USB device connected to USB port<br>If you click the area, the Components Status page opens, providing detailed status information. |

## 6.4 Configuring Web User Accounts

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

Each user account is based on the following:

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

**Table 6-9: Web User Access Levels and Privileges**

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

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

**Table 6-10: Pre-configured Web User Accounts**

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

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

**Notes:**

- For security, it's recommended that you change the default username and password of the pre-configured users (i.e., Security Administrator and Monitor users).
- The Security Administrator user can change all attributes of all Web user accounts. Web users with access levels other than Security Administrator can change only their username and password.
- To restore the two Web user accounts to default settings (usernames and passwords), set the *ini* file parameter ResetWebPassword to 1.
- To log in to the Web interface with a different Web user, click the **Log off** button and then login with with a different username and password.
- You can set the entire Web interface to read-only (regardless of Web user access levels) using the *ini* file parameter DisableWebConfig (see "Web and Telnet Parameters" on page 643).
- You can define additional Web user accounts using a RADIUS server (see "RADIUS Authentication" on page 227).

### 6.4.1 Basic User Accounts Configuration

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

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

**Table 6-11: Allowed Modifications per Web User Level**

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

**Notes:**

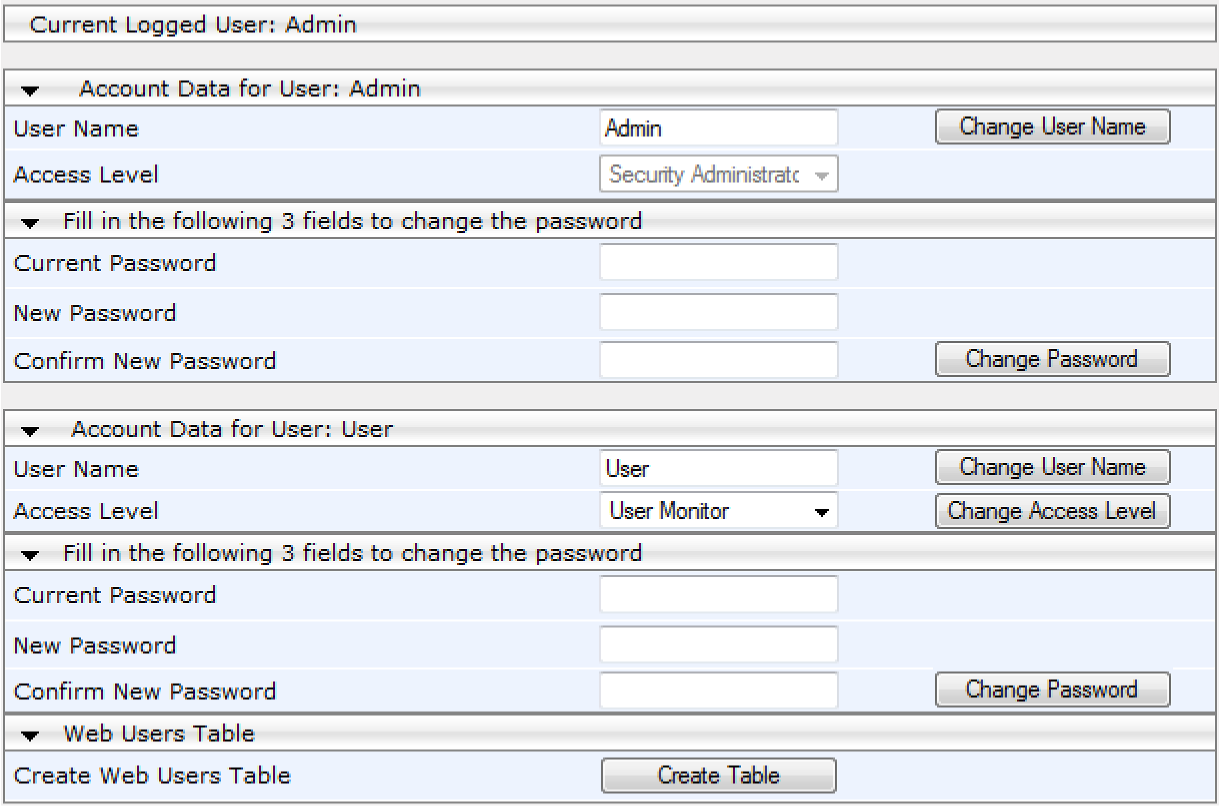
- The username and password can be a string of up to 19 characters and are case-sensitive.
- When only the basic user accounts are being used, up to two users can be concurrently logged in to the Web interface, and they can be the same user.

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

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

details of this user account are displayed.

**Figure 6-22: WEB User Accounts Page (for Users with 'Security Administrator' Privileges)**




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

## 6.4.2 Advanced User Accounts Configuration

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

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

- Master users

**Notes:**

- Only the Security Administrator user can **initially** access the Web Users table. Admin users have read-only privileges in the Web Users table. Monitor users have no access to this table.
- Only Security Administrator and Master users can add, edit, or delete users.
- For advanced user accounts, up to five users can be concurrently logged in to the Web interface, and they can be the same user.
- If you delete a user who is currently in an active Web session, the user is immediately logged off by the device.
- All user types can change their own passwords. This is done in the Web Security Settings page (see "Configuring Web Security Settings" on page 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 change only 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.)

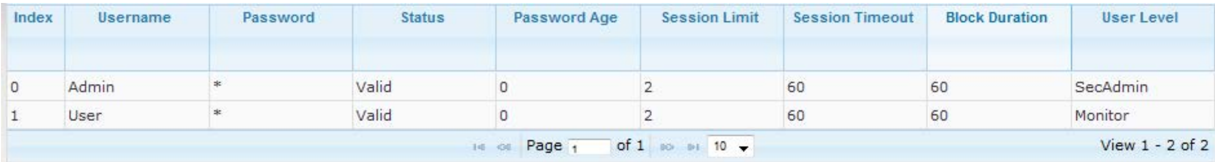
The following procedure describes how to configure Web users through the Web interface. You can also configure it through CLI (**configure system** > **create-users-table**).

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

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

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

**Figure 6-23: Web Users Table Page**

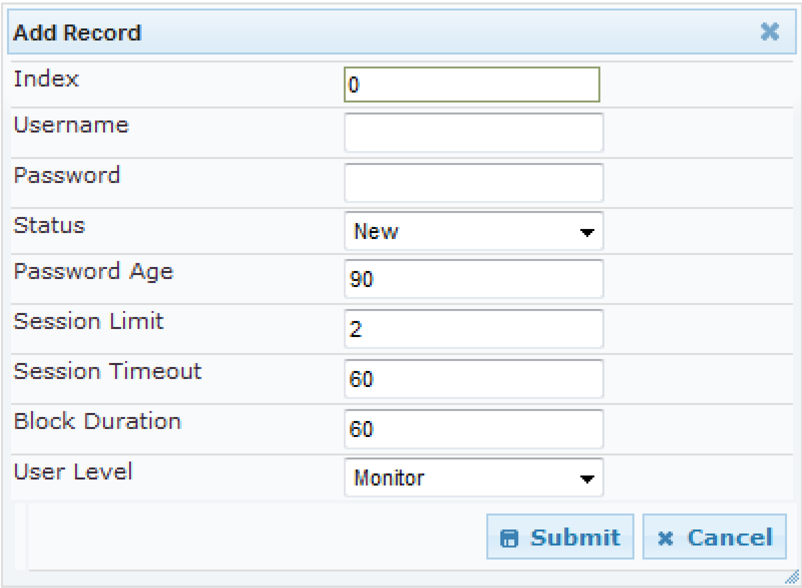


| Index | Username | Password | Status | Password Age | Session Limit | Session Timeout | Block Duration | User Level |
|---|---|---|---|---|---|---|---|---|
| 0 | Admin | * | Valid | 0 | 2 | 60 | 60 | SecAdmin |
| 1 | User | * | Valid | 0 | 2 | 60 | 60 | Monitor |

Page 1 of 1 10 View 1 - 2 of 2

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

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




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

**Table 6-12: Web User Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Username<br>`user-name` | Defines the Web user's username.<br>The valid value is a string of up to 40 alphanumeric characters, including the period ".", underscore "_", and hyphen "-" signs. |

| Parameter | Description |
|---|---|
| Password<br>`password` | Defines the Web user's password.<br>The valid value is a string of 8 to 40 ASCII characters. To ensure strong passwords, adhere to the following password complexity requirements:<br>▪ Contain at least eight characters.<br>▪ Contain at least two letters that are upper case (e.g., A).<br>▪ Contain at least two letters that are lower case (e.g., a).<br>▪ Contain at least two numbers (e.g., 4).<br>▪ Contain at least two symbols (non-alphanumeric characters) (e.g., $, #, %).<br>▪ No spaces.<br>▪ Contain at least four new characters that were not used in the previous password.<br>**Note:** To enforce the password complexity requirements mentioned above, configure the EnforcePasswordComplexity to 1. |
| Status<br>`status` | Defines the status of the Web user.<br>▪ 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.<br>▪ Valid = User can log in to the Web interface as normal.<br>▪ Failed Login = This state is automatically set for users that exceed a user-defined number of failed login attempts, set by the 'Deny Access on Fail Count' parameter (see "Configuring Web Session and Access Settings" on page 74). 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.<br>▪ Inactivity = This state is automatically set for users that have not accessed the Web interface for a user-defined number of days, set by the 'User Inactivity Timer' (see "Configuring Web Session and Access Settings" on page 74). 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.<br>**Notes:**<br>▪ The Inactivity status is applicable only to Admin and Monitor users; System Administrator and Master users can be inactive indefinitely.<br>▪ For security, it is recommended to set the status of a newly added user to New in order to enforce password change. |
| Password Age<br>`password-age` | Defines the duration (in days) of the validity of the password. When this duration elapses, the user is prompted to change the password; otherwise, access to the Web interface is blocked.<br>The valid value is 0 to 10000, where 0 means that the password is always valid. The default is 90. |

| Parameter | Description |
|---|---|
| Session Limit<br>`session-limit` | Defines the maximum number of concurrent Web interface sessions allowed for the specific user. For example, if configured to 2, the same user account can be logged into the device's Web interface (i.e., same username-password combination) from two different management stations (i.e., IP addresses) at any one time. Once the user logs in, the session is active until the user logs off (by clicking the **Log off** icon on the toolbar) or until the session expires if the user is inactive for a user-defined duration (see the 'Session Timeout' parameter below).<br>The valid value is 0 to 5. The default is 2.<br>**Note:** Up to 5 users can be concurrently logged in to the Web interface. |
| Session Timeout<br>`session-timeout` | Defines the duration (in minutes) of inactivity of a logged-in user in the Web interface, after which the the user is automatically logged off the Web session. In other words, the session expires when the user has not performed any operations (activities) in the Web interface for the configured timeout duration.<br>The valid value is 0 to 100000. A value of 0 means no timeout. The default value is according to the settings of the WebSessionTimeout global parameter (see "Configuring Web Session and Access Settings" on page 74). |
| Block Duration<br>`block-duration` | Defines the duration (in seconds) for which the user is blocked when the user exceeds a user-defined number of failed login attempts. This is configured by the 'Deny Access On Fail Count' parameter (see "Configuring Web Session and Access Settings" on page 74).<br>The valid value is 0 to 100000, where 0 means that the user can do as many login failures without getting blocked. The default is according to the settings of the 'Deny Authentication Timer' parameter (see "Configuring Web Session and Access Settings" on page 74).<br>**Note:** The 'Deny Authentication Timer' parameter relates to failed Web logins from specific IP addresses. |

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

## 6.5 Displaying Login Information upon Login

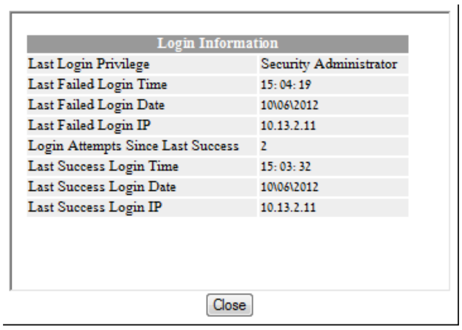
The device can display login information immediately upon Web login.

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

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

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

**Figure 6-25: Login Information Window**




## 6.6 Configuring Web Security Settings

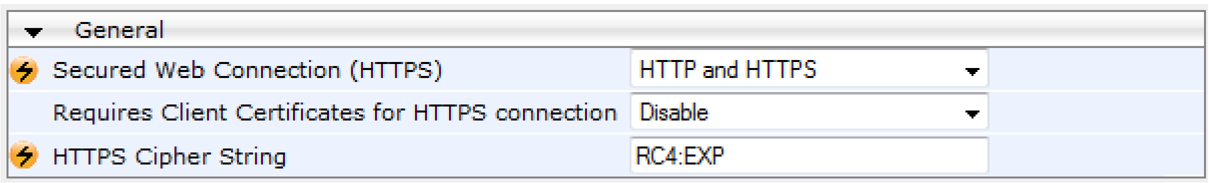
This section describes how to secure Web-based management.

### 6.6.1 Configuring Secured (HTTPS) Web

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

➢ **To configure secure Web access:**

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




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

effect.

For more information on secured Web-based management including TLS certificates, see "TLS for Remote Device Management" on page 115.

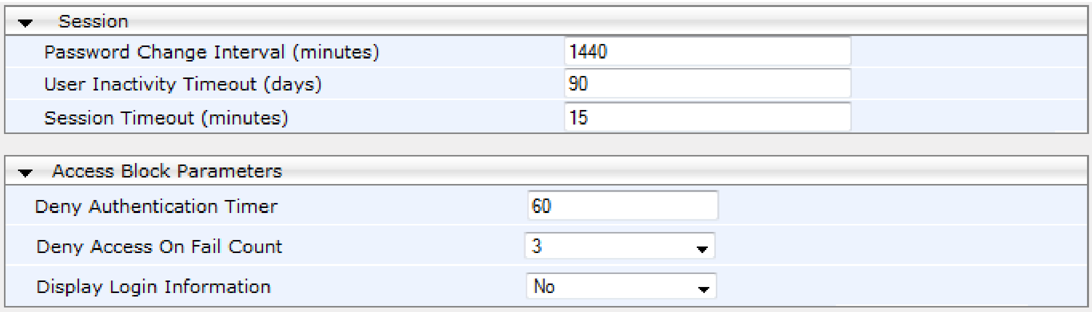
### 6.6.2 Configuring Web Session and Access Settings

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

➢ **To configure Web user sessions and access security:**

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

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




2. Web user sessions:
   a. 'Password Change Interval': Duration of the validity of Web login passwords. When the duration expires, the Web user must change the password in order to log in again.
   b. 'User Inactivity Timeout': If the user has not logged into the Web interface within this defined duration, the status of the user becomes inactive and the user can no longer access the Web interface. The user can only log in to the Web interface if its status is changed (to "New" or "Valid") by an Administrator or a Master user.
   c. 'Session Timeout': Duration of Web inactivity (i.e., no actions are performed in the Web interface) of a logged-in user, after which the Web session expires and the user is automatically logged off the Web interface and needs to log in again to continue the session. You can also configure the functionality per user in the Web Users table (see Advanced User Accounts Configuration on page 67), which overrides this global setting.
3. Web user access:
   a. 'Deny Authentication Timer': Interval (in seconds) that the user needs to wait before the user can attempt to log in from the same IP address after reaching the maximum number of failed login attempts (see next step).
   b. 'Deny Access On Fail Count': Number of failed login attempts after which the user is prevented access to the device for a user-defined duration (previous step).
4. Click **Submit**.

For a detailed description of the above parameters, see "Web Parameters" on page 644.

## 6.7 Web Login Authentication using Smart Cards

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

This feature is enabled using the EnableMgmtTwoFactorAuthentication parameter.

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

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

1. Insert the Common Access Card into the card reader.
2. Access the device using the following URL: https://<host name or IP address>; the device prompts for a username and password.
3. Enter the password only. As some browsers require that the username be provided, it's recommended to enter the username with an arbitrary value.

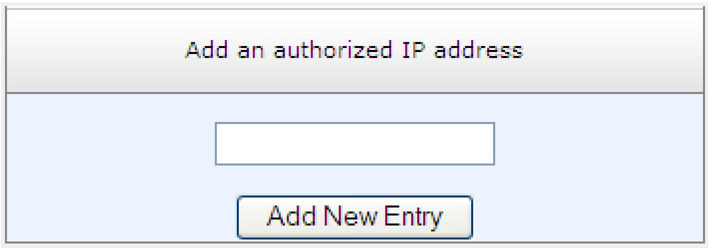
## 6.8 Configuring Web and Telnet Access List

The Web & Telnet Access List page is used to define IP addresses (up to ten) that are permitted to access the device's Web, Telnet, and SSH interfaces. Access from an undefined IP address is denied. If no IP addresses are defined, this security feature is inactive and the device can be accessed from any IP address. The Web and Telnet Access List can also be defined using the *ini* file parameter WebAccessList_x (see "Web Parameters" on page 644).

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

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

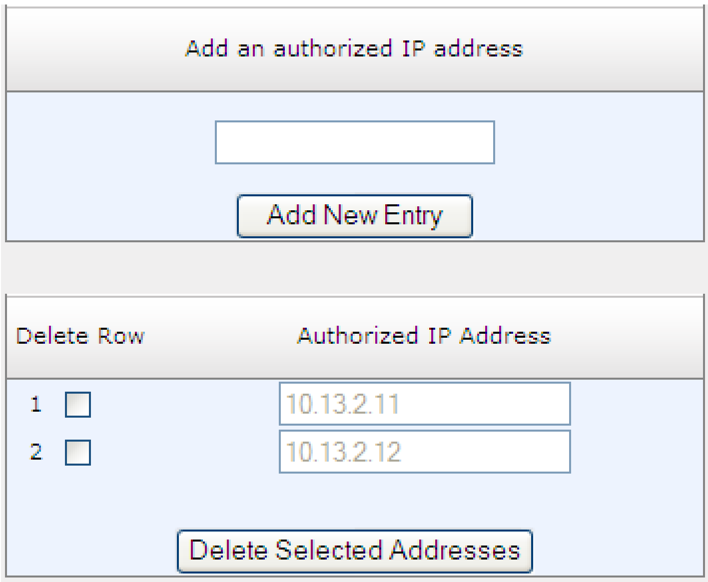
**Figure 6-27: Web & Telnet Access List Page - Add New Entry**




2. To add an authorized IP address, in the 'Add an authorized IP address' field, enter the required IP address, and then click **Add New Entry**; the IP address you entered is

added as a new entry to the Web & Telnet Access List table.

**Figure 6-28: Web & Telnet Access List Table**




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 490.

**Notes:**

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

# 7 CLI-Based Management

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

**Notes:**

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

## 7.1 Getting Familiar with CLI

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

### 7.1.1 Understanding Configuration Modes

Before you begin your CLI session, you should familiarize yourself with the CLI command modes. Each command mode provides different levels of access to commands, as described below:

- **Basic command mode**: This is the initial mode that is accessed upon a successful CLI login authentication. Any user level can access this mode and thus, the commands supported by this command tier are limited, as is interaction with the device itself. This mode allows you to view various information (using the show commands) and activate various debugging capabilities.

  ```
  Welcome to AudioCodes CLI
  Username: Admin
  Password:
  >
  ```

  The Basic mode prompt is ">".

- **Enable command mode:** This mode is the high-level tier in the command hierarchy, one step up from the Basic Mode. A password ("Admin", by default) is required to access this mode **after** you have accessed the Basic mode. This mode allows you to configure all the device's settings. The Enable mode is accessed by typing the following commands:

  ```
  > enable
  Password: <Enable mode password>
  #
  ```

  The Enable mode prompt is "#".

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

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

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

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

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

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

### 7.1.2 Using CLI Shortcuts

The CLI provides several editing shortcut keys to help you configure your device more easily, as listed in the table below.

**Table 7-1: CLI Editing Shortcut keys**

| Shortcut Key | Description |
|---|---|
| **Up** arrow key | Retypes the previously entered command. Continuing to press the **Up** arrow key cycles through all commands entered, starting with the most recent command. |
| **<Tab>** key | Pressing the **<Tab>** key after entering a partial (but unique) command automatically completes the command, displays it on the command prompt line, and waits for further input.<br>Pressing the **<Tab>** key after entering a partial and not unique command displays all completing options. |
| **?** (question mark) | ▪ Displays a list of all subcommands in the current mode, for example:<br>`(config-voip)# voip-network ?`<br>`  dns            Enter voip-network dns`<br>`  ip-group IP Group table`<br>`  nat-translation      NATTranslationtable`<br>`...`<br>▪ Displays a list of available commands beginning with certain letter(s), for example:<br>`(config)# voip-network d?`<br>`  dns            Enter voip-network dns`<br>▪ Displays syntax help for a specific command by entering the command, a space, and then a question mark (?). This includes the range of valid values and a brief description of the next parameter expected for that particular command. For example:<br>`(config)# voip-network dns srv2ip ?`<br>`  [0-9]       index`<br>If a command can be invoked (i.e., all its arguments have been entered), the question mark at its end displays "<cr>" to indicate that a carriage return (Enter) can now be entered to run the command, for example:<br>`(config)# logging host 10.1.1.1 ?`<br>`  <cr>` |
| <**Ctrl + A**> | Moves the cursor to the beginning of the command line. |

| Shortcut Key | Description |
|---|---|
| <**Ctrl + E**> | Moves the cursor to the end of the command line. |
| <**Ctrl + U**> | Deletes all the characters on the command line. |
| auto finish | You need only enter enough letters to identify a command as unique. For example, entering "int G 0/0" at the configuration prompt provides you access to the configuration parameters for the specified Gigabit-Ethernet interface. Entering "interface GigabitEthernet 0/0" would work as well, but is not necessary. |
| **Space Bar** at the --More--prompt | Displays the next screen of output. You can configure the size of the displayed output, as described in "Configuring Displayed Output Lines in CLI Terminal Window" on page 86. |

## 7.1.3 Common CLI Commands

The following table contains descriptions of common CLI commands.

**Table 7-2: Common CLI Commands**

| Command | Description |
|---|---|
| **do** | Provides a way to execute commands in other command sets without taking the time to exit the current command set. The following example shows the **do** command, used to view the GigabitEthernet interface configuration while in the virtual-LAN interface command set:<br>`(config)# interface vlan 1`<br>`(conf-if-VLAN 1)# do show interfaces GigabitEthernet 0/0` |
| **no** | Undoes an issued command or disables a feature. Enter **no** before the command:<br>`# no debug log` |
| **activate** | Activates a command. When you enter a configuration command in the CLI, the command is not applied until you enter the **activate** and **exit** commands.<br>**Note:** Offline configuration changes require a reset of the device. A reset can be performed at the end of the configuration changes. A required reset is indicated by an asterisk (*) before the command prompt. |
| **exit** | Leaves the current command-set and returns one level up. If issued on the top level, the session ends.<br>For online parameters, if the configuration was changed and no **activate** command was entered, the **exit** command applies the **activate** command automatically. If issued on the top level, the session will end:<br>`(config)# exit`<br>`# exit`<br>`(session closed)` |
| **display** | Displays the configuration of current configuration set. |
| **help** | Displays a short help how-to string. |
| **history** | Displays a list of previously run commands. |
| **list** | Displays the available command list of the current command-set. |
| **\| <filter>** | Applied to a command output. The filter should be typed after the command with a pipe mark (\|). |

| Command | Description |
|---|---|
| | Supported filters:<br>▪ **include <word>** – filter (print) lines which contain <word><br>▪ **exclude <word>** – filter lines which does not contain <word><br>▪ **grep <options>** - filter lines according to *grep* common Unix utility options<br>▪ **egrep <options>** - filter lines according to e*grep* common Unix utility options<br>▪ **begin <word>** – filter (print) lines which begins with <word><br>▪ **between <word1> <word2>** – filter (print) lines which are placed between <word1> and <word2><br>▪ **count** – show the output's line count<br>Example:<br>`# show system version \| grep Number`<br>`;Serial Number: 2239835;Slot Number: 1` |

### 7.1.4 Configuring Tables through CLI

Throughout the CLI, many configuration elements are in table format where each table row is represented by an index number. When you add a new row to a table, the device automatically assigns it the next consecutive, available index number. However, you can specify a different index number.

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

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

When you add a new table row, the device accesses the row's configuration mode. For example, if three rows are configured in the Account table (account-0, account-1, and account-2) and you then add a new row, account-3 is automatically created and its' configuration mode is accessed:

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

You can also add a new table row to any specific index number, even if a row has already been configured for that index number. The row that was previously assigned that index number is incremented to the next consecutive index number, as well as all the index rows listed below it in the table. To add a new table row to a specific index number, use the **insert** command:

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

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

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

**Note:** The insert table row feature is applicable only to tables that do not have "child" tables (sub-tables).

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

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

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

### 7.1.5 Understanding CLI Error Messages

The CLI provides feedback on commands by displaying informative messages:

- Failure reason of a run command. The failure message is identical to the notification failure message sent by Syslog. For example, an invalid Syslog server IP address is displayed in the CLI as follows:

```
(logging)# syslog-ip 1111.1.1.1
Parameter 'SyslogServerIP' does NOT accept the IP-Address:
1111.1.1.1, illegal IPAddress.
Configuration failed
Command Failed!
```

- "Invalid command" message: The command may not be valid in the current command mode, or you may not have entered sufficient characters for the command to be recognized. Use "?" to determine your error.
- "Incomplete command" message: You may not have entered all of the pertinent information required to make the command valid. Use "?" to determine your error.

# 7.2 Enabling CLI

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

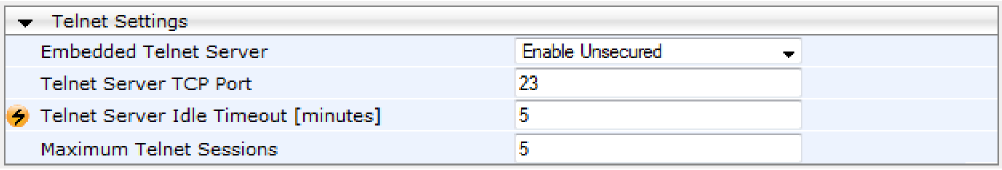
## 7.2.1 Enabling Telnet for CLI

The following procedure describes how to enable Telnet. You can enable a secured Telnet that uses Secure Socket Layer (SSL) where information is not transmitted in the clear. If SSL is used, a special Telnet client is required on your PC to connect to the Telnet interface over a secured connection; examples include C-Kermit for UNIX and Kermit-95 for Windows.

For security, some organizations require the display of a proprietary notice upon starting a Telnet session. You can use the configuration ini file parameter, WelcomeMessage to configure such a message (see "Creating a Login Welcome Message" on page 56).

➢ **To enable Telnet:**

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




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

For a detailed description of the Telnet parameters, see "Telnet Parameters" on page 648.

## 7.2.2 Enabling SSH with RSA Public Key for CLI

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

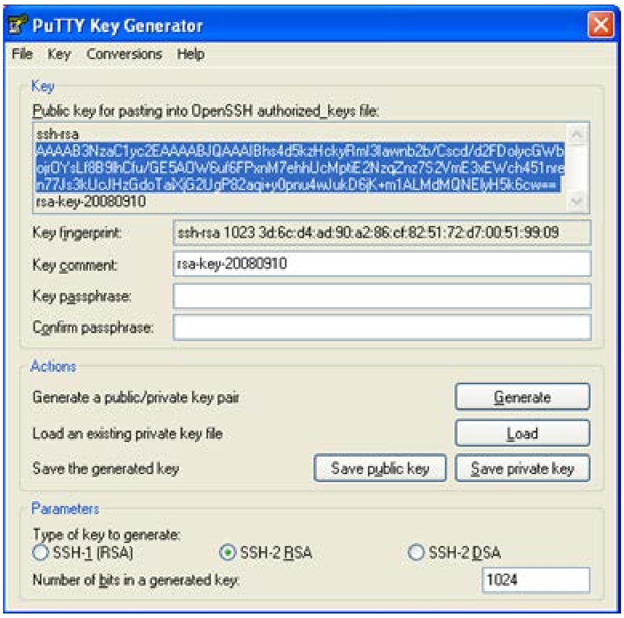
SSH requires appropriate client software for the management PC. Most Linux distributions have OpenSSH pre-installed; Windows-based PCs require an SSH client software such as PuTTY, which can be downloaded from http://www.chiark.greenend.org.uk/~sgtatham/putty/.

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

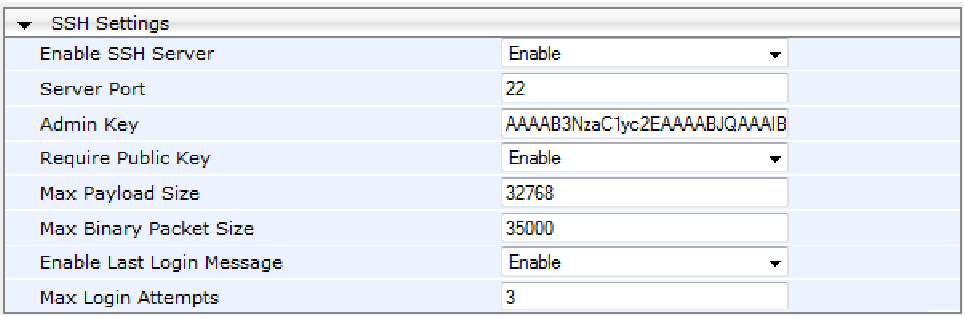
### ➢ To enable SSH and configure RSA public keys for Windows (using PuTTY SSH software):

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

**Figure 7-1: Selecting Public RSA Key in PuTTY**




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




   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 685.

e. Click **Submit**.

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

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

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

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

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

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

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

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

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

4. Click **Submit**.

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

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

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

## 7.3 Configuring Maximum Telnet/SSH Sessions

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

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

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

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

2. In the 'Maximum Telnet Sessions' field, enter the maximum number of concurrent sessions.

3. Click **Submit**.

## 7.4 Establishing a CLI Session

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

- RS-232: The device can be accessed through its RS-232 serial port, by connecting a VT100 terminal to it or using a terminal emulation program (e.g., HyperTerminal) with a PC. For connecting to the CLI through RS-232, see CLI on page 35.
- **Secure SHell (SSH):** The device can be accessed through its Ethernet interface by the SSH protocol using SSH client software. A popular and freeware SSH client software is Putty, which can be downloaded from http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
- **Telnet:** The device can be accessed through its Ethernet interface by the Telnet protocol using Telnet client software.

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

> **Note:** The CLI login credentials are the same as all the device's other management interfaces (such as Web interface). The default username and password is "Admin" and "Admin" (case-sensitive), respectively. For configuring login credentials, see "Configuring Web User Accounts" on page 65.

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

1. Connect the device to the network.
2. Establish a Telnet or SSH session using the device's OAMP IP address.
3. Log in to the session using the username and password assigned to the Admin user of the Web interface:
   a. At the Username prompt, type the username, and then press Enter:
      ```
      Username: Admin
      ```
   b. At the Password prompt, type the password, and then press Enter:
      ```
      Password: Admin
      ```
   c. At the prompt, type the following, and then press Enter:
      ```
      > enable
      ```
   d. At the prompt, type the password again, and then press Enter:
      ```
      Password: Admin
      ```

## 7.5 Viewing Current CLI Sessions

You can view users that are currently logged in to the device's CLI. This applies to users logged in to the CLI through RS-232 (console), Telnet, or SSH. For each logged-in user, the following is displayed: the type of interface (console, Telnet, or SSH), user's username, remote IP address from where the user logged in, and the duration (days and time) of the session. Each user is displayed with a unique index (session ID).

➢ **To view currently logged-in CLI users:**

```
# show users
[0]   console      Admin        local              0d00h03m15s
[1]   telnet       John         10.4.2.1           0d01h03m47s
[2]*  ssh          Alex         192.168.121.234    12d00h02m34s
```

The current session from which this show command was run is displayed with an asterisk (*).

**Note:** The device can display management sessions of up to 24 hours. After this time, the duration counter is reset.

## 7.6 Terminating a User's CLI Session

You can terminate users that are currently logged in to the device's CLI. This applies to users logged in to the CLI through RS-232 (console), Telnet, or SSH.

➢ **To terminate the CLI session of a specific CLI user:**

```
# clear user <session ID>
```

The *session ID* is a unique identification of each currently logged in user. You can view the session ID by running the **show users** command (see "Viewing Current CLI Sessions" on page 85).

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

## 7.7 Configuring Displayed Output Lines in CLI Terminal Window

You can configure the maximum number of lines (height) displayed in the terminal window for the output of CLI commands (Telnet and SSH). The number of displayed lines can be specified from 0 to 65,535, or determined by re-sizing the terminal window by mouse-dragging the window's border.

➢ **To configure a specific number of output lines:**

```
(config-system)# cli-terminal
<cli-terminal># window-height [0-65535]
```

If window-height is set to 0, the entire command output is displayed. In other words, even if the output extends beyond the visible terminal window length, the --MORE-- prompt is not displayed.

➢ **To configure the number of lines according to dragged terminal window:**

```
(config-system)# cli-terminal
<cli-terminal># window-height automatic
```

When this mode is configured, each time you change the height of the terminal window using your mouse (i.e., dragging one of the window's borders or corners), the number of displayed output command lines is changed accordingly.

# 8 SNMP-Based Management

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

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

This section provides configuration relating to SNMP management.

**Notes:**

- SNMP-based management is enabled by default.
- For more information on the device's SNMP support (e.g., SNMP traps), refer to the *SNMP User's Guide*.
- EMS support is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see "Software License Key" on page 510.
- For more information on using the EMS tool, refer to the *EMS User's Manual* and *EMS Server IOM Manual*.

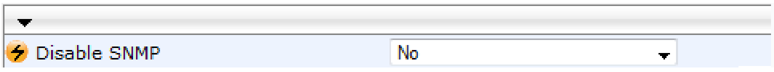
## 8.1 Disabling SNMP

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

➢ **To disable SNMP:**

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

**Figure 8-1: Disabling SNMP**




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

## 8.2 Configuring SNMP Community Strings

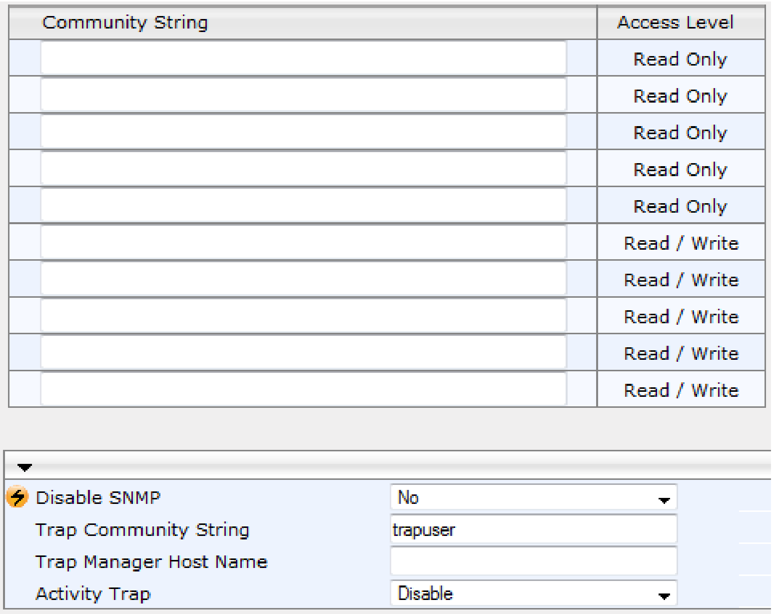
The SNMP Community String page lets you configure up to five read-only and up to five read-write SNMP community strings and to configure the community string that is used for sending traps. The SNMP community string determines the access privileges (read-only or read-write) of SNMP clients to the device's SNMP.

**Note:** SNMP community strings are used only for SNMPv1 and SNMPv2c; SNMPv3 uses username-password authentication along with an encryption key (see "Configuring SNMP V3 Users" on page 92).

For detailed descriptions of the SNMP parameters, see "SNMP Parameters" on page 649.

➢ **To configure SNMP community strings:**

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




2. Configure SNMP community strings according to the table below.
3. Click **Submit**, and then save ("burn") your settings to flash memory.

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

**Table 8-1: SNMP Community String Parameter Descriptions**

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

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

## 8.3 Configuring SNMP Trap Destinations

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

➢ **To configure SNMP trap destinations:**

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

**Figure 8-2: SNMP Trap Destinations Table**

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

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

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

**Notes:**

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

**Table 8-2: SNMP Trap Destinations Table Parameters Description**

| Parameter | Description |
|---|---|
| SNMP Manager<br>[SNMPManagerIsUsed_x] | Enables the SNMP Manager to receive traps and checks the validity of the configured destination (IP address and port number).<br>▪ **[0]** (check box cleared) = (Default) Disables SNMP Manager<br>▪ **[1]** (check box selected) = Enables SNMP Manager |

| Parameter | Description |
|---|---|
| IP Address<br>[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<br>[SNMPManagerTrapPort_x] | Defines the port number of the remote SNMP Manager. The device sends SNMP traps to this port.<br>The valid value range is 100 to 4000. The default is 162. |
| Trap User<br>[SNMPManagerTrapUser] | Associates a trap user with the trap destination. This determines the trap format, authentication level, and encryption level.<br>▪ v2cParams (default) = SNMPv2 user community string<br>▪ SNMPv3 user configured in "Configuring SNMP V3 Users" on page 92 |
| Trap Enable<br>[SNMPManagerTrapSendingEnable_x] | Activates the sending of traps to the SNMP Manager.<br>▪ **[0]** Disable<br>▪ **[1]** Enable (Default) |

## 8.4 Configuring SNMP Trusted Managers

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

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

### ➢ To configure SNMP Trusted Managers:

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

**Figure 8-3: SNMP Trusted Managers Table**

| Delete | Trusted Managers IP Address | |
|---|---|---|
| ☐ | SNMP Trusted Manager 1 | 0.0.0.0 |
| ☐ | SNMP Trusted Manager 2 | 0.0.0.0 |
| ☐ | SNMP Trusted Manager 3 | 0.0.0.0 |
| ☐ | SNMP Trusted Manager 4 | 0.0.0.0 |
| ☐ | SNMP Trusted Manager 5 | 0.0.0.0 |

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

## 8.5 Configuring SNMP V3 Users

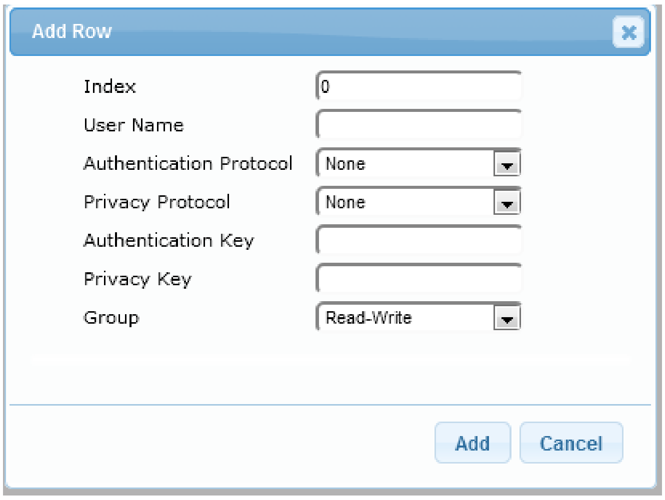
The SNMPv3 Users table lets you configure up to 10 SNMP v3 users for authentication and privacy.

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

➤ **To configure an SNMP v3 user:**

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

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




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

**Note:** If you delete a user that is associated with a trap destination (see "Configuring SNMP Trap Destinations" on page 90), the configured trap destination becomes disabled and the trap user reverts to default (i.e., SNMPv2).

**Table 8-3: SNMPv3 Users Table Parameters Description**

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

| Parameter | Description |
|---|---|
| Privacy Protocol<br>`priv-protocol`<br>[SNMPUsers_PrivProtocol] | Privacy protocol of the SNMP v3 user.<br>▪ **[0]** None (default)<br>▪ **[1]** DES<br>▪ **[2]** 3DES<br>▪ **[3]** AES-128<br>▪ **[4]** AES-192<br>▪ **[5]** AES-256 |
| Authentication Key<br>`auth-key`<br>[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<br>`priv-key`<br>[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<br>`group`<br>[SNMPUsers_Group] | The group with which the SNMP v3 user is associated.<br>▪ **[0]** Read-Only (default)<br>▪ **[1]** Read-Write<br>▪ **[2]** Trap<br>**Note:** All groups can be used to send traps. |

**This page is intentionally left blank.**

# 9 INI File-Based Management

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

**Notes:**

- For a list and description of the *ini* file parameters, see "Configuration Parameters Reference" on page 643.
- To restore the device to default settings through the *ini* file, see "Restoring Factory Defaults" on page 537.

## 9.1 INI File Format

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

- Individual parameters - see "Configuring Individual ini File Parameters" on page 95
- Table parameters - see "Configuring Table ini File Parameters" on page 95

### 9.1.1 Configuring Individual ini File Parameters

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

- An optional, subsection name (or group name) enclosed in square brackets "[...]". This is used to conveniently group similar parameters by their functionality.
- Parameter name, followed by an equal "=" sign and then its value.
- Comments must be preceded by a semicolon ";".

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

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

### 9.1.2 Configuring Table ini File Parameters

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

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

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

- **Format line:** Specifies the columns of the table (by their string names) that are to be configured.
    - The first word of the Format line must be "FORMAT", followed by the Index field name and then an equal "=" sign. After the equal sign, the names of the columns are listed.
    - Columns must be separated by a comma ",".
    - The Format line must only include columns that can be modified (i.e., parameters that are not specified as read-only). An exception is Index fields, which are mandatory.
    - The Format line must end with a semicolon ";".
- **Data line(s):** Contain the actual values of the columns (parameters). The values are interpreted according to the Format line.
    - The first word of the Data line must be the table's string name followed by the Index field.
    - Columns must be separated by a comma ",".
    - A Data line must end with a semicolon ";".
- **End-of-Table Mark:** Indicates the end of the table. The same string used for the table's title, preceded by a backslash "\", e.g., [\MY_TABLE_NAME].

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

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

The table ini file parameter formatting rules are listed below:

- Indices (in both the Format and the Data lines) must appear in the same order. The Index field must never be omitted.
- The Format line can include a subset of the configurable fields in a table. In this case, all other fields are assigned with the pre-defined default values for each configured line.
- The order of the fields in the Format line isn't significant (as opposed to the Index fields). The fields in the Data lines are interpreted according to the order specified in the Format line.
- The double dollar sign ($$) in a Data line indicates the default value for the parameter.
- The order of the Data lines is insignificant.
- Data lines must match the Format line, i.e., it must contain exactly the same number of Indices and Data fields and must be in exactly the same order.
- A row in a table is identified by its table name and Index field. Each such row may appear only once in the *ini* file.
- Table dependencies: Certain tables may depend on other tables. For example, one table may include a field that specifies an entry in another table. This method is used to specify additional attributes of an entity, or to specify that a given entity is part of a larger entity. The tables must appear in the order of their dependency (i.e., if Table X is referred to by Table Y, Table X must appear in the *ini* file before Table Y).

For general *ini* file formatting rules, see "General ini File Formatting Rules" on page 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_CoderSpecific;
CodersGroup0 0 = g711Alaw64k, 20, 0, 255, 0, 0;
CodersGroup0 1 = eg711Ulaw, 10, 0, 71, 0, 0;
[ \CodersGroup0 ]
```

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

### 9.1.3 General ini File Formatting Rules

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

- The *ini* file name must not include hyphens "-" or spaces; if necessary, use an underscore "_" instead.
- Lines beginning with a semi-colon ";" are ignored. These can be used for adding remarks in the *ini* file.
- A carriage return (i.e., Enter) must be done at the end of each line.
- The number of spaces before and after the equals sign "=" is irrelevant.
- Subsection names for grouping parameters are optional.
- If there is a syntax error in the parameter name, the value is ignored.
- Syntax errors in the parameter's value can cause unexpected errors (parameters may be set to the incorrect values).
- Parameter string values that denote file names (e.g., CallProgressTonesFileName) must be enclosed with inverted commas, e.g., CallProgressTonesFileName = 'cpt_usa.dat'.
- The parameter name is not case-sensitive.
- The parameter value is not case-sensitive, except for coder names.
- The *ini* file must end with at least one carriage return.

## 9.2 Configuring an ini File

There are different methods that you can use for configuring the ini file before you load it to the device.

- Modifying the device's current ini file. This method is recommended if you mainly need to change the settings of parameters that you have previously configured.
  1. Save the device's current configuration as an *ini* file on your computer, using the Web interface (see "Saving Configuration" on page 490).
  2. Open the file using a text file editor, and then modify the *ini* file as required.
  3. Save and close the file.
  4. Load the file to the device.
- Creating a new ini file that includes only updated configuration:
  1. Open a text file editor such as Notepad.
  2. Add only the required parameters and their settings.
  3. Save the file with the ini file extension name (e.g., myconfiguration.ini).

4. Load the file to the device.

For loading the ini file to the device, see "Loading an ini File to the Device" on page 98.

**Notes:**

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

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

- To restore the device to default settings through the *ini* file, see "Restoring Factory Defaults" on page 537.

## 9.3 Loading an ini File to the Device

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

- CLI:
  - Voice Configuration: # copy voice-configuration from <URL>
- Web interface:
  - Load Auxiliary Files page (see "Loading Auxiliary Files" on page 493): The device updates its configuration according to the loaded ini file, while preserving the remaining current configuration.
  - Configuration File page (see "Backing Up and Loading Configuration File" on page 517): The device updates its configuration according to the loaded ini file, and applies default values to parameters that were not included in the loaded ini file. Thus, all previous configuration is overridden.

When you load an ini file to the device, its configuration settings are saved to the device's non-volatile memory.

**Note:** Before you load an *ini* file to the device, make sure that the file extension name is *.ini*.

## 9.4 Secured Encoded ini File

The *ini* file contains sensitive information that is required for the functioning of the device. The file may be loaded to the device using HTTP. These protocols are not secure and are vulnerable to potential hackers. To overcome this security threat, the AudioCodes DConvert utility allows you to binary-encode (encrypt) the *ini* file before loading it to the device. For more information, refer to the *DConvert Utility User's Guide*.

**Note:** If you save an ini file from the device to a folder on your PC, an *ini* file that was loaded to the device encoded is saved as a regular *ini* file (i.e., unencoded).

## 9.5 Configuring Password Display in ini File

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

- Obscured: The password characters are concealed and displayed as encoded. The password is displayed using the syntax, *$1$<obscured password>*, for example, $1$S3p+fno=.
- Hidden: the password is replaced with an asterisk (*).

When you save an ini file from the device to a PC, the passwords are displayed according to the enabled format. When you load an ini file to the device, obscured passwords are parsed and applied to the device; hidden passwords are ignored.

By default, the enabled format is obscured passwords, thus enabling their full recovery in case of configuration restore or copy to another device.

When obscured password mode is enabled, you can enter a password in the ini file using any of the following formats:

- $1$<obscured password>: Password in obscured format as generated by the device; useful for restoring device configuration and copying configuration from one device to another.
- $0$<plain text>: Password can be entered in plain text; useful for configuring a new password. When the ini file is loaded to the device and then later saved from the device to a PC, the password is displayed obscured (i.e., $1$<obscured password>).

## 9.6 INI Viewer and Editor Utility

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

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

# Part III

## General System Settings

# 10 Configuring SSL/TLS Certificates

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

**Notes:**

- The device is shipped with an active, default TLS setup. Thus, configure certificates only if required.
- Since X.509 certificates have an expiration date and time, you must configure the device to use Network Time Protocol (NTP) to obtain the current date and time from an NTP server. Without the correct date and time, client certificates cannot work. For configuring NTP, see "Configuring Automatic Date and Time using SNTP" on page 119.
- Only **Base64 (PEM)** encoded X.509 certificates can be loaded to the device.

## 10.1 Configuring TLS Certificate Contexts

The TLS Contexts table lets you configure up to 12 TLS certificates, referred to as *TLS Contexts*. The Transport Layer Security (TLS), also known as Secure Socket Layer (SSL), is used to secure the device's SIP signaling connections, Web interface, and Telnet server. The TLS/SSL protocol provides confidentiality, integrity, and authenticity between two communicating applications over TCP/IP.

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

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

Each TLS Context can be configured with the following:

- Context ID and name
- TLS version - SSL 2.0 (only for TLS handshake), SSL 3.0, TLS 1.0, TLS 1.1, TLS 1.2)
- Encryption ciphers for server and client - DES, RC4 compatible, Advanced Encryption Standard (AES)
- Online Certificate Status Protocol (OCSP). Some Public-Key Infrastructures (PKI) can revoke a certificate after it has been issued. You can configure the device to check whether a peer's certificate has been revoked, using the OCSP. When OCSP is enabled, the device queries the OCSP server for revocation information whenever a peer certificate is received (TLS client mode, or TLS server mode with mutual authentication).
- Private key - externally created and then uploaded to device
- X.509 certificates - self-signed certificates or signed as a result of a certificate signing request (CSR)
- Trusted root certificate authority (CA) store (for validating certificates)

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

- **Incoming calls:**

1. Proxy Set: If the incoming call is successfully classified to an IP Group based on Proxy Set (i.e., IP address of calling party) and the Proxy Set is configured for TLS ('Transport Type' parameter is set to **TLS**), the TLS Context assigned to the Proxy Set is used. For configuring Proxy Sets, see "Configuring Proxy Sets" on page 329.
2. SIP Interface: If the Proxy Set is either not configured for TLS (i.e., the 'Transport Type' parameter is set to **UDP**) or not assigned a TLS Context, and/or classification to a Proxy Set fails, the device uses the TLS Context assigned to the SIP Interface used for the call. For configuring SIP Interfaces, see "Configuring SIP Interfaces" on page 319.
3. Default TLS Context (ID 0): If the SIP Interface is not assigned a TLS Context or no SIP Interface is used for the call, the device uses the default TLS Context.

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

**Notes:**

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

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

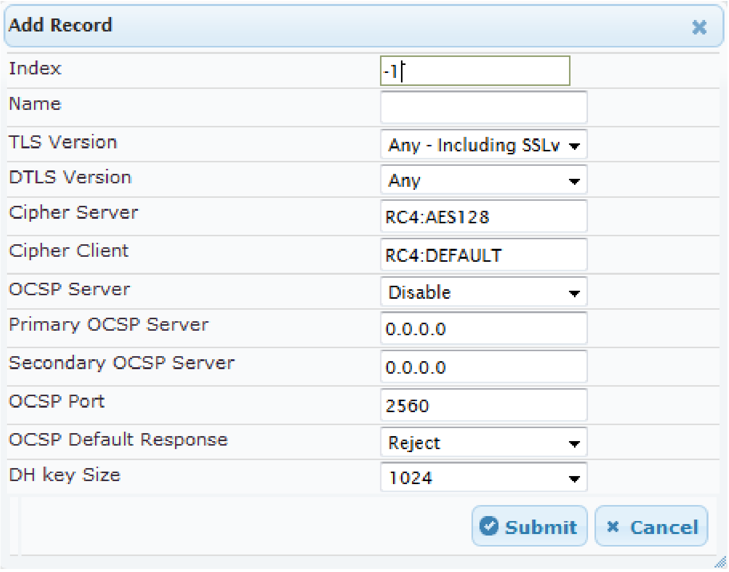
You can also configure TLS certificate expiry check, whereby the device periodically checks the validation date of the installed TLS server certificates and sends an SNMP trap event if a certificate is nearing expiry. This feature is configured globally for all TLS Contexts. For configuring TLS certificate expiry check, see "Configuring TLS Server Certificate Expiry Check" on page 117.

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

➢ **To configure a TLS Context:**

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

**Figure 10-1: TLS Contexts Table - Add Record Dialog Box**




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

**Table 10-1: TLS Context Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>`tls <ID>`<br>[TLSContexts_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>`name`<br>[TLSContexts_Name] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 31 characters. |
| TLS Version<br>`tls-version`<br>[TLSContexts_TLSVersion] | Defines the supported SSL/TLS protocol version. Clients attempting to communicate with the device using a TLS version that is not configured are rejected.<br>▪ [0] Any - Including SSLv3 = (Default) SSL 3.0 and all TLS versions are supported. SSL/TLS handshakes always start with an SSL 2.0-compatible handshake and then switch to the highest TLS version supported by both peers.<br>▪ [1] TLSv1.0 = Only TLS 1.0.<br>▪ [2] TLSv1.1 = Only TLS 1.1.<br>▪ [3] TLSv1.0 and TLSv1.1 = Only TLS 1.0 and TLS 1.1.<br>▪ [4] TLSv1.2 = Only TLS 1.2.<br>▪ [5] TLSv1.0 and TLSv1.2 = Only TLS 1.0 and TLS 1.2.<br>▪ [6] TLSv1.1 and TLSv1.2 = Only TLS 1.1 and TLS 1.2.<br>▪ [7] TLSv1.0 TLSv1.1 and TLSv1.2 = Only TLS 1.0, TLS 1.1 and TLS 1.2 (excludes SSL 3.0). |

| Parameter | Description |
|---|---|
| Cipher Server<br>`ciphers-server`<br>[TLSContexts_ServerCipherString] | Defines the supported cipher suite for the TLS server (in OpenSSL cipher list format).<br>The default is AES:RC4. For valid values, visit the OpenSSL website at https://www.openssl.org/docs/man1.0.2/apps/ciphers.html. |
| DTLS Version<br>[TLSContexts_DTLSVersion] | Defines the Datagram Transport Layer Security (DTLS) version, which is used to negotiate keys for WebRTC calls.<br>▪ [0] Any (default)<br>▪ [1] DTLSv1.0<br>▪ [2] DTLSv1.2<br>**Note:** The parameter is applicable only to the SBC application. |
| Cipher Client<br>`ciphers-client`<br>[TLSContexts_ClientCipherString] | Defines the supported cipher suite for TLS clients.<br>The default is DEFAULT.<br>For possible values and additional details, visit the OpenSSL website at https://www.openssl.org/docs/man1.0.2/apps/ciphers.html. |
| OCSP Server<br>`ocsp-server`<br>[TLSContexts_OcspEnable] | Enables or disables certificate checking using OCSP.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable |
| Primary OCSP Server<br>`ocsp-server-primary`<br>[TLSContexts_OcspServerPrimary] | Defines the IP address (in dotted-decimal notation) of the primary OCSP server.<br>The default IP address is 0.0.0.0. |
| Secondary OCSP Server<br>`ocsp-server-secondary`<br>[TLSContexts_OcspServerSecondary] | Defines the IP address (in dotted-decimal notation) of the secondary OCSP server (optional).<br>The default IP address is 0.0.0.0. |
| OCSP Port<br>`ocsp-port`<br>[TLSContexts_OcspServerPort] | Defines the OCSP server's TCP port number.<br>The default port number is 2560. |
| OCSP Default Response<br>`ocsp-default-response`<br>[TLSContexts_OcspDefaultResponse] | Determines whether the device allows or rejects peer certificates if it cannot connect to the OCSP server.<br>▪ **[0]** Reject (default)<br>▪ **[1]** Allow |
| DH Key Size<br>[TLSContexts_DHKeySize] | Defines the Diffie-Hellman (DH) key size (in bits). DH is an algorithm used chiefly for exchanging cryptography keys used in symmetric encryption algorithms such as AES.<br>▪ [1024] 1024 (default)<br>▪ [2048] 2048 |

## 10.2 Assigning CSR-based Certificates to TLS Contexts

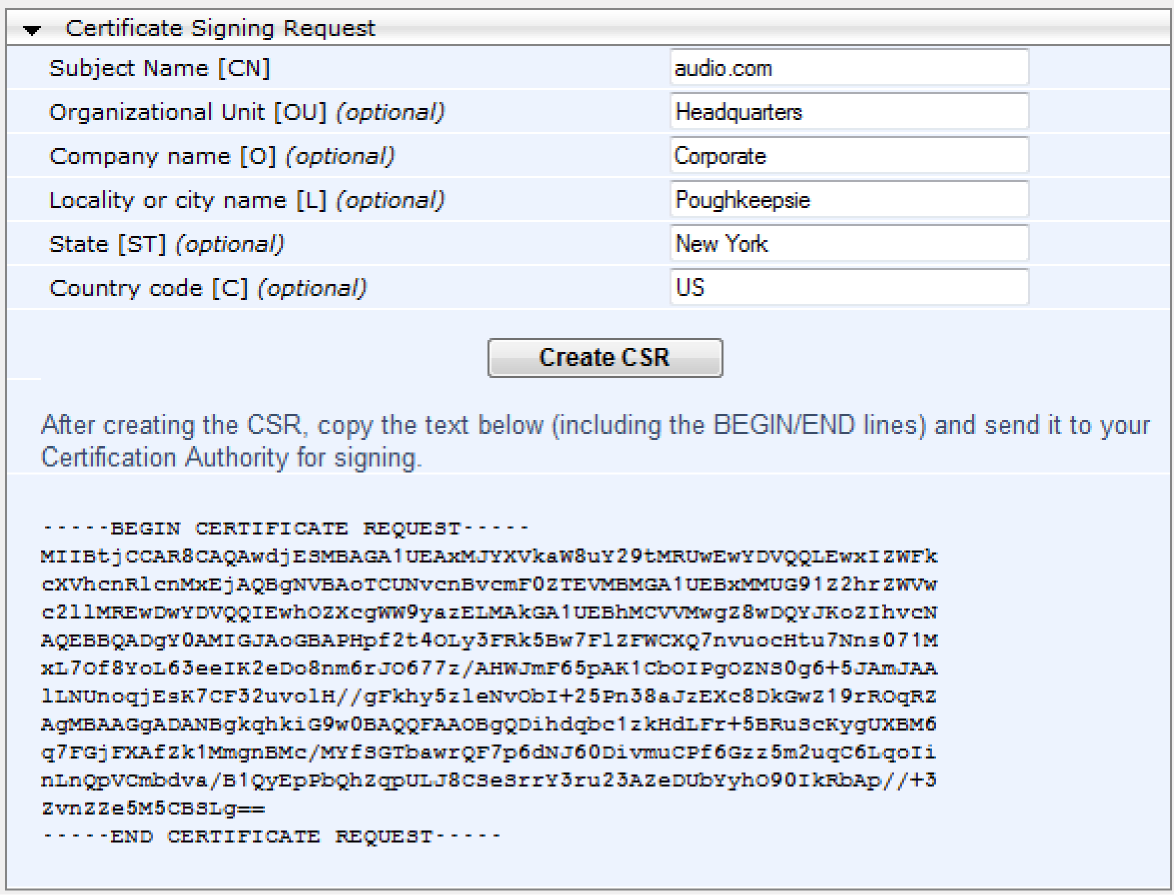
The following procedure describes how to request a digitally signed certificate from a Certification Authority (CA) for a TLS Context. This process is referred to as a certificate signing request (CSR) and is required if your organization employs a Public Key Infrastructure (PKI) system. The CSR contains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

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

1. Your network administrator should allocate a unique DNS name for the device (e.g., dns_name.corp.customer.com). This DNS name is used to access the device and therefore, must be listed in the server certificate.
2. Open the TLS Contexts page (**Configuration** tab > **System** menu > **TLS Contexts**).
3. In the table, select the required TLS Context index row, and then click the **TLS Context Certificate**  button, located below the table; the Context Certificates page appears.
4. Under the **Certificate Signing Request** group, do the following:
   a. In the 'Subject Name [CN]' field, enter the DNS name.
   b. From the 'Signature Algorithm' drop-down list, select the hash function algorithm (SHA-1, SHA-256, or SHA-512) with which to sign the certificate.
   c. Fill in the rest of the request fields according to your security provider's instructions.

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

**Figure 10-2: Certificate Signing Request Group**




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

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

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

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

8. After the certificate successfully loads to the device, save the configuration with a device reset.

9. Open the TLS Contexts page again, select the TLS Context index row, and then verify that under the **Certificate Information** group, the 'Private key' field displays "OK";

otherwise, consult your security administrator:

**Figure 10-3: Private key "OK" in Certificate Information Group**

| Certificate Information | |
|---|---|
| Certificate subject: | /CN=ACL_5967925 |
| Certificate issuer: | /CN=ACL_5967925 |
| Time to expiration: | 7246 days |
| Key size: | 1024 bits |
| Private key: | OK |

**Notes:**

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

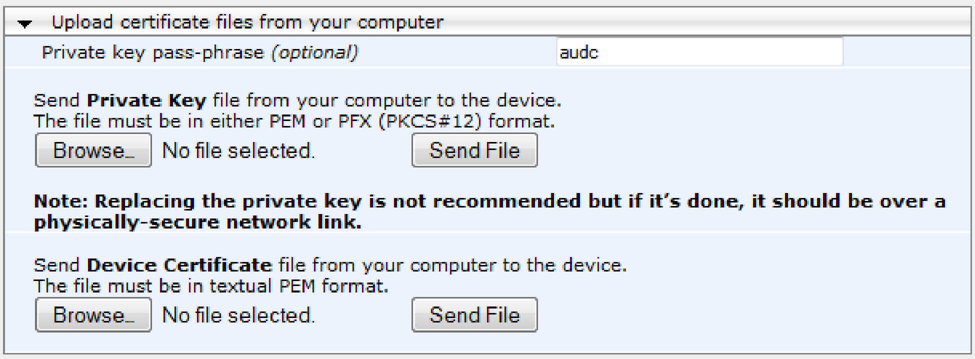
## 10.3 Assigning Externally Created Private Keys to TLS Contexts

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

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

1. Obtain a private key in either textual PEM (PKCS #7) or PFX (PKCS #12) format (typically provided by your security administrator). The file may be encrypted with a short pass-phrase.
2. Open the TLS Contexts page (**Configuration** tab > **System** menu > **TLS Contexts**).
3. In the table, select the required TLS Context index row, and then click the **TLS Context Certificate** button, located below the table; the Context Certificates page appears.
4. Scroll down to the **Upload certificate files from your computer** group.

**Figure 10-4: Upload Certificate Files from your Computer Group**




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

## 10.4 Generating Private Keys for TLS Contexts

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

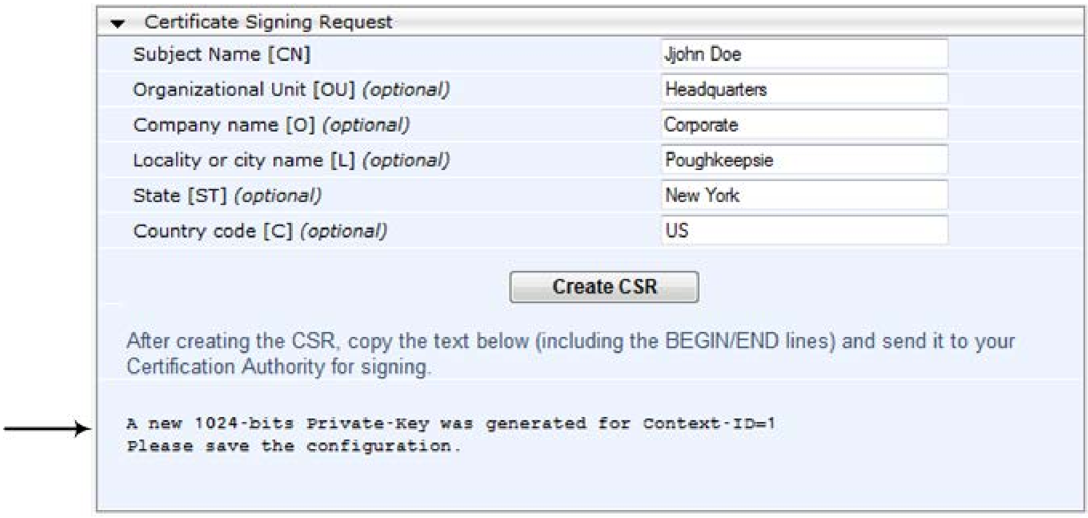
➢ **To generate a new private key for a TLS Context:**

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

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


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

**Figure 10-6: Indication of Newly Generated Private Key**



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

## 10.5 Creating Self-Signed Certificates for TLS Contexts

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

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

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

**Figure 10-7: Generate new private key and self-signed certificate Group**

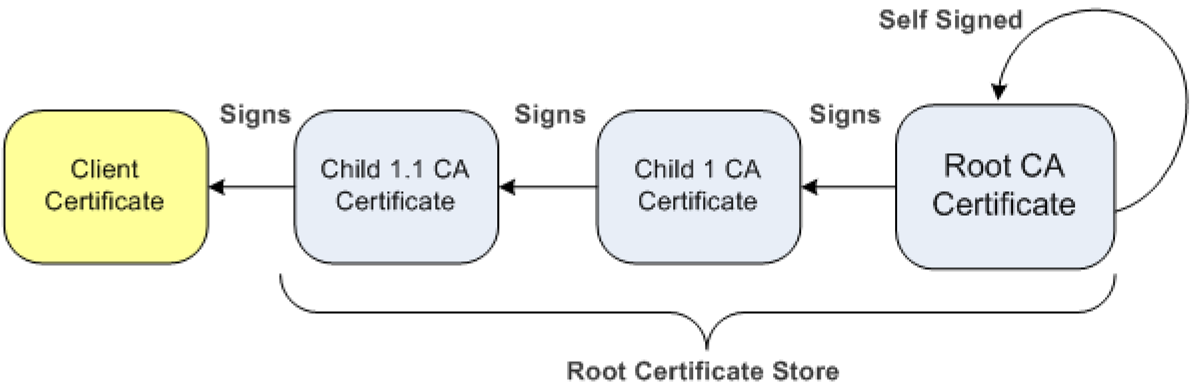

6. Click **Generate Self-Signed Certificate**; a message appears (after a few seconds) displaying the new subject name.
7. Save the configuration with a device reset for the new certificate to take effect.

## 10.6 Importing Certificates and Certificate Chain into Trusted Certificate Store

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

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

**Figure 10-8: Certificate Chain Hierarchy**




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

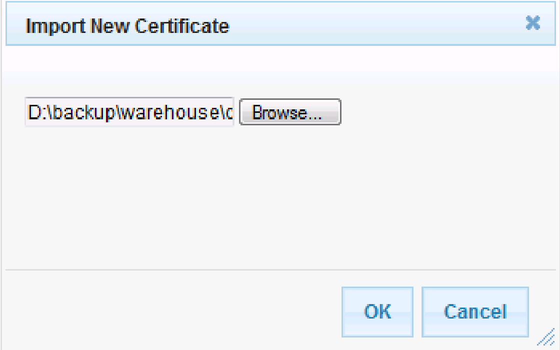
**Note:** Only Base64 (PEM) encoded X.509 certificates can be loaded to the device.

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

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

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

**Figure 10-9: Importing Certificate into Trusted Certificates Store**




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

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

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

# 10.7 Configuring Mutual TLS Authentication

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

## 10.7.1 TLS for SIP Clients

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

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

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

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

1. Enable two-way authentication on the specific SIP Interface:
   a. In the SIP Interface table (see "Configuring SIP Interfaces" on page 319), configure the 'TLS Mutual Authentication' parameter to **Enable** for the specific SIP Interface.
   b. Reset the device with a burn-to-flash for your settings to take effect.
2. Configure a TLS Context with the following certificates:
   - Import the certificate of the CA that signed the certificate of the SIP client into the Trusted Root Store so that the device can authenticate the client (see "Importing Certificates and Certificate Chain into Trusted Certificate Store" on page 113).
   - Make sure that the TLS certificate is signed by a CA that the SIP client trusts so that the client can authenticate the device.

## 10.7.2 TLS for Remote Device Management

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

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

1. On the Web Security Settings page (see "Configuring Web Security Settings" on page 73), configure the 'Secured Web Connection (HTTPS)' field to **HTTPS Only**. The setting ensures that you have a method for accessing the device in case the client certificate doesn't work. Restore the previous setting after testing the configuration.
2. In the TLS Contexts table (see "Configuring TLS Certificate Contexts" on page 103), select the required TLS Context row, and then click the **TLS Context Trusted Root Certificates** button, located below the table; the Trusted Certificates page appears.
3. Click the **Import** button, and then select the certificate file.
4. Wait until the import operation finishes successfully.
5. On the Web Security Settings page, configure the 'Requires Client Certificates for HTTPS connection' field to **Enable**.
6. Reset the device with a burn-to-flash for your settings to take effect.

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

- If the user has a client certificate from a CA that is listed in the Trusted Root Certificate file, the connection is accepted and the user is prompted for the system password.
- If both the CA certificate and the client certificate appear in the Trusted Root Certificate file, the user is not prompted for a password (thus, providing a single-sign-on experience - the authentication is performed using the X.509 digital signature).
- If the user does not have a client certificate from a listed CA or does not have a client certificate, the connection is rejected.

**Notes:**

- The process of installing a client certificate on your PC is beyond the scope of this document. For more information, refer to your operating system documentation, and/or consult your security administrator.
- The root certificate can also be loaded via the Automatic Update facility, using the HTTPSRootFileName *ini* file parameter.
- You can enable the device to check whether a peer's certificate has been revoked by an OCSP server per TLS Context (see "Configuring TLS Certificate Contexts" on page 103).

## 10.8 Configuring TLS Server Certificate Expiry Check

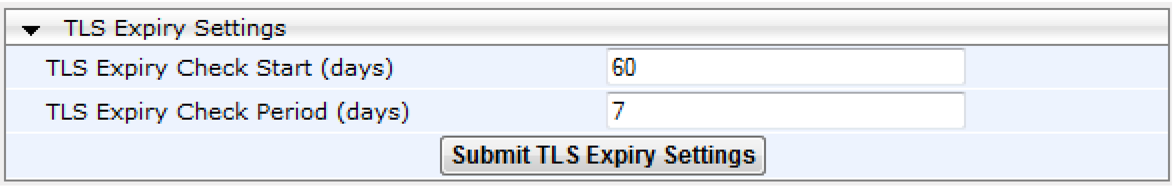
You can also configure the TLS Server Certificate Expiry Check feature, whereby the device periodically checks the validation date of the installed TLS server certificates. You can also configure the device to send a notification SNMP trap event (acCertificateExpiryNotification) at a user-defined number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

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

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

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

**Figure 10-10: TLS Expiry Settings Group**




3. In the 'TLS Expiry Check Start' field, enter the number of days before the installed TLS server certificate is to expire at which time the device sends an SNMP trap event to notify of this.
4. In the 'TLS Expiry Check Period' field, enter the periodical interval (in days) for checking the TLS server certificate expiry date. By default, the device checks the certificate every 7 days.
5. Click the **Submit TLS Expiry Settings** button.

**This page is intentionally left blank.**

# 11 Date and Time

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

## 11.1 Configuring Automatic Date and Time using SNTP

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

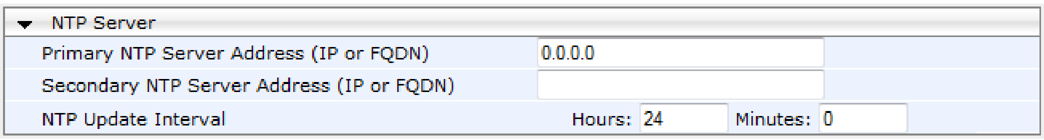
You can also configure the device to authenticate and validate the NTP messages received from the NTP server. Authentication is done using an authentication key with the MD5 cryptographic hash algorithm. When this feature is enabled, the device ignores NTP messages received without authentication.

The following procedure describes how to configure SNTP. For detailed descriptions of the configuration parameters, see "NTP and Daylight Saving Time Parameters" on page 667.

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

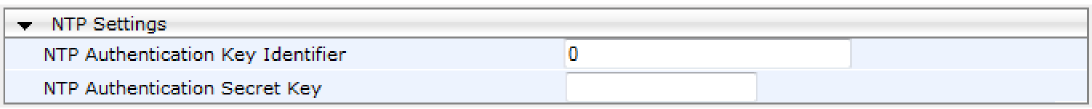
1. Open the Time and Date page (**Configuration** tab > **System** menu > **Time And Date**), and then scroll down to the 'NTP Sever' group:

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




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

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



5. Configure NTP message authentication:
   - In the 'NTP Authentication Key Identifier' field, configure the NTP authentication key identifier.
   - In the 'NTP Authentication Secret Key' field, configure the secret authentication key shared between the device and the NTP server.
6. Verify that the device has received the correct date and time from the NTP server. The date and time is displayed in the 'UTC Time' read-only field on the Time and Date page.

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

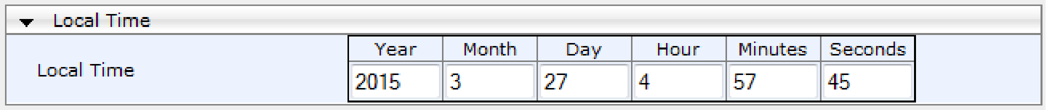
## 11.2 Configuring Date and Time Manually

You can manually configure the date and time of the device instead of using an NTP server (as described in ''Configuring Automatic Date and Time using SNTP'' on page 119).

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

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

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



Local Time

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

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

**Notes:**

- If the device is configured to obtain the date and time from an SNTP server, the fields on this page are read-only, displaying the date and time received from the NTP server.
- After performing a hardware reset, the date and time are returned to default values and thus, you should subsequently update the date and time.

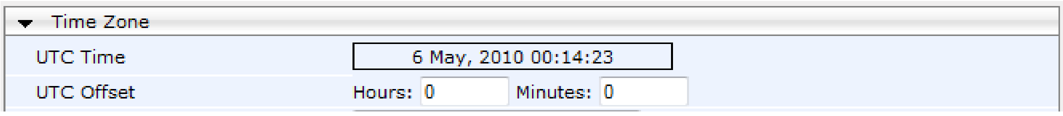
## 11.3 Configuring the Time Zone

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

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

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

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




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

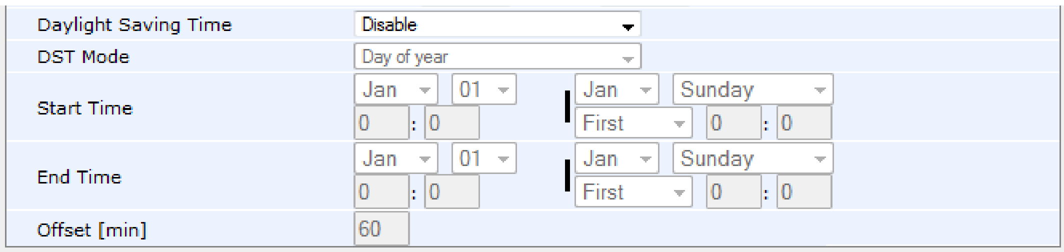
## 11.4 Configuring Daylight Saving Time

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

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

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

**Figure 11-5: Configuring DST**




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

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

**This page is intentionally left blank.**

# Part IV

## General VoIP Configuration

# 12 Network

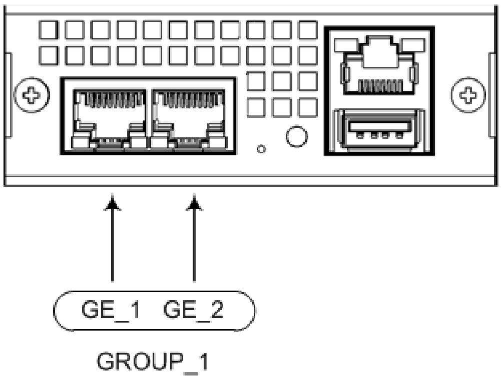
This section describes the network-related configuration.

## 12.1 Configuring Physical Ethernet Ports

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

The table also displays the status of the port (e.g., active) as well as the port group (*Ethernet Group*) to which the port belongs. You can assign up to two ports to an Ethernet Group. Ethernet Groups with two ports are used for 1+1 Ethernet port redundancy. For more information on Ethernet Groups and for assigning ports to Ethernet Groups, see "Configuring Ethernet Port Groups" on page 129.

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




To view the mapping of the physical ports to these logical ports (strings) as well as view port status, use the CLI command, **show voip ports**. This displays the MAC address and port status (up or down) of the physical port and its corresponding logical port.

**Note:** All the LAN ports have the same MAC address. This is the MAC address of the device itself.

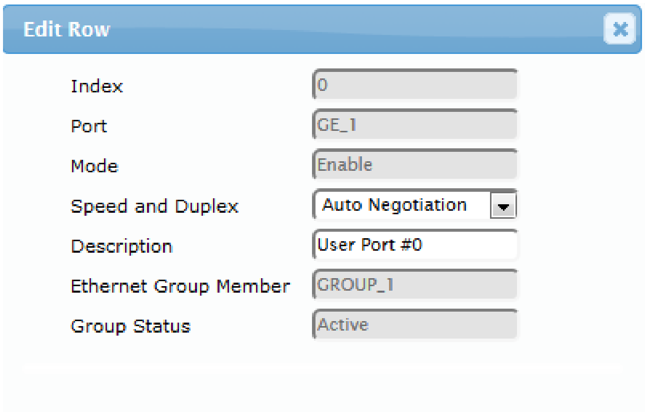
The following procedure describes how to configure the Ethernet ports through the Web interface. You can also configure it through ini file (PhysicalPortsTable) or CLI (configure voip > physical-port).

➢ **To configure the physical Ethernet ports:**

1. Open the Physical Ports Settings page (**Configuration** tab > **VoIP** menu > **Network** > **Physical Ports Table**).

2. Select a port that you want to configure by clicking its table row, and then clicking **Edit**; the following dialog box appears:

**Figure 12-1: Physical Ports Settings Table - Edit Row Dialog Box**




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

**Table 12-1: Physical Port Settings Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Port<br>`port`<br>[PhysicalPortsTable_Port] | (Read-only) Displays the Ethernet port number. The figure in the beginning of this section shows the mapping of this GUI port number to the physical port on the chassis. |
| Mode<br>`mode`<br>[PhysicalPortsTable_Mode] | (Read-only) Displays the mode of the port:<br>▪ **[0]** Disable<br>▪ **[1]** Enable (default) |
| Speed & Duplex<br>`speed-duplex`<br>[PhysicalPortsTable_SpeedDuplex] | Defines the speed and duplex mode of the port.<br>▪ **[0]** 10BaseT Half Duplex<br>▪ **[1]** 10BaseT Full Duplex<br>▪ **[2]** 100BaseT Half Duplex<br>▪ **[3]** 100BaseT Full Duplex<br>▪ **[4]** Auto Negotiation (default)<br>▪ **[6]** 1000BaseT Half Duplex<br>▪ **[7]** 1000BaseT Full Duplex |
| Description<br>`port-description`<br>[PhysicalPortsTable_PortDescription] | Defines an arbitrary description of the port.<br>By default, the value is "User Port #<row index>". |
| Group Member<br>`group-member`<br>[PhysicalPortsTable_GroupMember] | (Read-only) Displays the Ethernet Group to which the port belongs.<br>To assign the port to a different Ethernet Group, see "Configuring Ethernet Port Groups" on page 129. |
| Group Status<br>`group-status`<br>[PhysicalPortsTable_GroupStatus] | (Read-only) Displays the status of the port:<br>▪ Active port. When the Ethernet Group includes two ports and their transmit/receive mode is configured to 2RX 1TX or 2RX 2TX, both ports show "Active".<br>▪ "Redundant": Standby (redundant) port. |

## 12.2 Configuring Ethernet Port Groups

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

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

The maximum number of Ethernet Groups that can be configured is the same as the number of Ethernet ports provided by the device. Thus, the device supports up to 2 Ethernet Groups, each containing one port, or one Ethernet Group containing two ports. By default, an Ethernet Group is assigned the two ports; the other Ethernet Groups are empty.

You can assign Ethernet ports to IP network interfaces. This is done by first configuring an Ethernet Device with the required Ethernet Group containing the port or ports (see "Configuring Underlying Ethernet Devices" on page 132). Then by assigning the Ethernet Device to the IP network interface in the Interface table (see "Configuring IP Network Interfaces" on page 135). This enables physical separation of network interfaces, providing a higher level of segregation of sub-networks. Equipment connected to different physical ports is not accessible to one another; the only connection between them can be established by cross connecting them with media streams (VoIP calls).

The port names (strings) displayed in the Ethernet Group Settings table represent the physical ports on the device. For the mapping of these strings to the physical ports, see Configuring Physical Ethernet Ports on page 127.

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

**Notes:**

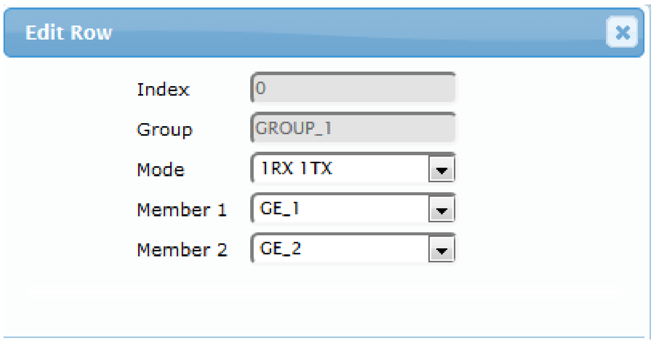
- Before you can re-assign a port to a different Ethernet Group, you must first remove the port from its current Ethernet Group. To remove the port, either set the 'Member' field to **None** or to a different port.
- As all the ports have the same MAC address, you must connect each port to a different Layer-2 switch.
- When implementing 1+1 Ethernet port redundancy, each port in the Ethernet Group (port pair) must be connected to a different switch, but in the same subnet.

➢ **To configure Ethernet Groups:**

1. Open the Ethernet Group Settings table (**Configuration** tab > **VoIP** menu > **Network** > **Ethernet Groups Table**).
2. If the port that you want to assign to a specific Ethernet Group is already associated with another Ethernet Group, you must first **remove** the port from the currently

associated Ethernet Group before you can associate it with the desired Ethernet Group:

a. Select the Ethernet Group to which the port is currently associated, and then click **Edit**; the following dialog box appears:




b. Set the 'Member 1' or 'Member 2' field (depending on where the port appears) to **None** (or assign it a different port).

c. Click **Submit**; the port is removed from the Ethernet Group.

3. Select the Ethernet Group that you want to configure and associate a port(s), and then click **Edit**.
4. Configure the Ethernet Group according to the parameters described in the table below.
5. Click **Submit**, and then save ("burn") your settings to flash memory.

**Table 12-2: Ethernet Group Settings Parameter Descriptions**

| Parameter | Description |
|---|---|
| Group<br>`group`<br>[EtherGroupTable_Group] | (Read-only) Displays the Ethernet Group number. |
| Mode<br>`mode`<br>[EtherGroupTable_Mode] | Defines the mode of operation of the ports in the Ethernet Group. This applies only to Ethernet Groups containing two ports.<br>▪ **[2]** 1RX/1TX = (Default) At any given time, only a single port in the Ethernet Group can transmit and receive packets. If a link exists on both ports, then the active one is either the first to have a link up or the lower-numbered port if both have the same link up from start.<br>▪ **[3]** 2RX/1TX = Both ports in the Ethernet Group can receive packets, but only one port can transmit. The transmitting port is determined arbitrarily by the device. If the selected port fails at a later stage, a switchover to the redundant port is done, which begins to transmit as well as receive.<br>▪ **[4]** 2RX/2TX = Both ports in the Ethernet Group can receive and transmit packets.<br>▪ **[5]** Single = If the Ethernet Group contains only one port, use this option.<br>▪ **[6]** None = If no port is assigned to the Ethernet Group, use this option.<br>**Notes:**<br>▪ It is recommended to use the 2RX/1TX option. In such a setup, the ports can be connected to the same LAN switch or each to a different switch where both are in the same subnet. |
| Member 1<br>`member1` | Assigns the first port to the Ethernet Group. To assign no port, set this field to **None**. |

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

## 12.3 Configuring Underlying Ethernet Devices

The Ethernet Device table lets you configure up to 16 *Ethernet Devices*. An Ethernet Device represents a Layer-2 bridging device and is assigned a VLAN ID and an Ethernet Port Group. Multiple Ethernet Devices can be associated with the same Ethernet Group.

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

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

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

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

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

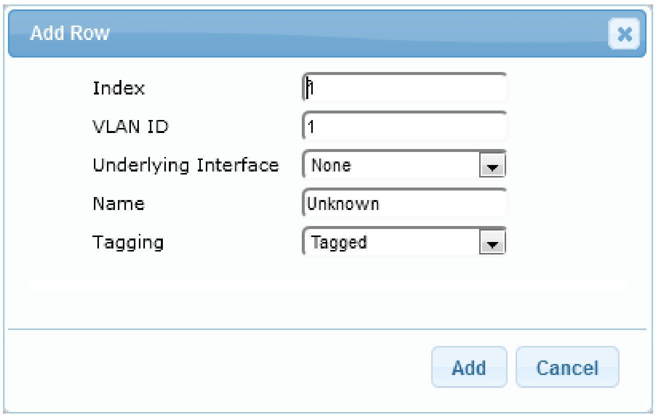
You can view configured Ethernet Devices that have been successfully applied to the device (saved to flash), in the Ethernet Device Status table. This page is accessed by clicking the **Ethernet Device Status Table** button, located at the bottom of the Ethernet Device table. The Ethernet Device Status table can also be accessed from the **Status & Diagnostics** tab > **VoIP Status** menu > **Ethernet Device Status Table** (see "Viewing Ethernet Device Status" on page 561).

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

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

- **To configure an Ethernet Device:**

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




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

**Table 12-3: Ethernet Device Table Parameter Descriptions**

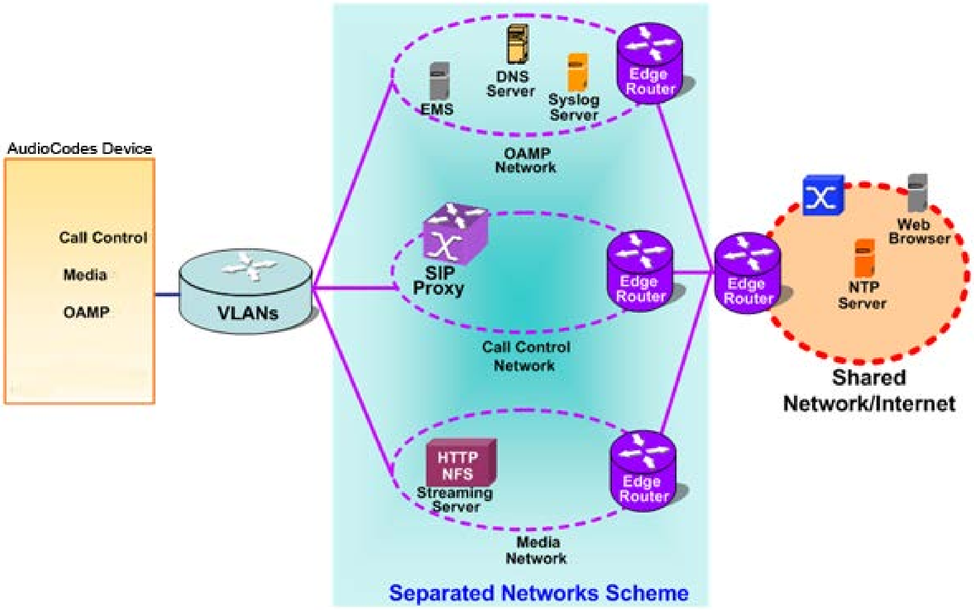
| Parameter | Description |
|---|---|
| Index<br>[DeviceTable_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| VLAN ID<br>`vlan-id`<br>[DeviceTable_VlanID] | Defines a VLAN ID for the Ethernet Device.<br>The valid value is 1 to 3999. The default value is 1.<br>**Note:** Each Ethernet Group must have a unique VLAN ID. |
| Underlying Interface<br>`underlying-if`<br>[DeviceTable_UnderlyingInterface] | Assigns an Ethernet Group to the Ethernet Device. For configuring Ethernet Groups, see Configuring Ethernet Port Groups on page 129.<br>**Note:** The parameter is mandatory. |
| Name<br>`name`<br>[DeviceTable_DeviceName] | Defines a name for the Ethernet Device. This name is used to associate the Ethernet Device with an IP network interface in the Interface table ('Underlying Device' field - see "Configuring IP Network Interfaces" on page 135) and/or with a static route in the Static Route table ('Device Name' field - see "Configuring Static IP Routing" on page 143). |

| Parameter | Description |
|---|---|
| Tagging<br>`tagging`<br>[DeviceTable_Tagging] | Defines VLAN tagging per Ethernet Device.<br>▪ [0] Untagged = (Default of pre-configured Ethernet Device) The Ethernet Device accepts untagged packets as well as packets with the same VLAN ID as configured for the Ethernet Device. Incoming untagged packets are assigned the VLAN ID of the Ethernet Device. The Ethernet Device sends these VLAN packets untagged (i.e., removes the VLAN ID).<br>▪ [1] Tagged = (Default for new Ethernet Devices) The Ethernet Device accepts packets that have the same VLAN ID as configured for the Ethernet Device and sends packets with this VLAN ID. For all Ethernet Devices that are associated with the same Ethernet Group ('Underlying Interface') and set to **Tagged**, incoming untagged packets received on this Ethernet Group are discarded.<br>**Note:** Only one Ethernet Device can be configured as Untagged per associated Ethernet Group (port group). In other words, if multiple Ethernet Devices are associated with the same Ethernet Group, only one of these Ethernet Devices can be set to **Untagged**; all the others must be set to **Tagged**. |

## 12.4 Configuring IP Network Interfaces

You can configure a single VoIP network interface for all applications, including OAMP (management traffic), call control (SIP signaling messages), and media (RTP traffic), or you can configure multiple logical, IP network interfaces for these applications. You may need to logically separated network segments for these applications for administration and security. This can be achieved by employing Layer-2 VLANs and Layer-3 subnets. The figure below illustrates a typical network architecture where the device is configured with three network interfaces, each representing the OAMP, call control, and media applications. The device is connected to a VLAN-aware switch for directing traffic from and to the device to the three separated Layer-3 broadcast domains according to VLAN tags (middle pane).

**Figure 12-2: Multiple Network Interfaces**




The device is shipped with a default OAMP interface. For more information, see "Default OAMP IP Address" on page 31. The Interface table lets you change this OAMP interface and configure additional network interfaces for control and media, if necessary. You can configure up to 12 interfaces, consisting of up to 11 Control and Media interfaces, and 1 OAMP interface. Each IP interface is configured with the following:

- Application type allowed on the interface:
  - Control: call control signaling traffic (i.e., SIP)
  - Media: RTP traffic
  - Operations, Administration, Maintenance and Provisioning (OAMP): management (i.e., Web, CLI, and SNMP based management)
- IP address (IPv4 and IPv6) and subnet mask (prefix length)
- For configuring Quality of Service (QoS), see "Configuring the QoS Settings" on page 146.

- Default Gateway: Traffic from this interface destined to a subnet that does not meet any of the routing rules (local or static) are forwarded to this gateway
- Primary and secondary domain name server (DNS) addresses (optional)
- Underlying Ethernet Device: Layer-2 bridging device and assigned a VLAN ID. As the Ethernet Device is associated with an Ethernet Group, this is useful for setting trusted and un-trusted networks on different physical Ethernet ports. Multiple entries in the Interface table may be associated with the same Ethernet Device, providing multi-homing IP configuration (i.e., multiple IP addresses on the same interface/VLAN).

Complementing the Interface table is the Static Route table, which lets you configure static routing rules for non-local hosts/subnets. For more information, see "Configuring Static IP Routing" on page 143.

**Note:** Before configuring IP interfaces, it is recommended that you read the IP interface configuration guidelines in "Interface Table Configuration Guidelines" on page 138.

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

### ➢ To configure IP network interfaces:

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

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

Page 1 of 1 | 10 | View 1 - 2 of 2

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

**Notes:**

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

To view configured network interfaces that are currently active, click the **IP Interface Status Table** button. For more information, see "Viewing Active IP Interfaces" on page 560.

**Table 12-4: Interface Table Parameters Description**

| Parameter | Description |
|---|---|
| **Table parameters** | |
| Index<br>`network-if`<br>[InterfaceTable_Index] | Table index row of the interface.<br>The range is 0 to 11. |
| Application Type<br>`application-type`<br>[InterfaceTable_ApplicationTypes] | Defines the applications allowed on the interface.<br>▪ **[0]** OAMP = Operations, Administration, Maintenance and Provisioning (OAMP) applications (e.g., Web, Telnet, SSH, and SNMP).<br>▪ **[1]** Media = Media (i.e., RTP streams of voice).<br>▪ **[2]** Control = Call Control applications (e.g., SIP).<br>▪ **[3]** OAMP + Media = OAMP and Media applications.<br>▪ **[4]** OAMP + Control = OAMP and Call Control applications.<br>▪ **[5]** Media + Control = Media and Call Control applications.<br>▪ **[6]** OAMP + Media + Control = All application types are allowed on the interface. |
| Interface Mode<br>[InterfaceTable_InterfaceMode] | Defines the method that the interface uses to acquire its IP address.<br>▪ [3] IPv6 Manual Prefix = IPv6 manual prefix IP address assignment. The IPv6 prefix (higher 64 bits) is set manually while the interface ID (the lower 64 bits) is derived from the device's MAC address.<br>▪ [4] IPv6 Manual = IPv6 manual IP address (128 bits) assignment.<br>▪ [10] IPv4 Manual = IPv4 manual IP address (32 bits) assignment. |
| IP Address<br>`ip-address`<br>[InterfaceTable_IPAddress] | Defines the IPv4/IPv6 address, in dotted-decimal notation. |
| Prefix Length<br>`prefix-length`<br>[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).<br>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, |

| Parameter | Description |
|---|---|
| | it uses the variable-length subnet masking technique to allow allocation on arbitrary-length prefixes.<br>The prefix length for IPv4 must be set to a value from 0 to 30. The prefix length for IPv6 must be set to a value from 0 to 64. |
| Default Gateway<br>`gateway`<br>[InterfaceTable_Gateway] | Defines the IP address of the default gateway for the interface. When traffic is sent from this interface to an unknown destination (i.e., not in the same subnet and not defined for any static routing rule), it is forwarded to this default gateway. |
| Interface Name<br>`name`<br>[InterfaceTable_InterfaceName] | Defines a name for the interface. This name is used in various configuration tables to associate the network interface with other configuration entities such as Media Realms. It is also displayed in management interfaces (Web, CLI, and SNMP) for clarity where it has no functional use.<br>The valid value is a string of up to 16 characters. |
| Primary DNS<br>`primary-dns`<br>[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.<br>By default, no IP address is defined. |
| Secondary DNS<br>`secondary-dns`<br>[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.<br>By default, no IP address is defined. |
| Underlying Device<br>`underlying-dev`<br>[InterfaceTable_UnderlyingDevice] | Assigns an Ethernet Device to the IP interface. An Ethernet Device is a VLAN ID associated with a physical Ethernet port (Ethernet Group). To configure Ethernet Devices, see Configuring Underlying Ethernet Devices on page 132. |

### 12.4.1 Assigning NTP Services to Application Types

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

### 12.4.2 Multiple Interface Table Configuration Summary and Guidelines

The Interface table configuration must adhere to the following rules:

- Multiple Control and Media interfaces can be configured with overlapping IP addresses and subnets.
- The prefix length replaces the dotted-decimal subnet mask presentation and **must** have a value of 0-30 for IPv4 addresses and a value of 0-64 for IPv6 addresses.
- **One** OAMP interface must be configured and this **must** be an IPv4 address. This OAMP interface can be combined with Media and Control.
- At least one Control interface **must** be configured.
- At least one Media interface **must** be configured.
- Multiple Media and/or Control interfaces can be configured with an IPv6 address.
- The network interface types can be combined:
    - Example 1:

- One combined OAMP-Media-Control interface with an IPv4 address
- Example 2:
  - One OAMP interface with an IPv4 address
  - One or more Control interfaces with IPv4 addresses
  - One or more Media interfaces with IPv4 interfaces
- Example 3:
  - One OAMP with an IPv4 address
  - One combined Media-Control interface with IPv4 address
  - One combined Media-Control interface with IPv6 address

- Each network interface can be configured with a Default Gateway. The address of the Default Gateway **must** be in the same subnet as the associated interface. Additional static routing rules can be configured in the Static Route table.
- The interface name **must** be configured (mandatory) and must be unique for each interface.
- For IPv4 addresses, the 'Interface Mode' column must be set to IPv4 Manual. For IPv6 addresses, this column must be set to IPv6 Manual or IPv6 Manual Prefix.

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

## 12.4.3 Networking Configuration Examples

This section provides configuration examples of networking interfaces.

### 12.4.3.1 One VoIP Interface for All Applications

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

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

**Table 12-5: Example of Single VoIP Interface in Interface Table**

| Index | Application Type | Interface Mode | IP Address | Prefix Length | Default Gateway | Underlying Device | Interface Name |
|---|---|---|---|---|---|---|---|
| 0 | OAMP, Media & Control | IPv4 | 192.168.0.2 | 16 | 192.168.0.1 | 1 | myInterface |

2. **Static Route table:** Two routes are configured for directing traffic for subnet 201.201.0.0/16 to 192.168.11.10, and all traffic for subnet 202.202.0.0/16 to 192.168.11.1:

**Table 12-6: Example of Static Route Table**

| Destination | Prefix Length | Gateway |
|---|---|---|
| 201.201.0.0 | 16 | 192.168.11.10 |
| 202.202.0.0 | 16 | 192.168.11.1 |

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

#### 12.4.3.2 VoIP Interface per Application Type

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

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

**Table 12-7: Example of VoIP Interfaces per Application Type in Interface Table**

| Index | Application Type | Interface Mode | IP Address | Prefix Length | Default Gateway | Underlying Device | Interface Name |
|---|---|---|---|---|---|---|---|
| 0 | OAMP | IPv4 Manual | 192.168.0.2 | 16 | 192.168.0.1 | 1 | ManagementIF |
| 1 | Control | IPv4 Manual | 200.200.85.14 | 24 | 200.200.85.1 | 200 | myControlIF |
| 2 | Media | IPv4 Manual | 211.211.85.14 | 24 | 211.211.85.1 | 211 | myMediaIF |

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

**Table 12-8: Example Static Route Table**

| Destination | Prefix Length | Gateway |
|---|---|---|
| 176.85.49.0 | 24 | 192.168.11.1 |

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

### 12.4.3.3 VoIP Interfaces for Combined Application Types

This example describes the configuration of multiple interfaces for the following applications:

- One interface for the OAMP application.
- Interfaces for Call Control and Media applications, where two of them are IPv4 interfaces and one is an IPv6 interface.

1. **Interface table:**

**Table 12-9: Example of VoIP Interfaces of Combined Application Types in Interface Table**

| Index | Application Type | Interface Mode | IP Address | Prefix Length | Default Gateway | Underlying Device | Interface Name |
|---|---|---|---|---|---|---|---|
| 0 | OAMP | IPv4 Manual | 192.168.0.2 | 16 | 192.168.0.1 | 1 | Mgmt |
| 1 | Media & Control | IPv4 Manual | 200.200.85.14 | 24 | 200.200.85.1 | 201 | MediaCntrl1 |
| 2 | Media & Control | IPv4 Manual | 200.200.86.14 | 24 | 200.200.86.1 | 202 | MediaCntrl2 |
| 3 | Media & Control | IPv6 Manual | 2000::1:200:200:86:14 | 64 | :: | 202 | V6CntrlMedia2 |

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

**Table 12-10: Example of Static Route Table**

| Destination | Prefix Length | Gateway |
|---|---|---|
| 176.85.49.0 | 24 | 192.168.0.10 |

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

```
EnableNTPasOAM = 1
```

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

#### 12.4.3.4 VoIP Interfaces with Multiple Default Gateways

Below is a configuration example using default gateways per IP network interface. In this example, the default gateway for OAMP is 192.168.0.1 and for Media and Control it is 200.200.85.1.

**Table 12-11: Configured Default Gateway Example**

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

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

**Table 12-12: Separate Static Route Table Example**

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

## 12.5 Configuring Static IP Routes

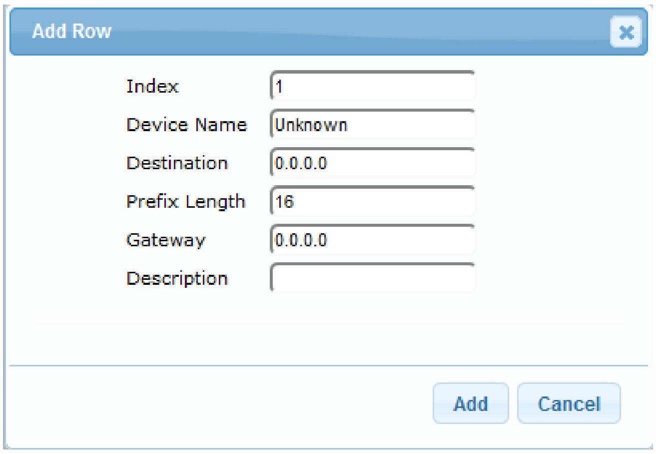
The Static Route table lets you configure up to 30 static IP routing rules. Using static routes lets you communicate with LAN networks that are not located behind the Default Gateway specified for the IP network interface, configured in the Interface table, from which the packets are sent. Before sending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

You can view the status of the configured static routes in the IP Routing Status table. This page can be accessed by clicking the **Static Route Status Table** button, located at the bottom of the Static Route table page, or it can be accessed from the Navigation tree under the **Status & Diagnostics** tab (see "Viewing Static Routes Status" on page 561).

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

➢ **To configure a static IP route:**

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




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

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

**Table 12-13: Static Route Table Parameter Descriptions**

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Index<br>[StaticRouteTable_Index]</td><td>Defines an index number for the new table row.<br>The valid value is 0 to 29.<br>Note: Each row must be configured with a unique index.</td></tr>
<tr><td>Device Name<br><code>device-name</code><br>[StaticRouteTable_DeviceName]</td><td>Assigns an IP network interface through which the static route's Gateway is reached. The Device Name (or underlying device) represents the IP network interface, including VLAN ID and associated physical port(s).<br>The value must be identical to the value in the 'Underlying Device' parameter of the required IP network interface in the Interface table (see Configuring IP Network Interfaces on page 135).<br>For configuring Ethernet Devices, see Configuring Underlying Ethernet Devices on page 132.</td></tr>
<tr><td>Destination<br><code>destination</code><br>[StaticRouteTable_Destination]</td><td>Defines the IP address of the destination host/network. The destination can be a single host or a whole subnet, depending on the prefix length configured for this routing rule.</td></tr>
<tr><td>Prefix Length<br><code>prefix-length</code><br>[StaticRouteTable_PrefixLength]</td><td>Defines the Classless Inter-Domain Routing (CIDR)-style representation of a dotted-decimal subnet notation of the destination host/network. The CIDR-style representation uses a suffix indicating the number of bits that are set in the dotted-decimal format. For example, the value 16 represents subnet 255.255.0.0. The value must be 0 to 31 for IPv4 interfaces and a value of 0 to 64 for IPv6 interfaces.</td></tr>
<tr><td colspan="2">The address of the host/network you want to reach is determined by an AND operation that is applied to the fields 'Destination' and 'Prefix Length'. For example, to reach the network 10.8.x.x, enter 10.8.0.0 in the 'Destination' field and 16 in the 'Prefix Length'. As a result of the AND operation, the value of the last two octets in the 'Destination' field is ignored. To reach a specific host, enter its IP address in the 'Destination' field and 32 in the 'Prefix Length' field.</td></tr>
<tr><td>Gateway<br><code>gateway</code><br>[StaticRouteTable_Gateway]</td><td>Defines the IP address of the Gateway (next hop) used for traffic destined to the subnet/host defined in the 'Destination' / 'Prefix Length' field.<br>Notes:<ul><li>The Gateway's address must be in the same subnet as the IP address of the network interface that is associated with the static route (using the 'Device Name' parameter - see above).</li><li>The IP network interface associated with the static route must be of the same IP address family (IPv4 or IPv6).</li></ul></td></tr>
<tr><td>Description<br><code>description</code><br>[StaticRouteTable_Description]</td><td>Defines an arbitrary name to easily identify the static route rule.<br>The valid value is a string of up to 20 characters.</td></tr>
</table>

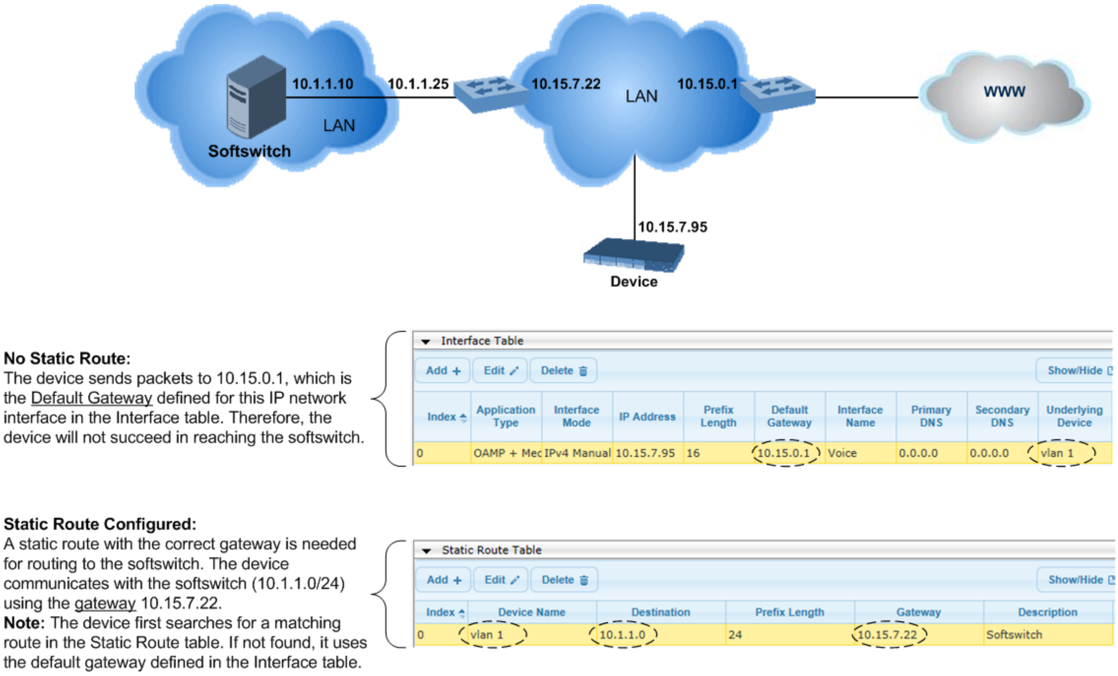
### 12.5.1 Configuration Example of Static IP Routes

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

Note the following configuration:

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

**Figure 12-3: Example of using a Static Route**




## 12.5.2 Troubleshooting the Routing Table

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

Common static routing configuration errors may include the following:

- The IP address specified in the 'Gateway' field is unreachable from the IP network interface associated with the static route.
- The same destination is configured in two different static routes.
- More than 30 static routes have been configured.

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

## 12.6 Configuring Quality of Service

The QoS Settings page lets you configure Layer-2 and Layer-3 Quality of Service (QoS). Differentiated Services (DiffServ) is an architecture providing different types or levels of service for IP traffic. DiffServ (according to RFC 2474), prioritizes certain traffic types based on priority, accomplishing a higher-level QoS at the expense of other traffic types. By prioritizing packets, DiffServ routers can minimize transmission delays for time-sensitive packets such as VoIP packets.

You can assign DiffServ to the following class of services (CoS) and assign VLAN priorities (IEEE 802.1p) to various values of DiffServ:

- Media Premium – RTP packets sent to the LAN
- Control Premium – control protocol (SIP) packets sent to the LAN
- Gold – HTTP streaming packets sent to the LAN
- Bronze – OAMP packets sent to the LAN

The Layer-3 QoS parameters define the values of the DiffServ field in the IP header of the frames related to a specific service class. The Layer-2 QoS parameters define the values for the 3 priority bits in the VLAN tag according to the value of the DiffServ field in the packet IP header (according to the IEEE 802.1p standard). The DiffServ table lets you configure up to 64 DiffServ-to-VLAN Priority mapping (Layer 2 class of service). For each packet sent to the LAN, the VLAN Priority of the packet is set according to the DiffServ value in the IP header of the packet.

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

**Table 12-14: Traffic/Network Types and Priority**

| Application | Traffic / Network Types | Class-of-Service (Priority) |
|---|---|---|
| **Debugging interface** | Management | Bronze |
| **Telnet** | Management | Bronze |
| **DHCP** | Management | Network |
| **Web server (HTTP)** | Management | Bronze |
| **SNMP GET/SET** | Management | Bronze |
| **Web server (HTTPS)** | Management | Bronze |
| **RTP traffic** | Media | Premium media |
| **RTCP traffic** | Media | Premium media |
| **T.38 traffic** | Media | Premium media |
| **SIP** | Control | Premium control |
| **SIP over TLS (SIPS)** | Control | Premium control |
| **Syslog** | Management | Bronze |
| **SNMP Traps** | Management | Bronze |

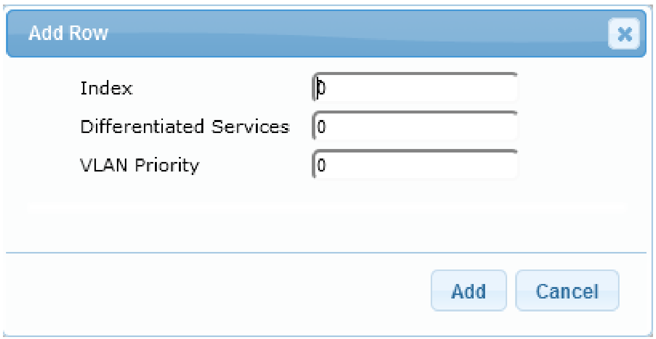
| Application | Traffic / Network Types | Class-of-Service (Priority) |
|---|---|---|
| **DNS client** | Varies according to DNS settings:<br>▪ OAMP<br>▪ Control | Depends on traffic type:<br>▪ Control: Premium Control<br>▪ Management: Bronze |
| **NTP** | Varies according to the interface type associated with NTP (see "Assigning NTP Services to Application Types" on page 138):<br>▪ OAMP<br>▪ Control | Depends on traffic type:<br>▪ Control: Premium control<br>▪ Management: Bronze |

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

➢ **To configure QoS:**

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

**Figure 12-4: DiffServ Table Page - Add Row Dialog Box**



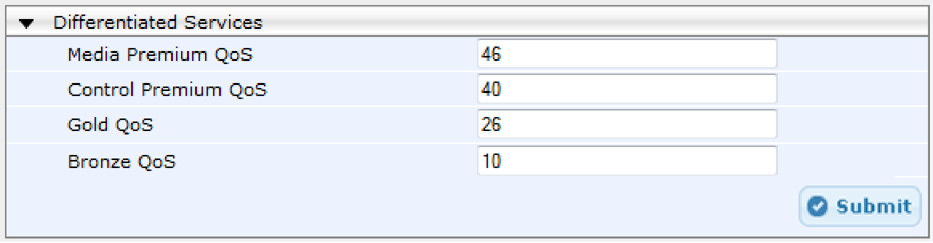

   b. Configure a DiffServ-to-VLAN priority mapping (Layer-2 QoS) according to the parameters described in the table below.
   c. Click Add, and then save ("burn") your settings to flash memory.

**Table 12-15: DiffServ Table Parameter Descriptions**

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

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

**Figure 12-5: QoS Settings Page - Differentiated Services**




## 12.7 Configuring ICMP Messages

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

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

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

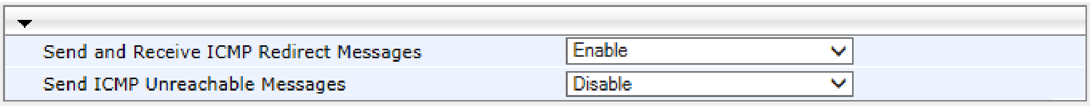
  This feature is applicable to IPv4 and IPv6 addressing schemes.

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

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

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

**Figure 12-6: Configuring ICMP Messaging in Network Settings Page**




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

# 12.8 DNS

You can use the device's embedded domain name server (DNS) or an external, third-party DNS to translate domain names into IP addresses. This is useful if domain names are used as the destination in call routing. The device supports the configuration of the following DNS types:

- Internal DNS table - see "Configuring the Internal DNS Table" on page 149
- Internal SRV table - see "Configuring the Internal SRV Table" on page 150

## 12.8.1 Configuring the Internal DNS Table

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

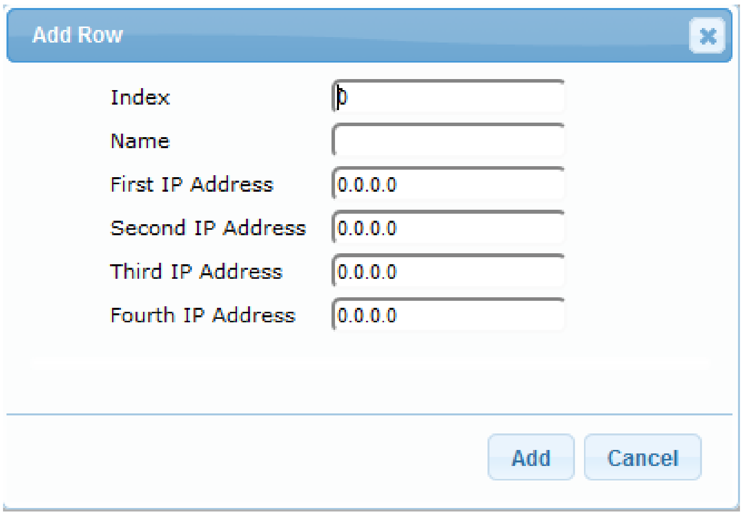
> **Note:** The device initially attempts to resolve a domain name using the Internal DNS table. If the domain name is not configured in the table, the device performs a DNS resolution using an external DNS server for the related IP network interface (see "Configuring IP Network Interfaces" on page 135).

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

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

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

**Figure 12-7: Internal DNS Table - Add Row Dialog Box**



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

**Table 12-16: Internal DNS Table Parameter Description**

| Parameter | Description |
|---|---|
| Domain Name<br>`domain-name`<br>[Dns2Ip_DomainName] | Defines the host name to be translated.<br>The valid value is a string of up to 31 characters. |
| First IP Address<br>`first-ip-address`<br>[Dns2Ip_FirstIpAddress] | Defines the first IP address (in dotted-decimal format notation) to which the host name is translated. The IP address can be configured as an IPv4 and/or IPv6 address. |
| Second IP Address<br>`second-ip-address`<br>[Dns2Ip_SecondIpAddress] | Defines the second IP address (in dotted-decimal format notation) to which the host name is translated. |
| Third IP Address<br>`third-ip-address`<br>[Dns2Ip_ThirdIpAddress] | Defines the third IP address (in dotted-decimal format notation) to which the host name is translated. |
| Fourth IP Address<br>`fourth-ip-address`<br>[Dns2Ip_FourthIpAddress] | Defines the fourth IP address (in dotted-decimal format notation) to which the host name is translated.<br>**Note:** Currently, this parameter is not supported. |

## 12.8.2 Configuring the Internal SRV Table

The Internal SRV table resolves host names to DNS A-Records. Three different A-Records can be assigned to each host name, where each A-Record contains the host name, priority, weight, and port.

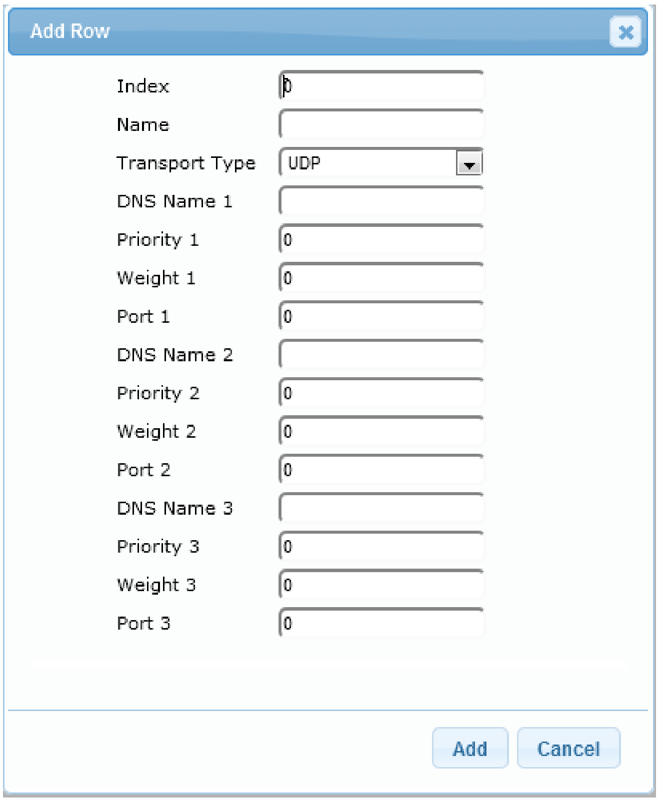
**Note:** If you configure the Internal SRV table, the device initially attempts to resolve a domain name using this table. If the domain is not configured in the table, the device performs a Service Record (SRV) resolution using an external DNS server, configured in the Interface table (see "Configuring IP Network Interfaces" on page 135).

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

➢ **To configure an SRV rule:**

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

**Figure 12-8: Internal SRV Table - Add Row Dialog Box**




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

**Table 12-17: Internal SRV Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Domain Name<br>`domain-name`<br>[Srv2Ip_InternalDomain] | Defines the host name to be translated.<br>The valid value is a string of up to 31 characters. By default, no value is defined. |
| Transport Type<br>`transport-type`<br>[Srv2Ip_TransportType] | Defines the transport type.<br>▪ **[0]** UDP (default)<br>▪ **[1]** TCP<br>▪ **[2]** TLS |
| DNS Name (1-3)<br>`dns-name-1\|2\|3`<br>[Srv2Ip_Dns1/2/3] | Defines the first, second or third DNS A-Record to which the host name is translated.<br>By default, no value is defined. |
| Priority (1-3)<br>`priority-1\|2\|3`<br>[Srv2Ip_Priority1/2/3] | Defines the priority of the target host. A lower value means that it is more preferred.<br>By default, no value is defined. |
| Weight (1-3)<br>`weight-1\|2\|3`<br>[Srv2Ip_Weight1/2/3] | Defines a relative weight for records with the same priority.<br>By default, no value is defined. |

| Parameter | Description |
|---|---|
| Port (1-3)<br>`port-1\|2\|3`<br>[Srv2Ip_Port1/2/3] | Defines the TCP or UDP port on which the service is to be found.<br>By default, no value is defined. |

# 12.9 Network Address Translation Support

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

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

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

## 12.9.1 Device Located behind NAT

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

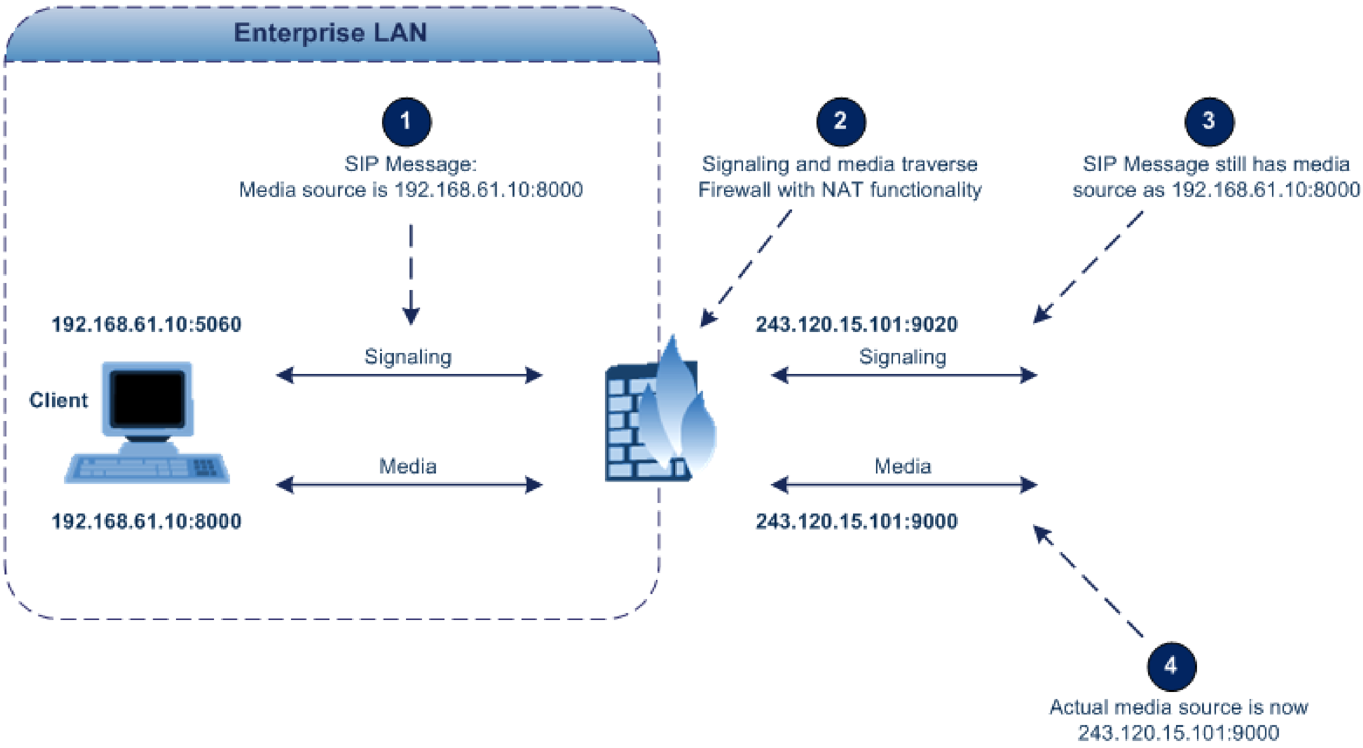
a. If configured, uses the single Static NAT IP address for all interfaces - see "Configuring a Static NAT IP Address for All Interfaces" on page 154.

b. If configured, uses the NAT Translation table which configures NAT per interface - see Configuring NAT Translation per IP Interface on page 154.

If NAT is not configured by any of the above-mentioned methods, the device sends the packet according to its IP address configured in the Interface table.

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

**Figure 12-9: Device behind NAT and NAT Issues**




## 12.9.1.1 Configuring a Static NAT IP Address for All Interfaces

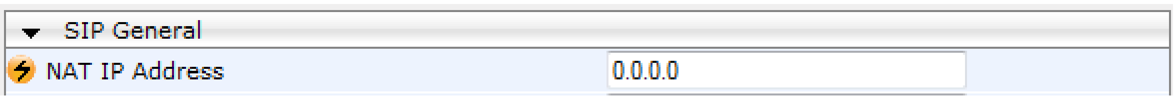
You can configure a global (public) IP address of the router to enable static NAT between the device and the Internet for all network interfaces. Thus, the device replaces the source IP address for media of all outgoing SIP messages sent on any of its network interfaces to this public IP address.

The following procedure describes how to configure a static NAT address through the Web interface. You can also configure it through ini file (StaticNATIP) or CLI (configure voip > sip-definition general-settings > nat-ip-addr).

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

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

**Figure 12-10: Configuring Static NAT IP Address in SIP General Parameters Page**




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

## 12.9.1.2 Configuring NAT Translation per IP Interface

The NAT Translation table lets you configure up to 32 network address translation (NAT) rules for translating source IP addresses into NAT IP addresses (*global* - *public*) when the device is located behind NAT. The device's NAT traversal mechanism replaces the source

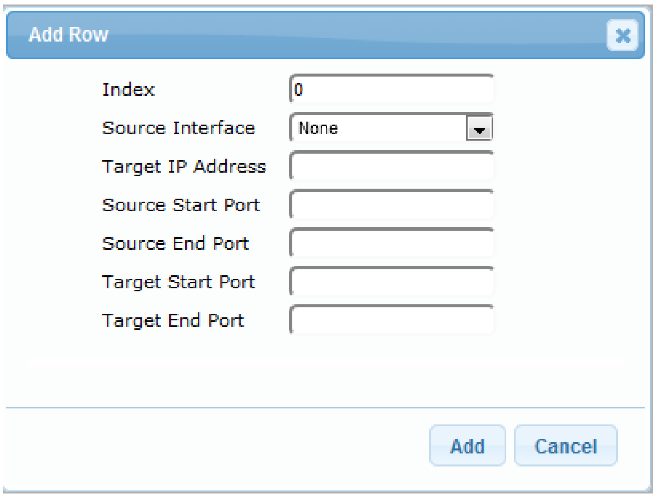
IP address of SIP messages sent from a specific VoIP interface (Control and/or Media) in the IP Interfaces table to a public IP address. This allows, for example, the separation of VoIP traffic between different ITSP's, and topology hiding of internal IP addresses from the "public" network. Each IP network interface, configured in the Interface table, can be associated with a NAT rule, translating the source IP address and port of the outgoing packet into the NAT address (IP address and port range).

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

- **To configure NAT translation rules:**

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

**Figure 12-11: NAT Translation Table - Add Row Dialog Box**




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

**Table 12-18: NAT Translation Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>`index`<br>[NATTranslation_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Source Interface<br>`src-interface-name`<br>[NATTranslation_SrcIPInterfaceName] | Assigns an IP network interface to the rule. Outgoing packets sent from the specified network interface are NAT'ed.<br>By default, no value is defined (**None**).<br>For configuring IP network interfaces, see "Configuring IP Network Interfaces" on page 135. |
| Target IP Address<br>`target-ip-address`<br>[NATTranslation_TargetIPAddress] | Defines the global (public) IP address. The device adds the address to the SIP Via header, Contact header, 'o=' SDP field, and 'c=' SDP field, in the outgoing packet. |

| Parameter | Description |
|---|---|
| Source Start Port<br>`src-start-port`<br>[NATTranslation_SourceStartPort] | Defines the optional starting port range (0-65535) of the IP interface, used as matching criteria for the NAT rule. If not configured, the match is done on the entire port range. Only IP addresses and ports of matched source ports will be replaced. |
| Source End Port<br>`src-end-port`<br>[NATTranslation_SourceEndPort] | Defines the optional ending port range (0-65535) of the IP interface, used as matching criteria for the NAT rule. If not configured, the match is done on the entire port range. Only IP addresses and ports of matched source ports will be replaced. |
| Target Start Port<br>`target-start-port`<br>[NATTranslation_TargetStartPort] | Defines the optional starting port range (0-65535) of the global address. If not configured, the ports are not replaced. Matching source ports are replaced with the target ports. This address is set in the SIP Via and Contact headers, as well as in the o= and c= SDP fields. |
| Target End Port<br>`target-end-port`<br>[NATTranslation_TargetEndPort] | Defines the optional ending port range (0-65535) of the global address. If not configured, the ports are not replaced. Matching source ports are replaced with the target ports. This address is set in the SIP Via and Contact headers, as well as in the o= and c= SDP fields. |

## 12.9.2 Remote UA behind NAT

### 12.9.2.1 SIP Signaling Messages

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

If necessary, you can also configure the device to always consider incoming SIP INVITE messages as sent from endpoints that are located behind NAT. When this is enabled, the device sends responses to the INVITE (to the endpoint), using the the source IP address of the packet (INVITE) initially received from the endpoint. This is especially useful in scenarios where the endpoint is located behind a NAT firewall and the device (for whatever reason) is unable to identify NAT using its regular NAT Detection mechanism. This feature is enabled per specific calls using IP Groups. To configure this feature, use the 'Always Use Source Address' parameter in the IP Group table (see "Configuring IP Groups" on page 323). If this feature is disabled, the device's NAT detection is according to the settings of the global parameter, 'SIP NAT Detection' parameter.

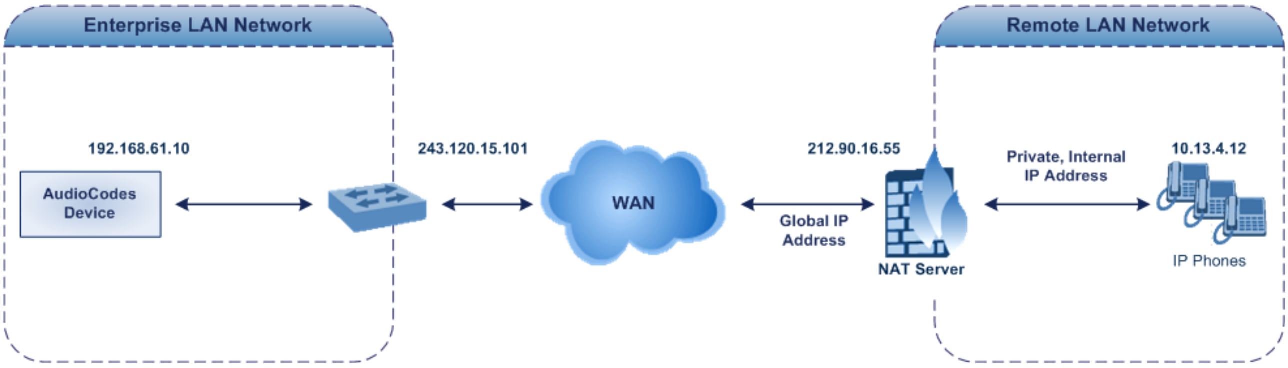
### 12.9.2.2 Media (RTP/RTCP/T.38)

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

- First Incoming Packet Mechanism - see "First Incoming Packet Mechanism" on page 157
- RTP No-Op packets according to the avt-rtp-noop draft - see "No-Op Packets" on page 158

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

**Figure 12-12: Remote UA behind NAT**




### 12.9.2.2.1 First Incoming Packet Mechanism

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

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

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

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

  Note: If the SIP session is established (ACK) and the device (not the UA) sends the first packet, it sends it to the address obtained from the SIP message and only after the device receives the first packet from the UA does it determine whether the UA is behind NAT.
- [1] Disable NAT: (Default) The device considers the UA as not located behind NAT and sends media packets to the UA using the IP address:port specified in the SDP 'c='

line (Connection) of the first received SIP message.

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

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

1. Open the General Settings page (**Configuration** tab > **VoIP** menu > **Media** > **General Media Settings**).
2. Set the 'NAT Mode' parameter (NATMode).
3. Click **Submit**.

##### 12.9.2.2.2 No-Op Packets

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

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

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

**Note:**

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

##### 12.9.2.2.3 Fax Transmission behind NAT

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

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

## 12.10 Robust Receipt of Media Streams by Media Latching

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

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

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

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

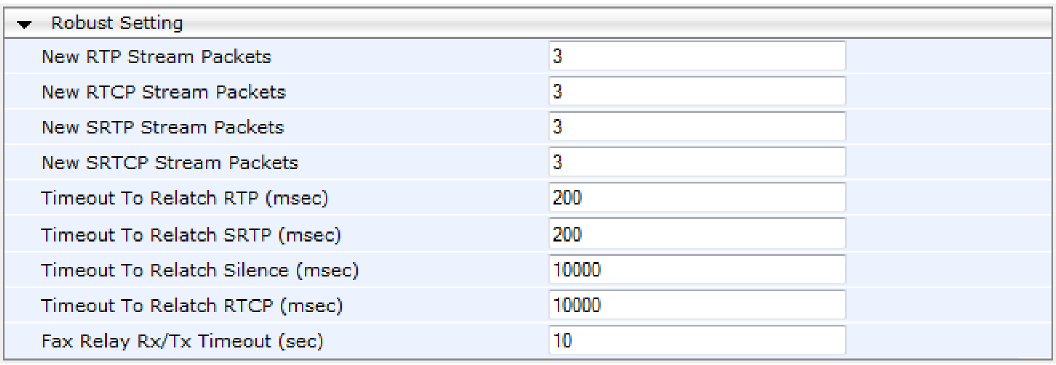
Latching onto a new T.38 stream is reported in CDR using the CDR fields, LatchedT38Ip (new IP address) and LatchedT38Port (new port). In addition, the SIP PUBLISH message updates the latched RTP SSRC, for example:

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

➢ **To configure media latching:**

1. Define the Robust Media method, using the InboundMediaLatchMode ini file parameter.
2. Open the General Settings page (Configuration tab > VoIP menu > Media > General Media Settings).

**Figure 12-13: General Settings Page - Robust Setting**



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

For a detailed description of the robust media parameters, see "General Security Parameters" on page 676.

## 12.11 Multiple Routers Support

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

When a set of routers operating within the same subnet serve as devices to that network and intercommunicate using a dynamic routing protocol, the routers can determine the shortest path to a certain destination and signal the remote host the existence of the better route. Using multiple router support, the device can utilize these router messages to change its next hop and establish the best path.

**Note:** Multiple Routers support is an integral feature that doesn't require configuration.

# 13 Security

This section describes the VoIP security-related configuration.

## 13.1 Configuring Firewall Settings

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

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

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

**Notes:**

- This firewall applies to a very low-level network layer and overrides all your other security-related configuration. Thus, if you have configured higher-level security features (e.g., on the Application level), you must also configure firewall rules to permit this necessary traffic. For example, if you have configured IP addresses to access the Web and Telnet interfaces in the Web Access List (see ''Configuring Web and Telnet Access List'' on page 75), you must configure a firewall rule that permits traffic from these IP addresses.
- Only users with Security Administrator or Master access levels can configure firewall rules.
- Setting the 'Prefix Length' field to **0** means that the rule applies to **all** packets, regardless of the defined IP address in the 'Source IP' field. Thus, it is highly recommended to set the parameter to a value other than 0.
- It is recommended to add a rule at the end of your table that blocks all traffic and to add firewall rules above it that allow required traffic (with bandwidth limitations). To block all traffic, use the following firewall rule:
  - √ Source IP: 0.0.0.0
  - √ Prefix Length: 0 (i.e., rule matches all IP addresses)
  - √ Start Port - End Port: 0-65535
  - √ Protocol: **Any**
  - √ Action Upon Match: **Block**

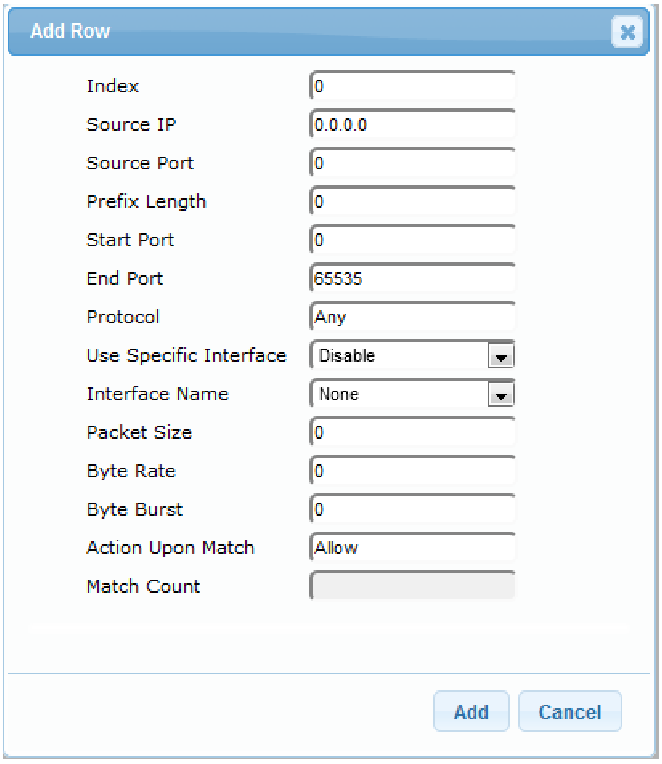
The following procedure describes how to configure Firewall rules through the Web interface. You can also configure it through ini file (AccessList) or CLI (configure voip > access-list).

➢ **To configure a Firewall rule:**

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

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

**Figure 13-1: Firewall Settings Table - Add Row Dialog Box**




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

**Table 13-1: Firewall Settings Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Source IP<br>`source-ip`<br>[AccessList_Source_IP] | Defines the IP address (or DNS name) or a specific host name of the source network from where the device receives the incoming packet.<br>The default is 0.0.0.0. |
| Source Port<br>`src-port`<br>[AccessList_Source_Port] | Defines the source UDP/TCP ports of the remote host from where the device receives the incoming packet.<br>The valid range is 0 to 65535. The default is 0.<br>**Note:** When set to 0, this field is ignored and any source port matches the rule. |

| Parameter | Description |
|---|---|
| Prefix Length<br>`prefixLen`<br>[AccessList_PrefixLen] | (**Mandatory**) Defines the IP network mask - 32 for a single host or the appropriate value for the source IP addresses.<br>▪ A value of 8 corresponds to IPv4 subnet class A (network mask of 255.0.0.0).<br>▪ A value of 16 corresponds to IPv4 subnet class B (network mask of 255.255.0.0).<br>▪ A value of 24 corresponds to IPv4 subnet class C (network mask of 255.255.255.0).<br>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'.<br>The default is 0 (i.e., applies to all packets). You **must** change this value to any of the above options.<br>**Note:** A value of 0 applies to **all** packets, regardless of the defined IP address. Therefore, you must set the parameter to a value other than 0. |
| Start Port<br>`start-port`<br>[AccessList_Start_Port] | Defines the first UDP/TCP port in the range of ports on the device on which the incoming packet is received. From the perspective of the remote IP entity, this is the destination port. To configure the last port in the range, see the 'End Port' parameter (below).<br>The valid range is 0 to 65535.<br>**Note:** When the protocol type isn't TCP or UDP, the entire range must be provided. |
| End Port<br>`end-port`<br>[AccessList_End_Port] | Defines the last UDP/TCP port in the range of ports on the device on which the incoming packet is received. From the perspective of the remote IP entity, this is the destination port. To configure the first port in the range, see the 'Start Port' parameter (above).<br>The valid range is 0 to 65535 (default).<br>**Note:** When the protocol type isn't TCP or UDP, the entire range must be provided. |
| Protocol<br>`protocol`<br>[AccessList_Protocol] | Defines the protocol type (e.g., **UDP**, **TCP**, **ICMP**, **ESP** or **Any**) or the IANA protocol number in the range of 0 (Any) to 255. The default is **Any**.<br>**Note:** The parameter also accepts the abbreviated strings "SIP" and "HTTP". Specifying these strings implies selection of the TCP or UDP protocols and the appropriate port numbers as defined on the device. |
| Use Specific Interface<br>`use-specific-interface`<br>[AccessList_Use_Specific_Interface] | Determines whether you want to apply the rule to a specific network interface defined in the Interface table (i.e., packets received from that defined in the Source IP field and received on this network interface):<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Notes:**<br>▪ If enabled, then in the 'Interface Name' field (described below), select the interface to which the rule is applied.<br>▪ If disabled, then the rule applies to all interfaces. |

| Parameter | Description |
|---|---|
| Interface Name<br>`network-interface-name`<br>[AccessList_Interface_x] | Defines the network interface to which you want to apply the rule. This is applicable if you enabled the 'Use Specific Interface' field. The list displays interface names as defined in the Interface table in "Configuring IP Network Interfaces" on page 135. |
| Packet Size<br>`packet-size`<br>[AccessList_Packet_Size] | Defines the maximum allowed packet size.<br>The valid range is 0 to 65535.<br>**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<br>`byte-rate`<br>[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.<br>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<br>`byte-burst`<br>[AccessList_Byte_Burst] | Defines the tolerance of traffic rate limit (number of bytes).<br>The default is 0. |
| Action Upon Match<br>`allow-type`<br>[AccessList_Allow_Type] | Defines the firewall action to be performed upon rule match.<br>▪ "Allow" = (Default) Permits these packets<br>▪ "Block" = Rejects these packets |
| Match Count<br>[AccessList_MatchCount] | (Read-only) Displays the number of packets accepted or rejected by the rule. |

The table below provides an example of configured firewall rules:

**Table 13-2: Configuration Example of Firewall Rules**

| Parameter | Firewall Rule | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| **Source IP** | 12.194.231.76 | 12.194.230.7 | 0.0.0.0 | 192.0.0.0 | 0.0.0.0 |
| **Prefix Length** | 16 | 16 | 0 | 8 | 0 |
| **Start Port and End Port** | 0-65535 | 0-65535 | 0-65535 | 0-65535 | 0-65535 |
| **Protocol** | Any | Any | icmp | Any | Any |
| **Use Specific Interface** | Enable | Enable | Disable | Enable | Disable |

| Parameter | Firewall Rule | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| **Interface Name** | WAN | WAN | None | Voice-Lan | None |
| **Byte Rate** | 0 | 0 | 40000 | 40000 | 0 |
| **Burst Bytes** | 0 | 0 | 50000 | 50000 | 0 |
| **Action Upon Match** | Allow | Allow | Allow | Allow | Block |

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

- **Rules 1 and 2:** Typical firewall rules that allow packets ONLY from specified IP addresses (e.g., proxy servers). Note that the prefix length is configured.
- **Rule 3:** A more "advanced” firewall rule - bandwidth rule for ICMP, which allows a maximum bandwidth of 40,000 bytes/sec with an additional allowance of 50,000 bytes. If, for example, the actual traffic rate is 45,000 bytes/sec, then this allowance would be consumed within 10 seconds, after which all traffic exceeding the allocated 40,000 bytes/sec is dropped. If the actual traffic rate then slowed to 30,000 bytes/sec, the allowance would be replenished within 5 seconds.
- **Rule 4:** Allows traffic from the LAN voice interface and limits bandwidth.
- **Rule 5:** Blocks all other traffic.

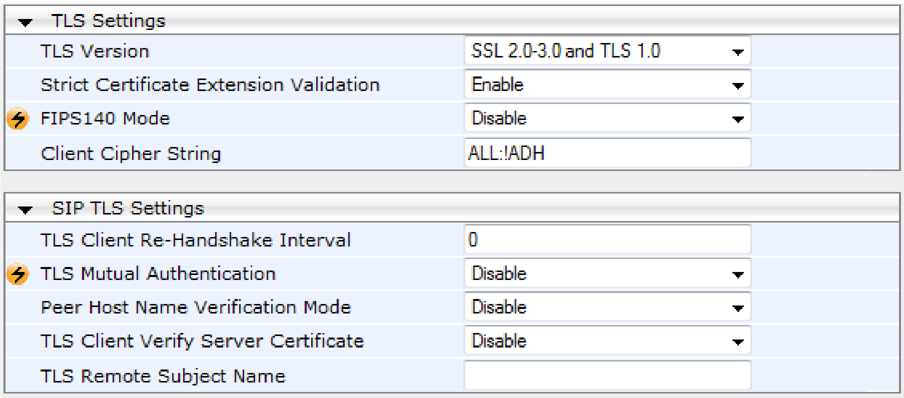
## 13.2 Configuring General Security Settings

The device uses TLS over TCP to encrypt and optionally, authenticate SIP messages. This is referred to as Secure SIP (SIPS). SIPS uses the X.509 certificate exchange process, as described in "Configuring SSL/TLS Certificates" on page 103, where you need to configure certificates (TLS Context).

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

**➢ To configure SIPS:**

1. Configure a TLS Context as required.
2. Assign the TLS Context to a Proxy Set or SIP Interface (see "Configuring Proxy Sets" on page 329 and "Configuring SIP Interfaces" on page 319, respectively).
3. Configure a SIP Interface with a TLS port number.
4. Configure various SIPS parameters in the General Security Settings page (**Configuration** tab > **VoIP** menu > **Security** > **General Security Settings**).




For a description of the TLS parameters, see "TLS Parameters" on page 683.

5. By default, the device initiates a TLS connection only for the next network hop. To enable TLS all the way to the destination (over multiple hops), set the 'Enable SIPS' (EnableSIPS) parameter to **Enable** in the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**).

## 13.3 Intrusion Detection System

The device's Intrusion Detection System (IDS) feature detects malicious attacks on the device and reacts accordingly. A remote host is considered malicious if it has reached or exceeded a user-defined threshold (counter) of specified malicious attacks.

If malicious activity is detected, the device can do the following:

- Block (blacklist) remote hosts (IP addresses / ports) considered by the device as malicious. The device automatically blacklists the malicious source for a user-defined period after which it is removed from the blacklist.
- Send SNMP traps to notify of malicious activity and/or whether an attacker has been added to or removed from the blacklist. For more information, see "Viewing IDS Alarms" on page 174.

The Intrusion Detection System (IDS) is an important feature for Enterprises to ensure legitimate calls are not being adversely affected by attacks and to prevent Theft of Service and unauthorized access.

There are many types of malicious attacks, the most common being:

- **Denial of service:** This can be Denial of Service (DoS) where an attacker wishing to prevent a server from functioning correctly directs a large amount of requests – sometimes meaningless and sometimes legitimate, or it can be Distributed Denial of Service (DDoS) where the attacker controls a large group of systems to coordinate a large scale DoS attack against a system:
  - Message payload tampering: Attacker may inject harmful content into a message, e.g., by entering meaningless or wrong information, with the goal of exploiting a buffer overflow at the target. Such messages can be used to probe for vulnerabilities at the target.
  - Message flow tampering: This is a special case of DoS attacks. These attacks disturb the ongoing communication between users. An attacker can then target the connection by injecting fake signaling messages into the communication channel (such as CANCEL messages).
  - Message Flooding: The most common DoS attack is where an attacker sends a huge amount of messages (e.g., INVITEs) to a target. The goal is to overwhelm the target’s processing capabilities, thereby rendering the target inoperable.
- **SPAM over Internet Telephony (SPIT):** VoIP spam is unwanted, automatically dialed, pre-recorded phone calls using VoIP. It is similar to e-mail spam.

- **Theft of Service (ToS):** Service theft can be exemplified by phreaking, which is a type of hacking that steals service (i.e., free calls) from a service provider, or uses a service while passing the cost to another person.

The IDS configuration is based on IDS Policies, where each policy can be configured with a set of IDS rules. Each rule defines a type of malicious attack to detect and the number of attacks during an interval (threshold) before an SNMP trap is sent. Each policy is then applied to a target under attack (SIP interface) and/or source of attack (Proxy Set and/or subnet address).

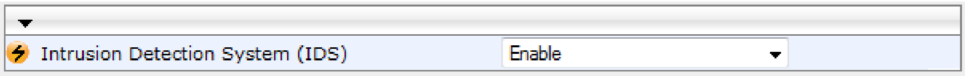
## 13.3.1 Enabling IDS

The following procedure describes how to enable IDS.

➢ **To enable IDS:**

1. Open the IDS Global Parameters page (**Configuration** tab > **VoIP** menu > **Security** > **Intrusion Detection and Prevention** > **Global Parameters**).

**Figure 13-2: Enabling IDS on IDS Global Parameters Page**




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

## 13.3.2 Configuring IDS Policies

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

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

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

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

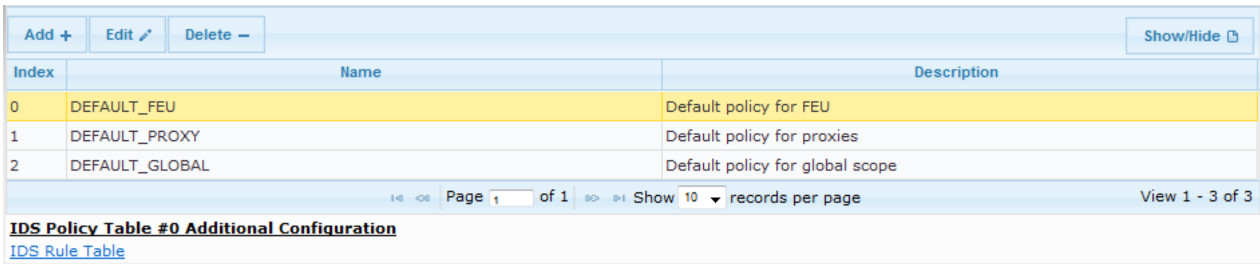
- "DEFAULT_FEU": IDS Policy for far-end users in the WAN
- "DEFAULT_PROXY": IDS Policy for proxy server
- "DEFAULT_GLOBAL": IDS Policy with global thresholds

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

➢ **To configure an IDS Policy:**

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

**Figure 13-3: IDS Policy Table with Default Rules**



Add + Edit Delete – Show/Hide

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

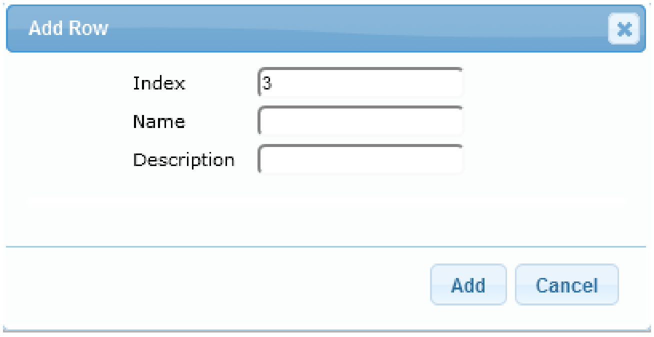
Page 1 of 1 Show 10 records per page View 1 - 3 of 3

**IDS Policy Table #0 Additional Configuration**

IDS Rule Table

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

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



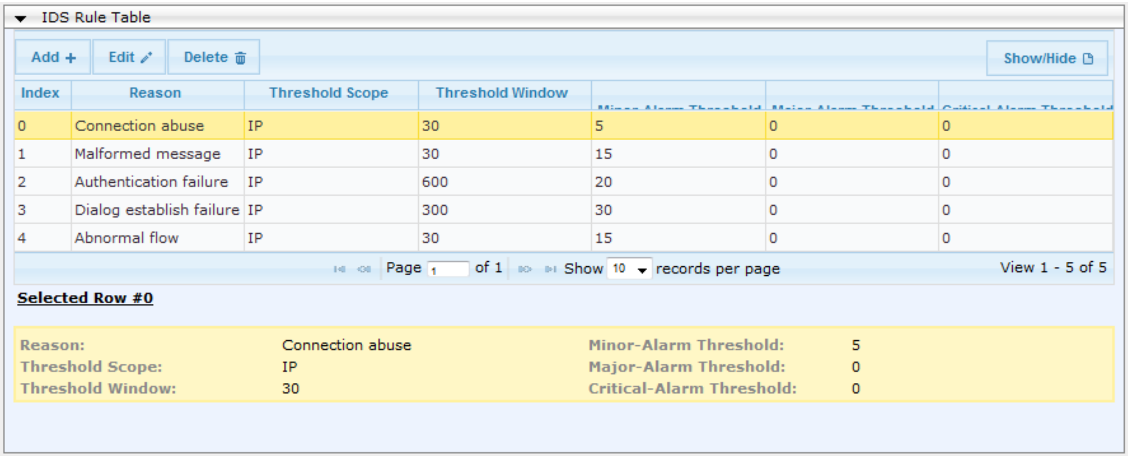

3. Configure an IDS Policy name according to the parameters described in the table below.
4. Click **Add**.

**Table 13-3: IDS Policy Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>`policy`<br>[IDSPolicy_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>`rule`<br>[IDSPolicy_Name] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters. |
| Description<br>[IDSPolicy_Description] | Defines a brief description for the IDS Policy.<br>The valid value is a string of up to 100 characters. |

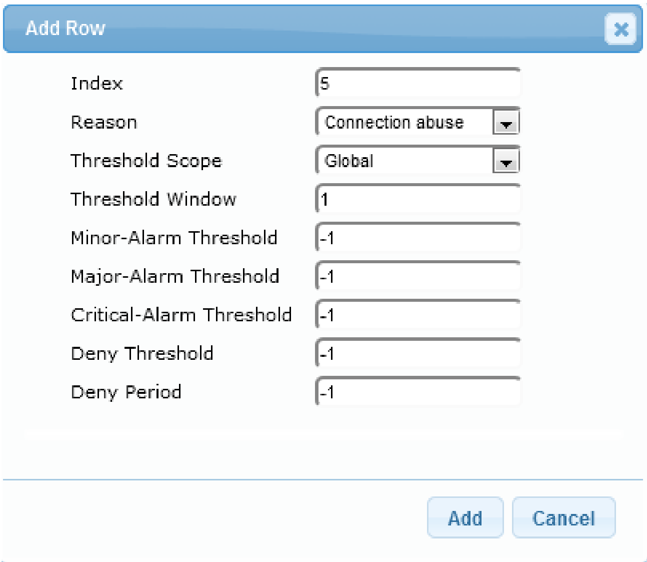
5. In the IDS Policy table, select the required IDS Policy row, and then click the **IDS Rule Table** link located below the table; the IDS Rule table opens:

**Figure 13-5: IDS Rule Table of Selected IDS Policy**



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

**Figure 13-6: IDS Rule Table - Add Record**




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

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

**Table 13-4: IDS Rule Table Parameter Descriptions**

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

| Parameter | Description |
|---|---|
| | ✓ Other local rejects (prior to SIP 180 response)<br>✓ Remote rejects (prior to SIP 180 response)<br>▪ **[5]** Abnormal flow =<br>✓ Requests and responses without a matching transaction user (except ACK requests)<br>✓ Requests and responses without a matching transaction (except ACK requests) |
| Threshold Scope<br>`threshold-scope`<br>[IDSRule_ThresholdScope] | Defines the source of the attacker to consider in the device's detection count.<br>▪ **[0]** Global = All attacks regardless of source are counted together during the threshold window.<br>▪ **[2]** IP = Attacks from each specific IP address are counted separately during the threshold window.<br>▪ **[3]** IP+Port = Attacks from each specific IP address:port are counted separately during the threshold window. This option is useful for NAT servers, where numerous remote machines use the same IP address but different ports. However, it is not recommended to use this option as it may degrade detection capabilities. |
| Threshold Window<br>`threshold-window`<br>[IDSRule_ThresholdWindow] | Defines the threshold interval (in seconds) during which the device counts the attacks to check if a threshold is crossed. The counter is automatically reset at the end of the interval.<br>The valid range is 1 to 1,000,000. The default is 1. |
| Minor-Alarm Threshold<br>`minor-alrm-thr`<br>[IDSRule_MinorAlarmThreshold] | Defines the threshold that if crossed a minor severity alarm is sent.<br>The valid range is 1 to 1,000,000. A value of 0 or -1 means not defined. |
| Major-Alarm Threshold<br>`major-alrm-thr`<br>[IDSRule_MajorAlarmThreshold] | Defines the threshold that if crossed a major severity alarm is sent.<br>The valid range is 1 to 1,000,000. A value of 0 or -1 means not defined. |
| Critical-Alarm Threshold<br>`critical-alrm-thr`<br>[IDSRule_CriticalAlarmThreshold] | Defines the threshold that if crossed a critical severity alarm is sent.<br>The valid range is 1 to 1,000,000. A value of 0 or -1 means not defined. |
| Deny Threshold<br>[IDSRule_DenyThreshold] | Defines the threshold that if crossed, the device blocks (blacklists) the remote host (attacker).<br>The default is -1 (i.e., not configured).<br>**Note:** The parameter is applicable only if the 'Threshold Scope' parameter is set to **IP** or **IP+Port**. |
| Deny Period<br>[IDSRule_DenyPeriod] | Defines the duration (in sec) to keep the attacker on the blacklist.<br>The valid range is 0 to 1,000,000. The default is -1 (i.e., not configured). |

### 13.3.3 Assigning IDS Policies

The IDS Match table lets you implement your configured IDS Policies. You do this by assigning IDS Policies to any, or a combination of, the following configuration entities:

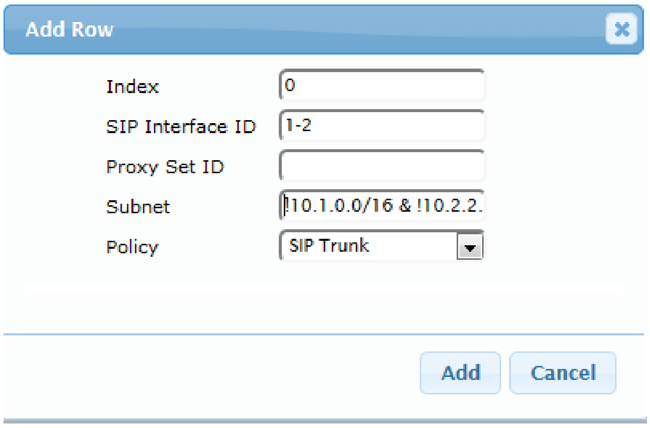
- **SIP Interface:** For detection of malicious attacks on specific SIP Interface(s). For configuring SIP Interfaces, see "Configuring SIP Interfaces" on page 319.
- **Proxy Sets:** For detection of malicious attacks from specified Proxy Set(s). For configuring Proxy Sets, see "Configuring Proxy Sets" on page 329.
- **Subnet addresses:** For detection of malicious attacks from specified subnet addresses.

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

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

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

**Figure 13-7: IDS Match Table - Add Row Dialog Box**




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

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

**Table 13-5: IDS Match Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[IDSMatch_Index] | Defines an index number for the new table record. |

| Parameter | Description |
|---|---|
| SIP Interface ID<br>`sip-interface`<br>[IDSMatch_SIPInterface] | Defines the SIP Interface(s) to which you want to assign the IDS Policy. This indicates the SIP Interfaces that are being attacked.<br>The valid value is the ID of the SIP Interface. The following syntax is supported:<br>▪ A comma-separated list of SIP Interface IDs (e.g., 1,3,4)<br>▪ A hyphen "-" indicates a range of SIP Interfaces (e.g., 3,4-7 means IDs 3, and 4 through 7)<br>▪ A prefix of an exclamation mark "!" means negation of the set (e.g., !3,4-7 means all indexes excluding 3, and excluding 4 through 7) |
| Proxy Set ID<br>`proxy-set`<br>[IDSMatch_ProxySet] | Defines the Proxy Set(s) to which the IDS Policy is assigned. This indicates the Proxy Sets from where the attacks are coming from. The following syntax is supported:<br>▪ A comma-separated list of Proxy Set IDs (e.g., 1,3,4)<br>▪ A hyphen "-" indicates a range of Proxy Sets (e.g., 3,4-7 means IDs 3, and 4 through 7)<br>▪ A prefix of an exclamation mark "!" means negation of the set (e.g., !3,4-7 means all indexes excluding 3, and excluding 4 through 7)<br>**Notes:**<br>▪ Only the IP address of the Proxy Set is considered (not port).<br>▪ If a Proxy Set has multiple IP addresses, the device considers the Proxy Set as one entity and includes all its IP addresses in the same IDS count. |
| Subnet<br>`subnet`<br>[IDSMatch_Subnet] | Defines the subnet to which the IDS Policy is assigned. This indicates the subnets from where the attacks are coming from. The following syntax can be used:<br>▪ Basic syntax is a subnet in CIDR notation (e.g., 10.1.0.0/16 means all sources with IP address in the range 10.1.0.0–10.1.255.255)<br>▪ An IP address can be specified without the prefix length to refer to the specific IP address.<br>▪ Each subnet can be negated by prefixing it with "!", which means all IP addresses outside that subnet.<br>▪ Multiple subnets can be specified by separating them with "&" (and) or "\|" (or) operations. For example:<br>✓ 10.1.0.0/16 \| 10.2.2.2: includes subnet 10.1.0.0/16 and IP address 10.2.2.2.<br>✓ !10.1.0.0/16 & !10.2.2.2: includes all addresses except those of subnet 10.1.0.0/16 and IP address 10.2.2.2. Note that the exclamation mark "!" appears before each subnet.<br>✓ 10.1.0.0/16 & !10.1.1.1: includes subnet 10.1.0.0/16, except IP address 10.1.1.1. |
| Policy<br>`policy`<br>[IDSMatch_Policy] | Assigns an IDS Policy (configured in "Configuring IDS Policies" on page 168). |

### 13.3.4 Viewing IDS Alarms

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

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

    If the severity level is raised, the alarm of the former severity is cleared and the device sends a new alarm with the new severity. The alarm is cleared after a user-defined period (configured by the ini file parameter, IDSAlarmClearPeriod) during which no thresholds have been crossed. However, this "quiet" period must be at least twice the 'Threshold Window' value (configured in "Configuring IDS Policies" on page 168). For example, if you set IDSAlarmClearPeriod to 20 sec and 'Threshold Window' to 15 sec, the IDSAlarmClearPeriod parameter is ignored and the alarm is cleared only after 30 seconds (2 x 15 sec).

    The figure below displays an example of IDS alarms in the Active Alarms table ("Viewing Active Alarms" on page 551). In this example, a Minor threshold alarm is cleared and replaced by a Major threshold alarm:

**Figure 13-8: IDS Alarms in Active Alarms Table**

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

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

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

- To view all active IDS alarms:

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

- To view all IP addresses that have crossed the threshold for an active IDS alarm:

  ```
  # show voip security ids active-alarm match <IDS Match Policy ID> rule <IDS Rule
  ID>
  ```

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

- To view the blacklist:

  ```
  # show voip security ids blacklist active
  ```

  For example:

  ```
  Active blacklist entries:
   10.33.5.110(NI:0) remaining 00h:00m:10s in blacklist
  ```

  Where SI is the SIP Interface and NI is the network interface.

The device also sends IDS notifications and alarms in Syslog messages to a Syslog server. This occurs only if you have configured Syslog (see "Enabling Syslog" on page 612). An example of a Syslog message with IDS alarms and notifications is shown below:

**Figure 13-9: Syslog Message Example with IDS Alarms and Notifications**

```
[S=92159] [SID:438286865] (     lgr_ids)(97420    )  IDS Event: reason=establish-fail,event=14003(establish-classify-fail),ip=10.13.45.200:5060(SI1),transport=udp
[S=92160] [SID:438286865] (     lgr_ids)(97421    )  IDS Counter (0,19995): IDSMatch#0/IDSRule#0,policy=3(TEST),reason=establish-fail,scope=ip,scope-val=10.13.45.200(SI1),value=6
[S=92161] [SID:438286865] (     lgr_ids)(97422    ) ?? [WARNING]  IDS Rule (0): Threshold cross. IDSMatch#0/IDSRule#0,policy=3(TEST),value=6,severity=2(major)
[S=92162] [SID:438286865] (     lgr_ids)(97423    ) ?? [WARNING]  IDS Rule (0): Threshold cross. IDSMatch#0/IDSRule#0,policy=3(TEST),value=6,severity=4(blacklist)
[S=92163] [SID:438286865] (     lgr_ids)(97424    ) ?? [WARNING]  IDS Blacklist: Added IP 10.13.45.200(NI:0) to blacklist
[S=92164] [SID:438286865] (  lgr_psbrdif)(97425    )  SNMP EVENT: IDS_BLACKLIST_NOTIFY "Added IP 10.13.45.200(NI:0) to blacklist"
[S=92165] RAISE-ALARM:acIDSBlacklistNotification; Textual Description: Added IP 10.13.45.200(NI:0) to blacklist; Severity:indeterminate; Source:; Unique ID:30;
[S=92166] [SID:438286865] (  lgr_psbrdex)(97426    )  InsertBoardEvent- event ADD_BLACKLIST_EV inserted channel -100
```

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

**Table 13-6: Types of Malicious Events and Syslog Text String**

| Type | Description | Syslog String |
|---|---|---|
| **Connection Abuse** | TLS authentication failure | abuse-tls-auth-fail |
| **Malformed Messages** | ▪ Message exceeds a user-defined maximum message length (50K)<br>▪ Any SIP parser error<br>▪ Message policy match<br>▪ Basic headers not present<br>▪ Content length header not present (for TCP)<br>▪ Header overflow | ▪ malformed-invalid-msg-len<br>▪ malformed-parse-error<br>▪ malformed-message-policy<br>▪ malformed-miss-header<br>▪ malformed-miss-content-len<br>▪ malformed-header-overflow |
| **Authentication Failure** | ▪ Local authentication ("Bad digest" errors)<br>▪ Remote authentication (SIP 401/407 is sent if original message includes authentication) | ▪ auth-establish-fail<br>▪ auth-reject-response |
| **Dialog Establishment Failure** | ▪ Classification failure<br>▪ Routing failure<br>▪ Other local rejects (prior to SIP 180 response)<br>▪ Remote rejects (prior to SIP 180 response) | ▪ establish-classify-fail<br>▪ establish-route-fail<br>▪ establish-local-reject<br>▪ establish-remote-reject |
| **Abnormal Flow** | ▪ Requests and responses without a matching transaction user (except ACK requests)<br>▪ Requests and responses without a matching transaction (except ACK requests) | ▪ flow-no-match-tu<br>▪ flow-no-match-transaction |

**This page is intentionally left blank.**

# 14 Media

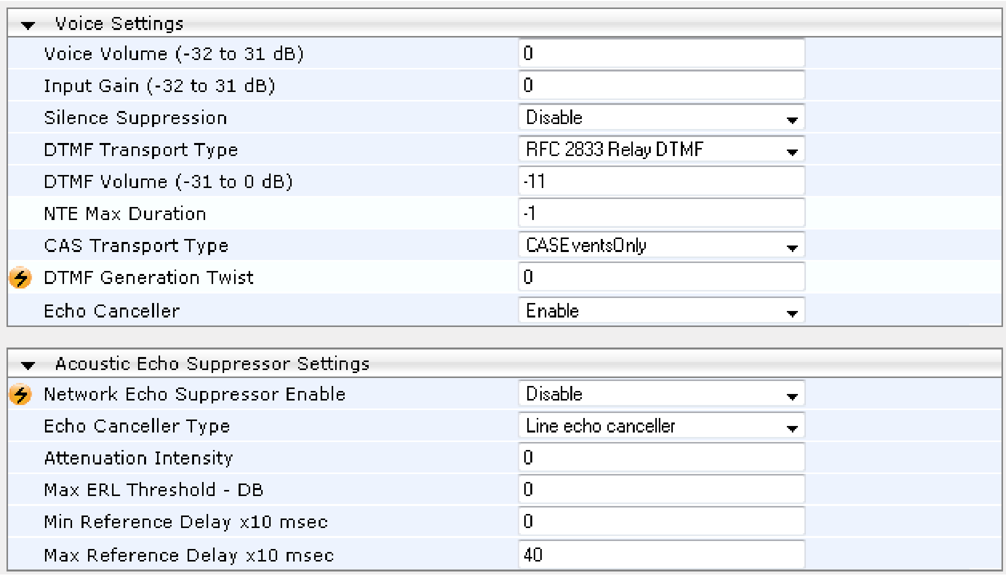
This section describes the media-related configuration.

## 14.1 Configuring Voice Settings

The Voice Settings page configures various voice parameters such as voice volume and DTMF transport type. For a detailed description of these parameters, see "Configuration Parameters Reference" on page 643.

➢ **To configure the voice parameters:**

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




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

### 14.1.1 Configuring Voice Gain (Volume) Control

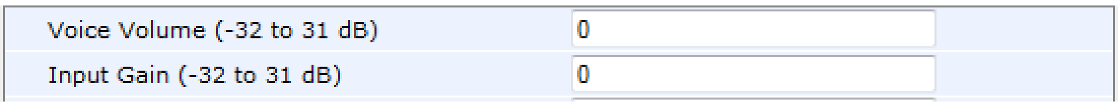
The device allows you to configure the level of the received (input gain) Tel-to-IP signal and the level of the transmitted (output gain) IP-to-Tel signal. The gain can be set between -32 and 31 decibels (dB).

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

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

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

**Figure 14-1: Voice Volume Parameters in Voice Settings Page**



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

### 14.1.2 Configuring Echo Cancellation

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

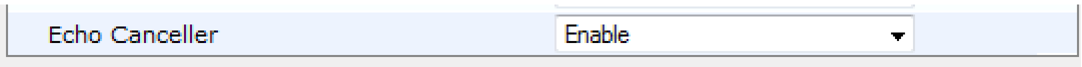
In this line echo, echoes are generated when two-wire telephone circuits (carrying both transmitted and received signals on the same wire pair) are converted to a four-wire circuit. Echoes are reflections of the transmitted signal, which result from impedance mismatch in the hybrid (bi-directional 2-wire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

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

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

➢ **To configure echo cancellation using the Web interface:**

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




   b. 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

## 14.2 Fax and Modem Capabilities

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

**Notes:**

- Unless otherwise specified, the configuration parameters mentioned in this section are available on this page.
- Some SIP parameters override these fax and modem parameters. For example, the IsFaxUsed parameter and V.152 parameters in Section "V.152 Support" on page 191.
- For a detailed description of the parameters appearing on this page, see "Configuration Parameters Reference" on page 643.

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

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

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

| Fax Relay Settings | |
|---|---|
| Fax Relay Redundancy Depth | 0 |
| Fax Relay Enhanced Redundancy Depth | 4 |
| Fax Relay ECM Enable | Enable |
| Fax Relay Max Rate (bps) | 33600bps |
| T38 Version | T.38 version 0 |

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

| V.150.1 Modem Relay Settings | |
|---|---|
| SSE Payload Type Rx | 105 |
| SSE Payload Type Tx | 105 |
| SSE Redundancy Depth | 3 |
| SPRT Payload Type Rx | 115 |
| SPRT Payload Type Tx | 115 |
| SPRT Transport Ch.0 Max Payload Size | 140 |
| SPRT Transport Ch.2 Max Payload Size | 132 |
| SPRT Transport Ch.2 Max Window Size | 8 |
| SPRT Transport Ch.3 Max Payload Size | 140 |

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

## 14.2.1 Fax/Modem Operating Modes

The device supports two modes of operation:

- Fax/modem negotiation that is not performed during the establishment of the call.

- Voice-band data (VBD) mode for V.152 implementation (see "V.152 Support" on page 191): fax/modem capabilities are negotiated between the device and the remote endpoint at the establishment of the call. During a call, when a fax/modem signal is detected, transition from voice to VBD (or T.38) is automatically performed and no additional SIP signaling is required. If negotiation fails (i.e., no match is achieved for any of the transport capabilities), fallback to existing logic occurs (according to the parameter IsFaxUsed).

## 14.2.2 Fax/Modem Transport Modes

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

- T.38 fax relay (see "T.38 Fax Relay Mode" on page 180)
- G.711 Transport: switching to G.711 when fax/modem is detected (see "G.711 Fax / Modem Transport Mode" on page 182)
- Fax fallback to G.711 if T.38 is not supported (see "Fax Fallback" on page 183)
- Fax and modem bypass: a proprietary method that uses a high bit rate coder (see "Fax/Modem Bypass Mode" on page 184)
- NSE Cisco's Pass-through bypass mode for fax and modem (see "Fax / Modem NSE Mode" on page 185)
- Transparent with events: passing the fax / modem signal in the current voice coder with adaptations (see "Fax / Modem Transparent with Events Mode" on page 186)
- Transparent: passing the fax / modem signal in the current voice coder (see "Fax / Modem Transparent Mode" on page 186)
- RFC 2833 ANS Report upon Fax/Modem Detection (see "RFC 2833 ANS Report upon Fax/Modem Detection" on page 187)

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

### 14.2.2.1 T.38 Fax Relay Mode

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

T.38 can be configured in the following ways:

- Switching to T.38 mode using SIP Re-INVITE messages (see "Switching to T.38 Mode using SIP Re-INVITE" on page 181)
- 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 181)

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

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

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

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

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

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

1. In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **T.38 Relay** (IsFaxUsed = 1).
2. In the Fax/Modem/CID Settings page, configure the following optional parameters:
   - 'Fax Relay Redundancy Depth' (FaxRelayRedundancyDepth)
   - 'Fax Relay Enhanced Redundancy Depth' (FaxRelayEnhancedRedundancyDepth)
   - 'Fax Relay ECM Enable' (FaxRelayECMEnable)
   - 'Fax Relay Max Rate' (FaxRelayMaxRate)

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

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

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

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

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

#### 14.2.2.1.3 Fax over IP using T.38 Transmission over RTP

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

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

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

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

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

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

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

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

### 14.2.2.2 G.711 Fax / Modem Transport Mode

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

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

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

A ‘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).

### 14.2.2.3 Fax Fallback

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

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

When the device initiates a fax session using G.711, a ‘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).

### 14.2.2.4 Fax/Modem Bypass Mode

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

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

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

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

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

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

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

### 14.2.2.5 Fax / Modem NSE Mode

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

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

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

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

Where *100* is the NSE payload type.

The Cisco gateway must include the following definition:

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

➢ **To configure NSE mode:**

1. In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **No Fax** (IsFaxUsed = 0).
2. In the Fax/Modem/CID Settings page, do the following:
   a. Set the 'Fax Transport Mode' parameter to **Bypass** (FaxTransportMode = 2).
   b. Set the 'V.21 Modem Transport Type' parameter to **Enable Bypass** (V21ModemTransportType = 2).
   c. Set the 'V.22 Modem Transport Type' parameter to **Enable Bypass** (V22ModemTransportType = 2).
   d. Set the 'V.23 Modem Transport Type' parameter to **Enable Bypass** (V23ModemTransportType = 2).
   e. Set the 'V.32 Modem Transport Type' parameter to **Enable Bypass** (V32ModemTransportType = 2).
   f. Set the 'V.34 Modem Transport Type' parameter to **Enable Bypass** (V34ModemTransportType = 2).
3. Set the ini file parameter, BellModemTransportType to 2 (Bypass).
4. Set the ini file parameter, NSEMode parameter to 1 (enables NSE).
5. Set the ini file parameter, NSEPayloadType parameter to 100.

#### 14.2.2.6 Fax / Modem Transparent with Events Mode

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

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

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

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

#### 14.2.2.7 Fax / Modem Transparent Mode

In this mode, fax and modem signals are transferred using the current voice coder without notifications to the user and without automatic adaptations. It's possible to use Profiles (see "Coders and Profiles" on page 357) 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 184) or Transparent with Events modes (see "Fax / Modem Transparent with Events Mode" on page 186) for modem.

#### 14.2.2.8 RFC 2833 ANS Report upon Fax/Modem Detection

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

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

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

### 14.2.3 V.34 Fax Support

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

- T38 Version 3 - V.34 fax relay mode
- Bypass mechanism for V.34 fax transmission (see "Bypass Mechanism for V.34 Fax Transmission" on page 188)
- T38 Version 0 relay mode, i.e., fallback to T.38 (see "Relay Mode for T.30 and V.34 Faxes" on page 188)

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

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

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

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

### 14.2.3.1 Bypass Mechanism for V.34 Fax Transmission

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

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

1. In the Fax/Modem/CID Settings page, do the following:
   a. Set the 'Fax Transport Mode' parameter to **Bypass** (FaxTransportMode = 2).
   b. Set the 'V.22 Modem Transport Type' parameter to **Enable Bypass** (V22ModemTransportType = 2).
   c. Set the 'V.23 Modem Transport Type' parameter to **Enable Bypass** (V23ModemTransportType = 2).
   d. Set the 'V.32 Modem Transport Type' parameter to **Enable Bypass** (V32ModemTransportType = 2).
   e. Set the 'V.34 Modem Transport Type' parameter to **Enable Bypass** (V34ModemTransportType = 2).
2. Set the ini file parameter, V34FaxTransportType to 2 (Bypass).

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

1. In the Fax/Modem/CID Settings page, do the following:
   a. Set the 'Fax Transport Mode' parameter to **T.38 Relay** (FaxTransportMode = 1).
   b. Set the 'V.22 Modem Transport Type' parameter to **Enable Bypass** (V22ModemTransportType = 2).
   c. Set the 'V.23 Modem Transport Type' parameter to **Enable Bypass** (V23ModemTransportType = 2).
   d. Set the 'V.32 Modem Transport Type' parameter to **Enable Bypass** (V32ModemTransportType = 2).
   e. Set the 'V.34 Modem Transport Type' parameter to **Enable Bypass** (V34ModemTransportType = 2).
2. Set the ini file parameter, V34FaxTransportType to 2 (Bypass).

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

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

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

1. In the Fax/Modem/CID Settings page, do the following:
   a. Set the 'Fax Transport Mode' parameter to **T.38 Relay** (FaxTransportMode = 1).
   b. Set the 'V.22 Modem Transport Type' parameter to **Disable** (V22ModemTransportType = 0).
   c. Set the 'V.23 Modem Transport Type' parameter to **Disable** (V23ModemTransportType = 0).
   d. Set the 'V.32 Modem Transport Type' parameter to **Disable** (V32ModemTransportType = 0).

e. Set the 'V.34 Modem Transport Type' parameter to **Disable** (V34ModemTransportType = 0).

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

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

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

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

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

### 14.2.3.3 V.34 Fax Relay for SG3 Fax Machines

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

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

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

**Notes:**

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

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

```
INVITE sip:2001@10.8.211.250;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.6.55;branch=z9hG4bKac1938966220
Max-Forwards: 70
From: <sip:318@10.8.6.55>;tag=1c1938956155
To: <sip:2001@10.8.211.250;user=phone>
Call-ID: 1938955292412000223310@10.8.6.55
CSeq: 1 INVITE
Contact: <sip:318@10.8.6.55:5060>
Supported: em,100rel,timer,replaces,path,resource-priority,sdp-
anat
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUB
SCRIBE,UPDATE
Remote-Party-ID:
<sip:318@10.8.211.250>;party=calling;privacy=off;screen=no;screen-
ind=0;npi=1;ton=0
Remote-Party-ID: <sip:2001@10.8.211.250>;party=called;npi=1;ton=0
User-Agent: Audiocodes-Sip-Gateway-/v.6.80A.227.005
Content-Type: application/sdp
Content-Length: 433
v=0
o=AudiocodesGW 1938931006 1938930708 IN IP4 10.8.6.55
s=Phone-Call
c=IN IP4 10.8.6.55
t=0 0
m=audio 6010 RTP/AVP 18 97
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:97 telephone-event/8000
a=fmtp:97 0-15
a=ptime:20
a=sendrecv
m=image 6012 udptl t38
a=T38FaxVersion:3
a=T38MaxBitRate:33600
a=T38FaxMaxBuffer:1024
a=T38FaxMaxDatagram:122
a=T38FaxRateManagement:transferredTCF
a=T38FaxUdpEC:t38UDPRedundancy
```

### 14.2.4 V.150.1 Modem Relay

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

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

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

**Notes:**

- The V.150.1 Modem Relay feature is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see "Software License Key" on page 510.
- V.150.1 modem relay feature support is a subset of the full V.150.1 protocol and is designed according to the US DoD requirement document. It therefore, cannot be used for general purposes.
- V.150.1 modem relay is applicable only to the Gateway application.
- The V.150.1 feature has been tested with certain IP phones. For more details, please contact your AudioCodes sales representative.
- The V.150.1 SSE Tx payload type is according to the offered SDP of the remote side.
- The V.150.1 SPRT Rx payload type is according to the 'Payload Type' field in the Coders table.
- The V.150.1 SPRT Tx payload type is according to the remote side offered SDP.

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

1. In the Coders Table (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **Coders**), set the coder to **V.150**.
2. On the Fax/Modem/CID Settings page, configure the V.150.1 parameters appearing under the 'V.150.1 Modem Relay Settings' group:
   a. Set the 'SSE Payload Type Rx' parameter to the V.150.1 SSE payload type that the device uses when it offers the SDP.
   b. Set the 'SSE Redundancy Depth' parameter to the number of sent SSE redundant packets. The parameter is important in case of network impairments.
   c. For additional V.150.1 related parameters, see "Fax and Modem Parameters" on page 739.

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

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

To support this functionality, you need to configure a Coders Group (in the Coders Group table - see "Configuring Coder Groups" on page 360) that includes the T.38, V.150, and G.711/VBD coders.

### 14.2.6 V.152 Support

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

For V.152 capability, the device supports T.38 as well as VBD codecs (i.e., G.711 A-law and G.711 µ-law). The selection of capabilities is performed using the coders table (see "Configuring Default Coders" on page 357).

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.

## 14.3 Configuring RTP/RTCP Settings

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

### 14.3.1 Configuring the Dynamic Jitter Buffer

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

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

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

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

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

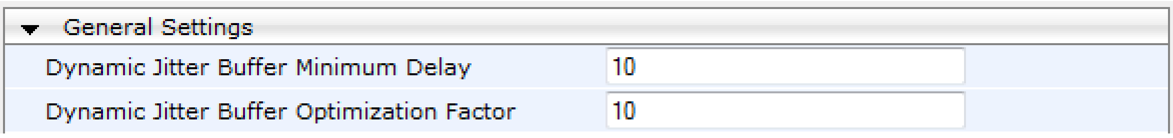
Fax and modem devices are sensitive to small packet losses or to added BFI packets. Therefore, to achieve better performance during modem and fax calls, the Optimization Factor should be set to 13. In this special mode the clock drift correction is performed less frequently - only when the Jitter Buffer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

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

### ➢ To configure jitter buffer using the Web interface:

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

**Figure 14-2: Jitter Buffer Parameters in the RTP/RTCP Settings Page**




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

## 14.3.2 Comfort Noise Generation

The device can generate artificial background noise, called *comfort* noise, in the voice channel during periods of silence (i.e. when no call party is speaking) for Gateway calls. This is useful in that it reassures the call parties that the call is still connected. The device detects silence using its Voice Activity Detection (VAD) mechanism. When Comfort Noise Generation 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 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 Comfort Noise Generation related parameters.

The following procedure describes how to configure Comfort Noise Generation using the Web interface.

### ➢ To configure Comfort Noise Generation using the Web interface:

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

**Figure 14-3: Comfort Noise Parameter in RTP/RTCP Settings Page**


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

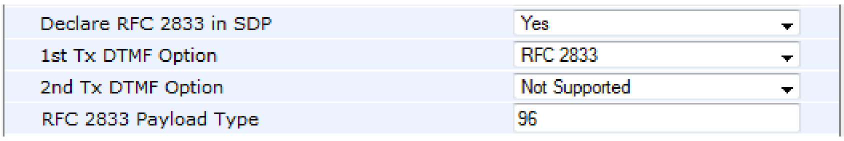
> **Note:** This feature is applicable only to the Gateway application.

## 14.3.3 Configuring DTMF Transport Types

The device supports various methods for transporting DTMF digits over the IP network to the remote endpoint. The methods and their configuration can be configured in the DTMF &

Dialing page (**Configuration** tab > **VoIP** menu > **Gateway** > **DTMF and Supplementary** > **DTMF & Dialing**):

**Figure 14-4: DTMF Transport Parameters in Web Interface**




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

  **Note:** 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 the mode:
  - **a.** Set the 'Declare RFC 2833 in SDP' parameter to **No** (RxDTMFOption = 0).
  - **b.** Set the 'First Tx DTMF Option' parameter to **INFO Cisco** (FirstTxDTMFOption = 3).

  **Note:** 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 the mode:
  - **a.** Set the 'Declare RFC 2833 in SDP' parameter to **No** (RxDTMFOption = 0).
  - **b.** Set the 'First Tx DTMF Option' parameter to **NOTIFY** (FirstTxDTMFOption = 2).

  **Note:** 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 the mode:
  - **a.** Set the 'Declare RFC 2833 in SDP' parameter to **Yes** (RxDTMFOption = 3).
  - **b.** Set the 'First Tx DTMF Option' parameter to **RFC 2833** (FirstTxDTMFOption = 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 the 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 the mode:
  - **a.** Set the 'Declare RFC 2833 in SDP' parameter to **No** (RxDTMFOption = 0).
  - **b.** Set the 'First Tx DTMF Option' parameter to **Not Supported** (FirstTxDTMFOption = 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:
  a. Set the 'Declare RFC 2833 in SDP' parameter to **No** (RxDTMFOption = 0).
  b. Set the 'First Tx DTMF Option' parameter to **INFO Cisco** (FirstTxDTMFOption = 3).

  **Note:** 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**.
- You can use the following parameters to configure DTMF digit handling:
  - √ FirstTxDTMFOption, SecondTxDTMFOption, RxDTMFOption, RFC2833TxPayloadType, and RFC2833RxPayloadType
  - √ MGCPDTMFDetectionPoint, DTMFVolume, DTMFTransportType, DTMFDigitLength, and DTMFInterDigitInterval

## 14.3.4 Configuring RFC 2833 Payload

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

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

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

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

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

2. Configure the following parameters:
   - 'RTP Redundancy Depth' (RTPRedundancyDepth) - enables the device to generate RFC 2198 redundant packets.
   - For Gateway application only: '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**.

## 14.3.5 Configuring RTP Base UDP Port

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

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

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

The port range is calculated using the following equation:

```
BaseUDPPort to 65,535
```

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

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

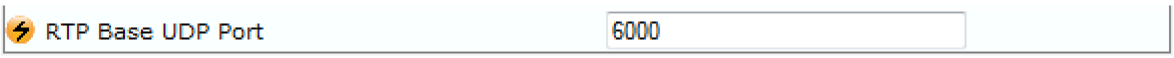
You can also configure specific port ranges for specific SIP entities, using Media Realms (see Configuring Media Realms on page 303). You can configure each Media Realm with a different UDP port range and then associate the Media Realm with a specific IP Group, for example. However, the port range of the Media Realm must be within the range configured by the BaseUDPPort parameter.

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

**➢ To configure the RTP base UDP port:**

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

**Figure 14-6: RTP Based UDP Port in RTP/RTCP Settings Page**




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

**Note:**

- The RTP port must be different from ports configured for SIP signaling traffic (i.e., ports configured for SIP Interfaces). For example, if the RTP port range is 6000 to 6999, the SIP port can either be less than 6000 or greater than 6999.
- The base UDP port number (BaseUDPPort parameter) must be greater than the highest UDP port configured for a SIP Interface (see Configuring SIP Interfaces on page 319). For example, if your highest configured UDP port for a SIP Interface is 6060, you must configure the BaseUDPPort parameter to any value greater than 6060.

## 14.4 Event Detection and Notification using X-Detect Header

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

The device can detect the following event types:

- Answering Machine Detection (AMD): Detects events that are related to the AMD feature. AMD detects whether an answering machine or live voice has answered the call. It can also be used to detect silence, or the beep sound played by an answering machine to indicate the end of the greeting message after which a voice message can be left. For more information on AMD, see "Answering Machine Detection (AMD)" on page 203.
- Call Progress Tone (CPT): Detects whether a specific tone, defined in the installed CPT file is received from the call. It can be used to detect the beep sound played by an answering machine (as mentioned above), Special Information Tones (SIT) which indicate call failure with a recorded announcement describing the call failure, and the busy, reorder and ring tones.
- Fax/modem machine: Detects whether a fax has answered the call (preamble, CED, CNG, and modem).
- PTT: Detects the start and end of voice.

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

- X-Detect header in the INVITE message received from the application server requesting a specific event detection:

```
X-Detect: Request=[event type to detect]
```

- X-Detect header in the SIP response message -- SIP 183 (for early dialogs) or 200 OK (for confirmed dialogs) -- sent by the device to the application server specifying which of the requested events it can detect (absence of the X-Detect header indicates that the device cannot detect any of the events):

```
X-Detect: Response=[supported event types]
```

- Each time the device detects the supported event, it sends an INFO message to the remote party with the following message body:

```
Content-Type: Application/X-Detect
Type = [event type]
Subtype = [subtype of each event type]
```

The table below lists the event types and subtypes that the device can detect. The text shown in the table are the actual strings that are used in the X-Detect header. The table also provides a summary of the required configuration.

**Table 14-1: Supported X-Detect Event Types**

| Event Type | Subtype | Description and Required Configuration |
|---|---|---|
| AMD | Voice (live voice)<br>Automata (answering machine)<br>Silence (no voice)<br>Unknown | Event detection using the AMD feature. For more information, see Answering Machine Detection (AMD) on page 203. |

| Event Type | Subtype | Description and Required Configuration |
|---|---|---|
| | Beep (greeting message of answering machine) | |
| CPT | SIT-NC<br>SIT-IC<br>SIT-VC<br>SIT-RO<br>Busy<br>Reorder<br>Ringtone<br>Beep (greeting message of answering message) | Event detection of tones using the CPT file.<br>1 Create a CPT file with the required tone types of the events that you want to detect.<br>2 Install the CPT file on the device.<br>3 For SIT detection:<br>a. Set the SITDetectorEnable parameter to 1.<br>b. Set the UserDefinedToneDetectorEnable parameter to 1.<br>**Notes:**<br>▪ For more information on SIT detection, see SIT Event Detection on page 199.<br>▪ For configuring beep detection, see Detecting Answering Machine Beep on page 200. |
| FAX | CED | ▪ Set the IsFaxUsed parameter to any value other than 0.<br>- or -<br>▪ Set the IsFaxUsed parameter to 0 and the FaxTransportMode parameter to any value other than 0. |
| | modem | Set the VxxModemTransportType parameter to 3. |
| PTT | voice-start<br>voice-end | Set the EnableDSPIPMDetectors parameter to 1. |

## 14.4.1 SIT Event Detection

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 14-2: 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 |

The following example shows a SIP INFO message sent by the device to a remote application server notifying it that SIT detection has been detected:

```
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
```

## 14.4.2 Detecting Answering Machine Beeps

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

- **AMD-based Detection:** The device uses its beep detector that is integrated in the AMD feature. You can configure the beep detection timeout and beep detection sensitivity level (for more information, see "Configuring AMD" on page 206). To enable the AMD beep detection, the received INVITE message must contain an X-Detect header with the value "Request=AMD",

  ```
  X-Detect: Request=AMD
  ```

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

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

  ```
  X-Detect: Request=CPT
  ```

  For more information on the CPT file, see "Call Progress Tones File" on page 496.

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

- AMD-detected Beep:

  ```
  Type= AMD
  SubType= Beep
  ```

- CPT-detected Beep:

  ```
  Type= CPT
  SubType=Beep
  ```

### 14.4.3 SIP Call Flow Examples of Event Detection and Notification

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

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

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

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

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

     ```
     INFO sip:sipp@172.22.2.9:5060 SIP/2.0
     Via: SIP/2.0/UDP 172.22.168.249;branch=z9hG4bKac482466515
     Max-Forwards: 70
     From: sut <sip:3000@172.22.168.249:5060>;tag=1c419779142
     To: sipp <sip:sipp@172.22.2.9:5060>;tag=1
     Call-ID: 1-29753@172.22.2.9
     CSeq: 1 INFO
     Contact: <sip:56700@172.22.168.249>
     Supported: em,timer,replaces,path,resource-priority
     ```

```
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,I
NFO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway/v.6.80A.227.005
Content-Type: application/x-detect
Content-Length: 34
Type= PTT
SubType= SPEECH-START
```

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

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

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

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

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

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

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

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

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

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

## 14.5 Answering Machine Detection (AMD)

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

When the device detects what answered the call (human or machine), it can notify this detection type to, for example, a third-party application server used for automatic dialing applications. The X-Detect SIP header is used for requesting event detection and notification. For more information, see "Event Detection and Notification using X-Detect Header" on page 198. The device can also detect beeps played by an answering machine at the end of its greeting message. For more information, see "Detecting Answering Machine Beeps" on page 200. You can also configure the device to disconnect IP-to-Tel calls upon detection of an answering machine on the Tel side. For more information, see Enabling IP-to-Tel Call Disconnection upon Detection of Answering Machine on page 206.

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

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

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

Statistical accuracy: The number of recorded calls should be as high as possible (at least 100) and varied. The calls must be made to different people. The calls must be made in the specific location in which the device's AMD feature is to operate.

- Real-life recording: The recordings should simulate real-life answering of a called person picking up the phone, and without the caller speaking.
- Normal environment interferences: The environment in which the recordings are done should simulate real-life scenarios, in other words, not sterile but not too noisy either. Interferences, for example, could include background noises of other people talking, spikes, and car noises.

Once you have provided AudioCodes with your database of recordings, AudioCodes compiles it into a loadable file. For a brief description of the file format and for installing the file on the device, see "AMD Sensitivity File" on page 509.

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

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

As Parameter Suite 1 provides a greater range of detection sensitivity levels (i.e., higher detection resolution), this may be the preferable suite to use in your deployment. The detected AMD type (human or machine) and success of detecting it correctly are sent in CDR and Syslog messages. For more information, see "Syslog Fields for Answering Machine Detection (AMD)" on page 608.

The Parameter Suite and sensitivity level can be applied globally for all calls, or for specific calls using IP Profiles. For enabling AMD and selecting the Parameter Suite and sensitivity level, see "Configuring AMD" on page 206.

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

**Table 14-3: Approximate AMD Normal Detection Sensitivity - Parameter Suite 0 (Based on North American English)**

| AMD Detection Sensitivity | Performance | |
|---|---|---|
| | Success Rate for Live Calls | Success Rate for Answering Machine |
| **0** (Best for Answering Machine) | - | - |
| **1** | 82.56% | 97.10% |

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

**Table 14-4: Approximate AMD High Detection Sensitivity - Parameter Suite 1 (Based on North American English)**

| AMD Detection Sensitivity | Performance | |
|---|---|---|
| | Success Rate for Live Calls | Success Rate for Answering Machine |
| **0** (Best for Answering Machine) | 72% | 97% |
| **1** | 77% | 96% |
| **2** | 79% | 95% |
| **3** | 80% | 95% |
| **4** | 84% | 94% |
| **5** | 86% | 93% |
| **6** | 87% | 92% |
| **7** | 88% | 91% |
| **8** | 90% | 89% |
| **9** | 90% | 88% |
| **10** | 91% | 87% |
| **11** | 94% | 78% |
| **12** | 94% | 73% |
| **13** | 95% | 65% |
| **14** | 96% | 62% |
| **15** (Best for Live Calls) | 97% | 46% |

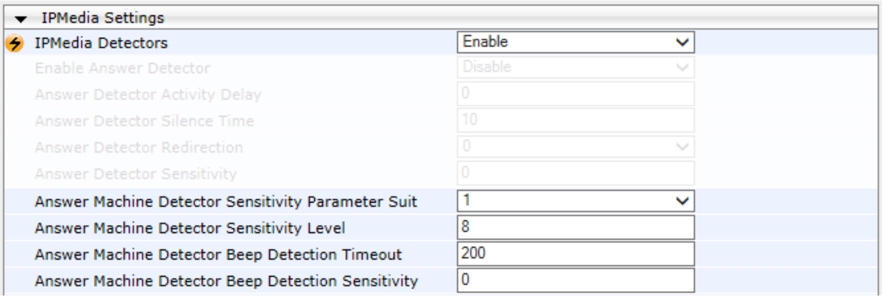
### 14.5.1 Configuring AMD

You can configure AMD for all calls using the global AMD parameters, or for specific calls using IP Profiles. The procedure below describes how to configure AMD for all calls. For configuring AMD for specific calls, use the AMD parameters in the IP Profile table (see "Configuring IP Profiles" on page 366). For Gateway calls, AMD can be configured per call based on the called number or Hunt Group. This is achieved by configuring AMD for a specific IP Profile and then assigning the IP Profile to a Hunt Group in the Inbound IP Routing table (see Configuring IP-to-Hunt Group Routing Rules on page 421).

➢ **To enable and configure AMD for all calls:**

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

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




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

For a complete list of AMD-related parameters, see "IP Media Parameters" on page 809.

## 14.6 Automatic Gain Control (AGC)

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

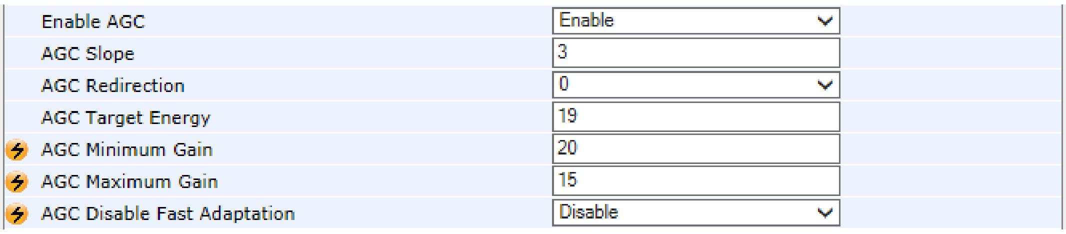
When the AGC first detects an incoming signal, it begins operating in Fast Mode, which allows the AGC to adapt quickly when a conversation starts. This means that the Gain Slope is 8 dB/sec for the first 1.5 seconds. After this period, the Gain Slope is changed to the user-defined value. You can disable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

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

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

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

**Figure 14-8: AGC Parameters in IPMedia Settings Page**




2. Configure the following parameters:
   - 'Enable AGC' (*EnableAGC*) - Enables the AGC mechanism.
   - 'AGC Slope' (*AGCGainSlope*) - Determines the AGC convergence rate.
   - 'AGC Redirection' (*AGCRedirection*) - Determines the AGC direction.
   - 'AGC Target Energy' - Defines the signal energy value (dBm) that the AGC attempts to attain.
   - 'AGC Minimum Gain' (AGCMinGain) - Defines the minimum gain (in dB) by the AGC when activated.
   - 'AGC Maximum Gain' (AGCMaxGain) - Defines the maximum gain (in dB) by the AGC when activated.
   - 'AGC Disable Fast Adaptation' (AGCDisableFastAdaptation) - Enables the AGC Fast Adaptation mode.
3. Click **Submit**.

# 14.7 Configuring Various Codec Attributes

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

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

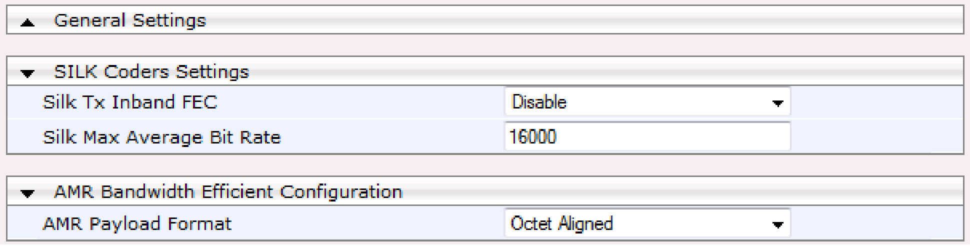
For a detailed description of these parameters and for additional codec parameters, see "Coder Parameters" on page 732.

➢ **To configure codec attributes:**

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

**General Media Settings**).

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




2. Configure the parameters as required, and then click click **Submit**.
3. To save the changes to flash memory, see "Saving Configuration" on page 490.

# 14.8 Configuring Analog Settings

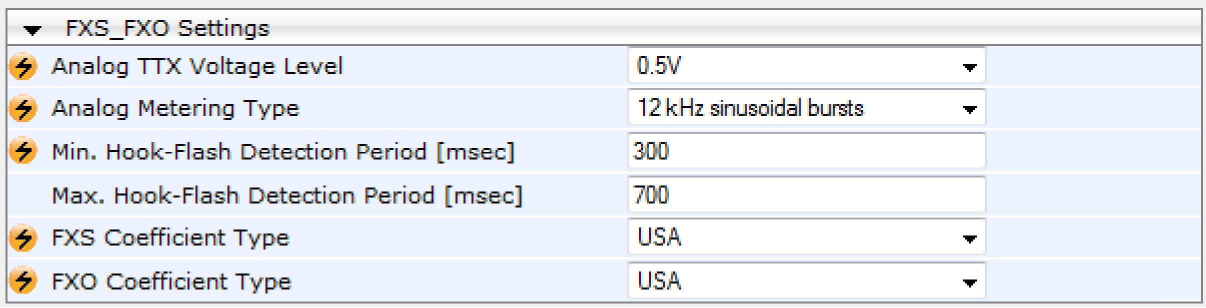
The Analog Settings page allows you to configure various analog parameters. For a detailed description of the parameters appearing on this page, see "Configuration Parameters Reference" on page 643.

This page also selects the type (USA or Europe) of FXS coefficient information. The FXS coefficient contains the analog telephony interface characteristics such as DC and AC impedance, feeding current, and ringing voltage.

➢ **To configure the analog parameters:**

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

**Figure 14-10: Analog Settings Page**




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

To save the changes to flash memory, see "Saving Configuration" on page 490.

# 14.9 Configuring Media (SRTP) Security

The device supports Secured RTP (SRTP) according to RFC 3711. SRTP is used to encrypt RTP and RTCP transport for protecting VoIP traffic. SRTP requires a cryptographic key exchange mechanism to negotiate the keys. To negotiate the keys, the device supports the Session Description Protocol Security Descriptions (SDES) protocol (according to RFC 4568). The key exchange is done by adding the 'a=crypto' attribute to the SDP. This attribute is used (by both sides) to declare the various supported cipher suites and to attach the encryption key. If negotiation of the encryption data is successful, the call is established.

SRTP supports the following cipher suites (all other suites are ignored):

- AES_CM_128_HMAC_SHA1_32
- AES_CM_128_HMAC_SHA1_80
- ARIA_CM_128_HMAC_SHA1_80
- ARIA_CM_192_HMAC_SHA1_80

When the device is the offering side (SDP offer), it can generate a Master Key Identifier (MKI). You can configure the MKI size globally (using the SRTPTxPacketMKISize parameter) or per SIP entity (using the IP Profile parameter, IpProfile_MKISize). The length of the MKI is limited to four bytes. If the remote side sends a longer MKI, the key is ignored.

The key lifetime field is not supported. However, if it is included in the key it is ignored and the call does not fail.

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

- UNENCRYPTED_SRTP
- UNENCRYPTED_SRTCP
- UNAUTHENTICATED_SRTP

Session parameters should be the same for the local and remote sides. When the device is the offering side, the session parameters are configured by the following parameter - 'Authentication On Transmitted RTP Packets', 'Encryption On Transmitted RTP Packets, and 'Encryption On Transmitted RTCP Packets'. When the device is the answering side, the device adjusts these parameters according to the remote offering. Unsupported session parameters are ignored, and do not cause a call failure.

Below is an example of crypto attributes usage:

```
a=crypto:1 AES_CM_128_HMAC_SHA1_80
inline:PsKoMpHlCg+b5X0YLuSvNrImEh/dAe
a=crypto:2 AES_CM_128_HMAC_SHA1_32
inline:IsPtLoGkBf9a+c6XVzRuMqHlDnEiAd
```

The device also supports symmetric MKI negotiation, whereby it can forward the MKI size received in the SDP offer 'a=crypto' line in the SDP answer. You can enable symmetric MKI globally (using the EnableSymmetricMKI parameter) or per SIP entity (using the IP Profile parameter, IpProfile_EnableSymmetricMKI. For more information on symmetric MKI, see "Configuring IP Profiles" on page 366.

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

You can also enable the device to validate the authentication of packets for SRTP tunneling for RTP and RTCP. This applies only to SRTP-to-SRTP SBC calls and where the endpoints use the same key. This is configured using the 'SRTP Tunneling Authentication for RTP' and 'SRTP Tunneling Authentication for RTCP' parameters.

**Notes:**

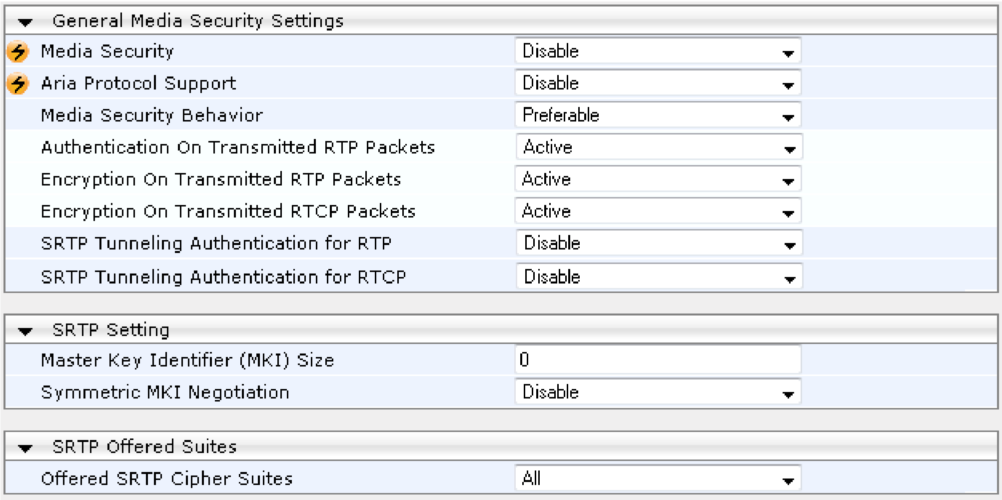
- For a detailed description of the SRTP parameters, see "Configuring IP Profiles" on page 366 and "SRTP Parameters" on page 680.
- When SRTP is used, the channel capacity may be reduced.

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

**➢ To enable and configure SRTP:**

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

**Figure 14-11: Media Security Page**



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

# 15 Services

This section describes configuration for various supported services.

## 15.1 DHCP Server Functionality

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

The DHCP server can service up to 800 DHCP clients. The DHCP clients are typically IP phones that are connected to the device's LAN port.

The DHCP server is activated when you configure a valid entry in the DHCP Servers table (see "Configuring the DHCP Server" on page 213) and associate it with an active IP network interface (listed in the Interface table). When an IP phone on the LAN requests an IP address, the DHCP server allocates one from the address pool. In scenarios of duplicated IP addresses on the LAN (i.e., an unauthorized network device using one of the IP addresses of the DHCP address pool), the DHCP server detects this condition using an Address Resolution Protocol (ARP) request and temporarily blacklists the duplicated address.

You can also configure the DHCP server to respond **only** to DHCPDiscover requests from DHCP clients that contain a specific value for Option 60 (Vendor Class Identification). For more information, see "Configuring the Vendor Class Identifier" on page 218.

### 15.1.1 Configuring the DHCP Server

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

**Table 15-1: Configurable DHCP Options in DHCP Servers Table**

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

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

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

- DHCP Vendor Class Identifier names (DHCP Option 60) - see "Configuring the Vendor Class Identifier" on page 218
- Additional DHCP Options - see "Configuring Additional DHCP Options" on page 219
- Static IP addresses for DHCP clients - see "Configuring Static IP Addresses for DHCP Clients" on page 221

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

To view and delete currently serviced DHCP clients, see "Viewing and Deleting DHCP Clients" on page 222.

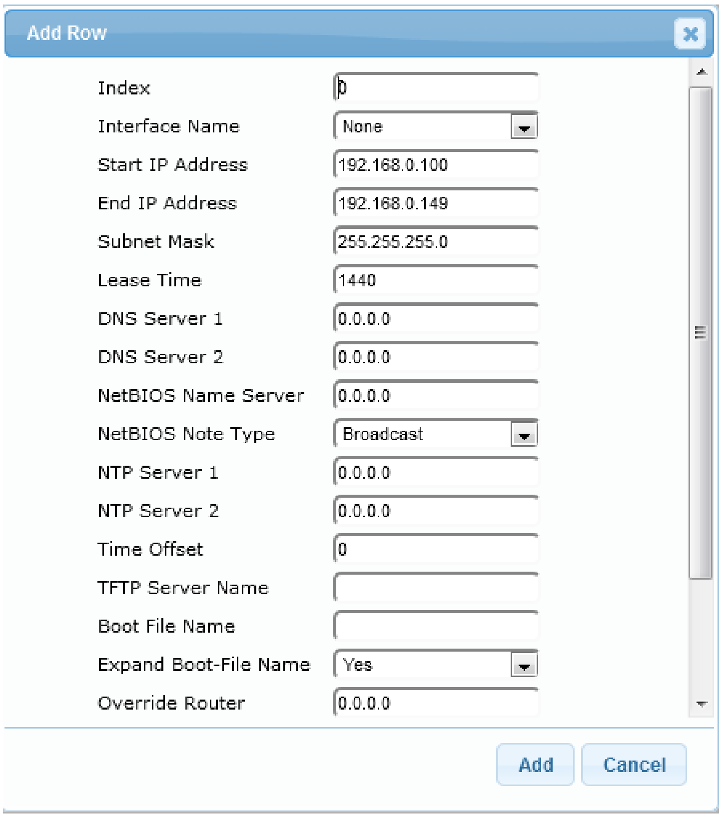
The following procedure describes how to configure the DHCP server through the Web interface. You can also configure it through ini file (DhcpServer) or CLI (configure voip > dhcp server <index>).

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

1. Open the DHCP Servers page (**Configuration** tab > **VoIP** menu > **Services** > **DHCP Severs**).

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

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




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

**Table 15-2: DHCP Servers Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>`dhcp server <index>` | Defines an index number for the new table row.<br>**Notes:**<br>▪ Each row must be configured with a unique index.<br>▪ Currently, only one index row can be configured. |
| Interface Name<br>`network-if`<br>[DhcpServer_InterfaceName] | Associates an IP interface on which the DHCP server operates. The IP interfaces are configured in the Interface table (see "Configuring IP Network Interfaces" on page 135).<br>By default, no value is defined. |
| Start IP Address<br>`start-address`<br>[DhcpServer_StartIPAddress] | Defines the starting IP address (IPv4 address in dotted-decimal format) of the IP address pool range used by the DHCP server to allocate addresses.<br>The default value is 192.168.0.100.<br>**Note:** The IP address must belong to the same subnet as the associated interface's IP address. |

| Parameter | Description |
|---|---|
| End IP Address<br>`end-address`<br>[DhcpServer_EndIPAddress] | Defines the ending IP address (IPv4 address in dotted-decimal format) of the IP address pool range used by the DHCP server to allocate addresses.<br>The default value is 192.168.0.149.<br>**Note:** The IP address must belong to the same subnet as the associated interface's IP address and must be "greater or equal" to the starting IP address defined in 'Start IP Address'. |
| Subnet Mask<br>`subnet-mask`<br>[DhcpServer_SubnetMask] | Defines the subnet mask (for IPv4 addresses) for the DHCP client. The value is sent in DHCP Option 1 (Subnet Mask).<br>The default value is 0.0.0.0.<br>**Note:** The value must be "narrower" or equal to the subnet mask of the associated interface's IP address. If set to "0.0.0.0", the subnet mask of the associated interface is used. |
| Lease Time<br>`lease-time`<br>[DhcpServer_LeaseTime] | Defines the duration (in minutes) of the lease time to a DHCP client for using an assigned IP address. The client needs to request a new address before this time expires. The value is sent in DHCP Option 51 (IP Address Lease Time).<br>The valid value range is 0 to 214,7483,647. The default is 1440. When set to 0, the lease time is infinite. |
| DNS Server 1<br>`dns-server-1`<br>[DhcpServer_DNSServer1] | Defines the IP address (IPv4) of the primary DNS server that the DHCP server assigns to the DHCP client. The value is sent in DHCP Option 6 (Domain Name Server).<br>The default value is 0.0.0.0. |
| DNS Server 2<br>`dns-server-2`<br>[DhcpServer_DNSServer2] | Defines the IP address (IPv4) of the secondary DNS server that the DHCP server assigns to the DHCP client. The value is sent in DHCP Option 6 (Domain Name Server). |
| NetBIOS Name Server<br>`netbios-server`<br>[DhcpServer_NetbiosNameServer] | Defines the IP address (IPv4) of the NetBIOS WINS server that is available to a Microsoft DHCP client. The value is sent in DHCP Option 44 (NetBIOS Name Server).<br>The default value is 0.0.0.0. |
| NetBIOS Node Type<br>`netbios-node-type`<br>[DhcpServer_NetbiosNodeType] | Defines the node type of the NetBIOS WINS server for a Microsoft DHCP client. The value is sent in DHCP Option 46 (NetBIOS Node Type).<br>▪ **[0]** Broadcast (default)<br>▪ **[1]** peer-to-peer<br>▪ **[4]** Mixed<br>▪ **[8]** Hybrid |
| NTP Server 1<br>`ntp-server-1`<br>[DhcpServer_NTPServer1] | Defines the IP address (IPv4) of the primary NTP server that the DHCP server assigns to the DHCP client. The value is sent in DHCP Option 42 (Network Time Protocol Server).<br>The default value is 0.0.0.0. |
| NTP Server 2<br>`ntp-server-2`<br>[DhcpServer_NTPServer2] | Defines the IP address (IPv4) of the secondary NTP server that the DHCP server assigns to the DHCP client. The value is sent in DHCP Option 42 (Network Time Protocol Server).<br>The default value is 0.0.0.0. |

| Parameter | Description |
|---|---|
| Time Offset<br>`time-offset`<br>[DhcpServer_TimeOffset] | Defines the Greenwich Mean Time (GMT) offset (in seconds) that the DHCP server assigns to the DHCP client. The value is sent in DHCP Option 2 (Time Offset).<br>The valid range is -43200 to 43200. The default is 0. |
| TFTP Server<br>`tftp-server-name`<br>[DhcpServer_TftpServer] | Defines the IP address or name of the TFTP server that the DHCP server assigns to the DHCP client. The TFTP server typically stores the boot file image, defined in the 'Boot file name' parameter (see below). The value is sent in DHCP Option 66 (TFTP Server Name).<br>The valid value is a string of up to 80 characters. By default, no value is defined. |
| Boot file name<br>`boot-file-name`<br>[DhcpServer_BootFileName] | Defines the name of the boot file image for the DHCP client. The boot file stores the boot image for the client. The boot image is typically the operating system the client uses to load (downloaded from a boot server). The value is sent in DHCP Option 67 (Bootfile Name). To define the server storing the file, use the 'TFTP Server' parameter (see above).<br>The valid value is a string of up to 256 characters. By default, no value is defined.<br>The name can also include the following case-sensitive placeholder strings that are replaced with actual values if the 'Expand Boot-file Name' parameter is set to **Yes**:<br>▪ <MAC>: Replaced by the MAC address of the client (e.g., *boot_<MAC>.ini*). The MAC address is obtained in the client's DHCP request.<br>▪ <IP>: Replaced by the IP address assigned by the DHCP server to the client. |
| Expand Boot-file Name<br>`expand-boot-file-name`<br>[DhcpServer_ExpandBootfileName] | Enables the use of the placeholders in the boot file name, defined in the 'Boot file name' parameter.<br>▪ **[0]** No<br>▪ **[1]** Yes (default) |
| Override Router<br>`override-router-address`<br>[DhcpServer_OverrideRouter] | Defines the IP address (IPv4 in dotted-decimal notation) of the default router that the DHCP server assigns the DHCP client. The value is sent in DHCP Option 3 (Router).<br>The default value is 0.0.0.0. If not specified (empty or “0.0.0.0”), the IP address of the default gateway configured in the Interface table for the IP network interface that you associated with the DHCP server (see the 'Interface Name' parameter above) is used. |
| SIP Server<br>`sip-server`<br>[DhcpServer_SipServer] | Defines the IP address or DNS name of the SIP server that the DHCP server assigns the DHCP client. The client uses this SIP server for its outbound SIP requests. The value is sent in DHCP Option 120 (SIP Server). After defining the parameter, use the 'SIP server type' parameter (see below) to define the type of address (FQDN or IP address).<br>The valid value is a string of up to 256 characters. The default is 0.0.0.0. |

| Parameter | Description |
|---|---|
| SIP server type<br>`sip-server-type`<br>[DhcpServer_SipServerType] | Defines the type of SIP server address. The actual address is defined in the 'SIP server' parameter (see above). Encoding is done per SIP Server Type, as defined in RFC 3361.<br>▪ **[0]** DNS names = (Default) The 'SIP server' parameter is configured with an FQDN of the SIP server.<br>▪ **[1]** IP address = The 'SIP server' parameter configured with an IP address of the SIP server. |

## 15.1.2 Configuring the Vendor Class Identifier

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

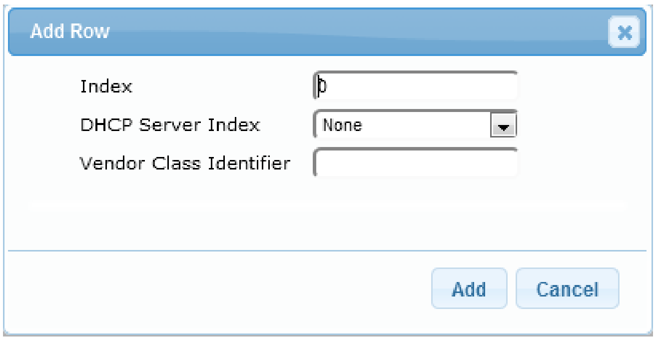
The VCI is a string that identifies the vendor and functionality of a DHCP client to the DHCP server. For example, Option 60 can show the unique type of hardware (e.g., "AudioCodes 440HD IP Phone") or firmware of the DHCP client. The DHCP server can then differentiate between DHCP clients and process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

The following procedure describes how to configure the DHCP VCIs through the Web interface. You can also configure it through ini file (DhcpVendorClass) or CLI (configure voip > dhcp vendor-class).

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

1. Open the DHCP Servers table (**Configuration** tab > **VoIP** menu > **Services** > **DHCP Severs**).
2. In the table, select the row of the desired DHCP server for which you want to configure VCIs, and then click the **DHCP Vendor Class Table** link, located below the table; the DHCP Vendor Class table opens.
3. Click **Add**; the following dialog box appears:

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




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

**Table 15-3: DHCP Vendor Class Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>dhcp vendor-class <index><br>[DhcpVendorClass_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| DHCP Server Index<br>dhcp-server-number<br>[DhcpVendorClass_DhcpServerIndex] | Associates the VCI table entry with a DHCP server that you configured in "Configuring the DHCP Server" on page 213.<br>**Note:** Currently, only one DHCP server (Index 0) can be configured and therefore, the parameter is always set at 0. |
| Vendor Class Identifier<br>vendor-class<br>[DhcpVendorClass_VendorClassId] | Defines the value of the VCI DHCP Option 60.<br>The valid value is a string of up to 80 characters. By default, no value is defined. |

### 15.1.3 Configuring Additional DHCP Options

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

The following procedure describes how to configure DHCP Options through the Web interface. You can also configure it through ini file (DhcpOption) or CLI (configure voip > dhcp option).

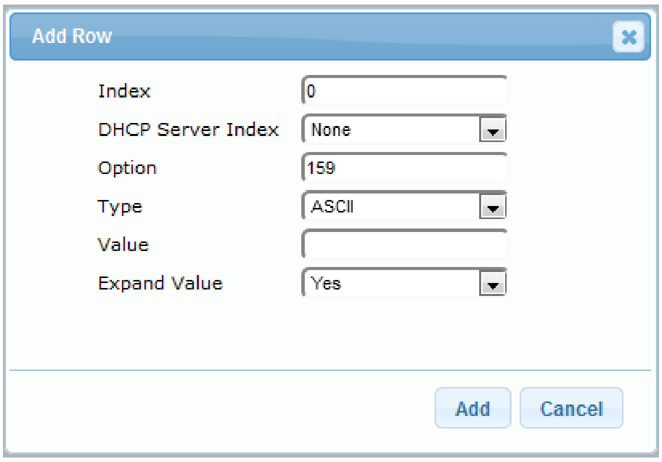
**Note:** The additional DHCP Options configured in the DHCP Option table override the default ones, which are configured in the DHCP Servers table. In other words, if you configure Option 67 in the DHCP Option table, the device uses the value configured in the DHCP Option table instead of the value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

➢ **To configure DHCP Options:**

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

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

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




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

**Table 15-4: DHCP Option Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>`dhcp option`<br>[DhcpOption_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| DHCP Server Index<br>`dhcp-server-number`<br>[DhcpOption_DhcpServerIndex] | Associates the DHCP Option table entry with a DHCP server that you configured in "Configuring the DHCP Server" on page 213.<br>**Note:** Currently, only one DHCP server (Index 0) can be configured and therefore, the parameter is always set at 0. |
| Option<br>`option`<br>[DhcpOption_Option] | Defines the code of the DHCP Option.<br>The valid value is 1 to 254. The default is 159.<br>For example, for DHCP Option 150 (Cisco proprietary for defining multiple TFTP server IP addresses), enter the value 150. |
| Type<br>`type`<br>[DhcpOption_Type] | Defines the format (type) of the DHCP Option value that is configured in the 'Value' parameter (see below).<br>▪ **[0]** ASCII = (Default) Plain-text string (e.g., when the value is a domain name).<br>▪ **[1]** IP address = IPv4 address.<br>▪ **[2]** Hexadecimal = Hexadecimal-encoded string.<br>For example, if you set the 'Value' parameter to "company.com", you need to set the 'Type' parameter to **ASCII**. |

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

### 15.1.4 Configuring Static IP Addresses for DHCP Clients

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

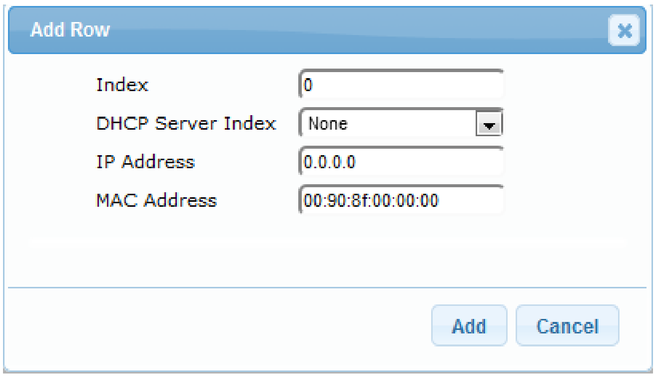
The following procedure describes how to configure static IP addresses for DHCP clients through the Web interface. You can also configure it through ini file (DhcpStaticIP) or CLI (configure voip > dhcp static-ip <index>).

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

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

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

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




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

**Table 15-5: DHCP Static IP Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>`dhcp static-ip <index>`<br>[DhcpStaticIP_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| DHCP Server Index<br>`dhcp-server-number`<br>[DhcpStaticIP_DhcpServerIndex] | Associates the DHCP Static IP table entry with a DHCP server that you configured in "Configuring the DHCP Server" on page 213.<br>**Note:** Currently, only one DHCP server (Index 0) can be configured and therefore, the parameter is always set at 0. |
| IP Address<br>`ip-address`<br>[DhcpStaticIP_IPAddress] | Defines the "reserved", static IP address (IPv4) to assign the DHCP client.<br>The default is 0.0.0.0. |
| MAC Address<br>`mac-address`<br>[DhcpStaticIP_MACAddress] | Defines the DHCP client by MAC address (in hexadecimal format).<br>The valid value is a string of up to 20 characters. The format includes six groups of two hexadecimal digits, each separated by a colon. The default MAC address is 00:90:8f:00:00:00. |

## 15.1.5 Viewing and Deleting DHCP Clients

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

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

- To view DHCP clients:

```
# show voip dhcp clients
```

- To view DHCP clients according to IP address:

```
# show voip dhcp ip
```

- To view DHCP clients according to MAC address:

```
# show voip dhcp mac
```

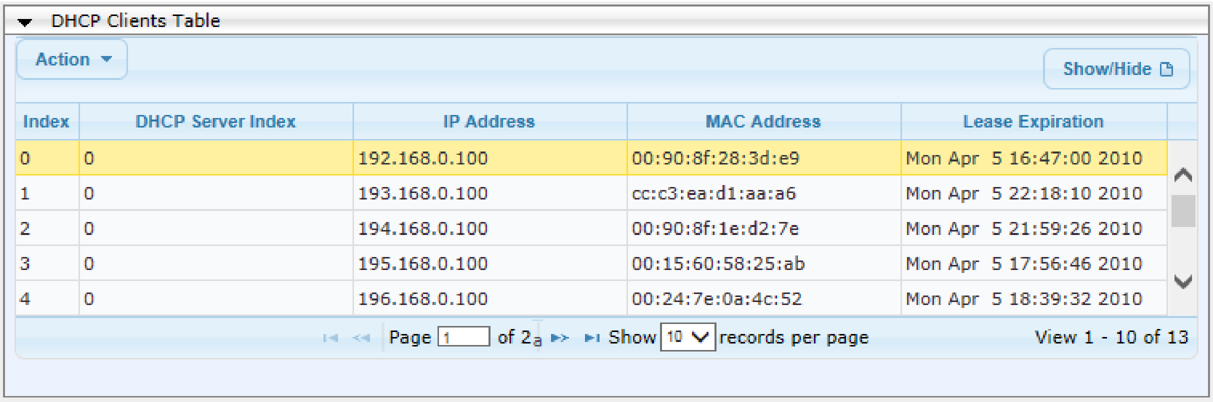
- To view DHCP clients that have been blacklisted from DHCP implementation (due to duplicated IP addresses in the network, where another device is using the same IP address as the one assigned to the client):

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

### ➢ To view or delete DHCP clients:

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

**Figure 15-5: DHCP Clients Table**



DHCP Clients Table

Action

Show/Hide

| Index | DHCP Server Index | IP Address | MAC Address | Lease Expiration |
|---|---|---|---|---|
| 0 | 0 | 192.168.0.100 | 00:90:8f:28:3d:e9 | Mon Apr 5 16:47:00 2010 |
| 1 | 0 | 193.168.0.100 | cc:c3:ea:d1:aa:a6 | Mon Apr 5 22:18:10 2010 |
| 2 | 0 | 194.168.0.100 | 00:90:8f:1e:d2:7e | Mon Apr 5 21:59:26 2010 |
| 3 | 0 | 195.168.0.100 | 00:15:60:58:25:ab | Mon Apr 5 17:56:46 2010 |
| 4 | 0 | 196.168.0.100 | 00:24:7e:0a:4c:52 | Mon Apr 5 18:39:32 2010 |

Page 1 of 2 Show 10 records per page View 1 - 10 of 13

The table displays the following per client:

- **Index:** Table index number.
- **DHCP Server Index:** The index number of the configured DHCP server scope in the DHCP Server table (see "Configuring the DHCP Server" on page 213) with which the client is associated.
- **IP Address:** IP address assigned to the DHCP client by the DHCP server.
- **MAC Address:** MAC address of the DHCP client.
- **Lease Expiration:** Date on which the lease of the DHCP client's IP address obtained from the DHCP server expires.

3. To delete a client:
   a. Select the table row index of the DHCP client that you want to delete.
   b. Click the **Action** button, and then from the drop-down menu, choose **Delete**; a confirmation message appears.
   c. Click **OK** to confirm deletion.

## 15.2 RADIUS-based Services

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

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

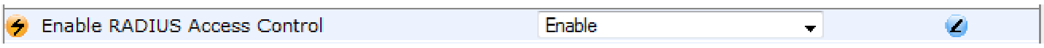
### 15.2.1 Enabling RADIUS Services

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

➢ **To enable RADIUS:**

1. Open the Call Detail Record Settings page (**Configuration** tab > **VoIP** menu > **Services** > **Call Detailed Record** > **Call Detail Record Settings**).

**Figure 15-6: Enabling RADIUS**




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

### 15.2.2 Configuring RADIUS Servers

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

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

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

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

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

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

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

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

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

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

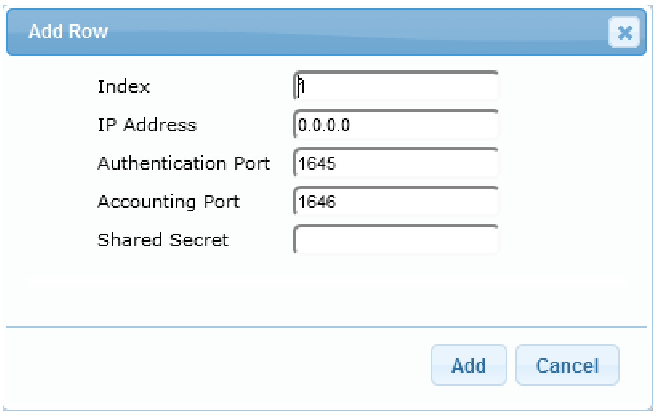
**Note:** To enable and configure RADIUS-based accounting, see "Configuring RADIUS Accounting" on page 588.

### ➢ To configure a RADIUS server:

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

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

**Figure 15-7: RADIUS Servers Table - Add Row Dialog Box**




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

**Table 15-6: RADIUS Servers Table Parameter Descriptions**

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

### 15.2.3 Configuring Interface for RADIUS Communication

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

**Note:** If set to Control, only one Control interface must be configured in the Interface table (see "Configuring IP Network Interfaces" on page 135); otherwise, RADIUS communication will fail.

### 15.2.4 Configuring General RADIUS Parameters

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

➢ **To configure general RADIUS parameters:**

1. Open the Authentication Settings page (**Configuration** tab > **System** menu > **Management** > **Authentication Settings**).
2. Scroll down the page to the RADIUS Settings group.
3. In the 'RADIUS VSA Vendor ID' field, enter the **same** vendor ID number as set on the third-party RADIUS server. The vendor-specific attribute (VSA) identifies the device to the RADIUS server using the Vendor ID. For an example of using the Vendor ID, see "Setting Up a Third-Party RADIUS Server" on page 228.
4. Configure RADIUS packet retransmission when no response is received from the RADIUS server:
    a. In the 'RADIUS Packets Retransmission' field (RADIUSRetransmission), enter the maximum number of RADIUS retransmissions that the device performs if no response is received from the RADIUS server.
    b. In the 'RADIUS Response Time Out' field (RadiusTO), enter the interval (in seconds) that the device waits for a response before sending a RADIUS retransmission.
5. Click **Submit**.

### 15.2.5 RADIUS-based Management User Authentication

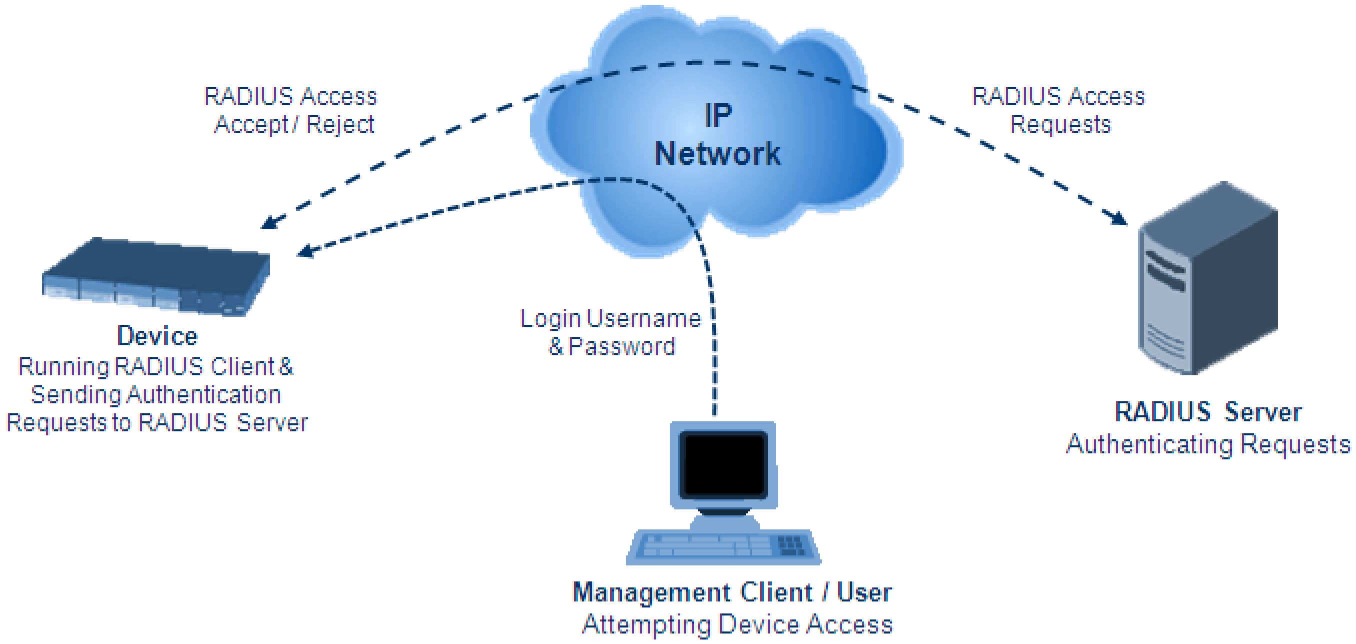
You can enhance security for your device by implementing Remote Authentication Dial-In User Service (RADIUS - RFC 2865) for authenticating multiple management user accounts of the device's embedded Web and Telnet (CLI) servers. Thus, RADIUS also prevents unauthorized access to your device.

When RADIUS authentication is not used, the user's login username and password are locally authenticated by the device in its Web Users table (database). However, the Web Users table can be used as a fallback mechanism in case the RADIUS server does not respond. For configuring local user accounts, see "Configuring Web User Accounts" on page 65.

When RADIUS authentication is used, the RADIUS server stores the user accounts - usernames, passwords, and access levels (authorization). When a management user (client) tries to access the device, the device sends the RADIUS server the user's username and password for authentication. The RADIUS server replies with an acceptance or a rejection notification. During the RADIUS authentication process, the device's Web interface is blocked until an acceptance response is received from the RADIUS server.

Note that communication between the device and the RADIUS server is done by using a shared secret, which is not transmitted over the network.

**Figure 15-8: RADIUS Login Authentication for Management**




For using RADIUS, you need to do the following:

- Set up a RADIUS server (third-party) to communicate with the device - see "Setting Up a Third-Party RADIUS Server" on page 228
- Configure the device as a RADIUS client for communication with the RADIUS server - see "Configuring RADIUS Authentication" on page 229

### 15.2.5.1 Setting Up a Third-Party RADIUS Server

The following procedure provides an example for setting up a third-party RADIUS sever, *FreeRADIUS*, which can be downloaded from www.freeradius.org. Follow the instructions on this Web site for installing and configuring the server. If you use a RADIUS server from a different vendor, refer to its appropriate documentation.

➢ **To set up a third-party RADIUS server (e.g., *FreeRADIUS*):**

1. Define the device as an authorized client of the RADIUS server, with the following:
   - Predefined *shared secret* (password used to secure communication between the device and the RADIUS server)
   - Vendor ID

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

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

2. If access levels are required, set up a Vendor-Specific Attributes (VSA) dictionary for the RADIUS server and select an attribute ID that represents each user's access level. The example below shows a dictionary file for FreeRADIUS that defines the attribute "ACL-Auth-Level" with "ID=35". For the device's user access levels and their corresponding numeric representation in RADIUS servers, see "Configuring Web User Accounts" on page 65.

```
#
# AudioCodes VSA dictionary
#
VENDOR AudioCodes 5003
ATTRIBUTE ACL-Auth-Level 35 integer AudioCodes
VALUE ACL-Auth-Level ACL-Auth-UserLevel 50
VALUE ACL-Auth-Level ACL-Auth-AdminLevel 100
VALUE ACL-Auth-Level ACL-Auth-SecurityAdminLevel 200
```

3. Define the list of users authorized to use the device, using one of the password authentication methods supported by the server implementation. The example below shows a user configuration file for FreeRADIUS using a plain-text password:

```
# users - local user configuration database

john    Auth-Type := Local, User-Password == "qwerty"
        Service-Type = Login-User,
        ACL-Auth-Level = ACL-Auth-SecurityAdminLevel

sue   Auth-Type := Local, User-Password == "123456"
        Service-Type = Login-User,
        ACL-Auth-Level = ACL-Auth-UserLevel
```

4. Record and retain the IP address, port number, shared secret code, vendor ID, and VSA access level identifier (if access levels are implemented) used by the RADIUS server.

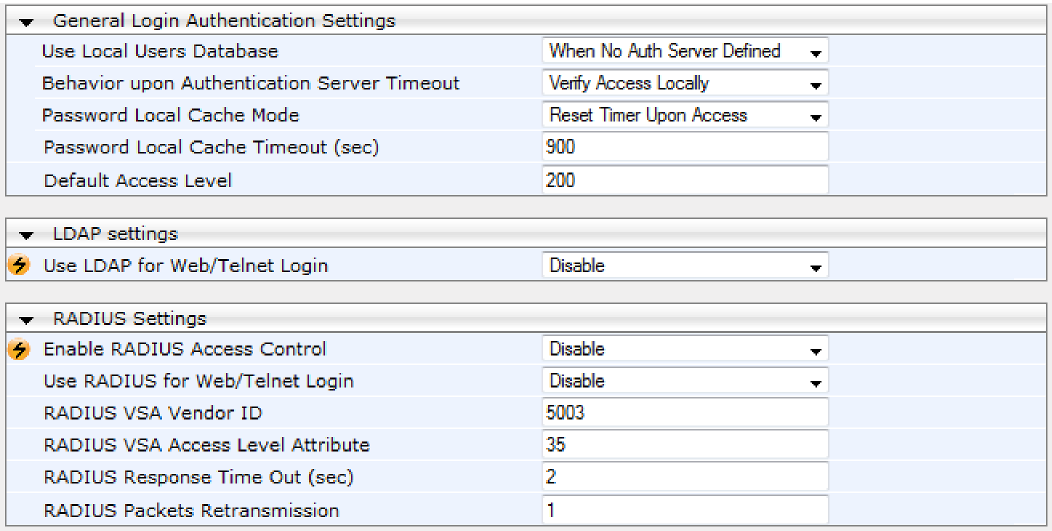
### 15.2.5.2 Configuring RADIUS-based User Authentication

The following procedure describes how to configure the RADIUS parameters specific to login authentication. For a detailed description of the RADIUS parameters, see "RADIUS Parameters" on page 814.

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

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

**Figure 15-9: Authentication Settings Page - RADIUS Configuration**




2. From the 'Use RADIUS for Web/Telnet Login' drop-down list, select **Enable** to enable RADIUS authentication for Web and Telnet login.
3. When implementing Web user access levels, do one of the following:
   - **If the RADIUS server response includes the access level attribute:** In the 'RADIUS VSA Access Level Attribute' field, enter the code that indicates the access level attribute in the VSA section of the received RADIUS packet. For defining the RADIUS server with access levels, see "Setting Up a Third-Party RADIUS Server" on page 228.
   - **If the RADIUS server response does not include the access level attribute:** In the 'Default Access Level' field, enter the default access level that is applied to all users authenticated by the RADIUS server.
4. Configure RADIUS timeout handling:
   a. From the 'Behavior upon Authentication Server Timeout' drop-down list, select the option if the RADIUS server does not respond within five seconds:
      - **Deny Access**: device denies user login access.
      - **Verify Access Locally**: device checks the username and password configured locally for the user (in the Web User Accounts page or Web Users table), and if correct, allows access.

   b. In the 'Password Local Cache Timeout' field, enter a time limit (in seconds) after which the username and password verified by the RADIUS server becomes invalid and a username and password needs to be re-validated with the RADIUS server.
   c. From the 'Password Local Cache Mode' drop-down list, select the option for the local RADIUS password cache timer:
      - **Reset Timer Upon Access**: upon each access to a Web page, the timer resets (reverts to the initial value configured in the previous step).
      - **Absolute Expiry Timer**: when you access a Web page, the timer doesn't reset, but continues its count down.

5. Configure when the Web Users table must be used to authenticate login users. From the 'Use Local Users Database' drop-down list, select one of the following:
    - **When No Auth Server Defined (default):** When no RADIUS server is configured or if a server is configured but connectivity with the server is down (if the server is up, the device authenticates the user with the server).
    - **Always:** First attempts to authenticate the user using the Web Users table, but if not found, it authenticates the user with the RADIUS server.
6. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

#### 15.2.5.3 Securing RADIUS Communication

RADIUS authentication requires HTTP basic authentication (according to RFC 2617). However, this is insecure as the usernames and passwords are transmitted in clear text over plain HTTP. Thus, as digest authentication is not supported with RADIUS, it is recommended that you use HTTPS with RADIUS so that the usernames and passwords are encrypted.

To configure the device to use HTTPS, set the 'Secured Web Connection (HTTPS)' parameter to **HTTPS Only**, in the Web Security Settings page (**Configuration** tab > **System** menu > **Management** > **Web Security Settings**).

#### 15.2.5.4 RADIUS-based User Authentication in URL

RADIUS authentication of the management user is typically done after the user accesses the Web interface by entering only the device's IP address in the Web browser's URL field (for example, http://10.13.4.12/) and then entering the username and password credentials in the Web interface's login screen. However, authentication with the RADIUS server can also be done immediately after the user enters the URL, if the URL also contains the login credentials. For example: http://10.4.4.112/Forms/RadiusAuthentication?WSBackUserName=John&WSBackPassword=1234

**Note:** This feature allows up to five simultaneous users only.

### 15.2.6 RADIUS-based CDR Accounting

Once you have configured a RADIUS server(s) for accounting in "Configuring RADIUS Servers" on page 224, you need to enable and configure RADIUS-based CDR accounting (see "Configuring RADIUS Accounting" on page 588).

## 15.3 LDAP-based Management and SIP Services

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

You can use LDAP for the following LDAP services:

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

  For the device to run a search, the path to the directory's subtree, known as the distinguished name (DN), where the search is to be done must be configured (see "Configuring LDAP DNs (Base Paths) per LDAP Server" on page 240). The search key (filter), which defines the exact DN to search, and one or more attributes whose values must be returned to the device must also be configured. For more information on configuring these attributes and search filters, see "Active Directory-based Routing for Microsoft Lync" on page 255.

  The device can store recent LDAP queries and responses in its local cache. The cache is used for subsequent queries and/or in case of LDAP server failure. For more information, see "Configuring the Device's LDAP Cache" on page 245.

  If connection with the LDAP server disconnects (broken), the device sends the SNMP alarm, acLDAPLostConnection. Upon successful reconnection, the alarm clears. If connection with the LDAP server is disrupted during the search, all search requests are dropped and an alarm indicating a failed status is sent to client applications.

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

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

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

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

  The device then assigns the user the access level configured for that group (in "Configuring Access Level per Management Groups Attributes" on page 242). The location in the directory where you want to search for the user's member group(s) is configured using the following:

- Search base object (distinguished name or DN, e.g., "ou=ABC,dc=corp,dc=abc,dc=com"), which defines the location in the directory from where the LDAP search begins, and is configured in "Configuring LDAP DNs (Base Paths) per LDAP Server" on page 240.
- Search filter, for example, (&(objectClass=person)(sAMAccountName=JohnD)), which filters the search in the subtree to include only the specific username. The search filter can be configured with the dollar ($) sign to represent the username, for example, (sAMAccountName=$). For configuring the search filter, see "Configuring the LDAP Search Filter Attribute" on page 241.
- Management attribute (e.g., memberOf), from where objects that match the search filter criteria are returned. This shows the user's member groups. The attribute is configured in the LDAP Configuration table (see "Configuring LDAP Servers" on page 236).

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

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

- Search method for searching DN object records between LDAP servers and within each LDAP server (see Configuring LDAP Search Methods).
- Default access level that is assigned to the user if the queried response does not contain an access level.
- Local users database (Web Users table) for authenticating users instead of the LDAP server (for example, when a communication problem occurs with the server). For more information, see "Configuring Local Database for Management User Authentication" on page 248.

## 15.3.1 Enabling the LDAP Service

Before you can configure LDAP support, you need to enable the LDAP service.

➢ **To enable LDAP:**

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

**Figure 15-10: Enabling LDAP on the LDAP Settings Page**


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

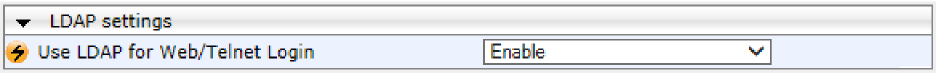
## 15.3.2 Enabling LDAP-based Web/CLI User Login Authentication and Authorization

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

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

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

**Figure 15-11: Authentication Settings Page - Enabling LDAP-based Login**




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

## 15.3.3 Configuring LDAP Server Groups

The LDAP Server Groups table lets you configure up to 41 LDAP Server Groups. An LDAP Server Group is a logical configuration entity that contains up to two LDAP servers. LDAP servers are assigned to LDAP Server Groups in the LDAP Configuration table (see "Configuring LDAP Servers" on page 236). To use a configured LDAP server, you must assign it to an LDAP Server Group.

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

To use an LDAP server for management user login authentication and authorization, you must configure its' LDAP Server Group as "Management" type. Additional LDAP-based management parameters need to be configured, as described in "Enabling LDAP-based Web/CLI User Login Authentication and Authorization" on page 234 and "Configuring LDAP Servers" on page 236.

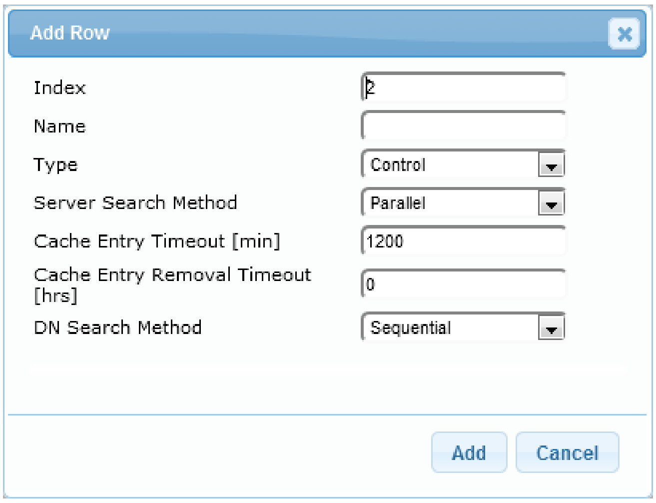
The following procedure describes how to configure an LDAP Server Group through the Web interface. You can also configure it through ini file (LDAPServersGroup) or CLI (configure voip/ldap/ldap-servers-group).

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

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

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

**Figure 15-12: LDAP Server Groups Table - Add Row Dialog Box**




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

**Table 15-7: LDAP Server Groups Table Parameter Descriptions**

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

| Parameter | Description |
|---|---|
| Cache Entry Removal Timeout<br>`cache-entry-removal-timeout`<br>[LdapServersGroup_CacheEntryRemovalTimeout] | Defines the duration (in hours) after which the LDAP entry is deleted from the device's LDAP cache.<br>The valid range is 0 to 596. The default is 0 (i.e., the entry is never deleted). |
| DN Search Method<br>`search-dn-method`<br>[LdapServersGroup_SearchDnsMethod] | Defines the method for querying the Distinguished Name (DN) objects within each LDAP server.<br>▪ [0] Sequential = (Default) The query is done in each DN object, one by one, until a result is returned. For example, a search for the DN object record "JohnD" is first run in DN object "Marketing" and if a result is not found, it searches in "Sales", and if not found, it searches in "Administration", and so on.<br>▪ [1] Parallel = The query is done in all DN objects at the same time. For example, a search for the DN object record "JohnD" is done at the same time in the "Marketing", "Sales" and "Administration" DN objects. |

## 15.3.4 Configuring LDAP Servers

The LDAP Configuration table lets you configure up to 82 LDAP servers. This table defines the address and connectivity settings of the LDAP server. The LDAP server can be configured for SIP-related queries (e.g., routing and manipulation) or LDAP-based management user login authentication and authorization (username-password).

The following procedure describes how to configure an LDAP server through the Web interface. You can also configure it through ini file (LdapConfiguration) or CLI (configure voip > ldap > ldap-configuration).

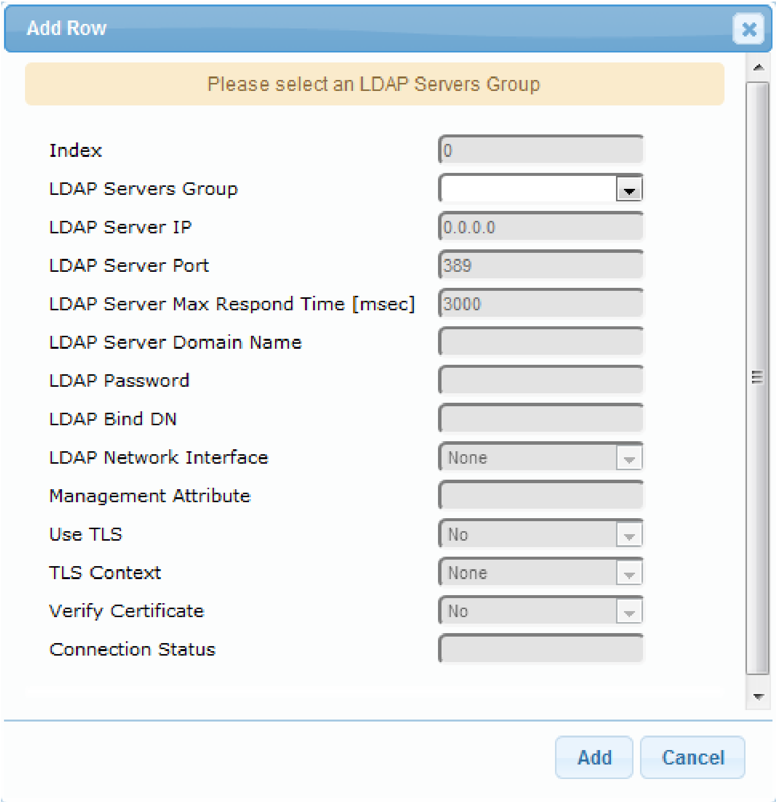
**Note:** When you configure an LDAP server, you need to assign it to an LDAP Server Group. Therefore, before you can configure an LDAP server in the table, you must first configure at least one LDAP Server Group in the LDAP Server Groups table (see "Configuring LDAP Server Groups" on page 234).

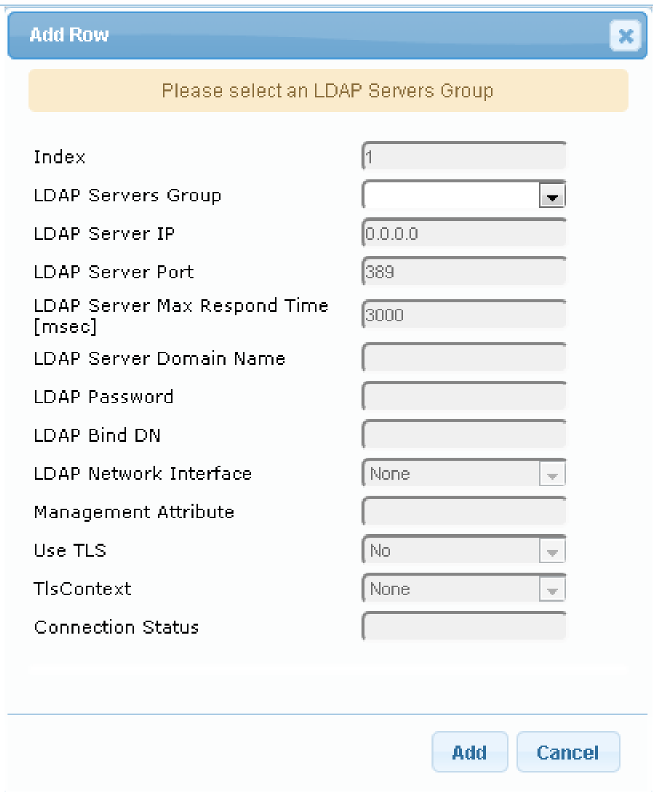
➢ **To configure an LDAP server:**

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

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

**Figure 15-13: LDAP Configuration Table - Add Row Dialog Box**




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

**Table 15-8: LDAP Configuration Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[LdapConfiguration_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| LDAP Servers Group<br>`server-group`<br>[LdapConfiguration_Group] | Assigns the LDAP server to an LDAP Server Group, configured in the LDAP Server Groups table (see "Configuring LDAP Server Groups" on page 234).<br>**Notes:**<br>▪ The parameter is mandatory and must be set before configuring the other parameters in the table.<br>▪ Up to two LDAP servers can be assigned to the same LDAP Server Group. |
| LDAP Server IP<br>`server-ip`<br>[LdapConfiguration_LdapConfServerIp] | Defines the IP address of the LDAP server (in dotted-decimal notation, e.g., 192.10.1.255).<br>By default, no IP address is defined.<br>**Notes:**<br>▪ The parameter is mandatory.<br>▪ If you want to use an FQDN for the LDAP server, leave the parameter undefined and configure the FQDN in the 'LDAP Server Domain Name' parameter (see below). |
| LDAP Server Port<br>`server-port`<br>[LdapConfiguration_LdapConfServerPort] | Defines the port number of the LDAP server.<br>The valid value range is 0 to 65535. The default port number is 389. |
| LDAP Server Max Respond Time<br>`max-respond-time`<br>[LdapConfiguration_LdapConfServerMaxRespondTime] | Defines the duration (in msec) that the device waits for LDAP server responses.<br>The valid value range is 0 to 86400. The default is 3000.<br>**Note:** If the response time expires, you can configure the device to use its local database (Web Users table) for authenticating the user. For more information, see "Configuring Local Database for Management User Authentication" on page 248. |
| LDAP Server Domain Name<br>`domain-name`<br>[LdapConfiguration_LdapConfServerDomainName] | Defines the domain name (FQDN) of the LDAP server. The device tries to connect to the LDAP server according to the IP address listed in the received DNS query. If there is no connection to the LDAP server or the connection to the LDAP server fails, the device tries to connect to the LDAP server with the next IP address in the DNS query list.<br>**Note:** If the 'LDAP Server IP' parameter is configured, the 'LDAP Server Domain Name' parameter is ignored. Thus, if you want to use an FQDN, leave the 'LDAP Server IP' parameter undefined. |
| LDAP Password<br>`password`<br>[LdapConfiguration_LdapConfPassword] | Defines the user password for accessing the LDAP server during connection and binding operations.<br>▪ LDAP-based SIP queries: The parameter is the password used by the device to authenticate itself, as a client, to obtain LDAP service from the LDAP server.<br>▪ LDAP-based user login authentication: The parameter represents the login password entered by the user during a login attempt. You can use the $ (dollar) sign in this value to enable the device to automatically replace the $ sign with the user's login password in the search filter, which it sends to |

| Parameter | Description |
|---|---|
| | the LDAP server for authenticating the user's username-password combination. For example, $. <br>**Notes:**<br>▪ The parameter is mandatory.<br>▪ By default, the device sends the password in clear-text format. You can enable the device to encrypt the password using TLS (see the 'Use SSL' parameter below). |
| LDAP Bind DN<br>`bind-dn`<br>[LdapConfiguration_LdapConfBindDn] | Defines the LDAP server's bind Distinguished Name (DN) or username.<br>▪ LDAP-based SIP queries: The DN is used as the username during connection and binding to the LDAP server. The DN is used to uniquely name an AD object. Below are example parameter settings:<br>✓ cn=administrator,cn=Users,dc=domain,dc=com<br>✓ administrator@domain.com<br>✓ domain\administrator<br>▪ LDAP-based user login authentication: The parameter represents the login username entered by the user during a login attempt. You can use the $ (dollar) sign in this value to enable the device to automatically replace the $ sign with the user's login username in the search filter, which it sends to the LDAP server for authenticating the user's username-password combination. An example configuration for the parameter is $@sales.local, where the device replaces the $ with the entered username, for example, JohnD@sales.local. The username can also be configured with the domain name of the LDAP server.<br>**Note:** By default, the device sends the username in clear-text format. You can enable the device to encrypt the username using TLS (see the 'Use SSL' parameter below). |
| LDAP Network Interface<br>`interface-type`<br>[LdapConfiguration_Interface] | Assigns one of the device's IP network interfaces through which communication with the LDAP server is done.<br>By default, no value is defined (**None**) and the device uses the OAMP network interface, configured in the Interface table.<br>For configuring IP network interfaces, see "Configuring IP Network Interfaces" on page 135.<br>**Note:** The parameter is mandatory. |
| Management Attribute<br>`mgmt-attr`<br>[LdapConfiguration_MngmAuthAtt] | Defines the LDAP attribute name to query, which contains a list of groups to which the user is a member. For Active Directory, this attribute is typically "memberOf". The attribute's values (groups) are used to determine the user's management access level; the group's corresponding access level is configured in "Configuring Access Level per Management Groups Attributes" on page 242.<br>**Notes:**<br>▪ The parameter is applicable only to LDAP-based login authentication and authorization (i.e., the 'Type' parameter is set to **Management**).<br>▪ If this functionality is not used, the device assigns the user the configured default access level. For more information, |

| Parameter | Description |
|---|---|
| | see "Configuring Access Level per Management Groups Attributes" on page 242. |
| Use TLS<br>`use-tls`<br>[LdapConfiguration_useTLS] | Enables the device to encrypt the username and password (for Control and Management related queries) using TLS when sending them to the LDAP server.<br>▪ **[0]** No = (Default) Username and password are sent in clear-text format.<br>▪ **[1]** Yes |
| TLS Context<br>[LdapConfiguration_ContextName] | Assigns a TLS Context for the connection with the LDAP server.<br>By default, no value is defined (**None**) and the device uses the default TLS Context (ID 0).<br>For configuring TLS Contexts, see "Configuring TLS Certificate Contexts" on page 103.<br>**Note:** The parameter is applicable only if the 'Use TLS' parameter is configured to **Yes**. |
| Verify Certificate<br>`verify-certificate`<br>[LdapConfiguration_VerifyCertificate] | Enables certificate verification when the connection with the LDAP server uses TLS.<br>▪ [0] No = (Default) No certificate verification is done.<br>▪ [1] Yes = The device verifies the authentication of the certificate received from the LDAP server. The device authenticates the certificate against the trusted root certificate store associated with the associated TLS Context (see 'TLS Context' parameter above) and if ok, allows communication with the LDAP server. If authentication fails, the device denies communication (i.e., handshake fails). The device can also authenticate the certificate by querying with an Online Certificate Status Protocol (OCSP) server whether the certificate has been revoked. This is also configured for the associated TLS Context.<br>**Note:** The parameter is applicable only if the 'Use TLS parameter is configured to **Yes**. |
| Connection Status<br>`connection-status`<br>[LdapConfiguration_ConnectionStatus] | (Read-only) Displays the connection status with the LDAP server.<br>▪ "Not Applicable"<br>▪ "LDAP Connection Broken"<br>▪ "Connecting"<br>▪ "Connected"<br>**Note:** For more information about a disconnected LDAP connection, see your Syslog messages generated by the device. |

### 15.3.5 Configuring LDAP DNs (Base Paths) per LDAP Server

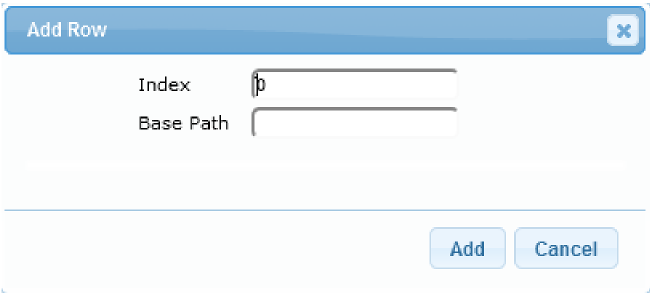
The LDAP Search DN table lets you configure LDAP base paths. The table is a "child" of the LDAP Configuration table (see "Configuring LDAP Servers" on page 236) and configuration is done per LDAP server. For the device to run a search using the LDAP service, the base path to the directory's subtree, referred to as the distinguished name object (or DN), where the search is to be done must be configured. For each LDAP server, you can configure up to three base paths.

The following procedure describes how to configure DNs per LDAP server through the Web interface. You can also configure it through ini file (LdapServersSearchDNs) or CLI (configure voip/ldap/ldap-servers-search-dns).

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

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

**Figure 15-14: LDAP Search Base DN Table - Add Row Dialog Box**




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

**Table 15-9: LDAP Server Search Base DN Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>`set internal-index`<br>[LdapServersSearchDNs_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Base Path<br>`set base-path`<br>[LdapServersSearchDNs_Base_Path] | Defines the full path (DN) to the objects in the AD where the query is done.<br>The valid value is a string of up to 256 characters.<br>For example: OU=NY,DC=OCSR2,DC=local. In this example, the DN path is defined by the LDAP names, OU (organizational unit) and DC (domain component). |

## 15.3.6 Configuring the LDAP Search Filter Attribute

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

- **Search base object (distinguished name or DN, e.g., "ou=ABC,dc=corp,dc=abc,dc=com"):** The DN defines the location in the directory from which the LDAP search begins and is configured in "Configuring LDAP DNs (Base Paths) per LDAP Server" on page 240.
- **Filter (e.g., "(&(objectClass=person)(sAMAccountName=johnd))"):** This filters the search in the subtree to include only the login username (and excludes others). This is configured by the 'LDAP Authentication Filter' parameter, as described in the following

procedure. You can use the dollar ($) sign to represent the username. For example, the filter can be configured as "(sAMAccountName=$)", where if the user attempts to log in with the username "SueM", the LDAP search is done only for the attribute sAMAccountName that equals "SueM".

- **Attribute (e.g., "memberOf") to return from objects that match the filter criteria:** The attribute is configured by the 'Management Attribute' parameter in the LDAP Configuration table (see "Configuring LDAP Servers" on page 236).

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

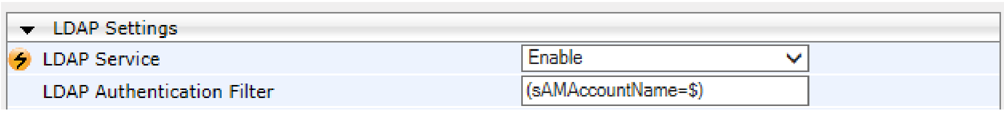
**Notes:**

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

**➢ To configure the LDAP search filter for management users:**

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

**Figure 15-15: LDAP Settings Page - LDAP Search Filter**




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

### 15.3.7 Configuring Access Level per Management Groups Attributes

The Management LDAP Groups table lets you configure LDAP group objects and their corresponding management user access level. The table is a "child" of the LDAP Configuration table (see "Configuring LDAP Servers" on page 236) and configuration is done per LDAP server. For each LDAP server, you can configure up to three table row entries of LDAP group(s) and their corresponding access level.

**Notes:**

- The Management LDAP Groups table is applicable only to LDAP-based login authentication and authorization queries.
- If the LDAP response received by the device includes multiple groups of which the user is a member and you have configured different access levels for some of these groups, the device assigns the user the highest access level. For example, if the user is a member of two groups where one has access level "Monitor" and the other "Administrator", the device assigns the user the "Administrator" access level.
- When the access level is unknown, the device assigns the default access level to the user, configured by the 'Default Access Level' parameter in the Authentication Settings page (**Configuration** tab > **System** menu > **Management** > **Authentication Settings**). This can occur in the following scenarios:
  - √ The user is not a member of any group.
  - √ The group of which the user is a member is not configured on the device (as described in this section).
  - √ The device is not configured to query the LDAP server for a management attribute (see "Configuring LDAP Servers" on page 236).

Group objects represent groups in the LDAP server of which the user is a member. The access level represents the user account's permissions and rights in the device's management interface (e.g., Web and CLI). The access level can either be Monitor, Administrator, or Security Administrator. For an explanation on the privileges of each level, see "Configuring Web User Accounts" on page 65.

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

- Search base object (distinguished name or DN, e.g., "ou=ABC,dc=corp,dc=abc,dc=com"), which defines the location in the directory from which the LDAP search begins. This is configured in "Configuring LDAP DNs (Base Paths) per LDAP Server" on page 240.
- Filter (e.g., "(&(objectClass=person)(sAMAccountName=johnd))"), which filters the search in the subtree to include only the login username (and excludes others). This is configured by the 'LDAP Authentication Filter' parameter.
- Attribute (e.g., "memberOf") to return from objects that match the filter criteria. This attribute is configured by the 'Management Attribute' parameter in the LDAP Configuration table.

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

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

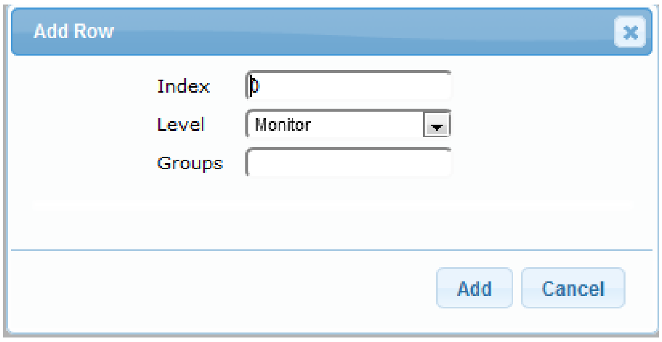
The device searches this LDAP response for the group names that you configured in the Management LDAP Groups table in order to determine the user's access level. If the device finds a group name, the user is assigned the corresponding access level and login is permitted; otherwise, login is denied. Once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

The following procedure describes how to configure an access level per management groups through the Web interface. You can also configure it through ini file (MgmntLDAPGroups) or CLI (configure voip > ldap > mgmt-ldap-groups).

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

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

**Figure 15-16: Management LDAP Groups Table - Add Row Dialog Box**




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

**Table 15-10: Management LDAP Groups Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[MgmntLDAPGroups_GroupIndex] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Level<br>`level`<br>[MgmntLDAPGroups_Level] | Defines the access level of the group(s).<br>▪ **[0]** Monitor (Default)<br>▪ **[1]** Admin<br>▪ **[2]** Security Admin |
| Groups<br>`groups`<br>[MgmntLDAPGroups_Group] | Defines the attribute names of the groups in the LDAP server.<br>The valid value is a string of up to 256 characters. To define multiple groups, separate each group name with a semicolon (;). |

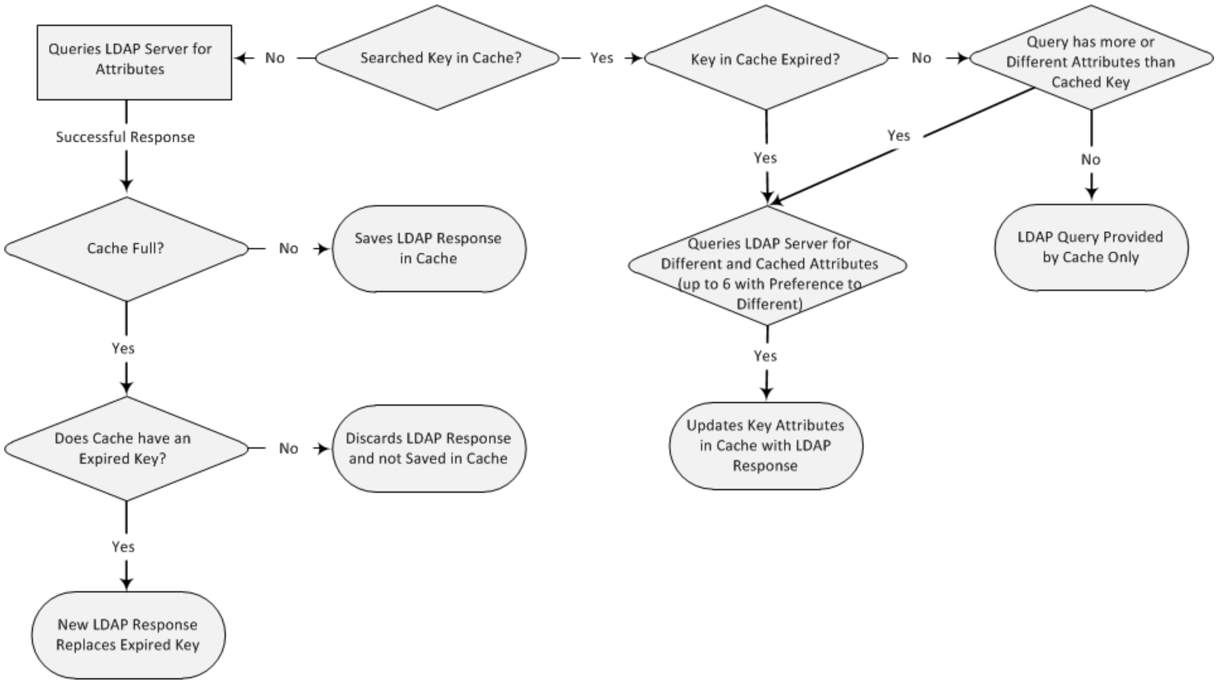
## 15.3.8 Configuring the Device's LDAP Cache

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

- Improves routing decision performance by using local cache for subsequent LDAP queries
- Reduces number of queries performed on an LDAP server and corresponding bandwidth consumption
- Provides partial survivability in case of intermittent LDAP server failure (or network isolation)

The handling of LDAP queries using the device's LDAP cache is shown in the flowchart below:

**Figure 15-17: LDAP Query Process with Local LDAP Cache**




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

**Table 15-11: Example of LDAP Query for Cached Attributes**

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

**Note:**

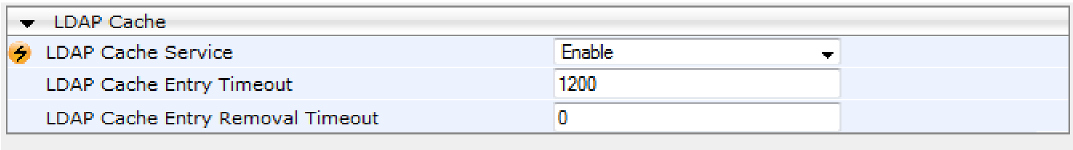
- The LDAP Cache feature is applicable only to LDAP-based SIP queries (Control).
- The maximum LDAP cache size is 10,000 entries.
- The device can save up to six LDAP Attributes in the cache per user (search LDAP key).
- The device also saves in the cache queried Attributes that do not have any values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

The following procedure describes how to configure the device's LDAP cache through the Web interface. For a full description of the cache parameters, see 'LDAP Parameters' on page 817.

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

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

**Figure 15-18: LDAP Settings Page - Cache Parameters**



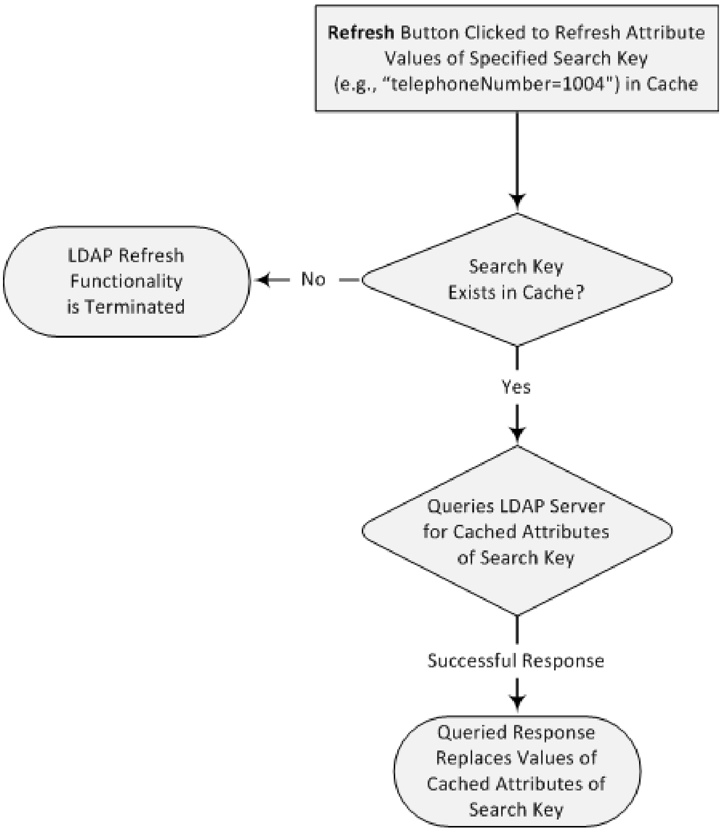

2. Under the Cache group, do the following:
   a. From the 'LDAP Cache Service' drop-down list, select **Enable** to enable LDAP cache.
   b. In the 'LDAP Cache Entry Timeout' field, enter the duration (in minutes) for which an entry in the LDAP cache is valid.
   c. In the 'LDAP Cache Entry Removal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

#### 15.3.8.1 Refreshing the LDAP Cache

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

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

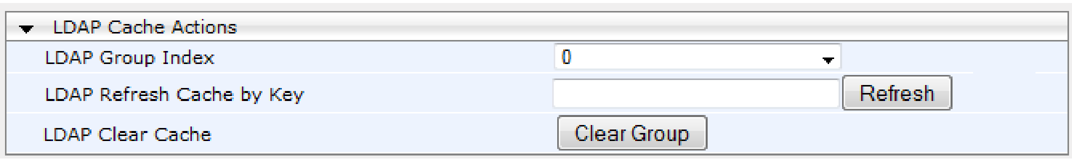
**Figure 15-19: LDAP Cache Refresh Flowchart**




### ➢ To refresh the LDAP cache per LDAP Server Group:

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

**Figure 15-20: Refreshing LDAP Cache**




2. Under the Cache Actions group, do the following:
   a. From the 'LDAP Group Index' drop-down list, select the required LDAP Server Group (see 'Configuring LDAP Server Groups' on page 234).
   b. In the 'LDAP Refresh Cache by Key' field, enter the LDAP search key that you want to refresh (e.g., telephoneNumber=1004).
   c. Click **Refresh**; if a request with the specified key exists in the cache, a request is sent to the LDAP server for the Attributes associated in the cache with the search key.

#### 15.3.8.2 Clearing the LDAP Cache

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

➢ **To clear the LDAP cache:**

1. Open the LDAP Settings page (**Configuration** tab > **VoIP** menu > **Services** > **LDAP** > **LDAP Settings**).
2. Under the Cache Actions group, do the following:
   a. From the 'LDAP Group Index' drop-down list, select the required LDAP Server Group (see 'Configuring LDAP Server Groups' on page 234).
   b. Click **Clear Group**.

### 15.3.9 Configuring Local Database for Management User Authentication

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

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

If user authentication using the Web Users table succeeds, the device grants management access to the user; otherwise access is denied. The access level assigned to the user is also determined by the Web Users table. To configure local Web/CLI users in the Web Users table, see "Configuring Web User Accounts" on page 65.

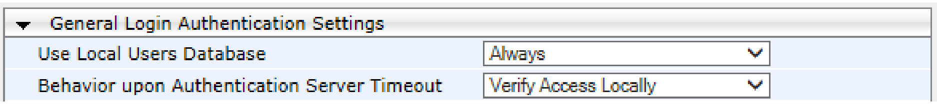
**Notes:**

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

➢ **To use the Web Users table for authenticating management users:**

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

**Figure 15-21: Authentication Settings Page - Local Database for Login Authentication**




2. Under General Login Authentication Settings:
   - Configure when the Web Users table must be used to authenticate login users. From the 'Use Local Users Database' drop-down list, select one of the following:
     - **When No Auth Server Defined (default):** When no LDAP/RADIUS server is configured or if a server is configured but connectivity with the server is down (if the server is up, the device authenticates the user with the server).
     - **Always:** First attempts to authenticate the user using the Web Users table,

but if not found, it authenticates the user with the LDAP/RADIUS server.

- Configure whether the Web Users table must be used to authenticate login users upon connection timeout with the server. From the 'Behavior upon Authentication Server Timeout' drop-down list, select one of the following:
  - **Deny Access:** User is denied access to the management platform.
  - **Verify Access Locally (default):** The device verifies the user's credentials in the Web Users table.

3. Click **Submit**.

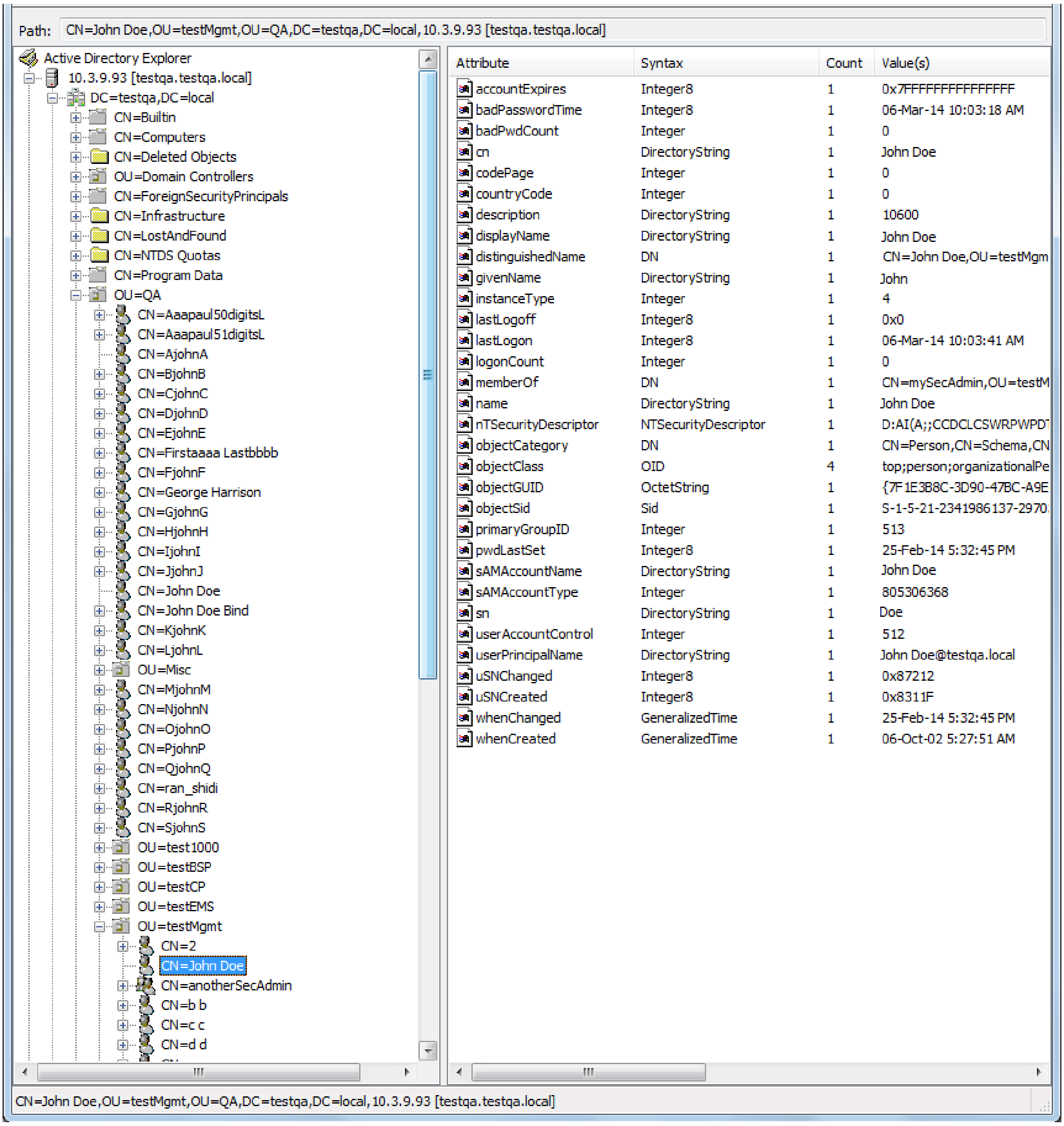
## 15.3.10 LDAP-based Login Authentication Example

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

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

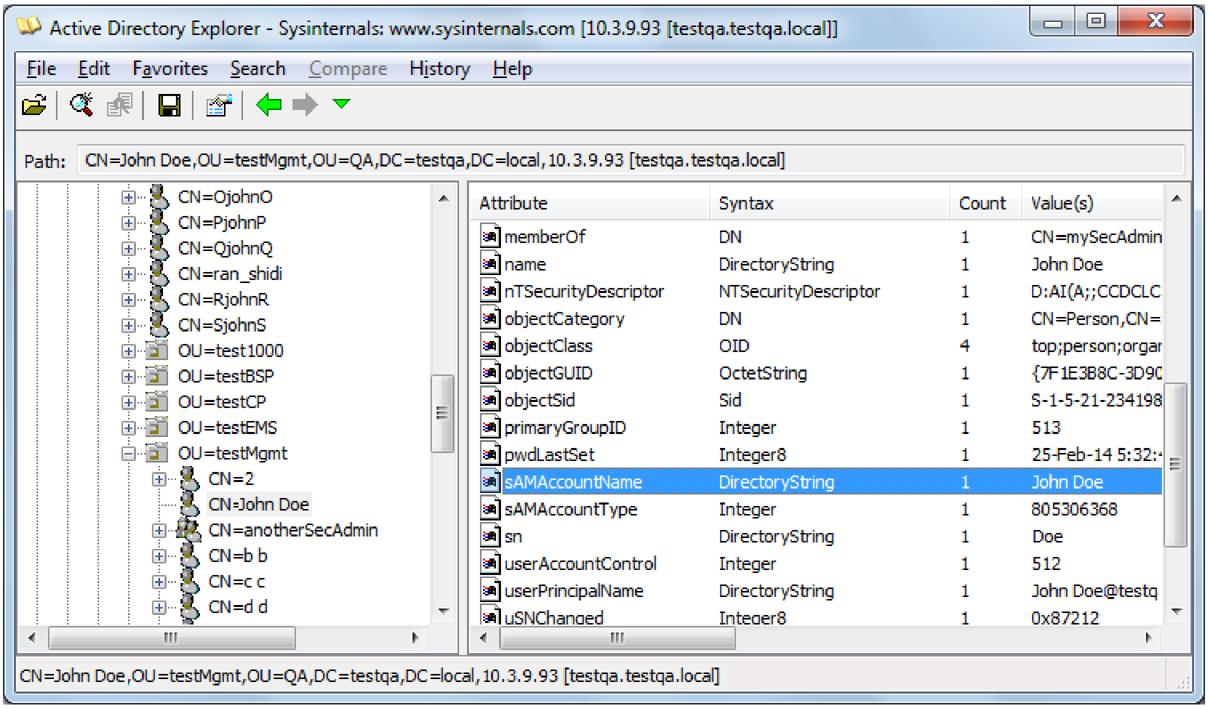
- **DN (base path):** OU=testMgmt,OU=QA,DC=testqa,DC=local. The DN path to search for the username in the directory is shown below:

**Figure 15-22: Base Path (DN) in LDAP Server**



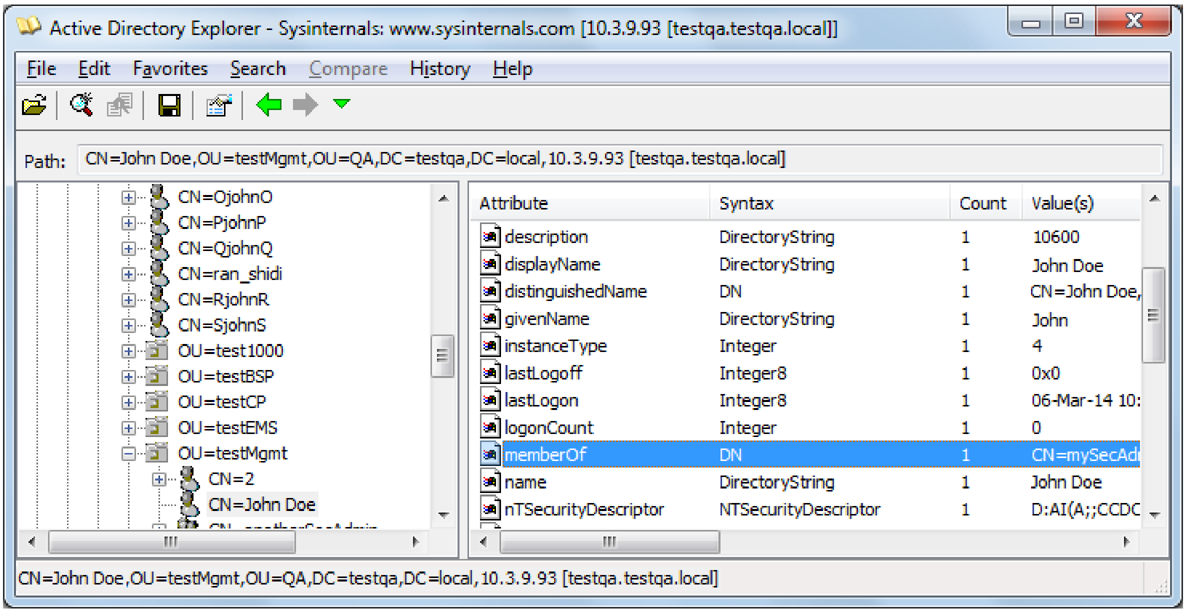
- **Search Attribute Filter:** (sAMAccountName=$). The login username is found based on this attribute (where the attribute's value equals the username):

**Figure 15-23: Username Found using sAMAccount Attribute Search Filter**



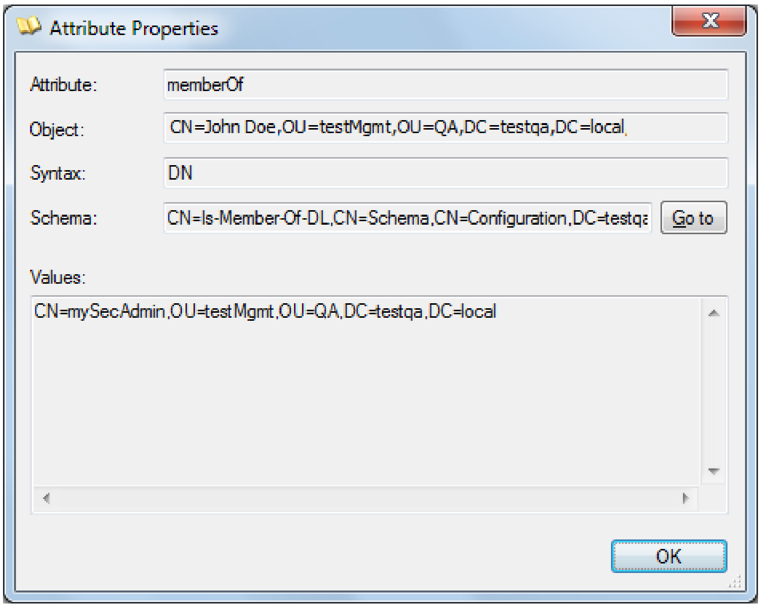

- **Management Attribute:** memberOf. The attribute contains the member groups of the user:

**Figure 15-24: User's memberOf Attribute**




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

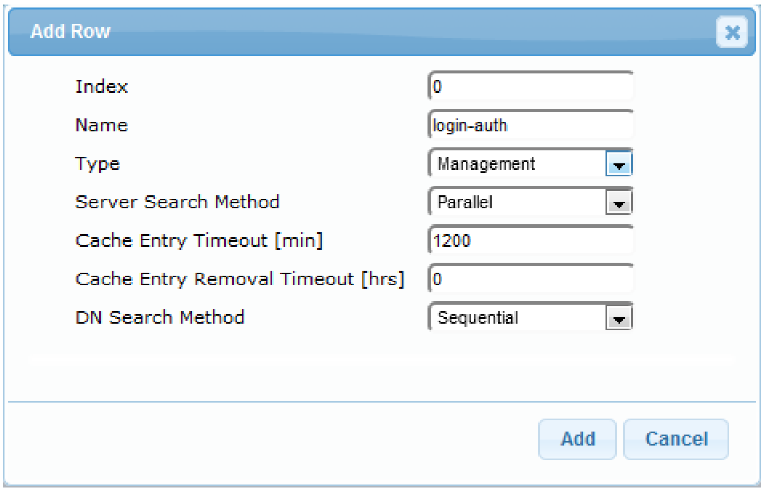
**Figure 15-25: User's mySecAdmin Group in memberOf Management Attribute**




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

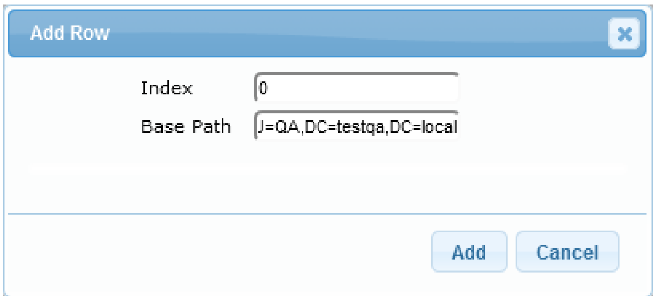
- LDAP-based login authentication (management) is enabled in the LDAP Server Groups table (see "Configuring LDAP Server Groups" on page 234):

**Figure 15-26: Configuring LDAP Server Group for Management**



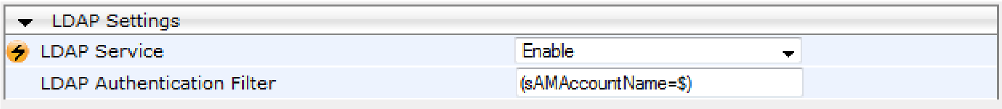

- The DN is configured in the LDAP Server Search Base DN table (see "Configuring LDAP DNs (Base Paths) per LDAP Server" on page 240):

**Figure 15-27: Configuring DN**



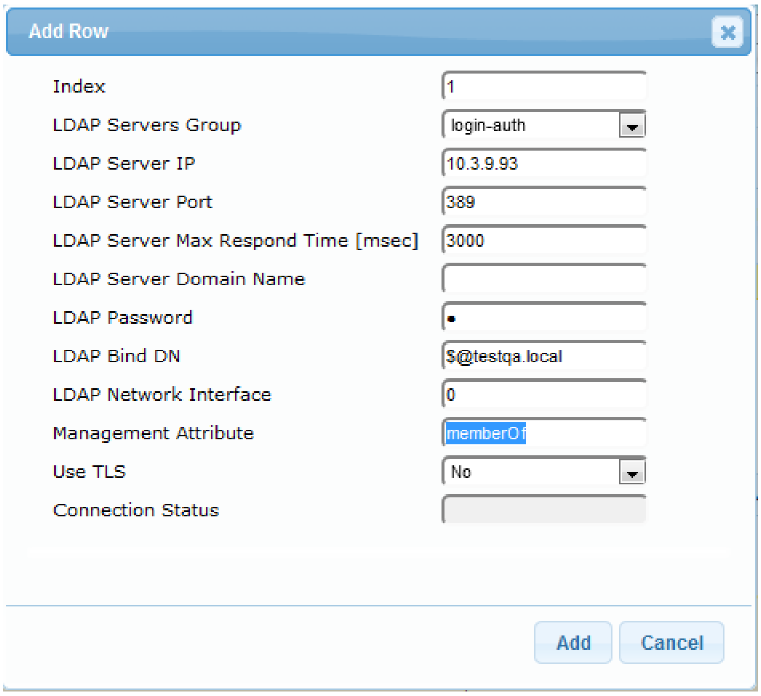
- The search attribute filter based on username is configured by the 'LDAP Authentication Filter' parameter in the LDAP Settings page (see "Configuring the LDAP Search Filter Attribute" on page 241):

**Figure 15-28: Configuring Search Attribute Filter**



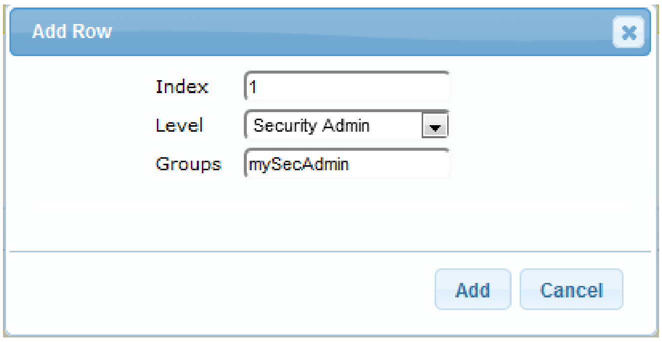

- The group management attribute is configured by the 'Management Attribute' parameter in the LDAP Configuration table:

**Figure 15-29: Configuring Management Attribute**




- The management group and its corresponding access level is configured in the Management LDAP Groups table (see "Configuring Access Level per Management Groups Attributes" on page 242):

**Figure 15-30: Configuring Management Group Attributes for Determining Access Level**




## 15.3.11 Enabling LDAP Searches for Numbers with Characters

Typically, the device performs LDAP searches in the AD for complete numbers where the digits are adjacent to one another (e.g., 5038234567). However, if the number is defined in the AD with characters (such as spaces, hyphens and periods) separating the digits (e.g., 503-823 4567), the LDAP query returns a failed result.

To enable the device to search the AD for numbers that may contain characters between its digits, you need to specify the Attribute (up to five) for which you want to apply this functionality, using the LDAPNumericAttributes parameter. For example, the telephoneNumber Attribute could be defined in AD with the telephone number "503-823-4567" (i.e., hyphens), "503.823.4567" (i.e., periods) or "503 823 4567" (i.e., spaces). If the device performs an LDAP search on this Attribute for the number 5038234567, the LDAP query will return results only if you configure the LDAPNumericAttributes parameter with the telephoneNumber Attribute (e.g., LDAPNumericAttributes=telephoneNumber). To search for the number with characters, the device inserts the asterisk (*) wildcard between all digits in the LDAP query (e.g., telephoneNumber = 5*0*3*8*2*3*4*5*6*7). As the AD server recognizes the * wildcard as representing any character, it returns all possible results to the device. Note that the wildcard represents only a character; a query result containing a digit in place of a wildcard is discarded and the device performs another query for the same Attribute. For example, it may return the numbers 533-823-4567 (second digit "3" and hyphens) and 503-823-4567. As the device discards query results where the wildcard results in a digit, it selects 503-823-4567 as the result. The correct query result is cached by the device for subsequent queries and/or in case of LDAP server failure.

## 15.3.12 Active Directory-based Routing for Microsoft Lync

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

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

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

### 15.3.12.1 Querying the AD and Routing Priority

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

**Table 15-12: Parameters for Configuring Query Attribute Key**

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

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

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

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

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

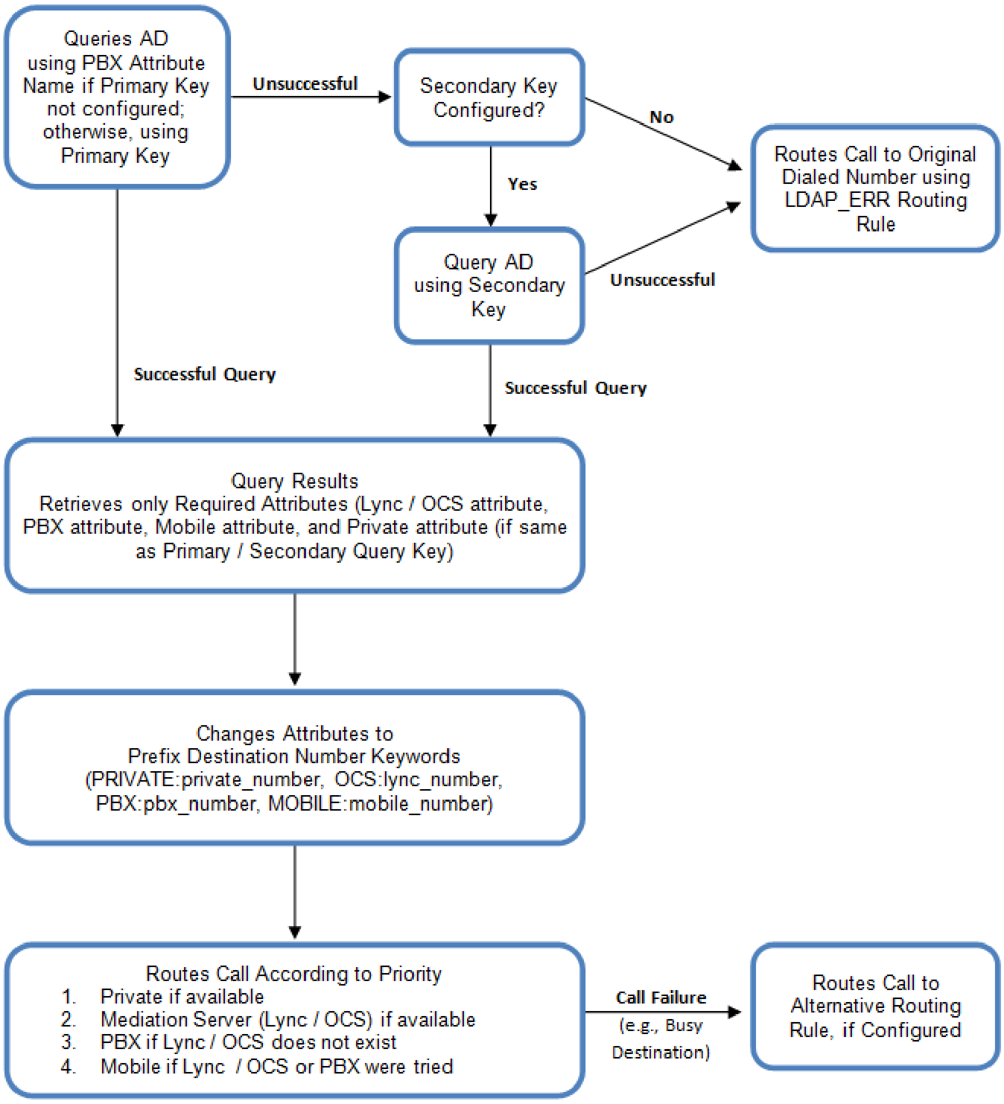
7. The device uses the Tel-to-IP Routing table to route the call based on the LDAP query result. The device routes the call according to the following priority:
   1. **Private line:** If the query is done for the private attribute and it's found, the device routes the call according to this attribute.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

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

> **Note:** For Enterprises implementing a PBX / IP PBX system, but yet to migrate to Lync Server, if the PBX / IP PBX system is unavailable or has failed, the device uses the AD query result for the user’s mobile phone number, routing the call through the PSTN to the mobile destination.

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

**Figure 15-31: LDAP Query Flowchart**



**Note:** If you are using the device's local LDAP cache, see "Configuring the Device's LDAP Cache" on page 245 for the LDAP query process.

### 15.3.12.2 Configuring AD-Based Routing Rules

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

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

1. Configure the LDAP server parameters, as described in "Configuring LDAP Servers" on page 236.
2. Configure the AD attribute names used in the LDAP query:
   a. Open the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**).

**Figure 15-32: LDAP Parameters for Microsoft Lync Server 2010**

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

   b. Configure the LDAP attribute names as desired.
3. Configure AD-based Tel-to-IP routing rules:
   a. Open the Tel-to-IP Routing table (Configuration tab > VoIP menu > Gateway > Routing > Tel to IP Routing). For more information, see Configuring Tel-to-IP Routing Rules on page 409.
   b. Configure query-result routing rules for each IP domain (private, PBX / IP PBX, Lync clients, and mobile), using the LDAP keywords (case-sensitive) for the prefix destination number:
      - PRIVATE: Private number
      - OCS: Lync client number
      - PBX: PBX / IP PBX number
      - MOBILE: Mobile number
      - LDAP_ERR: LDAP query failure
   c. Configure a routing rule for routing the initial Tel call to the LDAP server, using the value "LDAP" for denoting the IP address of the LDAP server.
   d. For alternative routing, enable the alternative routing mechanism and configure corresponding SIP reasons for alternative routing. For this feature, alternative routing starts from the table row located under the LDAP query row.

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

**Table 15-13: AD-Based Tel-to-IP Routing Rule Configuration Examples**

| Index | Destination Phone Prefix | Destination IP Address |
|---|---|---|
| 1 | PRIVATE: | 10.33.45.60 |
| 2 | PBX: | 10.33.45.65 |
| 3 | OCS: | 10.33.45.68 |
| 4 | MOBILE: | 10.33.45.100 |
| 5 | LDAP_ERR | 10.33.45.80 |
| 6 | * | LDAP |
| 7 | * | 10.33.45.72 |

The configured routing rule example is explained below:

- **Rule 1:** Sends call to private telephone line (at 10.33.45.60) upon successful AD query result for the private attribute.
- **Rule 2:** Sends call to IP PBX (at 10.33.45.65) upon successful AD query result for the PBX attribute.
- **Rule 3:** Sends call to Lync client (i.e., Mediation Server at 10.33.45.68) upon successful AD query result for the Lync attribute.
- **Rule 4:** Sends call to user's mobile phone number (to PSTN through the device's IP address at 10.33.45.100) upon successful AD query result for the Mobile attribute.
- **Rule 5:** Sends call to IP address of device (10.33.45.80) if AD query failure (e.g., no response from LDAP server or attribute not found).
- **Rule 6:** Sends query for original destination number of received call to the LDAP server.
- **Rule 7:** Alternative routing rule that sends the call of original dialed number to IP destination 10.33.45.72. This rule is applied in any of the following cases
  - LDAP functionality is disabled.
  - LDAP query is successful but call fails (due to, for example, busy line) to all the relevant attribute destinations (private, Lync, PBX, and mobile), and a relevant Tel-to-IP Release Reason (see Alternative Routing for Tel-to-IP Calls on page 429) has been configured.

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

### 15.3.12.3 Querying the AD for Calling Name

The device can retrieve the calling name (display name) from an LDAP-compliant server (for example, Microsoft Active Directory / AD) for Tel-to-IP calls that are received without a calling name.

The device uses the calling number (PBX or mobile number) for the LDAP query. Upon an incoming INVITE, the device queries the AD based on the Calling Number search key (tries to match the calling number with the appropriate "telephoneNumber" or "mobile" number AD attribute entry). It then searches for the corresponding calling name attribute, configured by the MSLDAPDisplayNameAttributeName parameter (e.g., “displayName”). The device uses the resultant calling name as the display name parameter in the SIP From header of the outgoing INVITE message.

To configure this feature, the following keywords are used in the Calling Name Manipulation Table for Tel-to-IP Calls table for the 'Prefix/Suffix to Add' fields, which can be combined with other characters:

- "$LDAP-PBX": LDAP query using the MSLDAPPBXAttrName parameter as the search key
- "$LDAP-MOBILE": LDAP query using MSLDAPMobileAttrName parameter as the search key

If the source (calling) number of the Tel-to-IP call matches the PBX / MOBILE (e.g., "telephoneNumber" and "mobile") number in the AD server, the device uses the resultant Display Name instead of the keyword(s).

For example, assume the following configuration in the Calling Name Manipulation Table for Tel-to-IP Calls:

- 'Source Prefix' field is set to "4".
- 'Prefix to Add' field is set to "$LDAP-PBX Office".

If the calling number is 4046 and the resultant LDAP query display name is "John Doe", the device sends the INVITE message with the following From header:

```
From: John Doe <sip:4064@company.com>
```

**Notes:**

- The Calling Name Manipulation Table for Tel-to-IP Calls table uses the numbers before manipulation, as inputs.
- The LDAP query uses the calling number after source number manipulation, as the search key value.

# 15.4 Least Cost Routing

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

## 15.4.1 Overview

The LCR feature enables the device to choose the outbound IP destination routing rule based on lowest call cost. This is useful in that it enables service providers to optimize routing costs for customers. For example, you may wish to define different call costs for local and international calls or different call costs for weekends and weekdays (specifying even the time of call). The device sends the calculated cost of the call to a Syslog server (as Information messages), thereby enabling billing by third-party vendors.

LCR is implemented by defining Cost Groups and assigning them to routing rules in the Tel-to-IP Routing table (Gateway calls). The device searches the routing table for matching routing rules and then selects the rule with the lowest call cost. If two routing rules have identical costs, the rule appearing higher up in the table is used (i.e., first-matched rule). If the selected route is unavailable, the device selects the next least-cost routing rule.

Even if a matched routing rule is not assigned a Cost Group, the device can select it as the preferred route over other matched rules that are assigned Cost Groups. This is determined according to the settings of the 'Default Call Cost' parameter configured for the Routing Policy. For configuring the Routing Policy, see Configuring a Gateway Routing Policy Rule on page 426.

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 15-14: Call Cost Comparison between Cost Groups for different Call Durations**

| Cost Group | Connection Cost | Minute Cost | Total Call Cost per Duration | |
|---|---|---|---|---|
| | | | 1 Minute | 10 Minutes |
| **A** | 1 | 6 | 7 | 61 |
| **B** | 0 | 10 | 10 | 100 |
| **C** | 0.3 | 8 | 8.3 | 80.3 |
| **D** | 6 | 1 | 7 | **16** |

If four matching routing rules are located in the routing table and each one is assigned a different Cost Group as listed in the table above, then the rule assigned Cost Group "D" is selected. Note that for one minute, Cost Groups "A" and "D" are identical, but due to the average call duration, Cost Group "D" is cheaper. Therefore, average call duration is an important factor in determining the cheapest routing role.

Below are a few examples of how you can implement LCR:

- **Example 1:** This example uses two different Cost Groups for routing local calls and international calls:

  Two Cost Groups are configured as shown below:

| Cost Group | Connection Cost | Minute Cost |
|---|---|---|
| 1. "Local Calls" | 2 | 1 |
| 2. "International Calls" | 6 | 3 |

  The Cost Groups are assigned to routing rules for local and international calls:

| Routing Index | Dest Phone Prefix | Destination IP | Cost Group ID |
|---|---|---|---|
| 1 | 2000 | x.x.x.x | 1 "Local Calls" |
| 2 | 00 | x.x.x.x | 2 "International Calls" |

- **Example 2:** This example shows how the device determines the cheapest routing rule in the Tel-to-IP Routing table:

  The 'Default Call Cost' parameter in the Routing Policy rule is configured to **Lowest Cost**, meaning that if the device locates other matching routing rules (with Cost Groups assigned), the routing rule without a Cost Group is considered the lowest cost route.

  - The following Cost Groups are configured:

| Cost Group | Connection Cost | Minute Cost |
|---|---|---|
| 1. "A" | 2 | 1 |
| 2. "B" | 6 | 3 |

  - The Cost Groups are assigned to routing rules:

| Routing Index | Dest Phone Prefix | Destination IP | Cost Group |
|---|---|---|---|
| 1 | 201 | x.x.x.x | "A' |
| 2 | 201 | x.x.x.x | "B" |
| 3 | 201 | x.x.x.x | 0 |
| 4 | 201 | x.x.x.x | "B" |

  The device calculates the optimal route in the following index order: 3, 1, 2, and then 4, due to the following logic:

  - Index 1 - Cost Group "A" has the lowest connection cost and minute cost
  - Index 2 - Cost Group "B" takes precedence over Index 4 entry based on the first-matched method rule
  - Index 3 - no Cost Group is assigned, but as the 'Default Call Cost' parameter is configured to **Lowest Cost**, it is selected as the cheapest route
  - Index 4 - Cost Group "B" is only second-matched rule (Index 1 is the first)

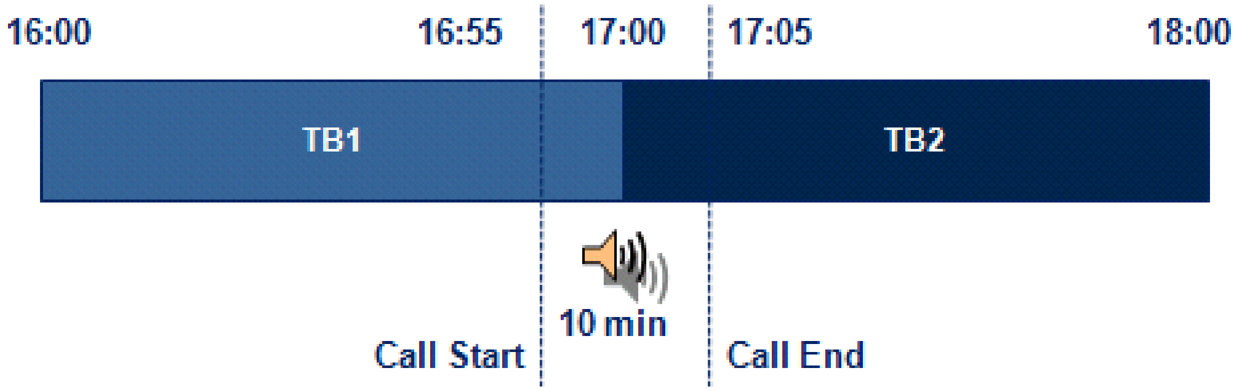
- **Example 3:** This example shows how the cost of a call is calculated if the call spans over multiple time bands:

  Assume a Cost Group, "CG Local" is configured with two time bands, as shown below:

| Cost Group | Time Band | Start Time | End Time | Connection Cost | Minute Cost |
|---|---|---|---|---|---|
| CG Local | TB1 | 16:00 | 17:00 | 2 | 1 |
| | TB2 | 17:00 | 18:00 | 7 | 2 |

  Assume that the call duration is 10 minutes, occurring between 16:55 and 17:05. In other words, the first 5 minutes occurs in time band "TB1" and the next 5 minutes occurs in "TB2", as shown below:

**Figure 15-33: LCR using Multiple Time Bands (Example)**




  The device calculates the call using the time band in which the call was initially established, regardless of whether the call spans over additional time bands:

  **Total call cost** = "TB1" Connection Cost + ("TB1" Minute Cost x call duration) = 2 + 1 x 10 min = 12

## 15.4.2 Configuring LCR

To configure LCR, perform the following main steps:

1. Enable LCR - see Configuring a Gateway Routing Policy Rule on page 426.
2. Configure Cost Groups - see "Configuring Cost Groups" on page 263.
3. Configure Time Bands for a Cost Group - see "Configuring Time Bands for Cost Groups" on page 264.
4. Assign Cost Groups to outbound IP routing rules - see "Assigning Cost Groups to Routing Rules" on page 266.

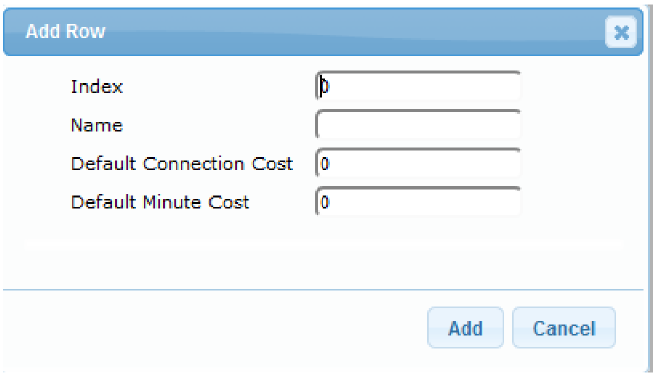
### 15.4.2.1 Configuring Cost Groups

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

The following procedure describes how to configure Cost Groups through the Web interface. You can also configure it through ini file (CostGroupTable) or CLI (configure voip > services least-cost-routing cost-group).

➢ **To configure a Cost Group:**

1. Open the Cost Group table (**Configuration** tab > **VoIP** menu > **Services** > **Least Cost Routing** > **Cost Group Table**).
2. Click **Add**; the following dialog box appears:




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

**Table 15-15: Cost Group Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[CostGroupTable_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>`cost-group-name`<br>[CostGroupTable_CostGroupName] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters.<br>**Note:** Each Cost Group must have a unique name. |
| Default Connection Cost<br>`default-connection-cost`<br>[CostGroupTable_DefaultConnectionCost] | Defines the call connection cost (added as a fixed charge to the call) for a call outside the time bands.<br>The valid value range is 0-65533. The default is 0.<br>**Note:** When calculating the cost of a call, if the current time of the call is not within a time band configured for the Cost Group, then this default connection cost is used. |
| Default Minute Cost<br>`default-minute-cost`<br>[CostGroupTable_DefaultMinuteCost] | Defines the call charge per minute for a call outside the time bands.<br>The valid value range is 0-65533. The default is 0.<br>**Note:** When calculating the cost of a call, if the current time of the call is not within a time band configured for the Cost Group, then this default charge per minute is used. |

### 15.4.2.2 Configuring Time Bands for Cost Groups

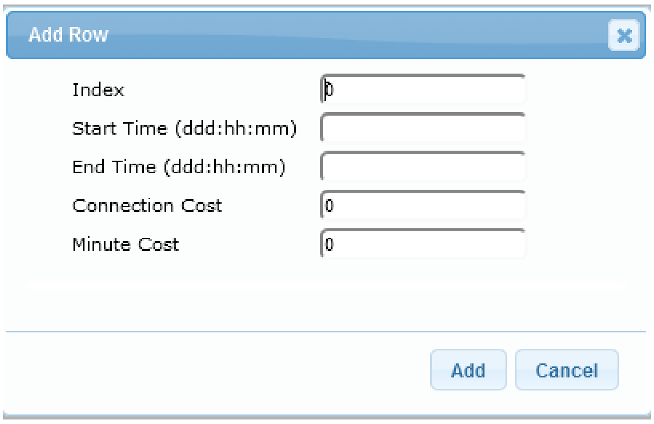
The Time Band table lets you configure Time Bands per Cost Group. A Time Band defines a day and time range (e.g., from Saturday 05:00 to Sunday 24:00), as well as the fixed call connection charge and call rate per minute for this interval. You can configure up to 70 Time Bands, where up to 21 Time Bands can be assigned to each Cost Group.

**Note:** You cannot configure overlapping Time Bands.

The following procedure describes how to configure Time Bands per Cost Group through the Web interface. You can also configure it through ini file (CostGroupTimebands) or CLI (configure voip >services least-cost-routing cost-group-time-bands).

➢ **To configure a Time Band per Cost Group:**

1. Open the Cost Group table (**Configuration** tab > **VoIP** menu > **Services** > **Least Cost Routing** > **Cost Group Table**).
2. Select a Cost Group for which you want to assign Time Bands, and then click the **Time Band** link located below the table; the Time Band table for the selected Cost Group appears.
3. Click **Add**; the following dialog box appears:




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

**Table 15-16: Time Band Table Description**

| Parameter | Description |
|---|---|
| Index<br>`timeband-index`<br>[CostGroupTimebands_TimebandIndex] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Start Time<br>`start-time`<br>[CostGroupTimebands_StartTime] | Defines the day and time of day from when this time band is applicable. The format is DDD:hh:mm, where:<br>▪ *DDD* is the day of the week, represented by the first three letters of the day in upper case (i.e., SUN, MON, TUE, WED, THU, FRI, or SAT).<br>▪ *hh* and *mm* denote the time of day, where *hh* is the hour (00-23) and *mm* the minutes (00-59)<br>For example, SAT:22:00 denotes Saturday at 10 pm. |
| End Time<br>`end-time`<br>[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. |

| Parameter | Description |
|---|---|
| Connection Cost<br>`connection-cost`<br>[CostGroupTimebands_ConnectionCost] | Defines the call connection cost during this time band. This is added as a fixed charge to the call.<br>The valid value range is 0-65533. The default is 0.<br>**Note:** The entered value must be a whole number (i.e., not a decimal). |
| Minute Cost<br>`minute-cost`<br>[CostGroupTimebands_MinuteCost] | Defines the call cost per minute charge during this timeband.<br>The valid value range is 0-65533. The default is 0.<br>**Note:** The entered value must be a whole number (i.e., not a decimal). |

### 15.4.2.3 Assigning Cost Groups to Routing Rules

To use your configured Cost Groups, you need to assign them to routing rules:

- Tel-to-IP Routing table - see Configuring Tel-to-IP Routing Rules on page 409

# 15.5 HTTP-based Remote Services

## 15.5.1 Configuring HTTP Services

The HTTP Remote Services table lets you configure up to seven HTTP-based services provided by third-party remote hosts (e.g., routing server). The following types of services can be offered by the remote host:

- **Routing:** Call routing service, whereby the remote host (e.g., routing server) determines the next hop of an incoming call on the path to the final destination. For more information on employing a third-party, remote routing server, see "Centralized Third-Party Routing Server" on page 272.
- **Call Status:** Call status of calls processed by the device. The call status is provided to the remote host through CDRs sent by the device.
- **Topology Status:** Status of device configuration (add, edit and delete). The device sends topology status to the HTTP host, using the REST TopologyStatus API command. To enable the functionality, configure the 'Topology Status' (RoutingServerGroupStatus) parameter to **Enable**. The parameter is located below the table.

  Topology status includes the following:

  - IP Groups: status is reported when the keep-alive mechanism (enabled for the associated Proxy Set) detects that the IP Group is unavailable, or when CAC thresholds (configured in the Admission Control table) are crossed.
  - Hunt Groups: status is reported when the Hunt Group's physical state indicates that the Hunt Group is unavailable.
  - Status is reported when IP Groups, Hunt Groups or SIP Interfaces that are configured to be used by HTTP-based services (i.e., the UsedByRoutingServer parameter is set to 1 - Used) are created or deleted. If you subsequently change the settings of the UsedByRoutingServer parameter or the 'Name' parameter, the device reports the change as a creation or deletion of the corresponding configuration entity.
- **Capture:** Recording of signaling and RTP packets, and Syslog. The remote host can be, for example, a Syslog server or AudioCodes SEM.

**Notes:**

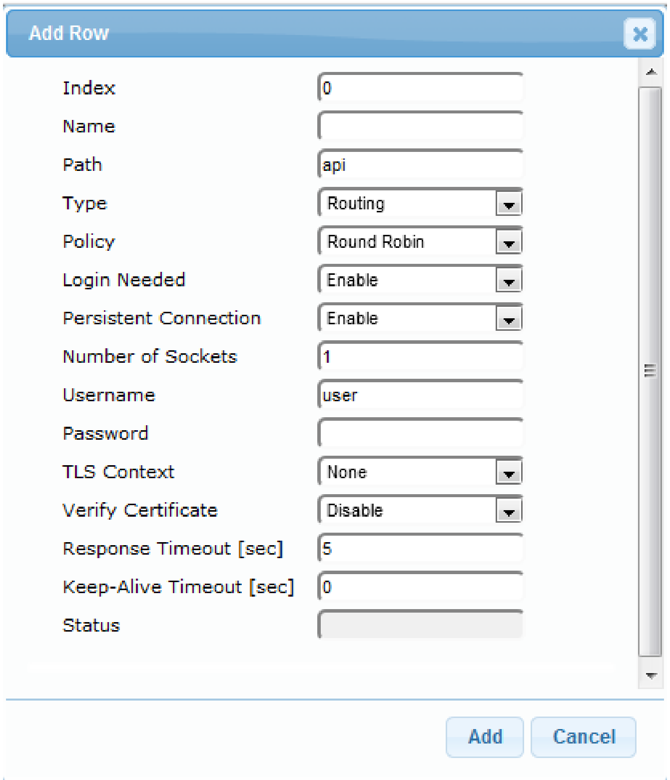
- You can configure only **one** HTTP Remote Service entry for Routing, for Call Status, and for Topology. However, you can configure up to four HTTP Remote Services for Capture.
- The Routing service also includes the Call Status and Topology Status services.
- Currently, the Capture service is not supported.
- The device supports HTTP redirect responses (3xx) only during connection establishment with the host. Upon receipt of a redirect response, the device attempts to open a new socket with the host and if this is successful, closes the current connection.

The following procedure describes how to configure HTTP Remote Services through the Web interface. You can also configure it through ini file (HTTPRemoteServices).

### ➢ To configure an HTTP-based service

1. Open the HTTP Remote Services table (**Configuration** tab > **VoIP** menu > **Services** > **HTTP Services** > **HTTP Remote Services**).
2. Click **Add**; the following dialog box appears:

**Figure 15-34: HTTP Remote Services Table - Add Row Dialog Box**




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

**Table 15-17: HTTP Remote Services Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[HTTPRemoteServices_Index] | Defines an index number for the new table row.<br>**Notes:**<br>▪ Each row must be configured with a unique index.<br>▪ The parameter is mandatory. |
| Name<br>[HTTPRemoteServices_Name] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters.<br>**Notes:**<br>▪ Each row must be configured with a unique name.<br>▪ The parameter is mandatory. |
| Path<br>[HTTPRemoteServices_Path] | Defines the path (prefix) to the REST APIs.<br>The valid value is a string of up to 80 characters. The default is "api". |

| Parameter | Description |
|---|---|
| Type<br>[HTTPRemoteServices_HTTPType] | Defines the type of service provided by the HTTP remote host:<br>▪ [0] Routing (default) = Routing service (also includes Call Status and Topology Status).<br>▪ [1] Call Status = Call status service.<br>▪ [2] Topology Status = Topology status service (e.g., change in configuration).<br>▪ [3] Capture = Recording of signaling and RTP packets, which can be sent to a remote host, for example, to a Syslog server or AudioCodes SEM.<br>**Notes:**<br>▪ You can only configure one HTTP service for each of the following service types: Routing, Call Status and Topology Status.<br>▪ For the **Topology Status** option to be functional, you must enable the RoutingServerGroupStatus parameter.<br>▪ Currently, the **Capture** option is not supported. |
| Policy<br>[HTTPRemoteServices_Policy] | Defines the mode of operation when you have configured multiple remote hosts (in the HTTP Remote Hosts table) for a specific HTTP service.<br>▪ [0] Round Robin = (Default) Load balancing of traffic across all configured hosts. Every consecutive message is sent to the next available host.<br>▪ [1] Sticky Primary = Device always attempts to send traffic to the first (primary) host. If the host does not respond, the device sends the traffic to the next available host. If the primary host becomes available again, the device sends the traffic to the primary host.<br>▪ [2] Sticky Next = Similar to **Sticky Primary**, but if the primary host does not respond, the device sends the traffic to the next available host and continues sending traffic to this host even if the primary host becomes available again. |
| Login Needed<br>[HTTPRemoteServices_LoginNeeded] | Enables the use of proprietary REST API Login and Logout commands for connecting to the remote host. The commands verify specific information (e.g., software version) before allowing connectivity with the device.<br>▪ [0] Disable = Commands are not used.<br>▪ [1] Enable (default) |
| Persistent Connection<br>[HTTPRemoteServices_PersistentConnection] | Defines whether the HTTP connection with the host remains open or is only opened per request.<br>▪ [0] Disable = Connection is not persistent and closes when the device detects inactivity. The device uses HTTP keep-alive messages to detect inactivity.<br>▪ [1] Enable = (Default) Connection remains open (persistent) even during inactivity. The device uses HTTP keep-alive / HTTP persistent connection messages to keep the connection open. |
| Number of Sockets<br>[HTTPRemoteServices_NumOfSockets] | Defines how many sockets (connection) are established per remote host.<br>The valid value is 1 to 10. The default is 1. |

| Parameter | Description |
|---|---|
| Username<br>[HTTPRemoteServices_AuthUserName] | Defines the username for HTTP authentication.<br>The valid value is a string of up to 80 characters. The default is "user". |
| Password<br>[HTTPRemoteServices_AuthPassword] | Defines the password for HTTP authentication.<br>The valid value is a string of up to 80 characters. The default is "password". |
| TLS Context<br>[HTTPRemoteServices_TLSContext] | Assigns a TLS Context for the connection with the HTTP service.<br>By default, no value is defined (**None**).<br>For configuring TLS Contexts, see "Configuring TLS Certificate Contexts" on page 103.<br>**Note:** The parameter is applicable only if the connection is HTTPS. |
| Verify Certificate<br>[HTTPRemoteServices_VerifyCertificate] | Enables certificate verification when the connection with the host is based on HTTPS.<br>▪ [0] Disable (default) = No certificate verification is done.<br>▪ [1] Enable = The device verifies the authentication of the certificate received from the HTTPS peer. The device authenticates the certificate against the trusted root certificate store associated with the associated TLS Context (see 'TLS Context' parameter above) and if ok, allows communication with the HTTPS peer. If authentication fails, the device denies communication (i.e., handshake fails). The device can also authenticate the certificate by querying with an Online Certificate Status Protocol (OCSP) server whether the certificate has been revoked. This is also configured for the associated TLS Context.<br>▪<br>**Note:** The parameter is applicable only if the connection is HTTPS. |
| Response Timeout<br>[HTTPRemoteServices_TimeOut] | Defines the TCP response timeout (in seconds) from the remote host. If one of the remote hosts does not respond to a request within the specified timeout, the device closes the corresponding socket and attempts to connect to the next remote host.<br>The valid value is 1 to 65535. The default is 5. |
| Keep-Alive Timeout<br>[HTTPRemoteServices_KeepAliveTimeOut] | Defines the duration/timeout (in seconds) in which HTTP-REST keep-alive messages are sent by the device if no other messages are sent. Keep-alive messages may be required for HTTP services that expire upon inactive sessions.<br>The valid value is 0 to 65535. The default is 0 (i.e., no keep-alive messages are sent).<br>**Note:** The parameter is applicable only if the 'Persistent Connection' parameter (in the table) is configured to **Enable**. |
| Topology Status<br>[HTTPRemoteServices_ServiceStatus] | Indicates the status of the host.<br>▪ "Connected": at least one of the hosts is connected.<br>▪ "Disconnected": all hosts are disconnected.<br>▪ "Not In Service": Configuration of the service is invalid. |

## 15.5.2 Configuring Remote HTTP Hosts

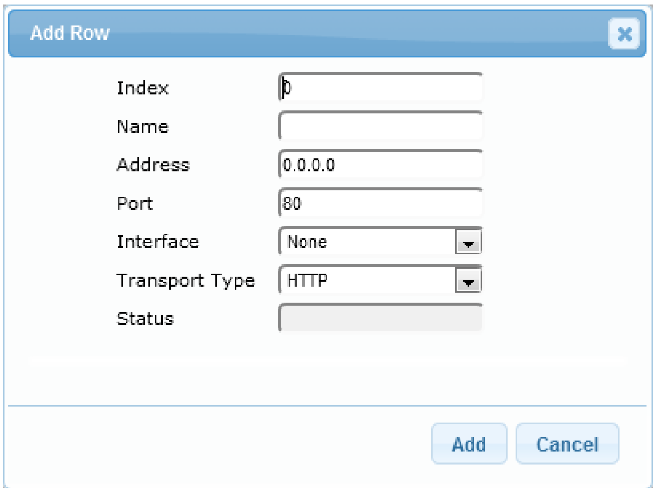
The HTTP Remote Hosts table lets you configure up to 10 remote HTTP hosts per HTTP Remote Service. The HTTP Remote Hosts table is a "child" of the HTTP Remote Services table (configured in "Configuring HTTP Services" on page 267).

The following procedure describes how to configure HTTP Remote hosts through the Web interface. You can also configure it through ini file (HTTPRemoteServices).

➢ **To configure an HTTP-based service**

1. Open the HTTP Remote Services table (**Configuration** tab > **VoIP** menu > **Services** > **HTTP Services** > **HTTP Remote Services**).
2. In the table, select the required HTTP Remote Service index row, and then click the **HTTP Remote Hosts** button, located below the table; the HTTP Remote Hosts page appears.
3. Click **Add**; the following dialog box appears:

**Figure 15-35: HTTP Remote Hosts Table - Add Row Dialog Box**




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

**Table 15-18: HTTP Remote Hosts Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[HTTPRemoteHosts_RemoteHostindex] | Defines an index number for the new table row.<br>**Notes:**<br>▪ Each row must be configured with a unique index.<br>▪ The parameter is mandatory. |
| Name<br>[HTTPRemoteHosts_Name] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters. By default, no value is defined.<br>**Notes:**<br>▪ Each row must be configured with a unique name.<br>▪ The parameter is mandatory. |
| Address | Defines the address (IP address or FQDN) of the host. |

| Parameter | Description |
|---|---|
| [HTTPRemoteHosts_Address] | The valid value is a string of up to 80 characters.<br>**Notes:**<br>▪ An IPv6 address can only be configured if the interface is a CONTROL type.<br>▪ If the address is an FQDN and the DNS resolution results in multiple IP addresses, the device device attempts to establish multiple connections (sessions) for each IP address. Only the first 10 resolved IP addresses are used regardless of the number of hosts.<br>▪ FQDN resolution is also performed (immediately) when connection is subsequently "closed" (by timeout or by the remote host) and connections are updated accordingly. In addition, the device periodically (every 15 minutes) performs DNS name resolution to ensure that the list of resolved IP addresses has not changed. If a change is detected, the device updates its' list of IP addresses and re-establishes connections accordingly.<br>▪ In addition to multiple HTTP sessions, the device establishes multiple (TCP) connections per session, thereby enhancing data exchange capabilities with the host. |
| Port<br>[HTTPRemoteHosts_Port] | Defines the port of the host.<br>The valid value is 0 to 65535. The default is 80. |
| Interface<br>[HTTPRemoteHosts_Interface] | Assigns one of the device's IP network interfaces through which communication with the remote host is done.<br>By default, no value is defined and the OAMP interface is used. |
| Transport Type<br>[HTTPRemoteHosts_HTTPTransportType] | Defines the protocol used for communicating with the host:<br>▪ [0] HTTP (default)<br>▪ [1] HTTPS |
| Status | Read-only field displaying the status of the connection.<br>▪ "Connected": The hosts is connected.<br>▪ "Disconnected": The host is disconnected.<br>▪ "Not In Service": Configuration of the host is invalid. |

### 15.5.3 Centralized Third-Party Routing Server

You can employ a remote, third-party Routing server to handle call routing decisions in deployments consisting of multiple AudioCodes devices. The Routing server can be used to handle Tel-to-IP, and IP-to-Tel calls. Employing a Routing server replaces the need for the device's routing tables (Tel-to-IP Routing table and IP-to-Tel Routing table for Tel-to-IP and IP-to-Tel calls respectively) to determine call destination.

When the device receives an incoming call (SIP INVITE, NOTIFY or MESSAGE), it disregards the routing tables and instead immediately requests the Routing server for an appropriate destination. The request is sent to the Routing server using an HTTP Get Route message. The request contains information about the call (SIP message and for IP-to-Tel calls, the source IP Group based on the associated Proxy Set).

The Routing server uses its own algorithms and logic in determining the best route path. The Routing server manages the call route between devices in "hops", which may be spread over different geographical locations. The destination to each hop (device) can be by IP address (with port), IP Group and/or Hunt Group. If the destination is an IP address, even though the destination type (in the IP-to-IP Routing table) is an IP Group, the device only uses the IP Group for profiling (i.e., associated IP Profile etc.). If multiple devices exist in the call routing path, the Routing server sends the IP address only to the last device ("node") in the path.

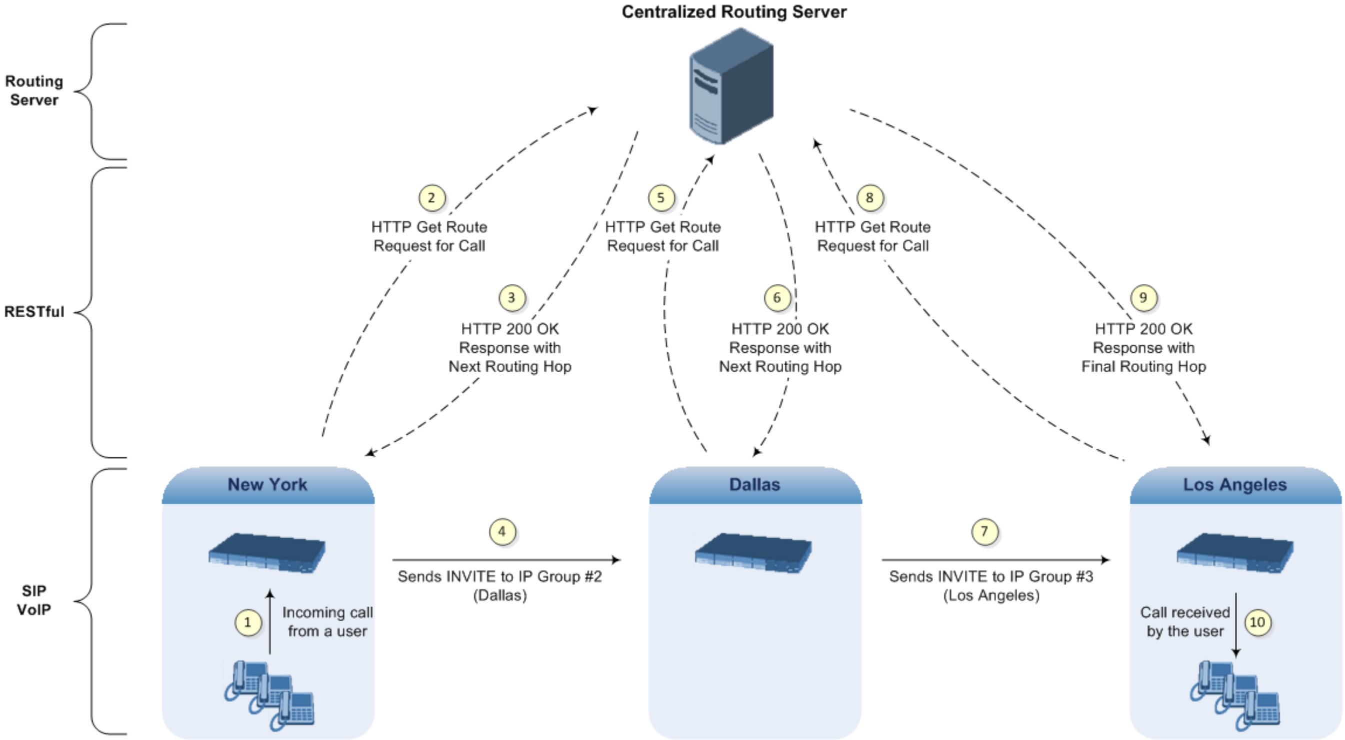
Once the device receives the resultant destination hop from the Routing server, it sends the call to that destination. The Routing server can provide the device with an appropriate route or reject the call. However, if for the initial request (first sent Get Route request for the call) the Routing server cannot find an appropriate route for the call or it does not respond, for example, due to connectivity loss (i.e., the Routing server sends an HTTP 404 "Not Found" message), the device routes the call using its routing tables. If the Get Route request is not the first one sent for the call (e.g., in call forwarding or alternative routing) and the Routing server responds with an HTTP 404 "Not Found" message, the device rejects the call.

This HTTP request-response transaction for the routing path occurs between Routing server and each device in the route path (hops) as the call traverses the devices to its final destination. Each device in the call path connects to the Routing server, which responds with the next hop in the route path. Each device considers the call as an incoming call from an IP Group or Hunt Group. The session ID (SID) is generated by the first device in the path and then passed unchanged down the route path, enabling the Routing server to uniquely identify requests belonging to the same call session.

Communication between the device and the Routing server is through the device's embedded Representational State Transfer (RESTful) API. The RESTful API is used to manage the routing-related information exchanged between the Routing server (RESTful server) and the device (RESTful client). When you have configured the device with connection settings of the Routing sever and the device starts-up, it connects to the Routing server and activates the RESTful API, which triggers the routing-related API commands.

The following figure provides an example of information exchange between devices and a Routing server for routing calls:

**Figure 15-36: Example of Call Routing Information Exchange between Devices and Routing Server**




The Routing server can also manipulate call data such as calling name, if required. It can also create new IP Groups and associated configuration entities, if necessary for routing.

Multiple Routing servers can also be employed, whereby each device in the chain path can use a specific Routing server. Alternatively, a single Routing server can be employed and used for all devices ("stateful" Routing server).

The device automatically updates (sends) the Routing server with its' configuration topology regarding SIP routing-related entities (Hunt Groups, SRDs, SIP Interfaces, and IP Groups) that have been configured for use by the Routing server. For example, if you add a new IP Group and enable it for use by the Routing server, the device sends this information to the Routing server. Routing of calls associated with routing-related entities that are disabled for use by the Routing server (default) are handled only by the device (not the Routing server).

In addition to regular routing, the routing server functionality also supports the following:

- **Alternative Routing:** If a call fails to be established, the device "closest" to the failure and configured to send "additional" routing requests (through REST API - "additionalRoute" attribute in HTTP Get Route request) to the Routing server, sends a new routing request to the Routing server. The Routing server may respond with a new route destination, thereby implementing alternative routing. Alternatively, it may enable the device to return a failure response to the previous device in the route path chain and respond with an alternative route to this device. Therefore, alternative routing can be implemented at any point in the route path. If the Routing server sends an HTTP 404 "Not Found" message for an alternative route request, the device rejects the call. If the Routing server is configured to handle alternative routing, the device does not make any alternative routing decisions based on its alternative routing tables.
- **Call Status:** The device can report call status to the Routing server to indicate whether a call has successfully been established and/or failed (disconnected). The device can also report when an IP Group (Proxy Set) is unavailable, detected by the keep-alive mechanism, or when the CAC thresholds permitted per IP Group have been crossed. For Hunt Groups, the device reports when the physical state indicates that the Hunt Group is unavailable.
- **Credentials for Authentication:** The Routing Server can provide user (e.g., IP Phone caller) credentials (username-password) in the Get Route response, which can be used by the device to authenticate outbound SIP requests if challenged by the outbound peer, for example, Microsoft Skype for Business (per RFC 2617 and RFC 3261). If multiple devices exist in the call routing path, the Routing server sends the credentials only to the last device ("node") in the path.

**➢ To configure routing based on Routing server:**

1. For each configuration entity (e.g., IP Group) that you want routing done by the Routing server, configure the entity's 'Used By Routing Server' parameter to **Used**.
2. Configure an additional Security Administrator user account in the Local Users table (see "Configuring Web User Accounts" on page 65), which is used by the Routing server (REST client) to log in to the device's management interface.
3. Configure the address and connection settings of the Routing server, referred to as a Remote Web Service and HTTP remote host. You must configure the 'Type' parameter of the Remote Web Service to **Routing**.
4. Enable routing based on Routing server, by configuring the GWRoutingServer parameter to 1.

## 15.6 HTTP-based Proxy Services

You can configure the device for the following HTTP-based proxy services:

- **HTTP Reverse Proxy for Managing Equipment behind NAT:**

  You can configure the device to function as a reverse HTTP proxy server. This functionality is required to enable administrators to manage communication equipment (such as IP Phones) over HTTP when the equipment is located behind NAT (e.g., in the LAN) and the administrator is located in a public domain (e.g., in the WAN). Thus, this functionality resolves NAT issues, enabling the administrator to access the IP Phone's management interface (e.g., embedded Web server).

  To support the functionality, the following configuration is required:

  1. Enable the HTTP Proxy application (see 'Enabling the HTTP Proxy Application' on page 276).
  2. Define a local, listening HTTP interface for the leg interfacing with the administrator (see 'Configuring HTTP Interfaces' on page 276).

> **Note:** It is recommended **not** to use port 80 as this is the default port used by IP Phones for their Web-based management interface.

  3. Define each HTTP-based managed equipment:
     a. Define the URL prefix for accessing the equipment's management interface (see 'Configuring HTTP Proxy Services' on page 278). To access the equipment's management interface, the administrator needs to enter the following URL in a Web browser:

        http://<device's WAN IP address:port>/url prefix/

     b. Define the IP address of the managed equipment (see 'Configuring HTTP Proxy Hosts' on page 279).

> **Note:** For this feature, no special configuration is required on the managed equipment.

- **HTTP-based EMS Services for AudioCodes Equipment behind NAT:**

  You can configure the device to act as an HTTP Proxy that enables AudioCodes EMS to manage AudioCodes equipment (such as IP Phones) over HTTP when the equipment is located behind NAT (e.g., in the LAN) and EMS is located in a public domain (e.g., in the WAN). Thus, the feature resolves NAT traversal issues. The IP Phones register with the device in order to allow communication between the IP Phones and the EMS.

  To support the functionality, the following configuration is required:

  1. Enable the HTTP Proxy application (see 'Enabling the HTTP Proxy Application' on page 276).
  2. Configure two local, listening HTTP interfaces - one for the EMS and one for the IP Phones (see 'Configuring HTTP Interfaces' on page 276).
  3. Configure the address of the EMS server (see 'Configuring an HTTP-based EMS Service' on page 281).

## 15.6.1 Enabling the HTTP Proxy Application

Before you can configure HTTP-based proxy services, you must enable the HTTP Proxy application, as described in the following procedure. Once enabled, the Web interface displays menus in the Navigation pane that are relevant to the HTTP Proxy application.

➢ **To enable the HTTP Proxy application:**

1. Open the Applications Enabling page (**Configuration** tab > **VoIP** menu > **Applications Enabling** > **Applications Enabling**).


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

## 15.6.2 Configuring HTTP Interfaces

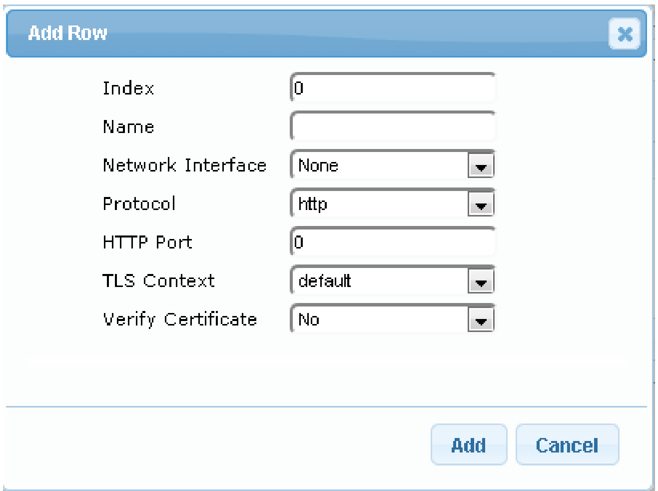
The HTTP Interfaces table lets you configure up to 10 HTTP Interfaces. An HTTP Interface represents a local, listening interface for receiving HTTP/S requests from HTTP-based (Web) clients such as managed equipment (e.g., IP Phones) and/or the EMS management tool for HTTP/S-based services.

The following procedure describes how to configure HTTP Interfaces through the Web interface. You can also configure it through ini file (HTTPInterface) or CLI (configure system > http-proxy > http-interface).

➢ **To configure an HTTP Interface:**

1. Open the HTTP Interfaces table (**Configuration** tab > **VoIP** menu > **Services** > **HTTP Proxy** > **HTTP Interfaces**).
2. Click **Add**; the following dialog box appears:

**Figure 15-37: HTTP Interfaces Table - Add Row Dialog Box**




3. Configure an HTTP Interface according to the parameters described in the table below.

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

**Table 15-19: HTTP Interfaces Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[HTTPInterface_Index] | Defines an index number for the new table row.<br>**Notes:**<br>▪ Each row must be configured with a unique index.<br>▪ The parameter is mandatory. |
| Name<br>`interface-name`<br>[HTTPInterface_InterfaceName] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters. By default, no value is defined.<br>**Notes:**<br>▪ Each row must be configured with a unique name.<br>▪ The parameter is mandatory. |
| Network Interface<br>`network-interface`<br>[HTTPInterface_NetworkInterface] | Assigns a local, network interface to the HTTP interface.<br>By default, no value is defined (**None**).<br>For configuring network interfaces, see Configuring IP Network Interfaces on page 135.<br>**Note:** The parameter is mandatory. |
| Protocol<br>`protocol`<br>[HTTPInterface_Protocol] | Defines the protocol type.<br>▪ [0] HTTP (default)<br>▪ [1] HTTPS |
| HTTP Port<br>`http-port`<br>[HTTPInterface_Port] | Defines the local, listening HTTP port.<br>The valid value is 0 to 65534. The default is 0.<br>**Note:** The parameter is mandatory. |
| TLS Context<br>`tls-context`<br>[HTTPInterface_TLSContext] | Assigns a TLS Context for the connection with the HTTP Proxy service.<br>By default, the default TLS Context (Index 0) is assigned.<br>For configuring TLS Contexts, see Configuring TLS Certificate Contexts on page 103.<br>**Note:** The parameter is applicable only if the connection protocol is HTTPS (defined using the 'Protocol' parameter, above). |

| Parameter | Description |
|---|---|
| Verify Certificate<br>`verify-cert`<br>[HTTPInterface_VerifyCert] | Enables TLS certificate verification when the connection with the proxy service is based on HTTPS.<br>▪ [0] No = (Default) No certificate verification is done.<br>▪ [1] Yes = The device verifies the authentication of the certificate received from the HTTPS peer. The device authenticates the certificate against the trusted root certificate store associated with the associated TLS Context (see 'TLS Context' parameter above) and if ok, allows communication with the HTTPS peer. If authentication fails, the device denies communication (i.e., handshake fails). The device can also authenticate the certificate by querying with an Online Certificate Status Protocol (OCSP) server whether the certificate has been revoked. This is also configured for the associated TLS Context.<br>▪<br>**Note:** The parameter is applicable only if the connection protocol is HTTPS (defined using the 'Protocol' parameter, above). |

## 15.6.3 Configuring HTTP Proxy Services

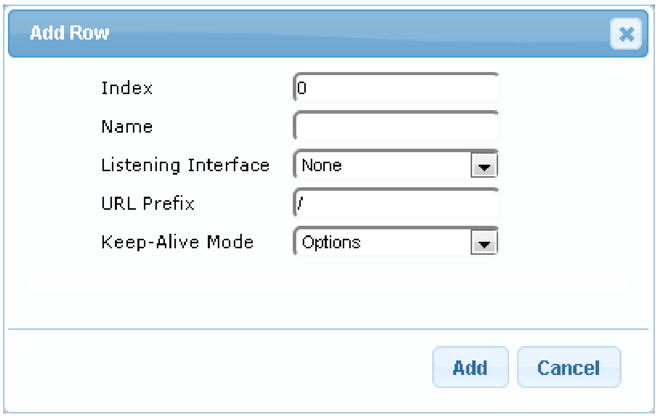
The HTTP Proxy Services table lets you configure up to 10 HTTP Proxy Services.

The following procedure describes how to configure HTTP Proxy Services through the Web interface. You can also configure it through ini file (HTTPProxyService) or CLI (configure system > http-proxy > http-proxy-serv).

➢ **To configure an HTTP Proxy Service:**

1. Open the HTTP Proxy Services table (**Configuration** tab > **VoIP** menu > **Services** > **HTTP Proxy** > **HTTP Proxy Services**).
2. Click **Add**; the following dialog box appears:

**Figure 15-38: HTTP Proxy Services Table - Add Row Dialog Box**




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

**Table 15-20: HTTP Proxy Services Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[HTTPProxyService_Index] | Defines an index number for the new table row.<br>**Notes:**<br>▪ Each row must be configured with a unique index.<br>▪ The parameter is mandatory. |
| Name<br>`service-name`<br>[HTTPProxyService_ServiceName] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters. By default, no value is defined.<br>**Notes:**<br>▪ Each row must be configured with a unique name.<br>▪ The parameter is mandatory. |
| Listening Interface<br>`listening-int`<br>[HTTPProxyService_ListeningInterface] | Assigns an HTTP Interface to the HTTP Proxy service. To configure HTTP Interfaces, see 'Configuring HTTP Interfaces' on page 276.<br>**Note:** The parameter is mandatory. |
| URL Prefix<br>`url-prefix`<br>[HTTPProxyService_URLPrefix] | Defines the URL prefix that is used to access the managed equipment's embedded Web server. The URL prefix is matched against the target of the HTTP requests sent by the client (such as GET and POST). If a match is located in the table, the device removes the prefix from the request and then forwards the HTTP request to the managed equipment without the prefix. For example, for the URL of GET /home/index.html HTTP/1.1 (which is part of the URL http://10.20.30.40/home/index.html), a URL prefix of "/home" can be configured. To match all URLs, configure the parameter to "/" (default). |
| Keep-Alive Mode<br>`keep-alive-mode`<br>[HTTPProxyService_KeepAliveMode] | Enables a keep-alive mechanism with the managed equipment:<br>▪ [0] Disable<br>▪ [1] Options = (Default) Enables keep-alive by sending HTTP OPTIONS messages. If no response is received from the HTTP host, the device stops forwarding HTTP requests to the host and raises an SNMP alarm (acHTTPProxyServiceAlarm). If you configured the address of the host as an FQDN (see 'Configuring HTTP Proxy Hosts' on page 279) and the DNS resolution results in multiple IP addresses, when no response is received from the keep-alive, the device checks connectivity with the next resolved IP address and so on, until a response is received. |

## 15.6.4 Configuring HTTP Proxy Hosts

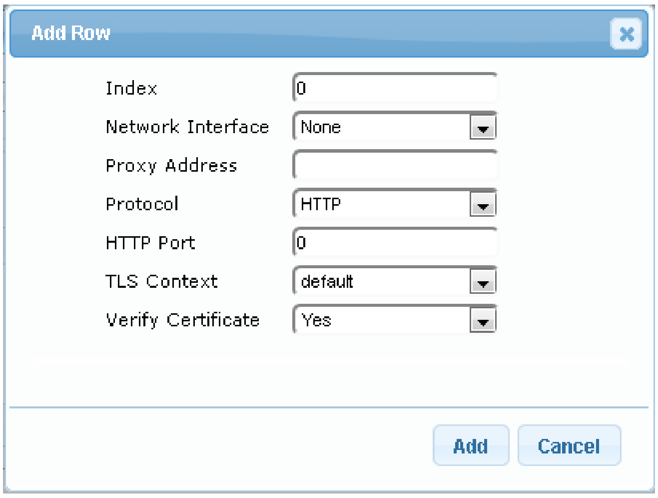
The HTTP Proxy Hosts table lets you configure up to 50 HTTP Proxy hosts (up to 5 per HTTP Proxy Service). The table is a "child" of the HTTP Proxy Services table (see 'Configuring HTTP Proxy Services' on page 278). An HTTP Proxy Host represents the HTTP-based managed equipment (e.g., IP Phone).

The following procedure describes how to configure HTTP Remote hosts through the Web interface. You can also configure it through ini file (HTTPProxyHost) or CLI (configure system > http-proxy > http-proxy-host).

### ➢ To configure an HTTP Proxy Host:

1. Open the HTTP Proxy Services table (**Configuration** tab > **VoIP** menu > **Services** > **HTTP Proxy** > **HTTP Proxy Services**).
2. In the table, select the required HTTP Proxy Service index row, and then click the **HTTP Proxy Hosts** link, located below the table; the HTTP Proxy Hosts table appears.
3. Click **Add**; the following dialog box appears:

**Figure 15-39: HTTP Proxy Hosts Table - Add Row Dialog Box**




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

**Table 15-21: HTTP Proxy Hosts Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index | Defines an index number for the new table row.<br>**Notes:**<br>▪ Each row must be configured with a unique index.<br>▪ The parameter is mandatory. |
| Network Interface<br>`network-interface`<br>[HTTPProxyHost_NetworkInterface] | Assigns a local, network interface to the HTTP Proxy Host.<br>By default, no value is defined (**None**).<br>For configuring network interfaces, see Configuring IP Network Interfaces on page 135.<br>**Note:** The parameter is mandatory. |
| Proxy Address<br>`proxy-address`<br>[HTTPProxyHost_IpAddress] | Defines the address of the managed equipment (host).<br>The valid value is an IP address in dotted-decimal notation or an FQDN (up to 100 characters). If the address is an FQDN, the device uses DNS to resolve it into an IP address. If the DNS resolution results in multiple IP addresses, the device uses the first available address (i.e., that responds to the keep-alive). |
| Protocol<br>`protocol`<br>[HTTPProxyHost_Protocol] | Defines the protocol type.<br>▪ [0] HTTP (default)<br>▪ [1] HTTPS |

| Parameter | Description |
|---|---|
| HTTP Port<br>`http-port`<br>[HTTPProxyHost_Port] | Defines the port of the managed equipment.<br>The default is 0.<br>**Note:** The parameter is mandatory. |
| TLS Context<br>`tls-context`<br>[HTTPProxyHost_TLSContext] | Assigns a TLS Context for the TLS connection with the HTTP Proxy host.<br>By default, the default TLS Context (Index 0) is assigned.<br>For configuring TLS Contexts, see Configuring TLS Certificate Contexts on page 103.<br>**Note:** The parameter is applicable only if the connection protocol is HTTPS (defined using the 'Protocol' parameter, above). |
| Verify Certificate<br>`verify-cert`<br>[HTTPProxyHost_VerifyCert] | Enables TLS certificate verification when the connection with the host is based on HTTPS.<br>▪ [0] No = No certificate verification is done.<br>▪ [1] Yes = (Default) The device verifies the authentication of the certificate received from the HTTPS peer. The device authenticates the certificate against the trusted root certificate store associated with the associated TLS Context (see 'TLS Context' parameter above) and if ok, allows communication with the HTTPS peer. If authentication fails, the device denies communication (i.e., handshake fails). The device can also authenticate the certificate by querying with an Online Certificate Status Protocol (OCSP) server whether the certificate has been revoked. This is also configured for the associated TLS Context.<br>**Note:** The parameter is applicable only if the connection protocol is HTTPS (defined using the 'Protocol' parameter, above). |

### 15.6.5 Configuring an HTTP-based EMS Service

The EMS Services table lets you configure a single HTTP-based EMS service. For a description of the EMS service, see 'HTTP-based Proxy Services' on page 275.

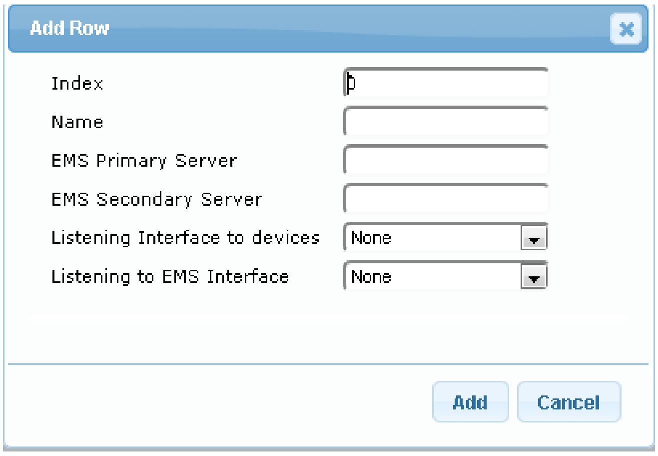
The following procedure describes how to configure an EMS Service through the Web interface. You can also configure it through ini file (EMSService) or CLI (configure system > http-proxy > ems-serv).

➢ **To configure an EMS Service:**

1. Open the EMS Services table (**Configuration** tab > **VoIP** menu > **Services** > **HTTP Proxy** > **EMS Services**).

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

**Figure 15-40: EMS Services Table - Add Row Dialog Box**




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

**Table 15-22: EMS Services Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[EMSService_Index] | Defines an index number for the new table row.<br>**Notes:**<br>▪ Each row must be configured with a unique index.<br>▪ The parameter is mandatory. |
| Name<br>`service-name`<br>[EMSService_ServiceName] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters. By default, no value is defined.<br>**Notes:**<br>▪ Each row must be configured with a unique name.<br>▪ The parameter is mandatory. |
| EMS Primary Server<br>`primary-server`<br>[EMSService_PrimaryServer] | Defines the address of the primary EMS server.<br>**Note:** The parameter is mandatory. |
| EMS Secondary Server<br>`secondary-server`<br>[EMSService_SecondaryServer] | Defines the address of the secondary EMS server. |
| Listening Interface to devices<br>`dev-login-int`<br>[EMSService_DeviceLoginInterface] | Assigns an HTTP Interface (local, listening HTTP interface:port) for communication with the client. To configure HTTP Interfaces, see 'Configuring HTTP Interfaces' on page 276.<br>By default, no value is defined (**None**).<br>**Note:** The parameter is mandatory. |

| Parameter | Description |
|---|---|
| Listing to EMS Interface<br>`ems-int`<br>[EMSService_EMSInterface] | Assigns an HTTP Interface (local, listening HTTP interface:port) for communication with the EMS. To configure HTTP Interfaces, see 'Configuring HTTP Interfaces' on page 276.<br>By default, no value is defined (**None**).<br>**Note:** The parameter is mandatory. |

## 15.7 Configuring Call Setup Rules

The Call Setup Rules table lets you configure up to 40 Call Setup rules. Call Setup rules define various sequences that are run upon the receipt of an incoming call (dialog) at call setup, before the device routes the call to its destination. Call Setup rules can be configured for any call direction (Tel-to-IP, or IP-to-Tel). Call Setup rules provides you with full flexibility in implementing simple or complex script-like rules that can be used for Lightweight Directory Access Protocol (LDAP) based routing as well as other advanced routing logic requirements such as manipulation. These Call Setup rules are assigned to routing rules.

Below is a summary of functions for which you can employ Call Setup rules:

- LDAP query rules: LDAP is used by the device to query Microsoft's Active Directory (AD) server for specific user details for routing, for example, office extension number, mobile number, private number, OCS (Lync) address, and display name. Call Setup rules provides full flexibility in AD-lookup configuration to suite just about any customer deployment requirement:
  - Routing based on query results.
  - Queries based on any AD attribute.
  - Queries based on any attribute value (alphanumeric), including the use of the asterisk (*) wildcard as well as the source number, destination number, redirect number. For example, the following Call Setup rule queries the attribute "proxyAddresses" for the record value "WOW:" followed by source number: "proxyAddresses=WOW:12345*"
  - Conditional LDAP queries, for example, where the query is based on two attributes (&(telephoneNumber=4064)(company=ABC).
  - Conditions for checking LDAP query results.
  - Manipulation of call parameters such as source number, destination number, and redirect number, while using LDAP query results.
  - Multiple LDAP queries.
- Manipulation (similar to the Message Manipulations table) of call parameters (such as source number, destination number, and redirect number).
- Conditions for routing, for example, if the source number equals a specific value, then use the call routing rule.

You configure Call Setup rules with a Set ID, similar to the Message Manipulations table, where multiple rules can be associated with the same Set ID. This lets you perform multiple Call Setup rules on the same call setup dialog.

To use your Call Setup rule(s), you need to assign the Call Setup Rules Set ID to the relevant routing rule. This is done using the 'Call Setup Rules Set ID' field in the routing table:

- Tel-to-IP routing rules - see Configuring Tel-to-IP Routing Rules on page 409
- IP-to-Tel routing rules - see "Configuring IP-to-Hunt Group Routing Rules" on page 421

If an incoming call matches the characteristics of a routing rule, the device **first** runs the assigned Call Setup Rules Set ID before routing the call according to the rule. The device uses the routing rule to route the call, depending on the result of the Call Setup Rules Set ID:

- **Rule's condition is met:** The device performs the rule's action and then runs the next rule in the Set ID until the last rule or until a rule with an **Exit** Action Type. If the **Exit** rule is configured with a "True" Action Value, the device uses the current routing rule. If the **Exit** rule is configured with a "False" Action Value, the device moves to the next routing rule. If an **Exit** Action Type is not configured and the device has run all the rules in the Set ID, the default Action Value of the Set ID is "True" (i.e., use the current routing rule).
- **Rule's condition is not met:** The device runs the next rule in the Set ID. When the device reaches the end of the Set ID and no **Exit** was performed, the Set ID ends with a "True" result.

You can also configure a Call Setup rule that determines whether the device must discontinue with the Call Setup Rules Set ID and route the call accordingly. This is done using the **Exit** optional value of the ‘Action Type’ parameter. When used, the ‘Action Value’ parameter can be configured to one of the following strings:

- “true”: Indicates that if the condition is met, the device routes the call according to the selected routing rule. Note that if the condition is not met, the device also uses the selected routing rule, unless the next Call Setup rule in the Set ID has an **Exit** option configured to “false” for an empty condition.
- “false”: Indicates that if the condition is met, the device attempts to route the call to the next matching routing rule (if configured). If the condition is not met, the device routes the call according to the selected routing rule.

As the default result of a Call Setup rule is always “true”, please adhere to the following guidelines when configuring the ‘Action Type’ field to **Exit**: If, for example, you want to exit the Call Setup Rule Set ID with "true" when LDAP query result is found and "false" when LDAP query result is not found:

- Incorrect -this rule will always exit with result = True:

  **Condition:** ldap.found exists **Action Type:** Exit **Action Value:** True
- Correct:
  - Single rule:

    **Condition:** ldap.found !exists **Action Type**: Exit **Action Value:** False
  - Set of rules:

    **Condition:** ldap.found exists **Action Type:** Exit **Action Value:** True

    **Condition:** <leave it blank> **Action Type:** Exit **Action Value:** False

**Note:** If the source and/or destination numbers are manipulated by the Call Setup rules, they revert to their original values if the device moves to the next routing rule.

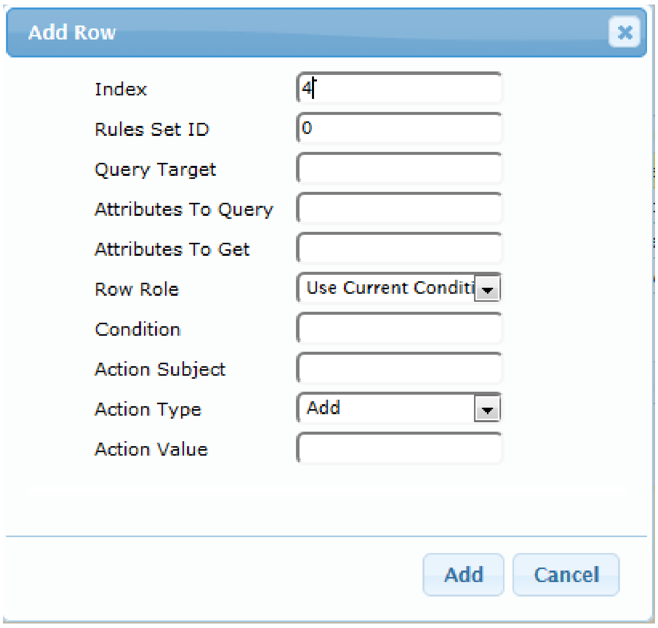
The following procedure describes how to configure Call Setup Rules through the Web interface. You can also configure it through ini file (CallSetupRules) or CLI (configure voip > services call-setup-rules).

**➢ To configure a Call Setup rule:**

1. Open the Call Setup Rules table (**Configuration** tab > **VoIP** menu > **Services** > **LDAP** > **Call Setup Rules**).

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

**Figure 15-41: Call Setup Rules Table - Add Row Dialog Box**




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

**Table 15-23: Call Setup Rules Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[CallSetupRules_Index] | Defines an index number for the new table record.<br>**Note:** Each rule must be configured with a unique index. |
| Rules Set ID<br>`rules-set-id`<br>[CallSetupRules_RulesSetID] | Defines a Set ID for the rule. You can define the same Set ID for multiple rules to create a group of rules. You can configure up to 10 Set IDs, where each Set ID can include up to 10 rules. The Set ID is used to assign the Call Setup rules to a routing rule in the routing table.<br>The valid value is 0 to 9. The default is 0. |
| Query Target<br>query-target<br>[CallSetupRules_QueryTarget] | Specifies an LDAP server (LDAP Server Group) on which to perform an LDAP query. To configure LDAP Server Groups, see Configuring LDAP Server Groups on page 234. |
| Attributes To Query<br>`attr-to-query`<br>[CallSetupRules_AttributesToQuery] | Defines the query string that the device sends to the LDAP server.<br>The valid value is a string of up to 100 characters. Combined strings and values can be configured like in the Message Manipulations table, using the '+' operator. Single quotes (') can be used for specifying a constant string (e.g., '12345').<br>For example:<br>▪ 'mobile=' + param.call.dst.user (searches for the AD attribute, "mobile" that has the value of the destination user part of the incoming call) |

| Parameter | Description |
|---|---|
| | ▪ 'telephoneNumber=' + param.call.redirect + '*' (searches for the AD attribute, "telephoneNumber" that has a redirect number) |
| Attributes To Get<br>`attr-to-get`<br>[CallSetupRules_AttributesToGet] | Defines the attributes of the queried LDAP record that the device must handle (e.g., retrieve value).<br>The valid value is a string of up to 100 characters. Up to five attributes can be defined, each separated by a comma (e.g., msRTCSIP-PrivateLine,msRTCSIP-Line,mobile).<br>**Note:** The device saves the retrieved attributes' values for future use in other rules, until the next LDAP query or until the call is connected. Thus, the device does not need to re-query the same attributes. |
| Row Role<br>`row-role`<br>[CallSetupRules_RowRole] | Determines which condition must be met in order for this rule to be performed.<br>▪ **[0]** Use Current Condition = The Condition configured for this rule must be matched in order to perform the configured action (default).<br>▪ **[1]** Use Previous Condition = The Condition configured for the rule located directly above this rule in the Call Setup table must be matched in order to perform the configured action. This option lets you configure multiple actions for the same Condition. |
| Condition<br>`condition`<br>[CallSetupRules_Condition] | Defines the condition that must exist for the device to perform the action.<br>The valid value is a string of up to 200 characters (case-insensitive). Regular Expression (regex) can also be used, for example:<br>▪ ldap.attr.mobile exists (attribute "mobile" exists in AD)<br>▪ param.call.dst.user == ldap.attr.msRTCSIP-PrivateLine (called number is the same as the number in the attribute "msRTCSIP-PrivateLine")<br>▪ ldap.found !exists (LDAP record not found)<br>▪ ldap.err exists (LDAP error exists) |
| Action Subject<br>`action-subject`<br>[CallSetupRules_ActionSubject] | Defines the element (header, parameter, or body) upon which you want to perform the action.<br>The valid value is a string of up to 100 characters (case-insensitive).<br>Examples:<br>▪ param.call.dst.user (called number)<br>▪ param.call.src.user (calling number)<br>▪ param.call.src.name (calling name)<br>▪ param.call.redirect (redirect number)<br>▪ param.call.src.host (source host)<br>▪ param.call.dst.host (destination host) |
| Action Type<br>`action-type`<br>[CallSetupRules_ActionType] | Defines the type of action to perform.<br>▪ **[0]** Add (default) = Adds new message header, parameter or body elements.<br>▪ **[1]** Remove = Removes message header, parameter, or body elements. |

| Parameter | Description |
|---|---|
| | ▪ **[2]** Modify = Sets element to the new value (all element types).<br>▪ **[3]** Add Prefix = Adds value at the beginning of the string (string element only).<br>▪ **[4]** Add Suffix = Adds value at the end of the string (string element only).<br>▪ **[5]** Remove Suffix = Removes value from the end of the string (string element only).<br>▪ **[6]** Remove Prefix = Removes value from the beginning of the string (string element only).<br>▪ **[20]** Run Rules Set = Performs a different Rule Set ID, specified in the 'Action Value' parameter (below).<br>▪ **[21]** Exit = Stops the Rule Set ID and returns a result ("True" or "False"). |
| Action Value<br>`action-value`<br>[CallSetupRules_ActionValue] | Defines a value that you want to use in the action.<br>The valid value is a string of up to 300 characters (case-insensitive).<br>Examples:<br>▪ '+9723976'+ldap.attr.alternateNumber<br>▪ '9764000'<br>▪ ldap.attr.displayName<br>▪ true (if the 'Action Type' is set to **Exit**)<br>▪ false (if the 'Action Type' is set to **Exit**) |

### 15.7.1 Call Setup Rule Examples

Below are configuration examples for using Call Setup Rules.

- **Example 1:** This example configures the device to replace (manipulate) the incoming call's source number with a number retrieved from the AD by an LDAP query. The device queries the AD server for the attribute record, "telephoneNumber" whose value is the same as the received source number (e.g., "telephoneNumber =4064"). If such an attribute is found, the device retrieves the number of the attribute record, "alternateNumber" and uses this number as the source number.
  - **Call Setup Rules table configuration:**
    - 'Rules Set ID': **1**
    - 'Attributes to Query': **'telephoneNumber=' + param.call.src.user**
    - 'Attributes to Get': **alternateNumber**
    - 'Row Role': **Use Current Condition**
    - 'Condition': **ldap.attr. alternateNumber exists**
    - 'Action Subject': **param.call.src.user**
    - 'Action Type': **Modify**
    - 'Action Value': **ldap.attr. alternateNumber**
  - **Routing table configuration:** A single routing rule is assigned the Call Setup Rule Set ID.
    - Index 1:
      - ✓ 'Call Setup Rules Set Id': **1**
- **Example 2:** This example configures the device to replace (manipulate) the incoming call's calling name (caller ID) with a name retrieved from the AD by an LDAP query. The device queries the AD server for the attribute record, "telephoneNumber" whose value is the same as the received source number (e.g., "telephoneNumber =5098"). If such an attribute is found, the device retrieves the name from the attribute record, "displayName" and uses this as the calling name in the incoming call.
  - **Call Setup Rules table configuration:**
    - 'Rules Set ID': **2**
    - 'Attributes to Query': **'telephoneNumber=' + param.call.src.user**
    - 'Attributes to Get': **displayName**
    - 'Row Role': **Use Current Condition**
    - 'Condition': **ldap.attr. displayName exists**
    - 'Action Subject': **param.call.src.name**
    - 'Action Type': **Modify**
    - 'Action Value': **ldap.attr. displayName**
  - **Routing table configuration:** A single routing rule is assigned the Call Setup Rule Set ID.
    - Index 1:
      - ✓ 'Call Setup Rules Set Id': **2**
- **Example 3:** This example configures the device to route the incoming call according to whether or not the source number of the incoming call also exists in the AD server. The device queries the AD server for the attribute record, "telephoneNumber" whose value is the same as the received source number (e.g., telephoneNumber=4064"). If such an attribute is found, the device sends the call to the Lync server; if the query fails, the device sends the call to the PBX.
  - **Call Setup Rules table configuration:**
    - 'Rules Set ID': **3**
    - 'Attributes to Query': **'telephoneNumber=' + param.call.src.user**

- 'Attributes to Get': **telephoneNumber**
- 'Row Role': **Use Current Condition**
- 'Condition': **ldap.found !exists**
- 'Action Subject': -
- 'Action Type': **Exit**
- 'Action Value': **false**

If the attribute record is found (i.e., condition is not met), the rule ends with a default exit result of true and uses the first routing rule (Lync). If the attribute record does not exist (i.e., condition is met), the rule exits with a false result and uses the second routing rule (PBX).

- **Routing table configuration:** Two routing rules are assigned with the same matching characteristics. Only the main routing rule is assigned a Call Setup Rules Set ID.
  - Index 1:
    - 'Call Setup Rules Set Id': **3**
    - 'Destination IP Group ID': **3** (IP Group for Lync)
  - Index 2:
    - 'Destination IP Group ID': **4** (IP Group of PBX)

This page is intentionally left blank.

# 16 Quality of Experience

This chapter describes how to configure the Quality of Experience feature.

## 16.1 Reporting Voice Quality of Experience to SEM

The device can be configured to report voice (media) Quality of Experience (QoE) to AudioCodes' Session Experience Manager (SEM) server, a plug-in for AudioCodes EMS. The reports include real-time metrics of the quality of the actual call experience, which are then processed by the SEM.

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

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

### 16.1.1 Configuring the SEM Server

The device can be configured to report QoE voice metrics to a single SEM server or to two SEM servers deployed in a Geographic Redundancy, High-Availability (HA) mode. Geographic Redundancy is when each SEM/EMS server is located in a different network subnet and has its own IP address. Thus, for the device to report QoE to both servers, you need to configure the IP address of each server. For normal HA mode, when both SEM/EMS servers are located in the same subnet, a single SEM/EMS server (global, virtual) IP address is used for all network components (EMS clients and managed devices). Thus, in such a setup, you need to configure only this IP address.

You can also configure the device to use a TLS connection with the SEM server. Before you can do this, configure a TLS Context (certificates) in the TLS Contexts table (see "Configuring TLS Certificate Contexts" on page 103). If no TLS Context is specified, the device uses the default TLS Context (ID 0).

You can also configure at what stage of the call the device must send the report to the SEM server. The report can be sent during the call or only at the end of the call. Reporting at the end of the call may be beneficial when network congestion occurs, as this reduces bandwidth usage over time.

> **Note:** If a QoE traffic overflow is experienced between SEM and the device, the device sends the QoE data only at the end of the call, regardless of your settings.

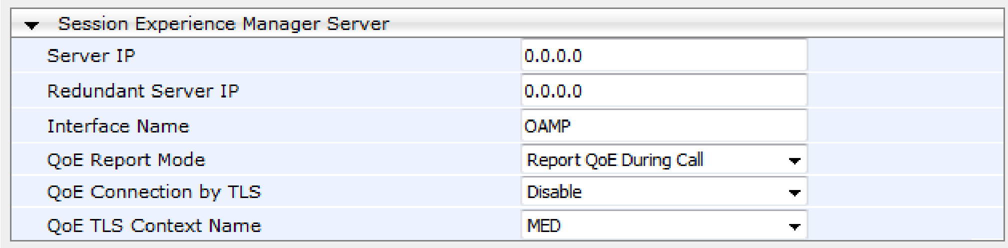
For a detailed description of the SEM parameters, see "Quality of Experience Parameters" on page 688.

➢ **To configure the SEM server address and other related features:**

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

> **Quality of Experience** > **Session Experience Manager Server**).

**Figure 16-1: Session Experience Manager Server Page**




2. Configure the address of the SEM server:
   a. In the 'Server IP' field, enter the primary SEM server's IP address.
   b. If Geographical-Redundancy HA mode exists, in the 'Redundant Server IP' field, enter the secondary SEM server's IP address.
   c. In the 'Interface Name' field, enter the device's IP network interface from which the device sends the reports to the SEM server.
3. From the 'QoE Report Mode' drop-down list, select when you want the device to send reports of a call to the SEM.
4. (Optional) Configure a TLS connection with the SEM server:
   a. From the 'QOE Connection by TLS' drop-down list, select **Enable**.
   b. From the 'Qoe TLS Context Name' drop-down list, select the desired TLS Context, which defines the TLS settings (e.g., certificates).
5. Click **Submit**, and then save ("burn") your settings to flash memory.

### 16.1.2 Configuring Clock Synchronization between Device and SEM

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

The NTP server can be one of the following:

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

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

To configure, the NTP server's address on the device, see "Configuring Automatic Date and Time using SNTP" on page 119.

### 16.1.3 Enabling RTCP XR Reporting to SEM

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

For enabling RTCP XR reporting, see "Configuring RTCP XR" on page 571. For configuring what to report to the SEM, see "Configuring Quality of Experience Profiles" on page 293.

## 16.2 Configuring Quality of Experience Profiles

The Quality of Experience feature lets you monitor the quality of voice calls traversing the device in your network. Voice-metric monitoring profiles (Quality of Experience Profiles) can be configured and applied to specific network links, including IP Groups (see "Configuring IP Groups" on page 323), Media Realms (see "Configuring Media Realms" on page 303), and Remote Media Subnets (see "Configuring Remote Media Subnets" on page 307).

The monitored voice metrics include the following:

- **Mean Opinion Score (MOS):** MOS is the average grade on a quality scale, expressed as a single number in the range of 1 to 5, where 1 is the lowest audio quality and 5 the highest audio quality.
- **Delay (or latency):** Time it takes for information to travel from source to destination (round-trip time).
- **Packet Loss:** Lost packets are RTP packets that are not received by the voice endpoint. Packet loss can result in choppy voice transmission.
- **Jitter:** Jitter can result from uneven delays between received voice packets. To space evenly, the device's jitter buffer adds delay. The higher the measurement, the greater the impact of the jitter buffer's delay on audio quality.
- **Residual Echo Return Loss (RERL):** An echo is a reflection of sound arriving at the listener at some time after the sound was initiated (often by the listener). Echo is typically caused by delay.

At any given time during a call, a voice metric can be in one of the following color-coded quality states:

- **Green:** Indicates good call quality
- **Yellow:** Indicates medium call quality
- **Red:** Indicates poor call quality

Quality of Experience Profiles let you configure quality thresholds per monitored voice metric. These are based on the following color-coded quality thresholds:

- **Green-Yellow threshold:** Lower threshold that indicates changes from Green to Yellow or vice versa when the threshold is crossed.
- **Yellow-Red threshold:** Higher threshold that indicates changes from Yellow to Red or vice versa when the threshold is crossed.

Hysteresis is also used to configure the threshold. This defines the amount of fluctuation from a threshold in order for the threshold to be considered as crossed (i.e., change in color state). Hysteresis is used to avoid false reports being sent by the device.

Each time a configured voice metric threshold is crossed (i.e., color changes), the device can do the following, depending on configuration:

- Report the change in the measured metrics to AudioCodes' Session Experience Manager (SEM) server. The SEM displays this call quality status for the associated SEM link (IP Group, Media Realm, or Remote Media Subnet). For configuring the SEM server's address, see "Configuring the SEM Server" on page 291.
- Determine access control and media enhancements based on measured metrics. Depending on the crossed threshold type, you can configure the device to accept or reject calls, or use an alternative IP Profile for the IP Group to which the call belongs. For more information, see "Configuring Media Enhancement Profiles" on page 300.
- Alternative routing based on measured metrics. If a call is rejected because of a crossed threshold, the device generates a SIP 806 response. You can configure this SIP response code as a reason for alternative routing (see "Configuring SIP Response Codes for Alternative Routing Reasons" on page 431).

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

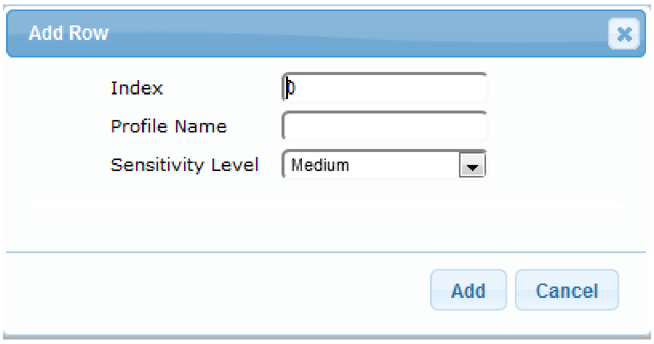
The following procedure describes how to configure Quality of Experience Profiles through the Web interface. You can also configure it through other management platforms:

- **Quality of Experience Profile table:** *ini* file (QoEProfile) or CLI (configure voip/qoe qoe-profile)
- **Quality of Experience Color Rules table:** *ini* file (QOEColorRules) or CLI (configure voip/qoe qoe-profile qoe-color-rules)

➢ **To configure a QoE Profile:**

1. Open the Quality of Experience Profile page (**Configuration** tab > **VoIP** menu > **Quality of Experience** > **Quality of Experience Profile**).
2. Click **Add**; the following dialog box appears:

**Figure 16-2: Quality of Experience Profile Table - Add Row Dialog Box**




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

**Table 16-1: Quality of Experience Profile Table Parameter Descriptions**

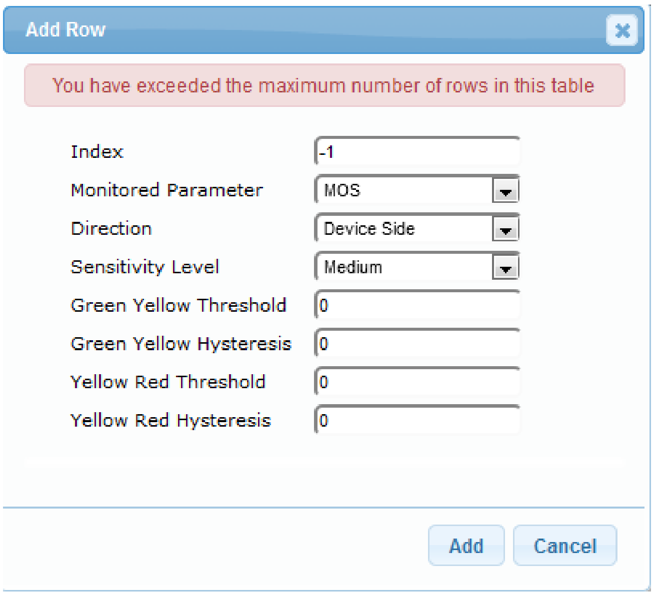
| Parameter | Description |
|---|---|
| Index<br>[QOEProfile_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Profile Name<br>name<br>[QOEProfile_Name] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 20 characters. |
| Sensitivity Level<br>sensitivity-level<br>[QOEProfile_SensitivityLevel] | Defines the pre-configured threshold profile to use.<br>▪ **[0]** User Defined = Need to define thresholds per monitored parameter in the Quality of Experience Color Rules table.<br>▪ **[1]** Low = Pre-configured low sensitivity thresholds.<br>▪ **[2]** Medium = (Default) Pre-configured medium sensitivity thresholds.<br>▪ **[3]** High = Pre-configured high sensitivity thresholds. Reporting is done for small fluctuations in parameter values. |

5. In the Quality of Experience Profile page, select the QoE Profile index row for which you

want to configure QoE thresholds, and then click the **Quality of Experience Color Rules** link located below the table; the Quality of Experience Color Rules page appears.

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

**Figure 16-3: Quality of Experience Table - Add Row Dialog Box**




The figure above shows a configuration example where if the MOS value changes by 0.1 (hysteresis) to 3.3 or 3.5, the Green-Yellow threshold is crossed. The device considers a change to 3.3 as a Yellow state (i.e., medium quality) and a change to 3.5 as a Green state.

7. Configure a QoE Color rule according to the parameters described in the table below.
8. Click **Add**, and then save ("burn") your settings to flash memory.

**Table 16-2: Quality of Experience Color Rules Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>`index`<br>[QOEColorRules_ColorRuleIndex] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Monitored Parameter<br>`monitored-parameter`<br>[QOEColorRules_monitoredParam] | Defines the parameter to monitor and report.<br>▪ **[0]** MOS (default)<br>▪ **[1]** Delay<br>▪ **[2]** Packet Loss<br>▪ **[3]** Jitter<br>▪ **[4]** RERL [Echo] |
| Direction<br>`direction`<br>[QOEColorRules_direction] | Defines the monitoring direction.<br>▪ **[0]** Device Side (default)<br>▪ **[1]** Remote Side |

| Parameter | Description |
|---|---|
| Sensitivity Level<br>`sensitivity-level`<br>[QOEColorRules_profile] | Defines the sensitivity level of the thresholds.<br>▪ **[0]** User Defined = Need to define the thresholds in the parameters described below.<br>▪ **[1]** Low = Pre-configured low sensitivity threshold values. Thus, reporting is done only if changes in parameters' values are significant.<br>▪ **[2]** Medium = (Default) Pre-configured medium sensitivity threshold values.<br>▪ **[3]** High = Pre-configured high sensitivity threshold values. Thus, reporting is done for small fluctuations in parameter values. |
| Green Yellow Threshold<br>`green-yellow-threshold`<br>[QOEColorRules_GreenYellowThreshold] | Defines the parameter threshold values between Green (good quality) and Yellow (medium quality) states.<br>The valid threshold values are as follows:<br>▪ MOS values are in multiples of 10. For example, to denote a MOS of 3.2, the value 32 (i.e., 3.2*10) must be entered.<br>▪ Delay values are in msec.<br>▪ Packet Loss values are in percentage (%).<br>▪ Jitter is in msec.<br>▪ Echo measures the Residual Echo Return Loss (RERL) in dB. |
| Green Yellow Hysteresis<br>`green-yellow-hysteresis`<br>[QOEColorRules_GreenYellowHysteresis] | Defines the fluctuation (change) from the value configured for the Green-Yellow threshold. When the threshold is exceeded by this hysteresis, the device sends a report to the SEM indicating this change.<br>**Note:** If the monitored parameter crosses two thresholds at once (e.g., from Green to Red), the device ignores the hysteresis value and reports the call state change to the SEM. |
| Yellow Red Threshold<br>`yellow-red-threshold`<br>[QOEColorRules_YellowRedThreshold] | Defines the parameter threshold values between Yellow (medium quality) and Red (poor quality) states.<br>The valid threshold values are as follows:<br>▪ MOS values are in multiples of 10. For example, to denote a MOS of 3.2, the value 32 (i.e., 3.2*10) must be entered.<br>▪ Delay values are in msec.<br>▪ Packet Loss values are in percentage (%).<br>▪ Jitter is in msec.<br>▪ Echo measures the Residual Echo Return Loss (RERL) in dB. |
| Yellow Red Hysteresis<br>`yellow-red-hysteresis`<br>[QOEColorRules_YellowRedHysteresis] | Defines the fluctuation (change) from the value configured for the Yellow-Red threshold. When the threshold is exceeded by this hysteresis value, the device sends a report to the SEM indicating this change.<br>**Note:** If the monitored parameter crosses two thresholds at once (e.g., from Green to Red), the device ignores the hysteresis value and reports the call state change to the SEM. |

## 16.3 Configuring Bandwidth Profiles

Bandwidth Profiles enhance the device's monitoring of bandwidth utilization. A Bandwidth Profile defines bandwidth utilization thresholds for audio and/or video traffic (incoming and outgoing). Bandwidth Profiles can be assigned to IP Groups (see "Configuring IP Groups" on page 323), Media Realms (see "Configuring Media Realms" on page 303), and Remote Media Subnets (see "Configuring Remote Media Subnets" on page 307).

Each time a configured bandwidth threshold is crossed, the device can do the following, depending on configuration:

- Determine access control and media enhancements based on bandwidth utilization. Depending on the crossed threshold type, you can configure the device to accept or reject calls, or use an alternative IP Profile for the IP Group to which the call belongs. For more information, see "Configuring Media Enhancement Profiles" on page 300.
- Alternative routing based on bandwidth utilization. If a call is rejected because of a crossed threshold, the device generates a SIP 806 response. You can configure this SIP response code as a reason for alternative routing (see "Configuring SIP Response Codes for Alternative Routing Reasons" on page 431).
- Send an SNMP alarm (acMediaRealmBWThresholdAlarm). The device clears the alarm when bandwidth utilization returns to normal (within the thresholds).

The thresholds of Bandwidth Profiles use the same color-coding as the Quality of Experience Profile:

- **Green-Yellow threshold:** Lower threshold that indicates that the bandwidth exceeded a user-defined percentage of the configured threshold. This is referred to as a "Warning" alarm (i.e., warning you that bandwidth is nearing the threshold). When bandwidth goes over the threshold, the device considers it as a Yellow state; when it goes below the threshold, it considers it as a Green state.
- **Yellow-Red threshold:** Indicates that bandwidth has exceeded the configured threshold. When bandwidth goes over the threshold, the device considers it as a Red state; when it goes below the threshold, it considers it as a Yellow state.

Hysteresis is also used to configure the threshold. This defines the amount of fluctuation from a threshold in order for the threshold to be considered as crossed (i.e., change in color state). Hysteresis is used to avoid false reports.

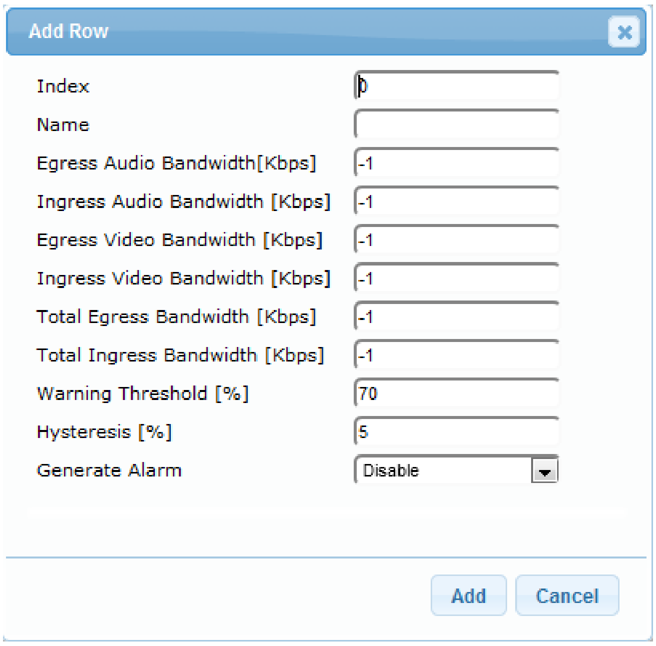
The following procedure describes how to configure Bandwidth Profiles through the Web interface. You can also configure it through ini file (BWProfile) or CLI (configure voip > qoe bw-profile).

➢ **To configure Bandwidth Profiles:**

1. Open the Bandwidth Profile page (**Configuration** tab > **VoIP** menu > **Quality of Experience** > **Bandwidth Profile**).

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

**Figure 16-4: Bandwidth Profile Table - Add Row Dialog Box**




The figure above shows a configuration example where if the outgoing voice traffic threshold of 64,000 increases by 80% (70% warning threshold plus 10% hysteresis) to 115,200 (64,000 plus 51,200), a Yellow state occurs and an alarm is sent. If the threshold increases by 10%, a Red state occurs and an alarm is sent.

3. Configure a Bandwidth Profile according to the parameters described in the table below.
4. Click **Add**, and then reset the device with a save ("burn") to flash memory.

**Table 16-3: Bandwidth Profile Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[BWProfile_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>`name`<br>[BWProfile_Name] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 20 characters. |
| Egress Audio Bandwidth<br>`egress-audio-bandwidth`<br>[BWProfile_EgressAudioBandwidth] | Defines the outgoing audio traffic threshold (in Kbps). |
| Ingress Audio Bandwidth<br>`ingress-audio-bandwidth`<br>[BWProfile_IngressAudioBandwidth] | Defines the incoming audio traffic threshold (in Kbps). |
| Egress Video Bandwidth<br>`egress-video-bandwidth`<br>[BWProfile_EgressVideoBandwidth] | Defines the outgoing video traffic threshold (in Kbps). |
| Ingress Video Bandwidth<br>`ingress-video-bandwidth`<br>[BWProfile_IngressVideoBandwidth] | Defines the incoming video traffic threshold (in Kbps). |

| Parameter | Description |
|---|---|
| Total Egress Bandwidth<br>`total-egress-bandwidth`<br>[BWProfile_TotalEgressBandwidth] | Defines the total (video and audio) outgoing bandwidth threshold (in Kbps). |
| Total Ingress Bandwidth<br>`total-ingress-bandwidth`<br>[BWProfile_TotalIngressBandwidth] | Defines the total (video and audio) incoming bandwidth threshold (in Kbps). |
| Warning Threshold<br>`warning-threshold`<br>[BWProfile_WarningThreshold] | Defines the threshold (in percentage) of the bandwidth thresholds that if exceeded is considered a Warning alarm (Green-Yellow threshold). This applies to any of the configured bandwidth thresholds. The Hysteresis is also added to this Warning threshold. For example, if set to 70% and the Hysteresis to 10%, when the current outgoing voice traffic exceeds 80% of the configured threshold, the Yellow state occurs and a Warning threshold alarm is sent if 'Generate Alarm' is set to **Enable**. |
| Hysteresis<br>`hysteresis`<br>[BWProfile_hysteresis] | Defines the bandwidth fluctuation (change) from the bandwidth threshold value (in percentage). The threshold is considered crossed if bandwidth exceeds the configured threshold plus this hysteresis, and a Red state occurs. For example, assume the parameter is set to 10% and the configured bandwidth threshold is set to 64000 Kbps. If current bandwidth reaches 70,400 Kbps (additional 10%), the threshold is considered crossed. |
| Generate Alarm<br>`generate-alarms`<br>[BWProfile_GenerateAlarms] | Enables the generation of an SNMP alarm if the threshold (with the hysteresis) is crossed.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>If enabled, an alarm is sent if one of the following scenarios occurs:<br>▪ Warning threshold is exceeded (Warning severity - Yellow threshold).<br>▪ Any configured bandwidth threshold is exceeded (Major severity - Red threshold). |

## 16.4 Configuring Media Enhancement Profiles

Media Enhancement Profiles provides support for access control and media quality enhancements based on call quality measurements (configured in "Configuring Quality of Experience Profiles" on page 293) and bandwidth utilization (configured in "Configuring Bandwidth Profiles" on page 297). These profiles contain color-coded thresholds that are used to trigger access control and/or media enhancements.

The Media Enhancement Profile table lets you configure any one of the following actions when a specific color-coded threshold (Green-Yellow or Yellow-Red) is crossed for a specific monitored voice metrics (e.g., MOS) or bandwidth (e.g., Egress Audio Bandwidth):

- Reject new calls until the voice metrics or bandwidth returns to below the threshold. This can be used, for example, to reject new calls when bandwidth threshold is exceeded.
- Use a different IP Profile. For example, if packet loss is detected, the IP Group (to which the Media Enhancement Rule is later assigned) can switch to an IP Profile configured with a higher RTP redundancy level. The ability to use a different IP Profile when call quality or bandwidth thresholds are crossed provides a wide range of options for media enhancement and traffic shaping. For example, it may be used to:
  - switch to a low bit-rate coder,
  - negotiate different p-time (and perform transrating if required),
  - increase RTP redundancy level,
  - or block video calls.
- Accept calls

A Media Enhancement Profile can later be assigned to an IP Group (in the IP Group table). However, when the device analyzes the call and determines whether Media Enhancement Profile should be applied or not, it searches for the "most relevant" Quality of Experience Profile or Bandwidth Profile in the following order: 1) Remote Media Subnet, 2) Media Realm, and then 3) IP Group. Thus, a Media Enhancement Profile associated with a specific IP Group may actually "respond" to Quality of Experience or bandwidth thresholds crossed at the Media Realm or Remote Media Subnet level.

**Notes:**

- The color-coded threshold is first calculated for the IP Group and only then for the Media Realm. The device uses the "worst" color-coded threshold crossing. For example, if a Media Realm crossed a Green-Yellow threshold and an IP Group a Yellow-Red threshold, the action defined for the Red color state is used.
- The device applies Media Enhancements Profiles on new calls **only**, based on the information gathered from previous and/or currently established calls.

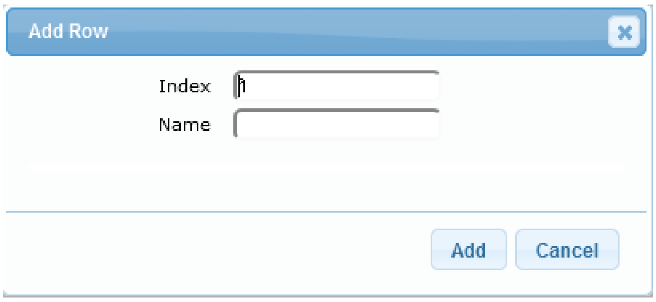
The following procedure describes how to configure Media Enhancement Profiles through the Web interface. You can also configure it through other management platforms:

- **Media Enhancement Profile table:** *ini* file (MediaEnhancementProfile) or CLI (configure voip/qoe media-enhancement)
- **Media Enhancement Rules table:** *ini* file (MediaEnhancementRules) or CLI (configure voip/qoe media-enhancement-rules)

### ➢ To configure a Media Enhancement Profile:

1. Open the Media Enhancement Profile page (**Configuration** tab > **VoIP** menu > **Quality of Experience** > **Media Enhancement Profile**).
2. Click **Add**; the following dialog box appears:

**Figure 16-5: Media Enhancement Profile Table - Add Row Dialog Box**



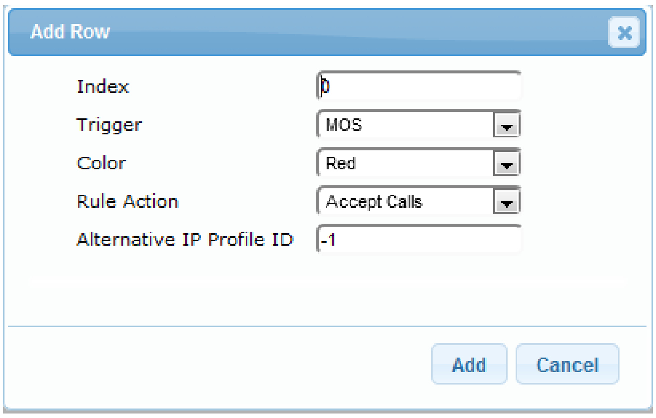

3. Configure a Media Enhancement Profile according to the parameters described in the table below.
4. Click **Add**.

**Table 16-4: Media Enhancement Profile Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[MediaEnhancementProfile_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>`profile-name`<br>[MediaEnhancementProfile_ProfileName] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 20 characters. |

5. In the Media Enhancement Profile table, select the required Media Enhancement Profile index row, and then click the **Media Enhancement Rules** link located below the table; the Media Enhancement Rules page appears.
6. Click **Add**; the following dialog box appears:

**Figure 16-6: Media Enhancement Rules Table - Add Row Dialog Box**




7. Configure a Media Enhancement Rule according to the parameters described in the table below.
8. Click **Add**, and then reset the device with a save ("burn") to flash memory.

**Table 16-5: Media Enhancement Rules Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>`rule-index`<br>[MediaEnhancementRules_RuleIndex] | Defines the index of the table row entry. |
| Trigger<br>`trigger`<br>[MediaEnhancementRules_Trigger] | Defines the monitored metrics parameter or bandwidth associated with this rule.<br>▪ **[0]** MOS (default)<br>▪ **[1]** Delay<br>▪ **[2]** Packet Loss<br>▪ **[3]** Jitter<br>▪ **[4]** Bandwidth |
| Color<br>`color`<br>[MediaEnhancementRules_Color] | Defines the color-coded threshold change of the monitored metrics or bandwidth (configured in the 'Trigger' parameter) for which this rule is done.<br>▪ **[0]** Red (default) = Yellow-to-Red threshold is crossed.<br>▪ **[1]** Yellow = Green-to-Yellow threshold is crossed. |
| Rule Action<br>`action-rule`<br>[MediaEnhancementRules_ActionRule] | Defines the action that the device performs when the color-coded threshold is crossed:<br>▪ **[0]** Accept Calls (default)<br>▪ **[1]** Reject Calls<br>▪ **[2]** Alternative IP Profile = An alternative IP Profile is used, as configured in the 'Value' field (below).<br>**Notes:**<br>▪ If the parameter is set to a restrictive action (i.e., **Reject Calls** or **Alternative IP Profile**) for Yellow and no action is set for Red, the device also applies the Yellow action to Red, if this color-coded threshold occurs.<br>▪ If the parameter is set to a permissive action (i.e., **Accept Calls**) for Red and no action is set for Yellow, the device applies the same action to Yellow, if this color-coded threshold occurs. |
| Alternative IP Profile ID<br>`value`<br>[MediaEnhancementRules_ActionValue] | Defines an alternative IP Profile ID for the IP Group that is associated with this rule, if this rule is applied. The parameter is applicable only if the 'Rule Action' parameter is set to **Alternative IP Profile**. |

# 17 Control Network

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

## 17.1 Configuring Media Realms

The Media Realm table lets you configure a pool of up to 64 SIP media interfaces, termed *Media Realms*. Media Realms lets you divide a Media-type interface (configured in the Interface table) into several media realms, where each realm is specified by a UDP port range. Media Realms also define the maximum number of permitted media sessions.

Once configured, to apply Media Realms to specific calls, you need to assign them to any of the following configuration entities:

- IP Groups (see "Configuring IP Groups" on page 323)
- SIP Interfaces (see "Configuring SIP Interfaces" on page 319)

You can also apply the device's Quality of Experience feature to Media Realms:

- **Quality of Experience Profile:** Call quality monitoring based on thresholds for voice metrics (e.g., MOS) can be applied per Media Realm. For example, if MOS is considered poor, calls on this Media Realm can be rejected. For configuring Quality of Experience Profiles, see "Configuring Quality of Experience Profiles" on page 293.
- **Bandwidth Profile:** Bandwidth utilization thresholds can be applied per Media Realm. For example, if bandwidth thresholds are crossed, the device can reject any new new calls on this Media Realm. For configuring Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 297.

The Media Realm table provides sub-tables ("child" tables) that let you configure the following:

- Remote Media Subnets: Defines remote destination subnets per Media Realm and assigns each subnet a Quality of Experience Profile and Bandwidth Profile. For more information, see "Configuring Remote Media Subnets" on page 307.
- Media Realm Extensions: Defines port ranges for multiple Media-type interfaces per Media Realm. For more information, see "Configuring Media Realm Extensions" on page 309.

**Notes:**

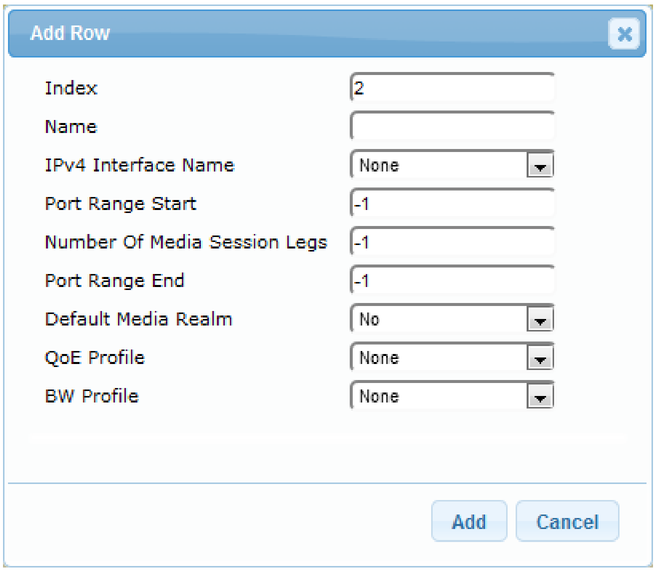
- The Media Realm assigned to an IP Group overrides any other Media Realm assigned to any other configuration entity associated with the call.
- If you modify a Media Realm that is currently being used by a call, the device does not perform Quality of Experience for the call.
- If you delete a Media Realm that is currently being used by a call, the device maintains the call until the call parties end the call.

The following procedure describes how to configure Media Realms through the Web interface. You can also configure it through ini file (CpMediaRealm) or CLI (configure voip > voip-network realm).

## ➢ To configure a Media Realm:

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

**Figure 17-1: Media Realm Table - Add Row Dialog Box**




3. Configure the Media Realm according to the parameters described in the table below.
4. Click **Add**.

**Table 17-1: Media Realm Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[CpMediaRealm_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>`name`<br>[CpMediaRealm_MediaRealmName] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters.<br>**Notes:**<br>▪ The parameter is mandatory.<br>▪ Each row must be configured with a unique name. |
| IPv4 Interface Name<br>`ipv4`<br>[CpMediaRealm_IPv4IF] | Assigns an IPv4 network interface to the Media Realm. This is the name of the interface as configured in the 'Interface Name' parameter in the Interface table.<br>By default, no value is defined (**None**). |
| IPv6 Interface Name<br>`ipv6if`<br>[CpMediaRealm_IPv6IF] | Assigns an IPv6 network interface to the Media Realm. This is the name of the interface as configured for the 'Interface Name' parameter in the Interface table.<br>By default, no value is defined (None). |

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Port Range Start<br><code>port-range-start</code><br>[CpMediaRealm_PortRangeStart]</td><td>Defines the starting port for the range of media interface UDP ports.<br>By default, no value is defined.<br>Notes:<ul><li>You must either configure all your Media Realms with port ranges or all without; not some with and some without.</li><li>The available UDP port range is according to the BaseUDPport parameter. For more information, see "Configuring RTP Base UDP Port" on page 197.</li><li>The base UDP port number (BaseUDPPort parameter) must be greater than the highest UDP port configured for a SIP Interface (see Configuring SIP Interfaces on page 319). For example, if your highest configured UDP port for a SIP Interface is 6060, you must configure the BaseUDPPort parameter to any value greater than 6060.</li><li>The port must be different from ports configured for SIP traffic (i.e., ports configured for SIP Interfaces). For example, if the RTP port range is 6000 to 6999, the SIP port can be less than 6000 or greater than 6999.</li></ul></td></tr>
<tr><td>Number of Media Session Legs<br><code>session-leg</code><br>[CpMediaRealm_MediaSessionLeg]</td><td>Defines the number of media sessions for the configured port range.<br>By default, no value is defined.</td></tr>
<tr><td>Port Range End<br><code>port-range-end</code><br>[CpMediaRealm_PortRangeEnd]</td><td>(Read-only field) Displays the ending port for the range of media interface UDP ports. The device automatically populates the parameter with a value, calculated by the summation of the 'Port Range Start' parameter and 'Number of Media Session Legs' parameter (multiplied by the port spacing) minus 1:<br><code>start port + (sessions * port spacing) - 1</code><br>For example, a port starting at 6,000, 5 sessions and 10 port spacing:<br><code>6,000 + (5 * 10) - 1 = 6,000 + (50) - 1 = 6,000 + 49 = 6,049</code><br>The device allocates the UDP ports for RTP, RTCP and T.38 in "jumps" (spacing) of 10 (default. For example, if the port range starts at 6000 and the UDP port spacing is 10, the available ports include 6000, 6010, 6020, 6030, and so on (depending on number of media sessions).<br>For RTCP and T.38 traffic, the port offset from the RTP port used for the voice session (channel) is one and two, respectively. For example, if the voice session uses RTP port 6000, the RTCP port and T.38 port for the session is 6001 and 6002, respectively. However, you can configure the device to use the same port for RTP and T.38 packets, by setting the T38UseRTPPort parameter to 1.</td></tr>
</table>

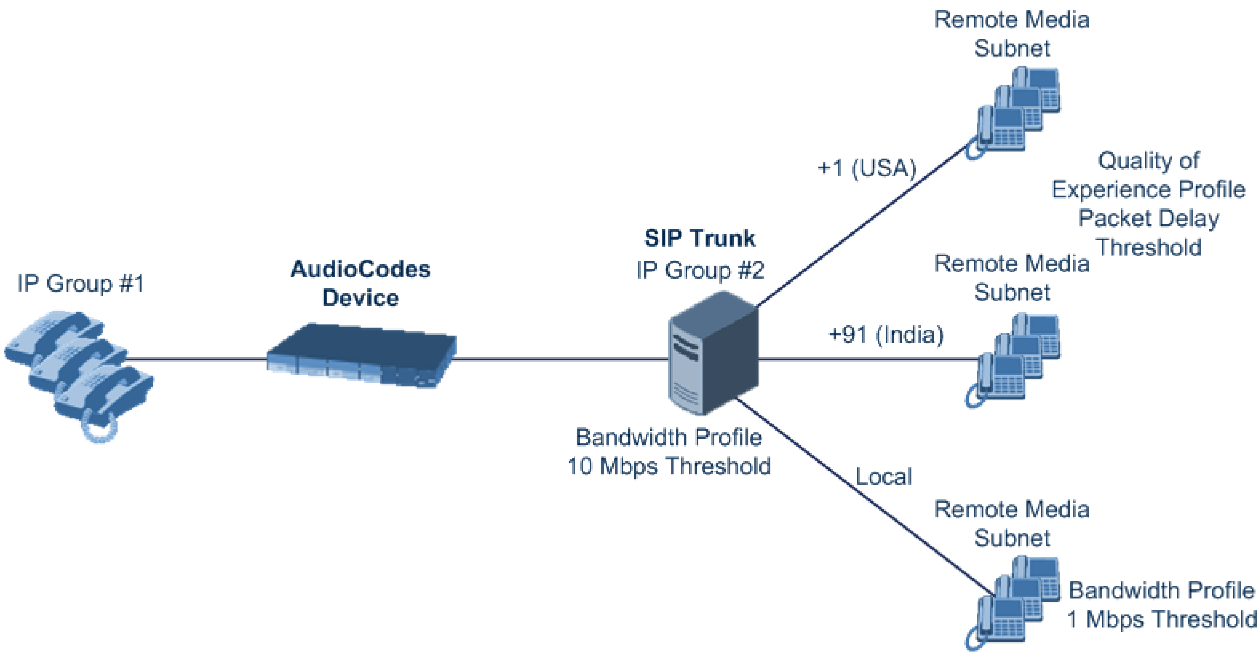
| Parameter | Description |
|---|---|
| Default Media Realm<br>`is-default`<br>[CpMediaRealm_IsDefault] | Defines the Media Realm as the default Media Realm. The default Media Realm is used for SIP Interfaces and IP Groups for which you have not assigned a Media Realm.<br>▪ **[0]** No (default)<br>▪ **[1]** Yes<br>**Notes:**<br>▪ You can configure the parameter to **Yes** for only **one** Media Realm; all the other Media Realms must be configured to **No**.<br>▪ If you do not configure the parameter (i.e., the parameter is **No** for all Media Realms), the device uses the first Media Realm in the table as the default.<br>▪ If the table is not configured, the default Media Realm includes all configured media interfaces. |
| QoE Profile<br>`qoe-profile`<br>[CpMediaRealm_QoeProfile] | Assigns a QoE Profile to the Media Realm.<br>By default, no value is defined (**None**).<br>For configuring QoE Profiles, see "Configuring Quality of Experience Profiles" on page 293. |
| BW Profile<br>`bw-profile`<br>[CpMediaRealm_BWProfile] | Assigns a Bandwidth Profile to the Media Realm.<br>By default, no value is defined (**None**).<br>For configuring Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 297. |

### 17.1.1 Configuring Remote Media Subnets

Remote Media Subnets define destination subnets for media (RTP/SRTP) traffic on a specific Media Realm. Each Remote Media Subnet can be assigned different call quality (Quality of Experience Profile) and bandwidth utilization (Bandwidth Profile) profiles. These profiles are configured in "Configuring Quality of Experience Profiles" on page 293 and "Configuring Bandwidth Profiles" on page 297, respectively. Thus, you can apply these profiles to remote media subnets instead of Media Realms or IP Groups. You can configure up to five Remote Media Subnets per Media Realm.

The figure below illustrates an example for implementing Remote Media Subnets. IP Group #2 represents a SIP Trunk which routes international (USA and India) and local calls. As international calls are typically more prone to higher delay than local calls, different Quality of Experience Profiles are assigned to them. This is done by creating Remote Media Subnets for each of these call destinations and assigning each Remote Media Subnet a different Quality of Experience Profile. A Quality of Experience Profile that defines a packet delay threshold is assigned to the international calls, which if crossed, a different IP Profile is used that defines higher traffic priority to voice over other traffic. In addition, IP Group #2 has a 10-Mbps bandwidth threshold and a "tighter" bandwidth limitation (e.g., 1 Mbps) is allocated to local calls. If this limit is exceeded, the device rejects new calls to this Remote Media Subnet.

**Figure 17-2: Remote Media Subnets Example**



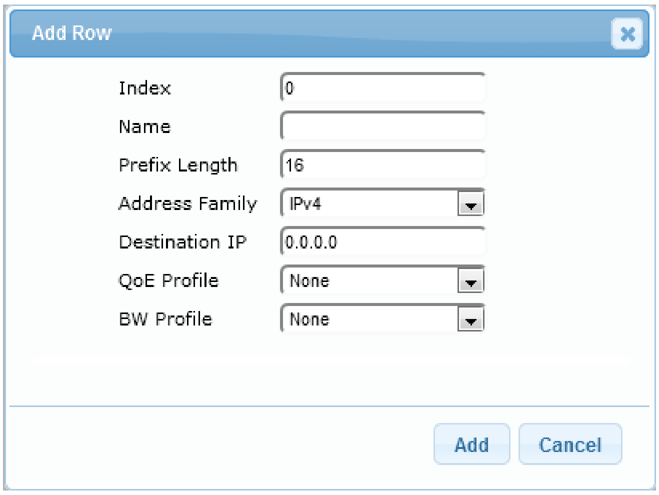

The following procedure describes how to configure Remote Media Subnets through the Web interface. You can also configure it through ini file (RemoteMediaSubnet) or CLI (configure voip > voip-network realm remote-media-subnet).

- **To configure a Remote Media Subnet:**

1. Open the Media Realm table (see "Configuring Media Realms" on page 303).
2. Select the Media Realm row for which you want to add Remote Media Subnets, and then click the **Remote Media Subnet** link located below the table; the Remote Media Subnet table appears.

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

**Figure 17-3: Remote Media Subnet Table - Add Row Dialog Box**




4. Configure the Remote Media Subnet according to the parameters described in the table below.
5. Click **Add**.

**Table 17-2: Remote Media Subnet Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[RemoteMediaSubnet_RemoteMediaSubnetIndex] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>`name`<br>[RemoteMediaSubnet_RemoteMediaSubnetName] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 20 characters.<br>**Note:** Each row must be configured with a unique name. |
| Prefix Length<br>`prefix-length`<br>[RemoteMediaSubnet_PrefixLength] | Defines the subnet mask in Classless Inter-Domain Routing (CIDR) notation. For example, 16 denotes 255.255.0.0.<br>The default is 16. |
| Address Family<br>`address-family`<br>[RemoteMediaSubnet_AddressFamily] | Defines the IP address protocol.<br>▪ **[2]** IPv4 (default)<br>▪ **[10]** IPv6 |
| Destination IP<br>`dst-ip-address`<br>[RemoteMediaSubnet_DstIPAddress] | Defines the IP address of the destination.<br>The default is 0.0.0.0. |
| QOE Profile Name<br>`qoe-profile`<br>[RemoteMediaSubnet_QOEProfileName] | Assigns a Quality of Experience Profile to the Remote Media Subnet.<br>By default, no value is defined (**None**).<br>For configuring QoE Profiles, see "Configuring Quality of Experience Profiles" on page 293. |

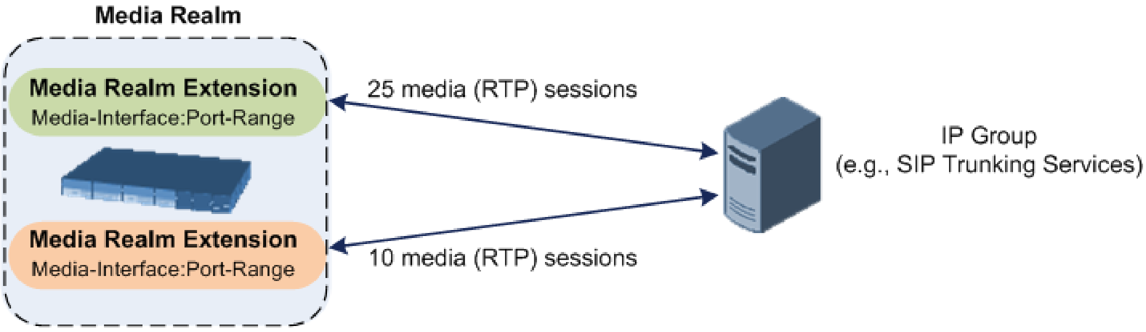
| Parameter | Description |
|---|---|
| BW Profile Name<br>`bw-profile`<br>[RemoteMediaSubnet_BWProfileName] | Assigns a Bandwidth Profile to the Remote Media Subnet.<br>By default, no value is defined (**None**).<br>For configuring Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 297. |

## 17.1.2 Configuring Media Realm Extensions

The Media Realm Extension table lets you configure 2 Media Realm Extensions. A Media Realm Extension is associated with a specific Media Realm and defines a port range and the number of media sessions for a specific Media-type network interface (configured in the IP Interfaces table). Therefore, a Media Realm Extension enhances a Media Realm by allowing you to define different port ranges, media sessions, and network interface than is defined by the associated Media Realm (i.e., the Media Realm is distributed across multiple interfaces).

Media Realm Extensions can be useful, for example, to overcome limitations of the maximum number of media ports supported per interface. Instead of configuring only a single Media Realm in the Media Realm table (see "Configuring Media Realms" on page 303), you can also configure additional "Media Realms" in the Media Realm Extensions table associated with the single Media Realm. An IP Group that is associated with a Media Realm configured with Media Realm Extensions, allocates its media sessions / ports between the different interfaces, as configured by the Media Real and its associated Media Realm Extensions. For example, two Media Realm Extensions could be configured, whereby one allocates 25 media sessions on interface "LAN-1" and another, 10 sessions on interface "LAN-2". The Media Realm associated with these Media Realm Extensions would be assigned to the relevant IP Group.

**Figure 17-4: Example of Implementation of Media Realm Extensions**



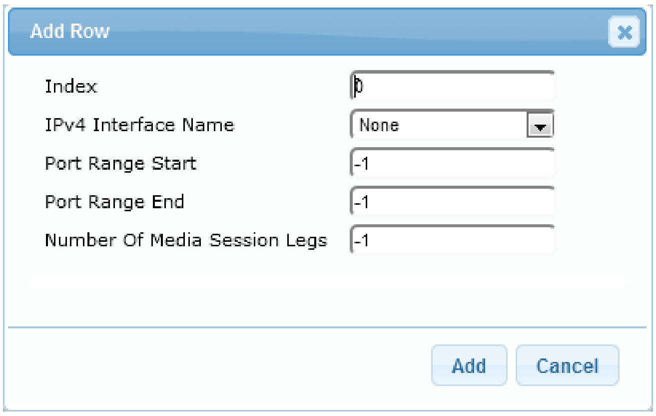

The following procedure describes how to configure Media Realm Extensions through the Web interface. You can also configure it through ini file (MediaRealmExtension).

➤ **To configure a Media Realm Extension:**

1. Open the Media Realm table (see "Configuring Media Realms" on page 303).
2. Select the Media Realm row for which you want to add Remote Media Extensions, and then click the **Media Realm Extension** link located below the table; the Media Realm Extension table appears.

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

**Figure 17-5: Media Realm Extension Table - Add Row Dialog Box**




4. Configure the Media Realm Extension according to the parameters described in the table below.
5. Click **Add**.

**Table 17-3: Media Realm Extension Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[MediaRealmExtension_ExtensionIndex] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| IPv4 Interface Name<br>[MediaRealmExtension_IPv4IF] | Assigns an IPv4 network interface (configured in the Interface table) to the Media Realm Extension.<br>By default, no value is defined (**None**).<br>For configuring IP network interfaces, see "Configuring IP Network Interfaces" on page 135.<br>**Note:**<br>▪ The parameter is mandatory.<br>▪ You must configure the Media Realm Extension with an IP network interface that has the same IP version(s) as the Media Realm to which the Media Realm Extension is associated. If the associated Media Realm is assigned both an IPv4 and IPv6 network interface, you also need to assign the Media Realm Extension with both an IPv4 and IPv6 network interface. For example, if the associated Media Realm is assigned only an IPv4 network interface, you also need to assign the Media Realm Extension with an IPv4 network interface. |

| Parameter | Description |
|---|---|
| IPv6 Interface Name<br>[MediaRealmExtension_IPv6IF] | Assigns an IPv6 network interface (configured in the Interface table) to the Media Realm Extension.<br>By default, no value is defined (None).<br>**Note:**<br>▪ The parameter is mandatory.<br>▪ You must configure the Media Realm Extension with an IP network interface that has the same IP version(s) as the Media Realm to which the Media Realm Extension is associated. If the associated Media Realm is assigned both an IPv4 and IPv6 network interface, you also need to assign the Media Realm Extension with both an IPv4 and IPv6 network interface. For example, if the associated Media Realm is assigned only an IPv6 network interface, you also need to assign the Media Realm Extension with an IPv6 network interface. |
| Port Range Start<br>[MediaRealmExtension_PortRangeStart] | Defines the first (lower) port in the range of media UDP ports for the Media Realm Extension.<br>By default, no value is defined.<br>**Notes:**<br>▪ You must either configure all your Media Realms with port ranges or all without; not some with and some without.<br>▪ The available UDP port range is according to the BaseUDPport parameter. For more information, see "Configuring RTP Base UDP Port" on page 197. |
| Port Range End<br>[MediaRealmExtension_PortRangeEnd] | Defines the last (upper) port in the range of media UDP ports for the Media Realm Extension.<br>**Note:** It is unnecessary to configure the parameter. The device automatically populates the parameter with a value, calculated by the summation of the 'Number of Media Session Legs' parameter (multiplied by the port chunk size) and the 'Port Range Start' parameter. After you have added the Media Realm Extension row to the table, the parameter is displayed with the calculated value. |
| Number Of Media Session Legs<br>[MediaRealmExtension_MediaSessionLeg] | Defines the number of media sessions for the port range. For example, 100 ports correspond to 10 media sessions, since ports are allocated in chunks of 10.<br>By default, no value is defined.<br>**Note:** The parameter is mandatory. |

## 17.2 Configuring SRDs

The SRD table lets you configure up to 41 signaling routing domains (SRD). The SRD is a logical representation of an entire SIP-based VoIP network (Layer 5) consisting of groups of SIP users and servers. The SRD is associated with all the configuration entities (e.g., SIP Interfaces and IP Groups) required for routing calls within the network. Typically, only a

**single** SRD is required (recommended) for most deployments. Multiple SRDs are only required for multi-tenant deployments, where the physical device is "split" into multiple logical devices. For more information on multi-tenant architecture, see "Multiple SRDs for Multi-tenant Deployments" on page 315.

As the device is shipped with a default SRD ("DefaultSRD" at Index 0), if your deployment requires only one SRD, you can use the default SRD instead of creating a new one. When only one SRD is employed and you create other related configuration entities (e.g., SIP Interfaces), the default SRD is automatically assigned to the new configuration entity. Therefore, when employing a single-SRD configuration topology, there is no need to handle SRD configuration (i.e., transparent).

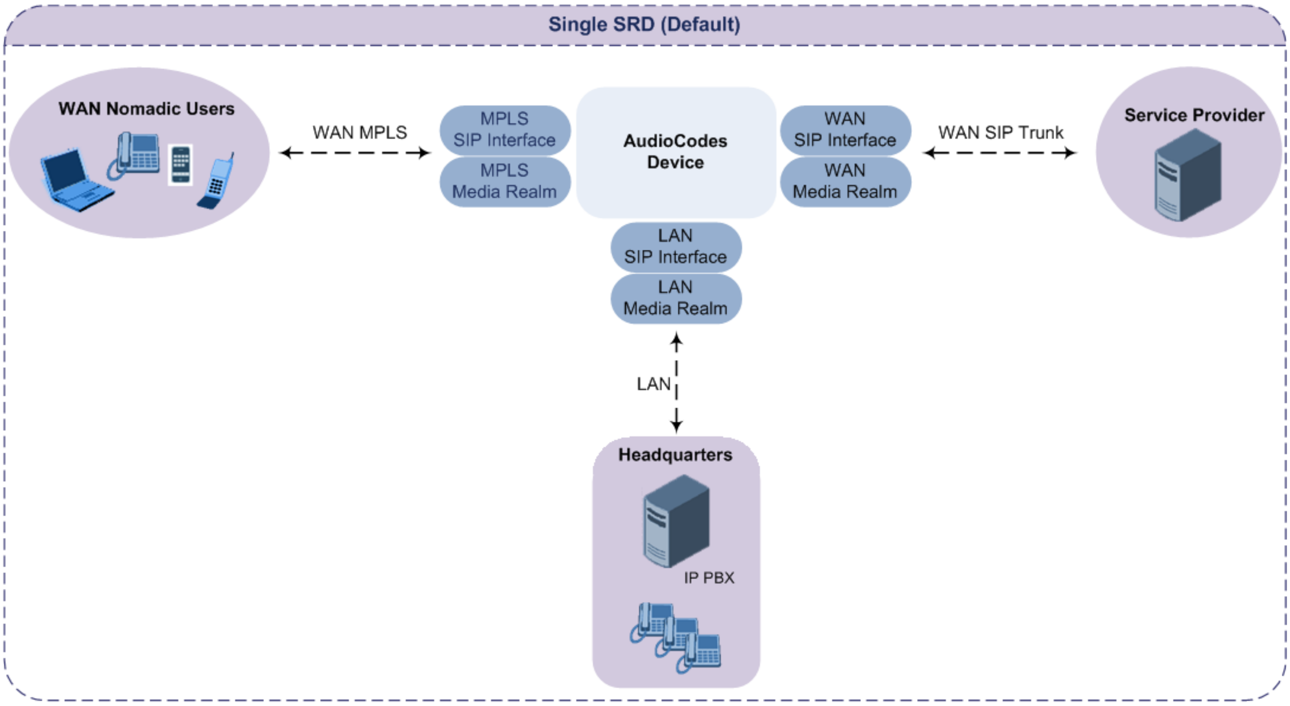
SRDs are associated with the following configuration entities:

- SIP Interface (mandatory) - see "Configuring SIP Interfaces" on page 319
- IP Group (mandatory) - see "Configuring IP Groups" on page 323
- Proxy Set (mandatory) - see "Configuring Proxy Sets" on page 329

As mentioned previously, if you use only a single SRD, the device automatically assigns it to the above-listed configuration entities.

As each SIP Interface defines a different Layer-3 network (see "Configuring SIP Interfaces" on page 319 for more information) on which to route or receive calls and as you can assign multiple SIP Interfaces to the same SRD, for most deployment scenarios (even for multiple Layer-3 network environments), you only need to employ a single SRD to represent your VoIP network (Layer 5). For example, if your VoIP deployment consists of an Enterprise IP PBX (LAN), a SIP Trunk (WAN), and far-end users (WAN), you would only need a single SRD. The single SRD would be assigned to three different SIP Interfaces, where each SIP Interface would represent a specific Layer-3 network (IP PBX, SIP Trunk, or far-end users) in your environment. The following figure provides an example of such a deployment:

**Figure 17-6: Deployment using a Single SRD**



**Notes:**

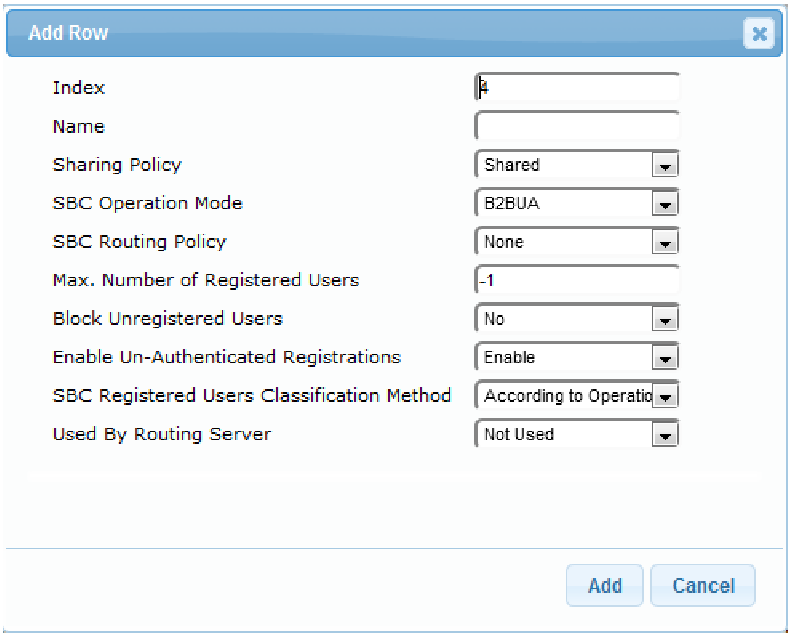
- It is recommended to use a single-SRD configuration topology, unless you are deploying the device in a multi-tenant environment, in which case, multiple SRDs are required.
- Each SIP Interface, Proxy Set, and IP Group can be associated with only one SRD.

The following procedure describes how to configure SRDs through the Web interface. You can also configure it through ini file (SRD) or CLI (configure voip > voip-network srd).

➢ **To configure an SRD:**

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

**Figure 17-7: SRD Table - Add Row Dialog Box**




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

**Table 17-4: SRD Table Parameter Descriptions**

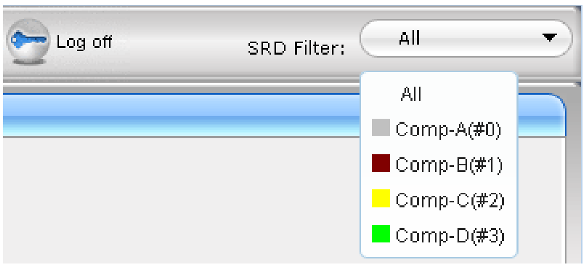
| Parameter | Description |
|---|---|
| Index<br>[SRD_Index] | Defines an index for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>`name`<br>[SRD_Name] | Defines an arbitrary name to easily identify the row.<br>The valid value can be a string of up to 40 characters.<br>**Notes:**<br>▪ The parameter is mandatory.<br>▪ Each row must be configured with a unique name. |

| Parameter | Description |
|---|---|
| Sharing Policy<br>`type`<br>[SRD_SharingPolicy] | Defines the sharing policy of the SRD, which determines whether the SRD shares its SIP resources (SIP Interfaces, Proxy Sets, and IP Groups) with all other SRDs (Shared and Isolated).<br>▪ [0] Shared = (Default) SRD shares its resources with other SRDs (Isolated and Shared) and calls can thus be routed between the SRD and other SRDs.<br>▪ [1] Isolated = SRD does not share its resources with other SRDs and calls cannot be routed between the SRD and other Isolated SRDs. However, calls can be routed between the SRD and other Shared SRDs.<br>For more information on SRD Sharing Policy, see Multiple SRDs for Multi-tenant Deployments on page 315. |
| Max. Number of Registered Users<br>`max-reg-users`<br>[SRD_MaxNumOfRegUsers] | Defines the maximum number of users belonging to the SRD that can register with the device.<br>The default is -1, which means that the number of allowed user registrations is unlimited. |
| Used By Routing Server<br>`used-by-routing-server`<br>[SIPInterface_UsedByRoutingServer] | Enables the SRD to be used by a third-party routing server for call routing decisions.<br>▪ [0] Not Used (default)<br>▪ [1] Used<br>For more information on the third-party routing server feature, see "Centralized Third-Party Routing Server" on page 272. |

## 17.2.1 Filtering Tables in Web Interface by SRD

When your configuration includes multiple SRDs, you can filter tables in the Web interface by a specific SRD. The filter is configured in the SRD Filter drop-down list, located on the Web interface's toolbar. The filter is applied throughout the GUI. The following figure shows the SRD Filter with an example of configured SRDs in its' drop-down list.

**Figure 17-8: SRD Filter**




When you select an SRD for filtering, the Web interface displays only table rows associated with the filtered SRD. In addition, if you add a new row to a table, the filtered SRD is automatically selected as the associated SRD (in the 'SRD' parameter of the Add Row dialog box). For example, if your SRD filter is set to "Comp-A" and you then add a new Proxy Set, the Proxy Set is automatically associated with SRD "Comp-A" (i.e., the 'SRD' parameter is set to "Comp-A"). All other parameters in the Add Row dialog box are also automatically set to values associated with the filtered SRD.

SRD filtering is especially useful in multi-tenant setups where multiple SRDs may be configured. In such a setup, SRD filtering eliminates configuration clutter by "hiding" SRDs that are irrelevant to the current configuration and facilitates configuration by automatically associating the filtered SRD, and other configuration elements associated with the filtered SRD, wherever applicable.

### 17.2.2 Multiple SRDs for Multi-tenant Deployments

The device can be deployed in a multi-tenant architecture, serving multiple customers (tenants) from a single, shared physical entity. The device's multi-tenant feature is fully scalable, offering almost “non-bleeding” partition per tenant, whereby users of one tenant can’t infringe on the space of users of another tenant. The device provides per tenant configuration, monitoring, reporting, analytics, alarms and interfacing. The device is a real-time multi-tenant system that provides each tenant with optimal real-time performance, as each session received by the device is classified and processed only through the tenant’s “orbit”.

While some enterprises are large enough to justify a dedicated standalone device, many enterprises require only a fraction of the device's capacity and capabilities. Service providers offering SIP Trunking services can funnel multiple enterprises into a single device and thereby, reap significant cost improvements over a device-per-customer model. Tenant size in a multi-tenant architecture can vary and therefore, the instance CPU, memory and interface allocations should be optimized so as not to waste resources for small-sized tenants on the one hand, and not to allocate too many instances for a single tenant/customer on the other. For example, it would be a waste to allocate a capacity of 100 concurrent sessions to a small tenant for which 10 concurrent sessions suffice.

In a multi-tenant deployment, each tenant is represented by a dedicated SRD. The different Layer-3 networks (e.g., LAN IP-PBX users, WAN SIP Trunk, and WAN far-end users) of the tenant are represented by SIP Interfaces, which are all associated with the tenant's SRD. As related configuration entities (SIP Interfaces, IP Groups, Proxy Sets) are associated with the specific SRD, each SRD has its own logically separated configuration tables (although configured in the same tables). Therefore, full logical separation (on the SIP application layer) between tenants is achieved by SRD.

To create a multi-tenant configuration topology that is as non-bleeding as possible, you can configure an SRD (tenant) as *Isolated* and *Shared*:

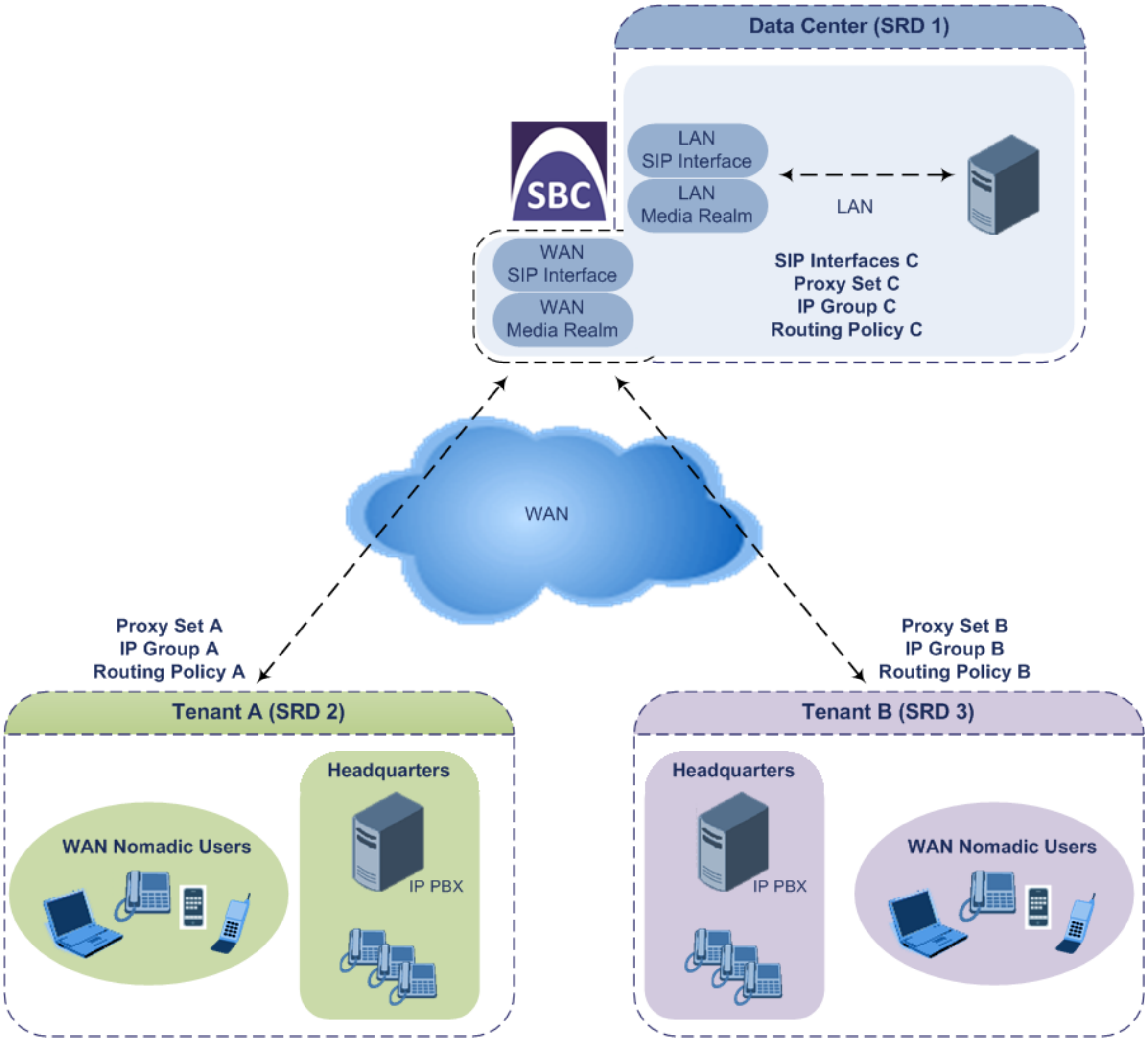
- **Isolated SRD:** An Isolated SRD has its own dedicated SIP resources (SIP Interfaces, Proxy Sets, and IP Groups). No other SRD can use the SIP resources of an Isolated SRD. Thus, call traffic of an Isolated SRD is kept separate from other SRDs (tenants), preventing any risk of traffic "leakage" with other SRDs.

  Isolated SRDs are more relevant when each tenant needs its own separate (dedicated) routing "table" for non-bleeding topology. Separate routing tables are implemented using Routing Policies. In such a non-bleeding topology, routing between Isolated SRDs is not possible. This enables accurate and precise routing per SRD, eliminating any possibility of erroneous call routing between SRDs, restricting routing to each tenant's (SRD's) sphere. Configuring only one Routing Policy that is shared between Isolated SRDs is not best practice for non-bleeding environments, since it allows routing between these SRDs.

- **Shared SRD:** Isolated SRDs have their own dedicated SIP resources (SIP Interfaces, Proxy Sets, and IP Groups). This may not be possible in some deployments. For example, in deployments where all tenants use the same SIP Trunking service, or use the same SIP Interface due to limited SIP interface resources (e.g., multiple IP addresses cannot be allocated and SIP port 5060 must be used). In contrast to Isolated SRDs, a Shared SRD can share its' SIP resources with all other SRDs (Shared and Isolated). This is typically required when tenants need to use common resources. In the SIP Trunk example, the SIP Trunk would be associated with a Shared SRD, enabling all tenants to route calls with the SIP Trunk.

Another configuration entity that can be used for multi-tenant deployments is the Routing Policy. Routing Policies allow each SRD (or tenant) to have its own routing rules, manipulation rules, Least Cost Routing (LCR) rules, and/or LDAP-based routing configuration. However, not all multi-tenant deployments need multiple Routing Policies and typically, their configuration is not required. Isolated SRDs are more relevant only when each tenant requires its own dedicated Routing Policy to create separate, dedicated routing "tables"; for all other scenarios, SRDs can be Shared.

The figure below illustrates a multi-tenant architecture with Isolated SRD tenants ("A" and "B") and a Shared SRD tenant ("Data Center") serving as a SIP Trunk:




To facilitate multi-tenant configuration through CLI, you can access a specific tenant "view". Once in a specific tenant view, all configuration commands apply only to the currently viewed tenant. Only table rows (indexes) belonging to the viewed tenant can be modified. New table rows are automatically associated with the viewed tenant (i.e., SRD name). The display of tables and show running-configuration commands display only rows relevant to the viewed tenant (and shared tenants). The show commands display only information relevant to the viewed tenant. To support this CLI functionality, use the following commands:

- To access a specific tenant view:

  ```
  # srd-view <SRD name>
  ```

  Once accessed, the tenant's name (i.e., SRD name) forms part of the CLI prompt, for example:

  ```
  # srd-view datacenter
  (srd-datacenter)#
  ```

- To exit the tenant view:

  ```
  # no srd-view
  ```

### 17.2.3 Cloning SRDs

You can clone (duplicate) existing SRDs. This is especially useful when operating in a multi-tenant environment and you need to add new tenants (SRDs). The new tenants can quickly and easily be added by simply cloning one of the existing SRDs. Once cloned, all you need to do is tweak configuration entities associated with the SRD clone.

When an SRD is cloned, the device adds the new SRD clone to the next available index row in the SRD table. The SRD clone is assigned a unique name in the following syntax format: <unique ID>_<original SRD index>_CopyOf_<name or index if no name of original SRD>. For example, if you clone SRD "SIP-Trunk" at index 2, the new SRD clone is assigned the name, "36454371_2_CopyOf_SIP-Trunk".

The SRD clone has identical settings as the original SRD. In addition, all configuration entities associated with the original SRD are also cloned and these clones are associated with the SRD clone. The naming convention of these entities is the same as the SRD clones (see above) and all have the same unique clone ID ("36454371" in the example above) as the cloned SRD. These configuration entities include IP Groups, SIP Interfaces, and Proxy Sets (without addresses). If the Routing Policy associated with the original SRD is not associated with any other SRD, the Routing Policy is also cloned and its' clone is associated with the SRD clone. All configuration entities associated with the original Routing Policy are also cloned and these clones are associated with the Routing Policy clone.

When any configuration entity is cloned as a result of a cloned SRD, all fields of the entity's row which "point" to other entities (e.g., SIP Interface, Source IP Group, and Destination IP Group) are replaced by their corresponding clones.

**Note:** For some cloned entities such as SIP Interfaces, some parameter values may change. This occurs in order to avoid the same parameter having the same value in more than one table row (index), which would result in invalid configuration. For example, a SIP Interface clone will have an empty Network Interface setting. After the clone process finishes, you thus need to update the Network Interface for valid configuration.

➢ **To clone an SRD:**

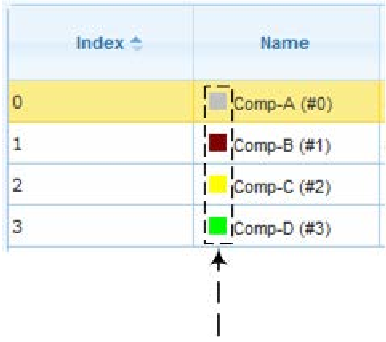
- Web interface: In the SRD table, select an SRD to clone, and then click the **Clone** button.
- CLI:

```
(config-voip)# voip-network srd clone <SRD index that you want cloned>
```

### 17.2.4 Color-Coding of SRDs in Web Interface

To easily identify your configured SRDs, the Web interface displays each SRD in a unique color. The color is automatically and randomly assigned to new SRDs, and is displayed in a box alongside the name of the SRD, in tables where the SRD is configured or assigned. This is applied throughout the Web interface's GUI. The following example shows SRDs assigned with unique color codes.

**Figure 17-9: Color-Coding of SRDs**



| Index | Name |
|---|---|
| 0 | Comp-A (#0) |
| 1 | Comp-B (#1) |
| 2 | Comp-C (#2) |
| 3 | Comp-D (#3) |

## 17.2.5 Automatic Configuration based on SRD

To facilitate configuration and eliminate possible flaws in configuration due to invalid associations between configuration entities, the Web interface automatically configures configuration entities based on SRD:

- If you delete an SRD (in the SRD table) that is associated with other configuration entities in other tables, the device automatically deletes the associated table rows. For example, if you delete an SRD that is associated with a Proxy Set, the device automatically deletes the Proxy Set.
- If you associate an SRD with a configuration entity in another table (i.e., other than the SRD table), the device automatically configures certain parameters of the configuration entity according to the SRD or associated SRD.
- If your configuration setup includes only a single SRD, the device automatically selects the SRD when adding related configuration entities. For example, when adding an IP Group, the single SRD is automatically selected in the Add Row dialog box.

## 17.3 Configuring SIP Interfaces

The SIP Interface table lets you configure up to 82 SIP Interfaces. A SIP Interface represents a Layer-3 network in your deployment environment, by defining a local, listening port number and type (e.g., UDP), and assigning an IP network interface for SIP signaling traffic. For example, if your deployment consists of an IP PBX in the LAN, a SIP Trunk in the WAN, and remote far-end users in the WAN, you would need to configure a SIP Interface for each of these SIP entities. SIP Interfaces are required for Tel-to-IP and IP-to-Tel calls. You can also configure various optional features for the SIP Interface such as assigning it a Media Realm, and blocking calls received on the SIP Interface from users not registered with the device.

Each SIP Interface can be associated with only one SRD. As the SRD configuration entity represents your VoIP deployment SIP network (Layer 5), you need to associate your SIP Interfaces with a specific SRD in order to represent your Layer-3 networks. For most deployments (except multi-tenant deployments), your SRD represents your entire network and thus, only one SRD is required. The device provides a default SRD and in such scenarios where only a single SRD is required, your SIP Interfaces are automatically assigned to the default SRD. Therefore, there is no need to even handle SRD configuration entity.

Once configured, you can apply SIP Interfaces to calls, by assigning them to the following configuration entities in their respective tables:

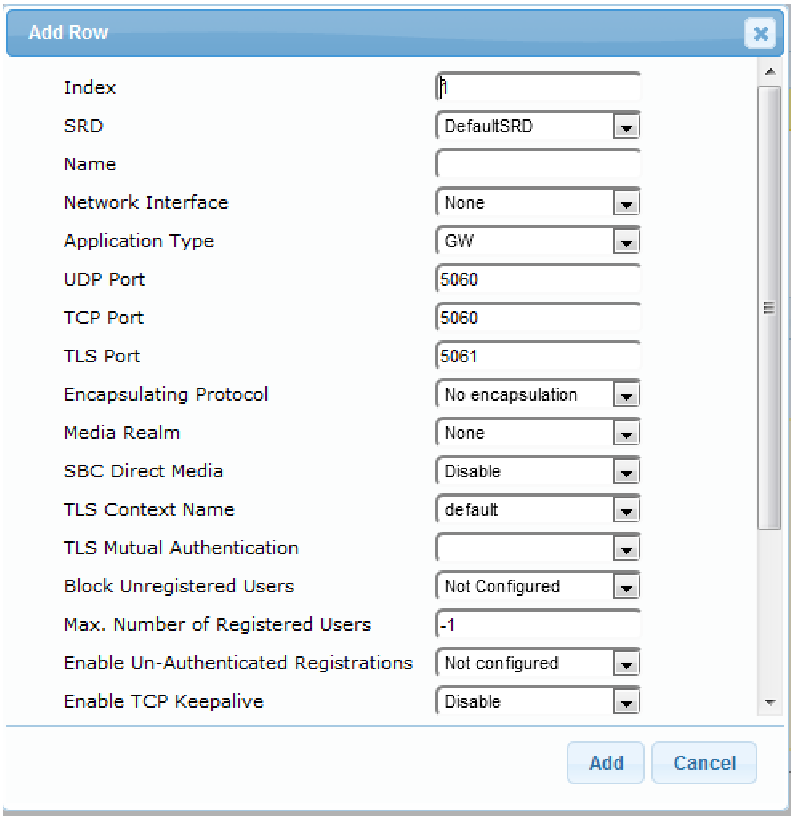
- (Mandatory) Proxy Set to specify the SIP Interface for communication with the proxy server (i.e., IP Group). For more information, see ''Configuring Proxy Sets'' on page 329.
- Tel-to-IP Routing rules for specifying the destination SIP Interface to where you want to route Tel-to-IP calls. For more information, see Configuring Tel-to-IP Routing Rules on page 409.
- IP-to-Hunt Group Routing rules for specifying the SIP Interface as a matching characteristics for the incoming IP call.
- Intrusion Detection System (IDS) for applying the IDS policy to a specific SIP Interface. For more information, see ''Configuring IDS Policies'' on page 168.

> **Note:** The device terminates active calls associated with a SIP Interface in the following scenarios:
>
> - If you delete the associated SIP Interface.
> - If you edit any of the following fields of the associated SIP Interface: 'Application Type', 'UDP Port, 'TCP Port', 'TLS Port' or 'SRD' fields.
> - If you edit or delete a network interface in the Interface table that is associated with the SIP Interface.

The following procedure describes how to configure SIP interfaces through the Web interface. You can also configure it through ini file (SIPInterface) or CLI (configure voip > voip-network sip-interface).

- **To configure a SIP Interface:**

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




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

**Table 17-5: SIP Interface Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[SIPInterface_Index] | Defines an index for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| SRD<br>`srd`<br>[SIPInterface_SRDName] | Assigns an SRD to the SIP Interface.<br>If only one SRD is configured in the SRD table, the SRD is assigned to the SIP Interface by default. If multiple SRDs are configured in the SRD table, no value is defined.<br>For configuring SRDs, see "Configuring SRDs" on page 311.<br>**Notes:**<br>▪ The parameter is mandatory.<br>▪ You can assign the same SRD to multiple SIP Interfaces. |
| Name<br>`interface-name`<br>[SIPInterface_InterfaceName] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters. By default, if you do not configure a name, the device automatically assigns the name "SIPInterface_<row index>" (e.g., "SIPInterface_1" when added to Index 1). |
| Network Interface<br>`network-interface`<br>[SIPInterface_NetworkInterface] | Assigns a Control-type IP network interface to the SIP Interface.<br>By default, no value is defined (**None**).<br>For configuring network interfaces, see "Configuring IP Network Interfaces" on page 135. |

| Parameter | Description |
|---|---|
| | **Note:** The parameter is mandatory. |
| Application Type<br>`application-type`<br>[SIPInterface_ApplicationType] | Defines the application for which the SIP Interface is used.<br>▪ [0] GW = (Default) Gateway application. |
| UDP Port<br>`udp-port`<br>[SIPInterface_UDPPort] | Defines the device's listening and source port for SIP signaling traffic over UDP.<br>The valid range is 1 to 65534. The default is 5060.<br>**Notes:**<br>▪ The port **must** be different from ports configured for RTP traffic (i.e., ports configured for Media Realms). For example, if the RTP port range is 6000 to 6999, the SIP port can either be less than 6000 or greater than 6999.<br>▪ The base UDP port number (BaseUDPPort parameter) for RTP traffic must be greater than the highest UDP port configured for a SIP Interface. For example, if your highest configured UDP port for a SIP Interface is 6060, you must configure the BaseUDPPort parameter to any value greater than 6060. For more information on base UDP port, see Configuring RTP Base UDP Port on page 197.<br>▪ Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping). |
| TCP Port<br>`tcp-port`<br>[SIPInterface_TCPPort] | Defines the device's listening port for SIP signaling traffic over TCP.<br>The valid range is 1 to 65534. The default is 5060.<br>**Notes:**<br>▪ The port **must** be different from ports configured for RTP traffic (i.e., ports configured for Media Realms). For example, if the RTP port range is 6000 to 6999, the SIP port can either be less than 6000 or greater than 6999.<br>▪ Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping). |
| TLS Port<br>`tls-port`<br>[SIPInterface_TLSPort] | Defines the device's listening port for SIP signaling traffic over TLS.<br>The valid range is 1 to 65534. The default is 5061.<br>**Notes:**<br>▪ The port **must** be different from ports configured for RTP traffic (i.e., ports configured for Media Realms). For example, if the RTP port range is 6000 to 6999, the SIP port can either be less than 6000 or greater than 6999.<br>▪ Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping). |
| Media Realm<br>`media-realm-name`<br>[SIPInterface_MediaRealm] | Assigns a Media Realm to the SIP Interface.<br>By default, no value is defined (**None**).<br>For configuring Media Realms, see "Configuring Media Realms" on page 303. |

| Parameter | Description |
|---|---|
| | **Note:** If you later delete the assigned Media Realm in the Media Realm table, this value becomes **None**. |
| TLS Context Name<br>`tls-context-name`<br>[SIPInterface_TLSContext] | Assigns a TLS Context (SSL/TLS certificate) to the SIP Interface.<br>The default TLS Context ("default" at Index 0) is assigned to the SIP Interface by default.<br>**Notes:**<br>▪ For incoming calls: The assigned TLS Context is used if no TLS Context is configured for the Proxy Set associated with the call or classification to an IP Group based on Proxy Set fails.<br>▪ For outgoing calls: The assigned TLS Context is used if no TLS Context is configured for the Proxy Set associated with the call.<br>For more information on TLS Contexts, see "Configuring SSL/TLS Certificates" on page 103. |
| TLS Mutual Authentication<br>`tls-mutual-auth`<br>[SIPInterface_TLSMutualAuthentication] | Enables TLS mutual authentication for the SIP Interface (when the device acts as a server).<br>▪ **[-1]** Not Configured = (Default) The SIPSRequireClientCertificate global parameter setting is applied.<br>▪ **[0]** Disable = Device does not request the client certificate for TLS connection on the SIP Interface.<br>▪ **[1]** Enable = Device requires receipt and verification of the client certificate to establish the TLS connection on the SIP Interface. |
| Block Unregistered Users<br>`block-un-reg-users`<br>[SIPInterface_BlockUnRegUsers] | Enables the device to block (reject) incoming calls (INVITE requests) from unregistered users belonging to the SIP Interface.<br>▪ [-1] Not Configured = (Default) The corresponding parameter in the SRD table (SRD_BlockUnRegUsers) of the SRD that is associated with the SIP Interface is applied.<br>▪ [0] No = Calls from unregistered users are allowed.<br>▪ [1] Yes = Calls from unregistered users are blocked.<br>**Notes:**<br>▪ When the device blocks a call, it sends a SIP 500 "Server Internal Error" response to the remote end.<br>▪ The parameter applies to calls belonging to a User-type IP Group.<br>▪ If configured to Yes or No, the parameter overrides the 'Block Unregistered Users' parameter of the associated SRD in the SRD table. |
| Enable TCP Keepalive<br>`tcp-keepalive-enable`<br>[SIPInterface_TCPKeepaliveEnable] | Enables the TCP Keep-Alive mechanism with the IP entity on this SIP Interface. TCP keep-alive can be used, for example, to keep a NAT entry open for clients located behind a NAT server, or simply to check that the connection to the IP entity is available.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Note:** For configuring TCP keepalive, use the following ini file parameters: TCPKeepAliveTime, TCPKeepAliveInterval, and TCPKeepAliveRetry. |
| Message Policy<br>`message-policy`<br>[SIPInterface_MessagePolicyName] | Assigns a SIP message policy to the SIP interface.<br>For configuring SIP Message Policy rules, see "Configuring SIP Message Policy Rules". |

| Parameter | Description |
|---|---|
| Used By Routing Server<br>`used-by-routing-server`<br>[SIPInterface_UsedByRouting Server] | Enables the SIP Interface to be used by a third-party routing server for call routing decisions.<br>▪ [0] Not Used (default)<br>▪ [1] Used<br>For more information on the third-party routing server feature, see Centralized Third-Party Routing Server on page 272. |

## 17.4 Configuring IP Groups

The IP Group table lets you configure up to 102 IP Groups. An IP Group represents a SIP entity in the network with which the device communicates. This can be a server (e.g., IP PBX or ITSP) or a group of users (e.g., LAN IP phones). For servers, the IP Group is typically used to define the server's IP address by associating it with a Proxy Set (see Configuring Proxy Sets on page 329).

You can use IP Groups for the following:

- SIP dialog registration and authentication (digest user/password) of specific IP Groups (Served IP Group, e.g., corporate IP-PBX) with other IP Groups (Serving IP Group, e.g., ITSP). This is configured in the Account table (see "Configuring Registration Accounts" on page 341).
- Call routing rules:
  - Tel-to-IP calls: The IP Group is used as the destination of the outgoing IP call and is used in Tel-to-IP call routing rules (see Configuring Tel-to-IP Routing Rules on page 409).
  - IP-to-Tel calls: The IP Group identifies the source of the IP call and is used in IP-to-Tel call routing rules (see Configuring IP-to-Hunt Group Routing Rules on page 421).
  - Number manipulation: The IP Group can be associated with a number manipulation rule (see Configuring Number Manipulation Tables on page 393).
- Included in routing decisions by a third-party routing server. If deemed necessary for routing, the routing server can even create an IP Group. For more information, see Centralized Third-Party Routing Server on page 272.

You can also apply the device's Quality of Experience feature to IP Groups:

- **Quality of Experience Profile:** Call quality monitoring based on thresholds for voice metrics (e.g., MOS) can be applied per IP Group. For example, if MOS is considered poor, calls belonging to this IP Group can be rejected. For configuring Quality of Experience Profiles, see "Configuring Quality of Experience Profiles" on page 293.
- **Bandwidth Profile:** Bandwidth utilization thresholds can be applied per IP Group. For example, if bandwidth thresholds are crossed, the device can reject any new calls on this IP Group. For configuring Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 297.

**Note:**

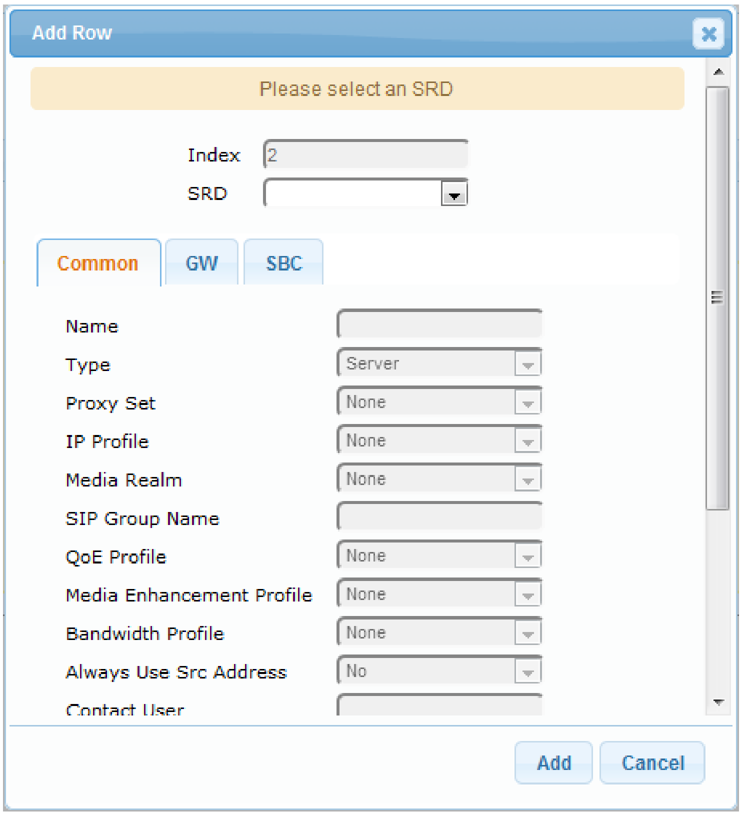
- IP Group ID 0 cannot be associated with Proxy Set ID 0.
- If you delete an IP Group or modify the 'Type' or 'SRD' parameters, the device immediately terminates currently active calls associated with the IP Group. In addition, all users belonging to this IP Group are removed from the device's users database.

The following procedure describes how to configure IP Groups through the Web interface. You can also configure it through ini file (IPGroup) or CLI (configure voip > control-network ip-group).

### ➢ To configure an IP Group:

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

**Figure 17-10: IP Group Table - Add Row Dialog Box**




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

**Table 17-6: IP Group Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| **Common Parameters** | |
| Index<br>[IPGroup_Index] | Defines an index for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| SRD<br>`srd-name`<br>[IPGroup_SRDName] | Assigns an SRD to the IP Group.<br>If only one SRD is configured in the SRD table, the SRD is assigned by default. If multiple SRDs are configured in the SRD table, no value is assigned by default.<br>For configuring SRDs, see Configuring SRDs on page 311.<br>**Notes:**<br>▪ The parameter is mandatory. |

| Parameter | Description |
|---|---|
| | ▪ For the parameter to take effect, a device reset is required. |
| Name<br>`name`<br>[IPGroup_Name] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters.<br>**Note:** Each row must be configured with a unique name. |
| Type<br>`type`<br>[IPGroup_Type] | Defines the type of IP Group:<br>▪ [0] Server = Applicable when the destination address of the IP Group (e.g., ITSP, Proxy, IP-PBX, or Application server) is known. The address is configured by the Proxy Set that is associated with the IP Group.<br>▪ [1] User = Represents a group of users such as IP phones and softphones where their location is dynamically obtained by the device when REGISTER requests and responses traverse (or are terminated) by the device. These users are considered remote (far-end).<br>Typically, this IP Group is configured with a Serving IP Group that represents an IP-PBX, Application or Proxy server that serves this User-type IP Group. Each SIP request sent by a user of this IP Group is proxied to the Serving IP Group. For registrations, the device updates its registration database with the AOR and contacts of the users.<br>Digest authentication using SIP 401/407 responses (if needed) is performed by the Serving IP Group. The device forwards these responses directly to the SIP users.<br>To route a call to a registered user, a rule must be configured in the Tel-to-IP Routing table. The device searches the dynamic database (by using the Request-URI) for an entry that matches a registered AOR or Contact. Once an entry is found, the IP destination is obtained from this entry and a SIP request is sent to the destination.<br>The device also supports NAT traversal for the SIP clients located behind NAT. In this case, the device must be defined with a global IP address. |
| Proxy Set<br>`proxy-set-id`<br>[IPGroup_ProxySetName] | Assigns a Proxy Set to the IP Group. All INVITE messages destined to the IP Group are sent to the IP address configured for the Proxy Set.<br>For configuring Proxy Sets, see "Configuring Proxy Sets" on page 329.<br>**Notes:**<br>▪ IP Group ID 0 cannot be associated with Proxy Set ID 0.<br>▪ The Proxy Set must be associated with the same SRD as that assigned to the IP Group.<br>▪ You can assign the same Proxy Set to multiple IP Groups.<br>▪ Proxy Sets are applicable only to Sever-type IP Groups. |
| IP Profile<br>`ip-profile-name`<br>[IPGroup_ProfileName] | Assigns an IP Profile to the IP Group.<br>By default, no value is defined (**None**).<br>For configuring IP Profiles, see "Configuring IP Profiles" on page 366. |

| Parameter | Description |
|---|---|
| Media Realm Name<br>`media-realm-name`<br>[IPGroup_MediaRealm] | Assigns a Media Realm to the IP Group. The Media Realm determines the UDP port range and maximum sessions on a specific interface for media traffic associated with the IP Group.<br>By default, no value is defined (None).<br>For configuring Media Realms, see Configuring Media Realms on page 303.<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ If you delete a Media Realm from the Media Realm table that is assigned to the IP Group, the parameter value reverts to None. |
| SIP Group Name<br>`sip-group-name`<br>[IPGroup_SIPGroupName] | Defines the SIP Request-URI host name in INVITE and REGISTER messages sent to this IP Group, or the host name in the From header of INVITE messages received from this IP Group. In other words, it replaces the original host name. .<br>The valid value is a string of up to 100 characters. By default, no value is defined.<br>**Notes:**<br>▪ If the parameter is not configured, the value of the global parameter, ProxyName is used instead (see "Configuring Proxy and Registration Parameters" on page 346).<br>▪ The parameter overrides inbound message manipulation rules that manipulate the host name in Request-URI, To, and/or From SIP headers. If you configure the parameter and you want to manipulate the host name in any of these SIP headers, you must apply your manipulation rule (Manipulation Set ID) to the IP Group as an Outbound Message Manipulation Set (see the IPGroup_OutboundManSet parameter), when the IP Group is the destination of the call. If you apply the Manipulation Set as an Inbound Message Manipulation Set (see the IPGroup_InboundManSet parameter), when the IP Group is the source of the call, the manipulation rule is overridden by the SIP Group Name parameter. |
| UUI Format<br>CLI: uui-format<br>**[IPGroup_UUIFormat]** | Enables the generation of the Avaya UCID value, adding it to the outgoing INVITE sent to this IP Group.<br>▪ [0] Disabled (default)<br>▪ [1] Enabled<br>This provides support for interworking with Avaya equipment by generating Avaya's UCID value in outgoing INVITE messages sent to Avaya's network. The device adds the UCID in the User-to-User SIP header.<br>Avaya's UCID value has the following format (in hexadecimal): 00 + FA + 08 + node ID (2 bytes) + sequence number (2 bytes) + timestamp (4 bytes)<br>This is interworked in to the SIP header as follows:<br>`User-to-User:`<br>`00FA080019001038F725B3;encoding=hex`<br>**Note:** To define the Network Node Identifier of the device for Avaya UCID, use the 'Network Node ID' (NetworkNodeId) parameter. |
| QoE Profile<br>`qoe-profile` | Assigns a Quality of Experience Profile rule.<br>By default, no value is defined (**None**). |

| Parameter | Description |
|---|---|
| [IPGroup_QOEProfile] | For configuring Quality of Experience Profiles, see "Configuring Quality of Experience Profiles" on page 293. |
| Media Enhancement Profile<br>`media-enhancement-profile`<br>[IPGroup_MediaEnhancementProfile] | Assigns a Media Enhancement Profile rule.<br>By default, no value is defined (**None**).<br>For configuring Media Enhancement Profiles, see "Configuring Media Enhancement Profiles" on page 300. |
| Bandwidth Profile<br>`bandwidth-profile`<br>[IPGroup_BWProfile] | Assigns a Bandwidth Profile rule.<br>By default, no value is defined (**None**).<br>For configuring Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 297. |
| Always Use Src Address<br>`always-use-source-addr`<br>[IPGroup_AlwaysUseSourceAddr] | Enables the device to always send SIP requests and responses, within a SIP dialog, to the source IP address received in the previous SIP message packet. This feature is especially useful in scenarios where the IP Group endpoints are located behind a NAT firewall (and the device is unable to identify this using its regular NAT mechanism).<br>▪ **[0]** No = (Default) The device sends SIP requests according to the settings of the global parameter, SIPNatDetection.<br>▪ **[1]** Yes = The device sends SIP requests and responses to the source IP address received in the previous SIP message packet.<br>For more information on NAT traversal, see "Remote UA behind NAT" on page 156. |
| Contact User<br>`contact-user`<br>[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.<br>By default, no value is defined.<br>**Notes:**<br>▪ The parameter is applicable only to Server-type IP Groups.<br>▪ The parameter is overridden by the 'Contact User' parameter in the Account table (see "Configuring Registration Accounts" on page 341). |
| Local Host Name<br>`local-host-name`<br>[IPGroup_ContactName] | Defines the host name (string) that the device uses in the SIP message's Via and Contact headers. This is typically used to define an FQDN as the host name. The device uses this string for Via and Contact headers in outgoing INVITE messages sent to a specific IP Group, and the Contact header in SIP 18x and 200 OK responses for incoming INVITE messages received from a specific IP Group. The Inbound IP Routing table can be used to identify the source IP Group from where the INVITE message was received.<br>If the parameter is not configured, these headers are populated with the device's dotted-decimal IP address of the network interface on which the message is sent.<br>By default, no value is defined.<br>**Note:** To ensure proper device handling, the parameter should be a valid FQDN. |

| Parameter | Description |
|---|---|
| Used By Routing Server<br>`used-by-routing-server`<br>[IPGroup_UsedByRoutingServer] | Enables the IP Group to be used by a third-party routing server for call routing decisions.<br>▪ [0] Not Used (default)<br>▪ [1] Used<br>For more information on the third-party routing server feature, see Centralized Third-Party Routing Server on page 272. |
| Created By Routing Server<br>[IPGroup_CreatedByRoutingServer] | (Read-only) Indicates whether the IP Group was created by a third-party routing server:<br>▪ [0] No<br>▪ [1] Yes<br>For more information on the third-party routing server feature, see Centralized Third-Party Routing Server on page 272. |
| GW Tab (Gateway Application) | |
| SIP Re-Routing Mode<br>`re-routing-mode`<br>[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).<br>▪ [-1] Not Configured (Default)<br>▪ [0] Standard = INVITE messages that are generated as a result of Transfer or Redirect are sent directly to the URI, according to the Refer-To header in the REFER message or Contact header in the 3xx response.<br>▪ [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.<br>▪ [2] Routing Table = Uses the Routing table to locate the destination and then sends a new INVITE to this destination.<br>**Notes:**<br>▪ When the 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].<br>▪ When the 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.<br>▪ When the parameter is set to [2], the XferPrefix parameter can be used to define different routing rules for redirected calls.<br>▪ The parameter is ignored if the parameter AlwaysSendToProxy is set to 1. |
| Always Use Route Table<br>`always-use-route-table`<br>[IPGroup_AlwaysUseRouteTable] | Defines the Request-URI host name in outgoing INVITE messages.<br>▪ [0] No (default).<br>▪ [1] Yes = The device uses the IP address (or domain name) defined in the Tel-to-IP Routing table (see Configuring the Tel to IP Routing on page 409) as the Request-URI host name in outgoing INVITE messages, instead of the value configured in the 'SIP Group Name' field.<br>**Note:** The parameter is applicable only to Server-type IP Groups. |
| **GW Group Status** | |

| Parameter | Description |
|---|---|
| GW Group Registered IP Address | (Read-only field) Displays the IP address of the IP Group entity (gateway) if registered with the device; otherwise, the field is blank.<br>**Note:** The field is applicable only to Gateway-type IP Groups (i.e., the 'Type' parameter is configured to Gateway). |
| GW Group Registered Status | (Read-only field) Displays whether the IP Group entity (gateway) is registered with the device ("Registered" or "Not Registered").<br>**Note:** The field is applicable only to Gateway-type IP Groups (i.e., the 'Type' parameter is configured to Gateway). |

## 17.5 Configuring Proxy Sets

The Proxy Sets table lets you configure up to 102 Proxy Sets. A Proxy Set defines the address and transport type (e.g., UDP or TCP) of a SIP server (e.g., SIP proxy and SIP registrar server). The Proxy Set represents the destination (address) of the IP Group configuration entity. Each Proxy Set can be configured with up to 10 addresses configured as an IP address and/or DNS host name (FQDN), enabling you to implement load balancing and redundancy (Proxy Hot-Swap feature) between multiple servers. If you configure the address as an FQDN, you can configure the method (A-record DNS, SRV, or NAPTR) for resolving the domain name to an IP address. The device supports up to 30 DNS-resolved IP addresses. (If the DNS resolution provides more than this number, the device uses the first 30 IP addresses in the received list and ignores the rest.) Each Proxy Set can be assigned a specific SSL/TLS certificate the (TLS Context), enabling you to use different TLS certificates per SIP entity (IP Group). In addition, each Proxy Set must be assigned a SIP Interface (and SRD), which determines, amongst others, the device's local network interface through which communication with the Proxy Set is done.

You can enable the device's keep-alive feature per Proxy Set, which determines whether proxies (addresses) configured for the Proxy Set are online or offline. If offline, the device will not route the call to the specific proxy. You can configure the device to send either SIP OPTIONS or REGISTER messages for the keep-alive. The keep-alive feature is required when using the proxy load-balancing or redundancy feature. For load-balancing, the device performs keep-alive on all proxies. For Parking-type redundancy, the device performs keep-alive only on the currently active proxy. For Homing-type redundancy, the device performs keep-alive on the current proxy as well as the "main" proxy. When using SIP OPTIONS, you can configure the device to consider the proxy as offline if specific SIP response codes are received from the keep-alive messages. To ensure that a previously offline proxy is now online, you can configure the number of required consecutive successful keep-alive messages (SIP OPTIONS only) before the device considers the proxy as being online. This mechanism avoids the scenario in which the device falsely detects a proxy as being online when it is actually offline, resulting in call routing failure. To view the connectivity status of Proxy Sets, see Viewing Proxy Set Status on page 567.

To use a configured Proxy Set, you need to assign it to an IP Group in the IP Group table (see "Configuring IP Groups" on page 323). When the device sends INVITE messages to an IP Group, it sends it to the address configured for the Proxy Set. You can assign the same Proxy Set to multiple IP Groups (belonging to the same SRD).

**Note:** You can view the device's connectivity status with proxy servers in the Tel-to-IP Routing table, for Tel-to-IP routing rules whose destination is an IP Group that is associated with a Proxy Set. The status is only displayed for Proxy Sets enabled with the Proxy Keep-Alive feature.

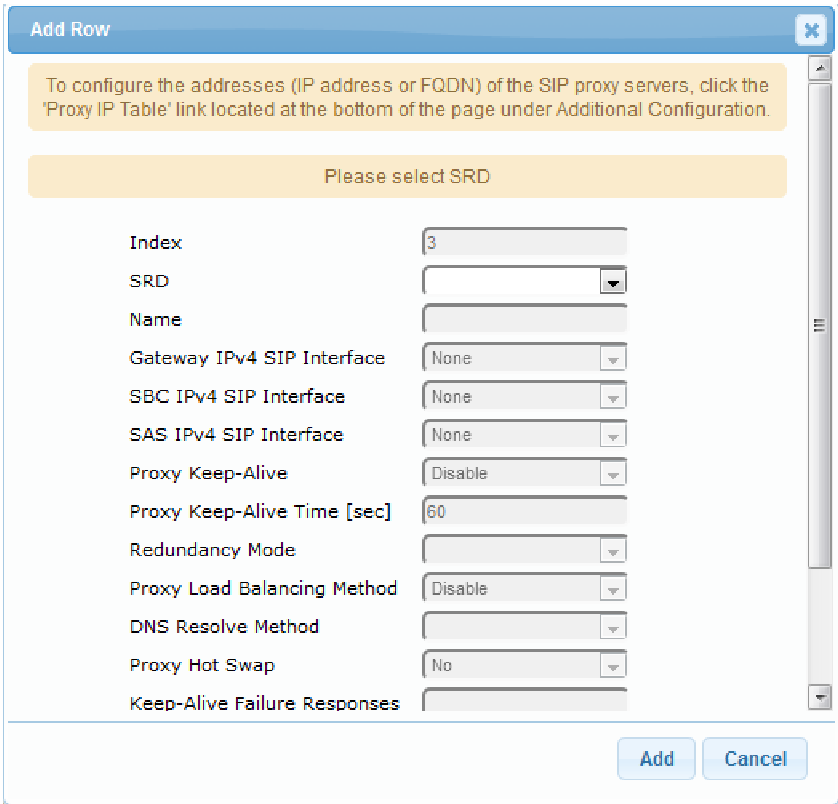
The Proxy Set is configured using two tables, one a "child" of the other:

- Proxy Sets table: Defines the attributes of the Proxy Set such as associated SIP Interface and redundancy features - ini file parameter, ProxySet or CLI command, configure voip > voip-network proxy-set
- Proxy Set Address table ("child"): Defines the addresses of the Proxy Set - table ini file parameter, ProxyIP or CLI command, configure voip > voip-network proxy-ip > proxy-set-id

### ➢ To configure a Proxy Set:

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

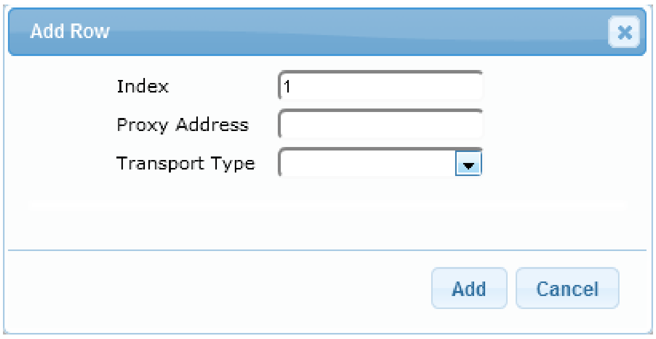
**Figure 17-11: Proxy Sets Table - Add Row Dialog Box**




3. Configure a Proxy Set according to the parameters described in the table below.
4. Click **Add**.
5. Select the index row of the Proxy Set that you added, and then click the **Proxy Address Table** link located below the table; the Proxy Address table opens.

6. Click **Add**; the following dialog box appears:

**Figure 17-12: Proxy Address Table - Add Row Dialog Box**




7. Configure the address of the Proxy Set according to the parameters described in the table below.
8. Click **Add**.

**Table 17-7: Proxy Sets Table and Proxy Address Table Parameter Description**

| Parameter | Description |
|---|---|
| **Proxy Sets Table** | |
| Index<br>`configure voip > voip-network proxy-set`<br>[ProxySet_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| SRD<br>`voip-network proxy-set > srd-id`<br>[ProxySet_SRDName] | Assigns an SRD to the Proxy Set.<br>**Notes:**<br>▪ The parameter is mandatory and must be configured first before you can configure the other parameters in the table.<br>▪ To configure SRDs, see Configuring SRDs on page 311. |
| Name<br>`proxy-name`<br>[ProxySet_ProxyName] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters.<br>**Note:**<br>▪ Each row must be configured with a unique name.<br>▪ The value cannot include a "/" forward slash. |
| Gateway IPv4 SIP Interface<br>`gwipv4-sip-int-name`<br>[ProxySet_GWIPv4SIPInterfaceName] | Assigns an IPv4-based SIP Interface for Gateway calls to the Proxy Set.<br>**Notes:**<br>▪ At least one SIP Interface must be assigned to the Proxy Set.<br>▪ The parameter appears only if you have configured a network interface with an IPv4 address in the Interface table (see Configuring IP Network Interfaces on page 135).<br>▪ To configure SIP Interfaces, see Configuring SIP Interfaces on page 319. |
| Gateway IPv6 SIP Interface<br>`gwipv6-sip-int-name`<br>[ProxySet_GWIPv6SIPInterfaceName] | Assigns an IPv6-based SIP Interface for Gateway calls to the Proxy Set.<br>**Notes:**<br>▪ At least one SIP Interface must be assigned to the Proxy Set.<br>▪ The parameter appears only if you have configured a network interface with an IPv6 address in the Interface table. |
| Proxy Keep-Alive<br>`proxy-enable-keep-alive`<br>[ProxySet_EnableProxyKeepAlive] | Enables the device's Proxy Keep-Alive feature, which checks communication with the proxy server.<br>▪ **[0]** Disable (default).<br>▪ **[1]** Using OPTIONS = Enables the Proxy Keep-Alive feature using SIP OPTIONS messages. The device sends an OPTIONS message every user-defined interval, configured by the 'Proxy Keep-Alive Time' parameter (in this table). If the device receives a SIP response code that is configured |

| Parameter | Description |
|---|---|
| | in the 'Keep-Alive Failure Responses' parameter (in this table), the device considers the proxy as down. You can also configure whether to use the device's IP address or string name ("gateway name") in the OPTIONS message (see the UseGatewayNameForOptions parameter).<br>▪ **[2]** Using REGISTER = Enables the Proxy Keep-Alive feature using SIP REGISTER messages. The device sends a REGISTER message every user-defined interval, configured by the RegistrationTime parameter. Any SIP response from the proxy - success (200 OK) or failure (4xx response) - is considered as if the proxy is "alive". If the proxy does not respond to INVITE messages sent by the device, the proxy is considered as down (offline).<br>**Notes:**<br>▪ Proxy keep-alive using REGISTER messages (**Using REGISTER** option) is applicable only to the Parking redundancy mode ('Redundancy Mode' parameter configured to **Parking**).<br>▪ For Survivability mode for User-type IP Groups, you must enable this Proxy Keep-Alive feature.<br>▪ If you enable this Proxy Keep-Alive feature and the proxy uses the TCP/TLS transport type, you can enable CRLF Keep-Alive feature, using the UsePingPongKeepAlive parameter.<br>▪ If you enable this Proxy Keep-Alive feature, the device can operate with multiple proxy servers (addresses) for redundancy and load balancing (see the 'Proxy Load Balancing Method' parameter). |
| Proxy Keep-Alive Time<br>`proxy-keep-alive-time`<br>[ProxySet_ProxyKeepAliveTime] | Defines the interval (in seconds) between keep-alive messages sent by the device when the Proxy Keep-Alive feature is enabled (see the 'Proxy Keep-Alive' parameter in this table).<br>The valid range is 5 to 2,000,000. The default is 60.<br>**Note:** The parameter is applicable only if the 'Proxy Keep-Alive' parameter is set to **Using Options**. |
| Success Detection Retries<br>success-detect-retries<br>[ProxySet_SuccessDetectionRetries] | Defines the minimum number of consecutive, successful keep-alive messages that the device sends to an offline proxy, before the device considers the proxy as being online.<br>The valid range is 1 to 10. The default is 1.<br>Note: The parameter is applicable only if the 'Proxy Keep-Alive' parameter is set to Using Options. |
| Success Detection Interval<br>success-detect-int<br>[ProxySet_SuccessDetectionInterval] | Defines the interval (in seconds) between each keep-alive retries (as configured by the 'Success Detection Retries' parameter) that the device performs for offline proxies.<br>The valid range is 1 to 30. The default is 10. |

| Parameter | Description |
|---|---|
| | Note: The parameter is applicable only if the 'Proxy Keep-Alive' parameter is set to Using Options. |
| Failure Detection Retransmissions<br>fail-detect-rtx<br>[ProxySet_FailureDetectionRetransmissions] | Defines the maximum number of UDP retransmissions that the device sends to an offline proxy, before the device considers the proxy as being offline.<br>The valid range is -1 to 255. The default is -1 (i.e., the settings of the global parameter SIPMaxRtxis applied).<br>Note: The parameter is applicable only if the 'Proxy Keep-Alive' parameter is set to Using Options. |
| Redundancy Mode<br>`proxy-redundancy-mode`<br>[ProxySet_ProxyRedundancyMode] | Determines whether the device switches from a redundant proxy to the primary proxy when the primary proxy becomes available again.<br>▪ **[-1]** Not configured = (Default) Proxy redundancy method is according to the settings of the global parameter, ProxyRedundancyMode.<br>▪ **[0]** Parking = The device continues operating with the redundant (now active) proxy even if the primary proxy returns to service. If the redundant proxy subsequently becomes unavailable, the device operates with the next configured redundant proxy.<br>▪ **[1]** Homing = The device always attempts to operate with the primary proxy. The device switches back to the primary proxy whenever it becomes available.<br>**Notes:**<br>▪ To enable this functionality, you must also enable the Proxy Keep-Alive feature (see the 'Proxy Keep-Alive' parameter in this table).<br>▪ The **Homing** option can only be used if the 'Proxy Keep-Alive' parameter is set to **Using Options.** |
| Proxy Load Balancing Method<br>`proxy-load-balancing-method`<br>[ProxySet_ProxyLoadBalancingMethod] | Enables load balancing between proxy servers of the Proxy Set.<br>▪ **[0]** Disable = (Default) Disables proxy load balancing.<br>▪ **[1]** Round Robin = A list of all possible proxy IP addresses is compiled. This list includes all IP addresses of the Proxy Set after DNS resolutions (including NAPTR and SRV, if configured). After this list is compiled, the Proxy Keep-Alive feature (enabled by the 'Proxy Keep-Alive' and 'Proxy Keep-Alive Time' parameters in this table) tags each entry as "offline" or "online". Load balancing is only performed on proxy servers that are tagged as "online". All outgoing messages are equally distributed across the list of IP addresses. REGISTER messages are also distributed unless a RegistrarIP is configured. The IP address list is refreshed every user-defined interval (see the |

| Parameter | Description |
|---|---|
| | ProxyIPListRefreshTime parameter). If a change in the order of the IP address entries in the list occurs, all load statistics are erased and balancing starts over again.<br>▪ **[2]** Random Weights = The outgoing requests are not distributed equally among the Proxies. The weights are received from the DNS server, using SRV records. The device sends the requests in such a fashion that each proxy receives a percentage of the requests according to its' assigned weight. A single FQDN should be configured as a proxy IP address. Random Weights Load Balancing is not used in the following scenarios:<br>✓ More than one IP address has been configured for the Proxy Set.<br>✓ The proxy address is not configured as an FQDN (only IP address).<br>✓ SRV is disabled (see the DNSQueryType parameter).<br>✓ The SRV response includes several records with a different Priority value. |
| Min. Active Servers for Load Balancing<br>min-active-serv-lb<br>[ProxySet_MinActiveServersLB] | Defines the minimum number of proxies in the Proxy Set that must be online for the device to consider the Proxy Set as online, when proxy load balancing is used.<br>The valid value is 1 to 15. The default is 1.<br>**Note:** The parameter is applicable only if proxy load balancing is enabled (see the 'Proxy Load Balancing Method' parameter, above). |
| DNS Resolve Method<br>`dns-resolve-method`<br>[ProxySet_DNSResolveMethod] | Defines the DNS query record type for resolving the proxy server's host name (FQDN) into an IP address(es).<br>▪ **[-1]** = DNS resolution method is according to the settings of the global parameter, ProxyDNSQueryType.<br>▪ **[0]** A-Record = (Default) DNS A-record query is used to resolve DNS to IP addresses.<br>▪ **[1]** SRV = If the proxy address is configured with a domain name without a port (e.g., domain.com), an SRV query is done. The SRV query returns the host names (and their weights). The device then performs DNS A-record queries per host name (according to the received weights). If the configured proxy address contains a domain name with a port (e.g., domain.com:5080), the device performs a regular DNS A-record query.<br>▪ **[2]** NAPTR = NAPTR query is done. If successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is done according to the configured transport type. If the configured proxy address contains a domain name with a port (e.g., domain.com:5080), the device |

| Parameter | Description |
|---|---|
| | performs a regular DNS A-record query. If the transport type is configured for the proxy address, a NAPTR query is not performed.<br>▪ **[3]** MS-Lync = SRV query as required by Microsoft when the device is deployed in a Microsoft Lync environment. The device sends a special SRV query to the DNS server according to the transport protocol configured in the 'Transport Type' parameter (described later in this section):<br>✓ TLS: "_sipinternaltls_tcp.<domain>" and "_sip_tls.<domain>". For example, if the configured domain name (in the 'Proxy Address' parameter) is "ms-server.com", the device queries for "_sipinternaltls_tcp.ms-server.com" and "_sip_tls.ms-server.com".<br>✓ TCP: "_sipinternal._tcp.<domain>" and "_sip_tcp.<domain>".<br>✓ Undefined: "_sipinternaltls_tcp.<domain>", "_sipinternal_tcp.<domain>", "_sip_tls.<domain>" and "_sip_tcp.<domain>".<br>The SRV query returns the host names (and their weights). The device then performs DNS A-record queries per host name (according to the received weights) to resolve into IP addresses.<br>**Note:** An SRV query can return up to four host names. For each host name, the subsequent DNS A-record query can resolve into up to 15 IP addresses. However, the device supports up to 30 DNS-resolved IP addresses. If the device receives more than this number of IP addresses, it uses the first 30 IP addresses in the received list and ignores the rest. |
| Proxy Hot Swap<br>`is-proxy-hot-swap`<br>[ProxySet_IsProxyHotSwap] | Enables the Proxy Hot-Swap feature, whereby the device switches to a redundant proxy upon a failure in the primary proxy (no response is received).<br>▪ **[0]** No = (Default) Disables the Proxy Hot-Swap feature. If a failure occurs in te primary proxy, the device does not connect with any other address (proxy) configured for the Proxy Set.<br>▪ **[1]** Yes = The device sends SIP INVITE/REGISTER messages to the first address listed in the Proxy Address table that is configured for the Proxy Set. If a SIP response is received and this response code is configured in the Reasons for Tel-to-IP Alternative Routing table (see Alternative Routing Based on SIP Responses on page 431) for Gateway, the device assumes that the proxy is down and sends the message to the next available proxy (address) in the list. |

| Parameter | Description |
|---|---|
| Keep-Alive Failure Responses<br>`keepalive-fail-resp`<br>[ProxySet_KeepAliveFailureResp] | Defines SIP response codes that if any is received in response to a keep-alive message using SIP OPTIONS, the device considers the proxy as down.<br>Up to three response codes can be configured, where each code is separated by a comma (e.g., 407,404). By default, no response code is defined. If no response code is configured, or if response codes received are not those configured, the proxy is considered "alive".<br>**Note:** The SIP 200 response code is not supported for this feature. |
| TLS Context Index<br>`tls-context-index`<br>[ProxySet_TLSContextName] | Assigns a TLS Context (SSL/TLS certificate) to the Proxy Set.<br>By default, no TLS Context is assigned. If you assign a TLS Context, the TLS Context is used as follows:<br>▪ **Incoming calls:** If the 'Transport Type' parameter (in this table) is set to **TLS** and the incoming call is successfully classified to an IP Group based on the Proxy Set, this TLS Context is used. If the 'Transport Type' parameter is set to **UDP** or classification to this Proxy Set fails, the TLS Context is not used. Instead, the device uses the TLS Context configured for the SIP Interface (see "Configuring SIP Interfaces" on page 319) used for the call; otherwise, the default TLS Context (ID 0) is used.<br>▪ **Outgoing calls:** If the 'Transport Type' parameter is set to **TLS** and the outgoing call is sent to an IP Group that is associated with this Proxy Set, this TLS Context is used. Instead, the device uses the TLS Context configured for the SIP Interface used for the call; otherwise, the default TLS Context (ID 0) is used. If the 'Transport Type' parameter is set to **UDP**, the device uses UDP to communicate with the proxy and no TLS Context is used.<br>For configuring TLS Contexts, see "Configuring TLS Certificate Contexts" on page 103. |
| **Proxy Address Table**<br>`configure voip > voip-network proxy-ip > proxy-set-id` | |
| Index<br>`proxy-ip-index`<br>[ProxyIp_ProxyIpIndex] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Proxy Address<br>`proxy-address`<br>[ProxyIp_IpAddress] | Defines the address of the proxy.<br>Up to 10 addresses can be configured per Proxy Set. The address can be defined as an IP address in dotted-decimal notation (e.g., 201.10.8.1) or FQDN. You can also specify the port using the following format:<br>▪ IPv4 address: <IP address>:<port> (e.g., 201.10.8.1:5060) |

| Parameter | Description |
| --- | --- |
| | ▪ IPv6 address: <[IPV6 address]>:<port> (e.g., [2000::1:200:200:86:14]:5060) |
| Transport Type<br>`transport-type`<br>[ProxyIp_TransportType] | Defines the transport type for communicating with the proxy.<br>▪ **[0]** UDP<br>▪ **[1]** TCP<br>▪ **[2]** TLS<br>▪ **[-1]** = (Default) The transport type is according to the settings of the global parameter, SIPTransportType. |

**This page is intentionally left blank.**

# 18 SIP Definitions

This section describes configuration of various SIP-related functionalities.

## 18.1 Configuring SIP Parameters

Many of the stand-alone SIP parameters associated with various features can be configured in the following pages:

- **SIP General Parameters page:** Provides SIP parameters for configuring general SIP features. To access this page, use the following path: **Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**.
- **SIP Advanced Parameters page:** Provides SIP parameters for configuring advanced SIP features. To access this page, use the following path: **Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**.

For a description of these parameters, refer to the section corresponding to the feature or see ''Configuration Parameters Reference'' on page 643.

## 18.2 Configuring Registration Accounts

The Account table lets you configure up to 102 Accounts. An Account defines registration information for registering and authenticating (digest) Hunt Groups (e.g., PBX) with a "serving" IP Group (e.g., ITSP).

The device initiates registration with a "serving" IP Group on behalf of the "served" Hunt Group. Therefore, Accounts are typically required when the "served" Hunt Group is unable to register or authenticate itself for whatever reason. Registration information includes username, password, host name (AOR), and contact user name (AOR). The device includes this information in the REGISTER message sent to the serving IP Group. Up to 10 Accounts can be configured per "served" Hunt Group. A Hunt Group can register to more than one IP Group (e.g., multiple ITSPs). This is done by configuring multiple entries in the Account table for the same served Hunt Group, but with different serving IP Groups, username/password, host name, and contact user values.

Authentication is typically required for INVITE messages sent to the "serving" IP Group. If the device receives a SIP 401 (Unauthorized) in response to a sent INVITE, the device checks for a matching "serving" and "served" entry in the Account table. If a matching row exists, the device authenticates the INVITE by providing the corresponding MD5 authentication username and password to the "serving" IP Group.

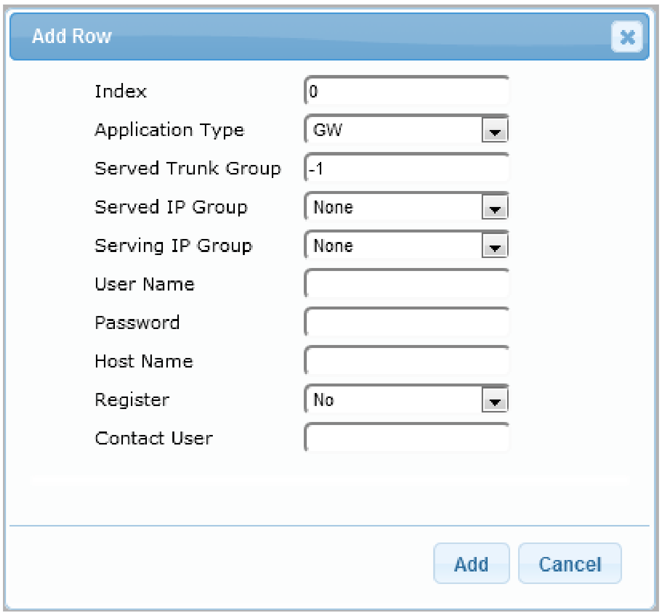
> **Note:** If no match is found in the Account table for incoming or outgoing calls, the username and password is taken from:
>
> - FXS interfaces: Authentication table (see Configuring Authentication on page 472 per Port)
> - 'UserName' and 'Password' parameters on the Proxy & Registration page

The following procedure describes how to configure Accounts through the Web interface. You can also configure it through ini file (Account) or CLI (configure voip > sip-definition account).

➢ **To configure an Account:**

1. Open the Account table (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Account Table**).

2. Click **Add**; the following dialog box appears:




3. Configure an account according to the parameters described in the table below.
4. Click **Add**.

Once you have configured Accounts, you can register or un-register them, as described below:

### ➢ To register or un-register an Account:

1. In the table, select the required Account entry row.
2. From the **Action** drop-down list, choose one of the following commands:
   - **Register** to register the Account.
   - **Un-Register** to un-register an Account.

To view Account registration status, see "Viewing Registration Status" on page 566.

If all channels belonging to the Hunt Group are down, the device un-registers them. 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 channels that are out of service.

**Table 18-1: Account Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index | Defines an index for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Served Trunk Group<br>`served-trunk-group`<br>[Account_ServedTrunkGroup] | Defines the Hunt Group ID that you want to register and/or authenticate.<br>▪ For Tel-to-IP calls, the served Hunt Group is the source Hunt Group from where the call originated.<br>▪ For IP-to-Tel calls, the served Hunt Group is the Hunt Group ID to where the call is sent. |
| Serving IP Group<br>`serving-ip-group-name`<br>[Account_ServingIPGroupName] | Defines the IP Group (*Serving IP Group*) to where the device sends the SIP REGISTER requests (if enabled) for registration and authentication (of the Served IP Group).<br>▪ Tel-to-IP calls: The serving IP Group is the destination IP Group configured in the Hunt Group Settings table or Tel-to- |

| Parameter | Description |
|---|---|
| | IP Routing table (see Configuring Tel-to-IP Routing Rules on page 409).<br>▪ IP-to-Tel calls: The serving IP Group is the 'Source IP Group ID' configured in the IP to Hunt Group Routing table (see "Configuring IP-to-Hunt Group Routing Rules" on page 421).<br>**Note:** By default, only IP Groups associated with the SRD to which the Served IP Group is associated are displayed, as well as IP Groups of Shared SRDs. However, if you filter the Web display by SRD (using the SRD Filter box), only IP Groups associated with the filtered SRD are displayed, as well as IP Groups of Shared SRDs. |
| User Name<br>`user-name`<br>[Account_Username] | Defines the digest MD5 Authentication username.<br>The valid value is a string of up to 50 characters. |
| Password<br>`password`<br>[Account_Password] | Defines the digest MD5 Authentication password.<br>The valid value is a string of up to 50 characters. |
| Host Name<br>`host-name`<br>[Account_HostName] | Defines the Address of Record (AOR) host name. The host name appears in SIP REGISTER From/To headers as ContactUser@HostName. For a successful registration, the host name is also included in the URI of the INVITE From header.<br>The valid value is a string of up to 49 characters.<br>**Note:** If the parameter is not configured or if registration fails, the 'SIP Group Name' parameter value configured in the IP Group table is used instead. |
| Register<br>`register`<br>[Account_Register] | Enables registration.<br>▪ **[0]** No= (Default) The device only performs authentication (not registration). Authentication is typically done for INVITE messages sent to the "serving" IP Group. If the device receives a SIP 401 (Unauthorized) in response to a sent INVITE, the device checks for a matching "serving" and "served" entry in the table. If a matching row exists, the device authenticates the INVITE by providing the corresponding MD5 authentication username and password to the "serving" IP Group.<br>▪ **[1]** Regular = Regular registration process. For more information, see "Regular Registration Mode" on page 344.<br>▪ **[2]** GIN = Registration for legacy PBXs, using Global Identification Number (GIN). For more information, see "Single Registration for Multiple Phone Numbers using GIN" on page 344.<br>**Notes:**<br>▪ To enable registration, you also need to set the 'Registration Mode' parameter to Per Account in the Hunt Group Settings table (see Configuring Hunt Group Settings on page 386).<br>▪ The account registration is not affected by the IsRegisterNeeded parameter. |
| Contact User<br>`contact-user` | Defines the AOR username. This appears in REGISTER From/To headers as ContactUser@HostName, and in |

| Parameter | Description |
|---|---|
| [Account_ContactUser] | INVITE/200 OK Contact headers as ContactUser@<device's IP address>.<br>**Notes:**<br>▪ If the parameter is not configured, the 'Contact User' parameter in the IP Group table is used instead.<br>▪ If registration fails, the user part in the INVITE Contact header contains the source party number. |
| Application Type<br>`application-type`<br>[Account_ApplicationType] | Defines the application type:<br>▪ [0] GW = (Default) Gateway application. |

### 18.2.1 Regular Registration Mode

When you configure the registration mode in the Account table to **Regular**, the device sends REGISTER requests to the Serving IP Group. The host name (in the SIP From/To headers) and contact user (user in From/To and Contact headers) are taken from the configured Account table upon successful registration. See the example below:

```
REGISTER sip:xyz SIP/2.0
Via: SIP/2.0/UDP 10.33.37.78;branch=z9hG4bKac1397582418
From: <sip:ContactUser@HostName>;tag=1c1397576231
To: <sip: ContactUser@HostName >
Call-ID: 1397568957261200022256@10.33.37.78
CSeq: 1 REGISTER
Contact: <sip:ContactUser@10.33.37.78>;expires=3600
Expires: 3600
User-Agent: Sip-Gateway/v.6.80A.227.005
Content-Length: 0
```

### 18.2.2 Single Registration for Multiple Phone Numbers using GIN

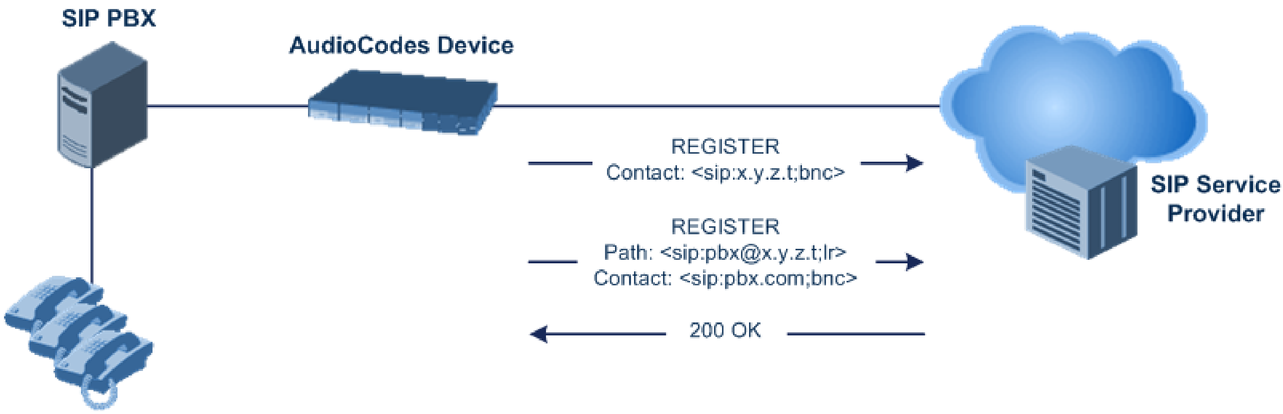
When you configure the registration mode in the Account table to **GIN**, the Global Identifiable Number (GIN) registration method is used, according to RFC 6140. The device performs GIN-based registration of users to a SIP registrar on behalf of a SIP PBX. In effect, the PBX registers with the service provider, just as a directly hosted SIP endpoint would register. However, because a PBX has multiple user agents, it needs to register a contact address on behalf of each of these. Rather than performing a separate registration procedure for each user agents, GIN registration mode does multiple registrations using a single REGISTER transaction.

According to this mechanism, the PBX delivers to the service provider in the Contact header field of a REGISTER request a template from which the service provider can construct contact URIs for each of the AORs assigned to the PBX and thus, can register these contact URIs within its location service. These registered contact URIs can then be used to deliver to the PBX inbound requests targeted at the AORs concerned. The mechanism can be used with AORs comprising SIP URIs based on global E.164 numbers and the service provider's domain name or sub-domain name.

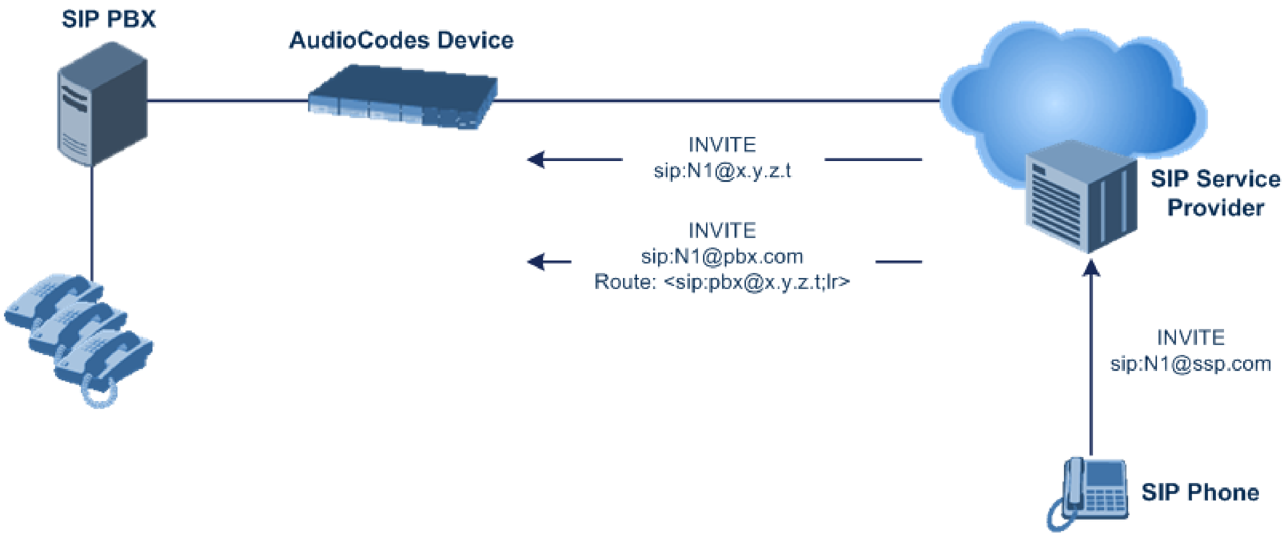
The SIP REGISTER request sent by the device for GIN registration with a SIP server provider contains the Require and Proxy-Require headers. These headers contain the token 'gin'. The Supported header contains the token 'path' and the URI in the Contact header contains the parameter 'bnc' without a user part:

```
Contact: <sip:198.51.100.3;bnc>;
```

The figure below illustrates the GIN registration process:




The figure below illustrates an incoming call using GIN:



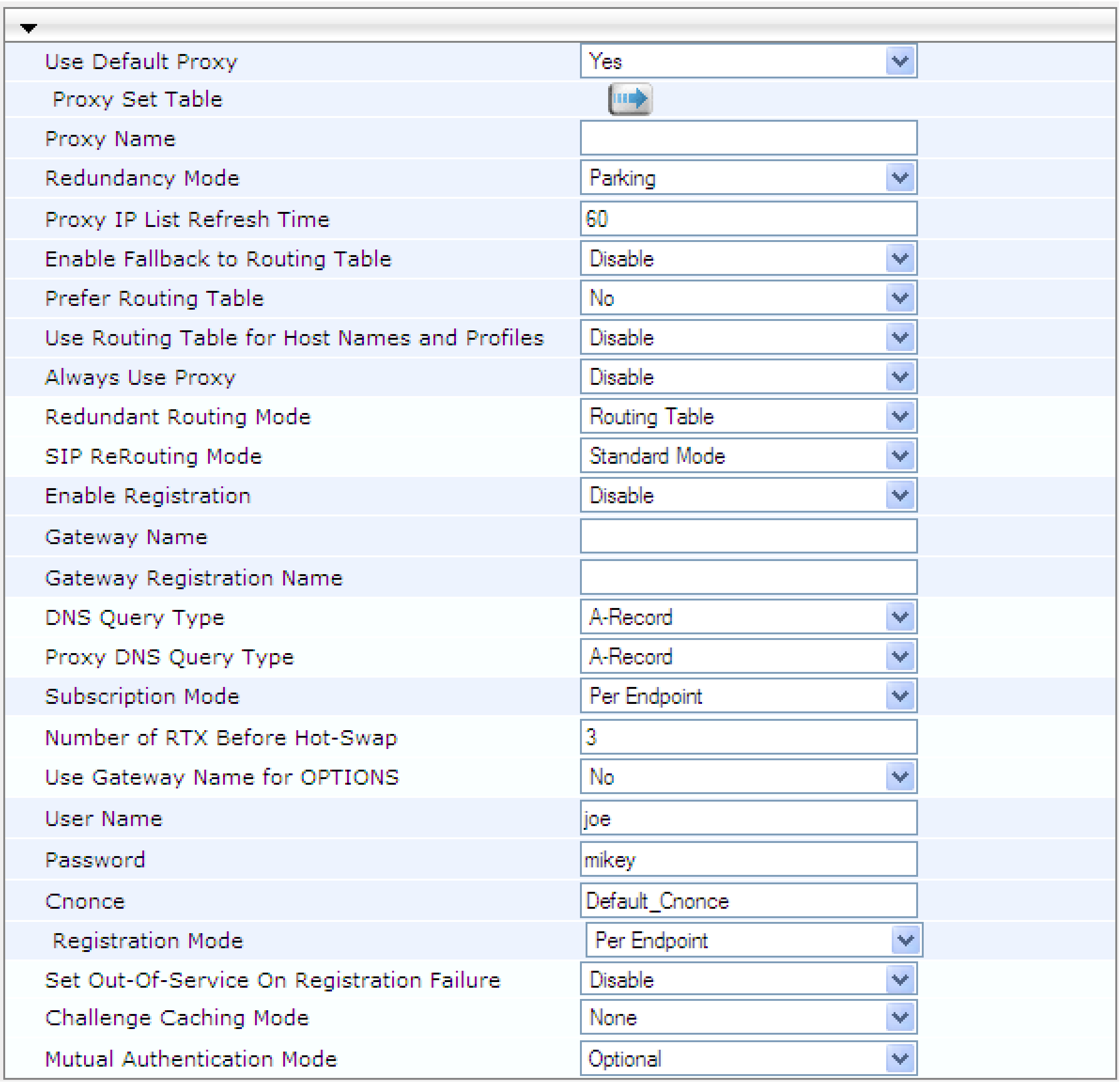
## 18.3 Configuring Proxy and Registration Parameters

The Proxy & Registration page allows you to configure the Proxy server and registration parameters. For a description of the parameters appearing on this page, see "Configuration Parameters Reference" on page 643.

**Note:** To view the registration status of endpoints with a SIP Registrar/Proxy server, see "Viewing Registration Status" on page 566.

➢ **To configure the Proxy and registration parameters:**

1. Open the Proxy & Registration page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Proxy & Registration**).




2. Configure the parameters as required.
3. Click **Submit**.

➢ **To register or un-register the device to a Proxy/Registrar:**

- Click the **Register** button to register.
- Click **Un-Register** button to un-register.

Instead of registering the entire device, you can register specific entities as listed below by using the **Register** button located on the page in which these entities are configured:

- FXS endpoints, Hunt Groups - Hunt Group table (see Configuring Hunt Groups on page 385)
- Accounts - Account table (see "Configuring Registration Accounts" on page 341)

Click the **Proxy Set Table** button to Open the Proxy Sets table to configure groups of proxy addresses. Alternatively, you can open this page from the **Proxy Sets Table** page item (see "Configuring Proxy Sets" on page 329 for a description of this page).

### 18.3.1 SIP Message Authentication Example

The device supports basic and digest (MD5) authentication types, according to SIP RFC 3261. A proxy server might require authentication before forwarding an INVITE message. A Registrar/Proxy server may also require authentication for client registration. A proxy replies to an unauthenticated INVITE with a 407 Proxy Authorization Required response, containing a Proxy-Authenticate header with the form of the challenge. After sending an ACK for the 407, the user agent can then re-send the INVITE with a Proxy-Authorization header containing the credentials.

User agents, Redirect or Registrar servers typically use the SIP 401 Unauthorized response to challenge authentication containing a WWW-Authenticate header, and expect the re-INVITE to contain an Authorization header.

The following example shows the Digest Authentication procedure, including computation of user agent credentials:

1. The REGISTER request is sent to a Registrar/Proxy server for registration:

```
REGISTER sip:10.2.2.222 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c17940
To: <sip: 122@10.1.1.200>
Call-ID: 634293194@10.1.1.200
User-Agent: Sip-Gateway/Mediant 800B Gateway and E-
SBC/v.6.80A.227.005
CSeq: 1 REGISTER
Contact: sip:122@10.1.1.200:
Expires:3600
```

2. Upon receipt of this request, the Registrar/Proxy returns a 401 Unauthorized response:

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 10.2.1.200
From: <sip:122@10.2.2.222 >;tag=1c17940
To: <sip:122@10.2.2.222 >
Call-ID: 634293194@10.1.1.200
Cseq: 1 REGISTER
Date: Mon, 30 Jul 2012 15:33:54 GMT
Server: Columbia-SIP-Server/1.17
Content-Length: 0
WWW-Authenticate: Digest realm="audiocodes.com",
nonce="11432d6bce58ddf02e3b5e1c77c010d2",
stale=FALSE,
algorithm=MD5
```

3. According to the sub-header present in the WWW-Authenticate header, the correct

REGISTER request is created.

4. Since the algorithm is MD5:
   - The username is equal to the endpoint phone number "122".
   - The realm return by the proxy is "audiocodes.com".
   - The password from the *ini* file is "AudioCodes".
   - The equation to be evaluated is "122:audiocodes.com:AudioCodes". According to the RFC, this part is called A1.
   - The MD5 algorithm is run on this equation and stored for future usage.
   - The result is "a8f17d4b41ab8dab6c95d3c14e34a9e1".
5. The par called A2 needs to be evaluated:
   - The method type is "REGISTER".
   - Using SIP protocol "sip".
   - Proxy IP from *ini* file is "10.2.2.222".
   - The equation to be evaluated is "REGISTER:sip:10.2.2.222".
   - The MD5 algorithm is run on this equation and stored for future usage.
   - The result is "a9a031cfddcb10d91c8e7b4926086f7e".
6. Final stage:
   - A1 result: The nonce from the proxy response is "11432d6bce58ddf02e3b5e1c77c010d2".
   - A2 result: The equation to be evaluated is "A1:11432d6bce58ddf02e3b5e1c77c010d2:A2"**.**
   - The MD5 algorithm is run on this equation. The outcome of the calculation is the response needed by the device to register with the Proxy.
   - The response is "b9c45d0234a5abf5ddf5c704029b38cf".

   At this time, a new REGISTER request is issued with the following response:

```
REGISTER sip:10.2.2.222 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c23940
To: <sip: 122@10.1.1.200>
Call-ID: 654982194@10.1.1.200
Server: Audiocodes-Sip-Gateway/Mediant 800B Gateway and E-
SBC/v.6.80A.227.005
CSeq: 1 REGISTER
Contact: sip:122@10.1.1.200:
Expires:3600
Authorization: Digest, username: 122,
realm="audiocodes.com”,
nonce="11432d6bce58ddf02e3b5e1c77c010d2",
uri=”10.2.2.222”,
response=“b9c45d0234a5abf5ddf5c704029b38cf”
```

7. Upon receiving this request and if accepted by the Proxy, the Proxy returns a 200 OK response, completing the registration transaction:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c23940
To: <sip: 122@10.1.1.200>
Call-ID: 654982194@10.1.1.200
Cseq: 1 REGISTER
Date: Thu, 26 Jul 2012 09:34:42 GMT
Server: Columbia-SIP-Server/1.17
Content-Length: 0
```

```
Contact: <sip:122@10.1.1.200>; expires="Thu, 26 Jul 2012
10:34:42 GMT"; action=proxy; q=1.00
Contact: <122@10.1.1.200:>; expires="Tue, 19 Jan 2038 03:14:07
GMT"; action=proxy; q=0.00
Expires: Thu, 26 Jul 2012 10:34:42 GMT
```

## 18.4 Configuring SIP Message Manipulation

The Message Manipulations table lets you configure up to 102 Message Manipulation rules. A Message Manipulation rule defines a manipulation sequence for SIP messages. SIP message manipulation enables the normalization of SIP messaging fields between communicating network segments. For example, it allows service providers to design their own policies on the SIP messaging fields that must be present before a SIP call enters their network. Similarly, enterprises and small businesses may have policies for the information that can enter or leave their networks for policy or security reasons from a service provider. SIP message manipulations can also be implemented to resolve incompatibilities between SIP devices inside the enterprise network.

Each Message Manipulation rule is configured with a Manipulation Set ID. You can create groups (sets) of Message Manipulation rules by assigning each of the relevant Message Manipulation rules to the same Manipulation Set ID. The Manipulation Set ID is then used to assign the rules to specific calls:

- Message Manipulation rules are applied to calls as follows:
  - Manipulating Inbound SIP INVITE Messages: Message manipulation can be applied only to all inbound calls (not specific calls). This is done by assigning a Manipulation Set ID to the "global" ini file parameter, GWInboundManipulationSet.
  - Manipulating Outbound SIP INVITE Messages:
    a. Message manipulation can be done for specific calls, by assigning a Manipulation Set ID to an IP Group in the IP Group table, using the 'Outbound Message Manipulation Set' parameter.
    b. Message manipulation can be applied to all outbound calls (except for IP Groups that have been assigned a Manipulation Set ID). This is done by assigning a Manipulation Set ID to the "global" ini file parameter, GWOutboundManipulationSet.

The device also supports a built-in SIP message normalization feature that can be enabled per Message Manipulation rule. The normalization feature removes unknown SIP message elements before forwarding the message. These elements can include SIP headers, SIP header parameters, and SDP body fields.

The SIP message manipulation feature supports the following:

- Manipulation on SIP message type (Method, Request/Response, and Response type)
- Addition of new SIP headers
- Removal of SIP headers ("black list")
- Modification of SIP header components such as values, header values (e.g., URI value of the P-Asserted-Identity header can be copied to the From header), call's parameter values
- Deletion of SIP body (e.g., if a message body is not supported at the destination network this body is removed)
- Translating one SIP response code to another
- Topology hiding (generally present in SIP headers such as Via, Record Route, Route and Service-Route).
- Configurable identity hiding (information related to identity of subscribers, for example,

P-Asserted-Identity, Referred-By, Identity and Identity-Info)

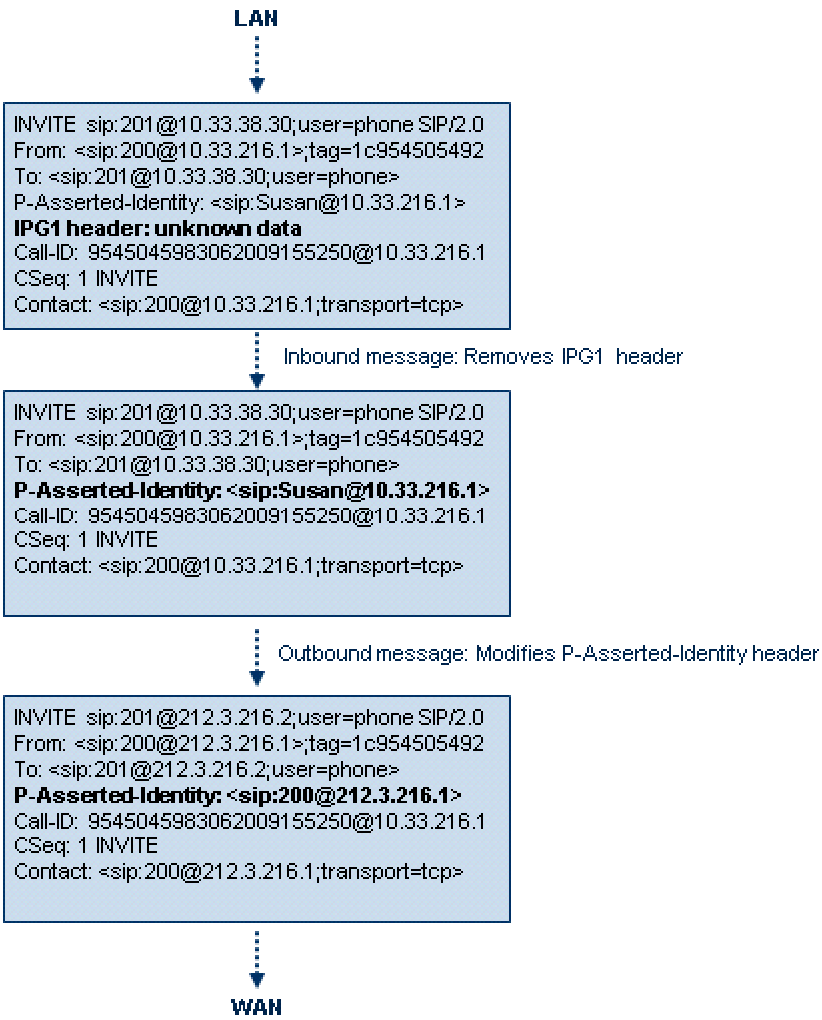
- Apply conditions per rule - the condition can be on parts of the message or call's parameters
- Multiple manipulation rules on the same SIP message
- Multiple manipulation rules using the same condition. The following figure shows a configuration example where rules 1 and 2 ('Row Rule' configured to **Use Previous Condition**) use the condition configured for rule 0 ('Row Rule' configured to **Use Current Condition**). For more information, see the description of the 'Row Rule' parameter in this section.

**Figure 18-1: Configuration Example of Message Manipulation Rules uising Same Condition**

| Index | Name | Manipulation Set ID | Message Type | Condition | Action Subject | Action Type | Action Value | Row Role |
|---|---|---|---|---|---|---|---|---|
| 0 | To header urgent | 0 | invite.request | header.request-uri.ı | header.to | Modify | header.to + ':urgent=1' | Use Current Condition |
| 1 | Add emergency | 0 | | | header.priority | Add | 'emergency' | Use Previous Condition |
| 2 | User-Agent | 0 | | | header.user-agent | Modify | 'trunk-a' | Use Previous Condition |

The figure below illustrates a SIP message manipulation example:

**Figure 18-2: SIP Header Manipulation Example**



**Notes:**

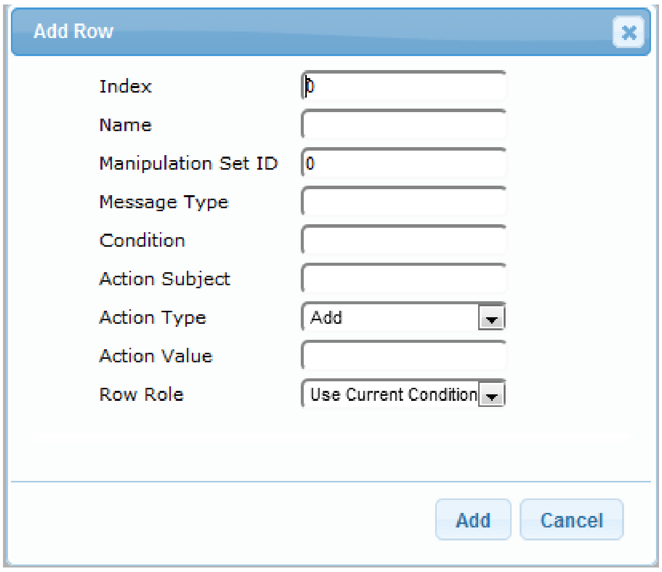
- For a detailed description of the syntax used for configuring Message Manipulation rules, refer to the *SIP Message Manipulations Quick Reference Guide*.
- Each message can be manipulated twice - on the source leg and on the destination leg (i.e., source and destination IP Groups).
- Unknown SIP parts can only be added or removed.
- SIP manipulations do not allow you to remove or add mandatory SIP headers. They can only be modified and only on requests that initiate new dialogs. Mandatory SIP headers include To, From, Via, CSeq, Call-Id, and Max-Forwards.
- The SIP Group Name (IPGroup_SIPGroupName) parameter overrides inbound message manipulation rules that manipulate the host name in Request-URI, To, and/or From SIP headers. If you configure a SIP Group Name for the IP Group (see Configuring IP Groups on page 323) and you want to manipulate the host name in any of these SIP headers, you must apply your manipulation rule (Manipulation Set ID) to the IP Group as an Outbound Message Manipulation Set (IPGroup_OutboundManSet), when the IP Group is the destination of the call. If you apply the Manipulation Set as an Inbound Message Manipulation Set (IPGroup_InboundManSet), when the IP Group is the source of the call, the manipulation rule will be overridden by the SIP Group Name.

The following procedure describes how to configure Message Manipulation rules through the Web interface. You can also configure it through ini file (MessageManipulations) or CLI (configure voip > sbc manipulations message-manipulations).

### ➢ To configure SIP message manipulation rules:

1. Open the Message Manipulations page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Msg Policy & Manipulation** > **Message Manipulations**).
2. Click **Add**; the following dialog box appears:

**Figure 18-3: Message Manipulations Table - Add Row Dialog Box**




3. Configure a Message Manipulation rule according to the parameters described in the table below.
4. Click **Add**.

An example of configured message manipulation rules are shown in the figure below:

**Figure 18-4: Message Manipulations Page**

Add + | Insert + | Edit | Delete | Up | Down | Show/Hide

| Index | Manipulation Name | Manipulation Set ID | Message Type | Condition | Action Subject | Action Type | Action Value |
|---|---|---|---|---|---|---|---|
| 0 | ITSP A | 1 | invite.response.200 | | header.to.url.user | Add Suffix | '.com' |
| 1 | | 1 | invite.response.200 | | header.from.url.user | Modify | header.p-asserted-id.url.user |
| 2 | | 1 | invite.request | | header.from.url.user | Modify | '200' |
| 3 | | 2 | invite.request | header.from.url.user=='Unkown' | header.from.url.user | Modify | param.ipg.src.user |
| 4 | | 2 | invite.request | | header.priority | Remove | |

Page 1 of 1 Show 10 records per page View 1 - 5 of 5

- Index 0: Adds the suffix ".com" to the host part of the To header.
- Index 1: Changes the user part of the From header to the user part of the P-Asserted-ID.
- Index 2: Changes the user part of the SIP From header to "200".
- Index 3: If the user part of the From header equals "unknown", then it is changed according to the srcIPGroup call's parameter.
- Index 4: Removes the Priority header from an incoming INVITE message.

**Table 18-2: Message Manipulations Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[MessageManipulations_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>`manipulation-name`<br>[MessageManipulations_ManipulationName] | Defines an arbitrary name to easily identify the rule.<br>The valid value is a string of up to 40 characters. |
| Manipulation Set ID<br>`manipulation-set-id`<br>[MessageManipulations_ManSetID] | Defines a Manipulation Set ID for the rule. You can define the same Manipulation Set ID for multiple rules to create a group of rules. The Manipulation Set ID is used to assign the manipulation rules to an IP Group (in the IP Group table) for inbound and/or outbound messages.<br>The valid value is 0 to 19. The default is 0. |
| **Matching Characteristics** | |
| Message Type<br>`message-type`<br>[MessageManipulations_MessageType] | Defines the SIP message type that you want to manipulate.<br>The valid value is a string (case-insensitive) denoting the SIP message.<br>For example:<br>▪ Empty = rule applies to all messages<br>▪ Invite = rule applies to all INVITE requests and responses<br>▪ Invite.Request = rule applies to INVITE requests<br>▪ Invite.Response = rule applies to INVITE responses<br>▪ subscribe.response.2xx = rule applies to SUBSCRIBE confirmation responses<br>**Note:** Currently, SIP 100 Trying messages cannot be manipulated. |
| Condition<br>`condition` | Defines the condition that must exist for the rule to apply.<br>The valid value is a string (case-insensitive). |

| Parameter | Description |
|---|---|
| [MessageManipulations_Condition] | For example:<br>▪ header.from.url.user== '100' (indicates that the user part of the From header must have the value "100")<br>▪ header.contact.param.expires > '3600'<br>▪ header.to.url.host contains 'domain'<br>▪ param.call.dst.user != '100' |
| **Operation** | |
| Action Subject<br>`action-subject`<br>[MessageManipulations_ActionSubject] | Defines the SIP header upon which the manipulation is performed.<br>The valid value is a string (case-insensitive). |
| Action Type<br>`action-type`<br>[MessageManipulations_ActionType] | Defines the type of manipulation.<br>▪ **[0]** Add (default) = Adds new header/param/body (header or parameter elements).<br>▪ **[1]** Remove = Removes header/param/body (header or parameter elements).<br>▪ **[2]** Modify = Sets element to the new value (all element types).<br>▪ **[3]** Add Prefix = Adds value at the beginning of the string (string element only).<br>▪ **[4]** Add Suffix = Adds value at the end of the string (string element only).<br>▪ **[5]** Remove Suffix = Removes value from the end of the string (string element only).<br>▪ **[6]** Remove Prefix = Removes value from the beginning of the string (string element only).<br>▪ **[7]** Normalize = Removes unknown SIP message elements before forwarding the message. |
| Action Value<br>`action-value`<br>[MessageManipulations_ActionValue] | Defines a value that you want to use in the manipulation.<br>The default value is a string (case-insensitive) in the following syntax:<br>▪ string/<message-element>/<call-param> +<br>▪ string/<message-element>/<call-param><br>For example:<br>▪ 'itsp.com'<br>▪ header.from.url.user<br>▪ param.call.dst.user<br>▪ param.call.dst.host + '.com'<br>▪ param.call.src.user + '<' + header.from.url.user + '@' + header.p-asserted-id.url.host + '>'<br>**Note:** Only single quotation marks must be used. |
| Row Role<br>`row-role`<br>[MessageManipulations_RowRole] | Determines which message manipulation condition (configured by the 'Condition' parameter) to use for the rule.<br>▪ [0] Use Current Condition = (Default) The condition configured in the table row of the rule is used.<br>▪ [1] Use Previous Condition = The condition configured in the first table row above the rule that is configured to **Use** |

| Parameter | Description |
|---|---|
| | **Current Condition** is used. For example, if Index 3 is configured to **Use Current Condition** and Index 4 and 5 are configured to **Use Previous Condition**, Index 4 and 5 use the condition configured for Index 3. A configuration example is shown in the beginning of this section. The option allows you to use the same condition for multiple manipulation rules.<br>**Notes:**<br>▪ When configured to **Use Previous Condition**, the 'Message Type' and 'Condition' parameters are not applicable and if configured are ignored.<br>▪ When multiple manipulation rules apply to the same header, the next rule applies to the resultant string of the previous rule. |

## 18.5 Configuring SIP Message Policy Rules

The Message Policy table lets you configure up to 20 SIP Message Policy rules. SIP Message Policy rules are used to block (blacklist) unwanted incoming SIP messages or permit (whitelist) receipt of desired SIP messages. You can configure legal and illegal characteristics of a SIP message. This feature is helpful against VoIP fuzzing (also known as robustness testing), which sends different types of packets to its "victims" for finding bugs and vulnerabilities. For example, the attacker might try sending a SIP message containing either an oversized parameter or too many occurrences of a parameter.

To apply SIP Message Policy rules, you need to assign them to SIP Interfaces associated with the relevant IP Groups (see "Configuring SIP Interfaces" on page 319).

Each Message Policy rule can be configured with the following:

- Maximum message length
- Maximum header length
- Maximum message body length
- Maximum number of headers
- Maximum number of bodies
- Option to send 400 "Bad Request" response if message request is rejected
- Blacklist and whitelist for defined methods (e.g., INVITE)
- Blacklist and whitelist for defined bodies

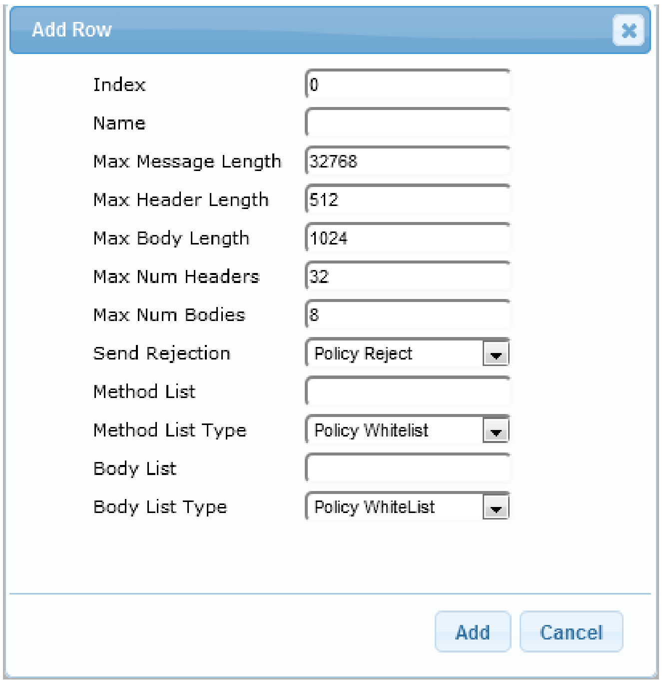
The following procedure describes how to configure Message Policy rules through the Web interface. You can also configure it through ini file (MessagePolicy) or CLI (configure voip > sbc message-policy).

➢ **To configure SIP Message Policy rules:**

1. Open the Message Policy table (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Msg Policy & Manipulation** > **Message Policy Table**).

2. Click **Add**; the following dialog box appears:

**Figure 18-5: Message Policy Table - Add Row Dialog Box**




3. Configure a Message Policy rule according to the parameters described in the table below.
4. Click **Add**.

**Table 18-3: Message Policy Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[MessagePolicy_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>`name`<br>[MessagePolicy_Name] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters.<br>**Note:** Each row must be configured with a unique name. |
| Max Message Length<br>`max-message-length`<br>[MessagePolicy_MaxMessageLength] | Defines the maximum SIP message length.<br>The valid value is up to 32,768 characters. The default is 32,768. |
| Max Header Length<br>`max-header-length`<br>[MessagePolicy_MaxHeaderLength] | Defines the maximum SIP header length.<br>The valid value is up to 512 characters. The default is 512. |
| Max Body Length<br>`max-body-length`<br>[MessagePolicy_MaxBodyLength] | Defines the maximum SIP message body length. This is the value of the Content-Length header.<br>The valid value is up to 1,024 characters. The default is 1,024. |

| Parameter | Description |
|---|---|
| Max Num Headers<br>`max-num-headers`<br>[MessagePolicy_MaxNumHeaders] | Defines the maximum number of SIP headers.<br>The valid value is any number up to 32. The default is 32.<br>**Note:** The device supports up to 20 SIP Record-Route headers that can be received in a SIP INVITE request or a 200 OK response. If it receives more than this, it responds with a SIP 513 'Message Too Large' response. |
| Max Num Bodies<br>`max-num-bodies`<br>[MessagePolicy_MaxNumBodies] | Defines the maximum number of bodies (e.g., SDP) in the SIP message.<br>The valid value is any number up to 8. The default is 8. |
| Send Rejection<br>`send-rejection`<br>[MessagePolicy_SendRejection] | Determines whether the device sends a 400 "Bad Request" response if a message request is rejected.<br>▪ **[0]** Policy Reject = (Default) If the message is a request, the device sends a response to reject the request.<br>▪ **[1]** Policy Drop = The device ignores the message without sending any response. |
| **SIP Method Blacklist-Whitelist Policy** | |
| Method List<br>`method-list`<br>[MessagePolicy_MethodList] | Defines SIP methods (e.g., INVITE\BYE) to blacklist or whitelist.<br>Multiple methods are separated by a backslash (\). The method values are case-insensitive. |
| Method List Type<br>`method-list-type`<br>[MessagePolicy_MethodListType] | Defines the policy (blacklist or whitelist) for the SIP methods specified in the 'Method List' parameter (above).<br>▪ **[0]** Policy Blacklist = The specified methods are rejected.<br>▪ **[1]** Policy Whitelist = (Default) Only the specified methods are allowed; the others are rejected. |
| **SIP Body Blacklist-Whitelist Policy** | |
| Body List<br>`body-list`<br>[MessagePolicy_BodyList] | Defines the SIP body type (i.e., value of the Content-Type header) to blacklist or whitelist. For example, application/sdp.<br>The values of the parameter are case-sensitive. |
| Body List Type<br>`body-list-type`<br>[MessagePolicy_BodyListType] | Defines the policy (blacklist or whitelist) for the SIP body specified in the 'Body List' parameter (above).<br>▪ **[0]** Policy Blacklist =The specified SIP body is rejected.<br>▪ **[1]** Policy Whitelist = (Default) Only the specified SIP body is allowed; the others are rejected. |

# 19 Coders and Profiles

This section describes configuration of the coders and SIP profiles parameters.

## 19.1 Configuring Default Coders

The Coders table lets you configure up to 11 voice coders for the device. This is the default Coder Group, which is used by the device for all calls, unless a different Coder Group, configured in the Coder Group Settings table (see "Configuring Coder Groups" on page 360) is assigned to specific calls, using Tel or IP Profiles.

Each coder can be configured with packetization time (ptime), bit rate, payload type, and silence suppression. The first coder configured in the table has the highest priority and is used by the device whenever possible. If the remote side cannot use the first coder, the device attempts to use the next coder in the table, and so on.

**Notes:**

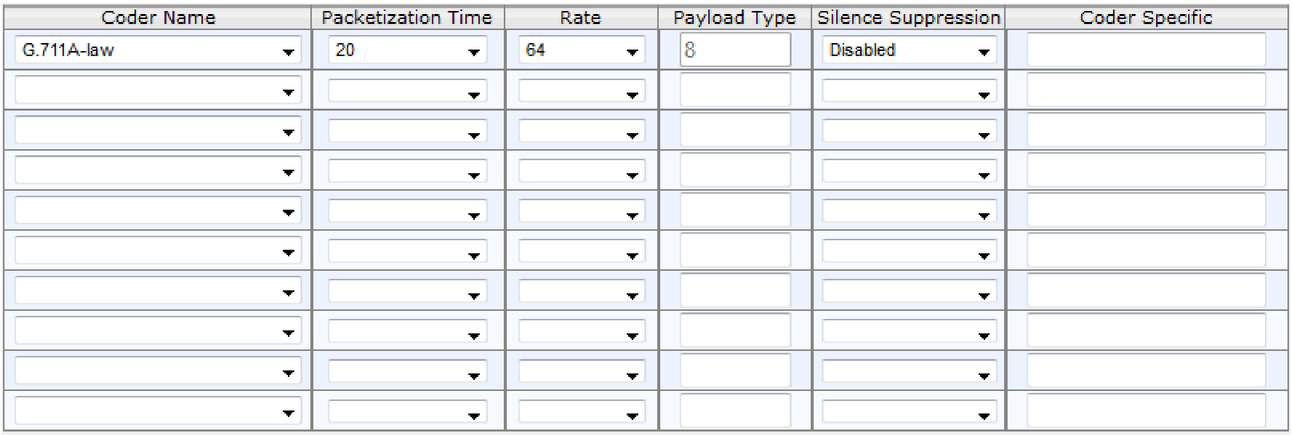
- Some coders are license-dependent and are available only if purchased from AudioCodes and included in the Software License Key installed on your device. For more information, contact your AudioCodes sales representative.
- Only the packetization time of the first coder in the coder list is declared in INVITE/200 OK SDP, even if multiple coders are defined. The device always uses the packetization time requested by the remote side for sending RTP packets. If not specified, the packetization time is assigned the default value.
- The value of several fields is hard-coded according to common standards (e.g., payload type of G.711 U-law is always 0). Other values can be set dynamically. If no value is specified for a dynamic field, a default value is assigned. If a value is specified for a hard-coded field, the value is ignored.
- The G.722 coder provides Packet Loss Concealment (PLC) capabilities, ensuring higher voice quality.
- Gateway calls always use the narrowband Opus coder.
- For information on V.152 and implementation of T.38 and VBD coders, see "Supporting V.152 Implementation" on page 191.

The following procedure describes how to configure the Coders table through the Web interface. You can also configure it through ini file (CodersGroup) or CLI (configure voip > coders-and-profiles coders-group).

➢ **To configure coders:**

1. Open the Coders page (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **Coders**).

**Figure 19-1: Coders Table Page**




2. Configure coders according to the parameters described in the table below.
3. Click **Submit**, and then reset the device with a save ("burn") to flash memory.

**Table 19-1: Coders Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Coder Name<br>`name`<br>[CodersGroup0_Name] | Defines the coder.<br>**Note:** Each coder type (e.g., G.729) can be configured only once in the table. |
| Packetization Time<br>`p-time`<br>[CodersGroup0_pTime] | Defines the packetization time (in msec) for the coder. The packetization time determines how many coder payloads are combined into a single RTP packet. |
| Rate<br>`rate`<br>[CodersGroup0_rate] | Defines the bit rate (in kbps) for the coder. |
| Payload Type<br>`payload-type`<br>[CodersGroup0_PayloadType] | Defines the payload type if the payload type (i.e., format of the RTP payload) for the coder is dynamic. |
| Silence Suppression<br>`silence-suppression`<br>[CodersGroup0_Sce] | Enables silence suppression for the coder.<br>▪ **[0]** Disable (Default)<br>▪ **[1]** Enable<br>▪ **[2]** Enable w/o Adaptation =Applicable only to G.729.<br>**Notes:**<br>▪ If G.729 is configured with silence suppression disabled, the device includes 'annexb=no' in the SDP of the relevant SIP messages. If silence suppression is enabled or set to **Enable w/o Adaptations**, 'annexb=yes' is included. For the Gateway application, an exception to this logic is when the remote gateway is Cisco equipment (IsCiscoSCEMode). |

| Parameter | Description |
|---|---|
| Coder Specific<br>`coder-specific`<br>[CodersGroup0_CoderSpecific] | Defines additional settings specific to the coder.<br>Currently, the parameter is applicable only to the AMR coder and is used to configure the payload format type.<br>▪ [0] 0 = Bandwidth Efficient<br>▪ [1] 1 = Octet Aligned (default)<br>**Note:** The AMR payload type can be configured globally using the AmrOctetAlignedEnable parameter. However, the Coder Group configuration overrides the global parameter. |

The table below lists the supported coders:

**Table 19-2: Supported Coders**

| Coder Name | Packetization Time (msec) | Rate (kbps) | Payload Type | Silence Suppression |
|---|---|---|---|---|
| G.711 A-law<br>**[g711Alaw64k]** | 10, 20 (default), 30, 40, 50, 60, 80, 100, 120 | 64 | 8 | ▪ **[0]** Disable<br>▪ **[1]** Enable |
| G.711 U-law<br>**[g711Ulaw64k]** | 10, 20 (default), 30, 40, 50, 60, 80, 100, 120 | 64 | 0 | ▪ **[0]** Disable<br>▪ **[1]** Enable |
| G.711A-law_VBD<br>**[g711AlawVbd]** | 10, 20 (default), 30, 40, 50, 60, 80, 100, 120 | 64 | 8 or Dynamic (default 118) | N/A |
| G.711U-law_VBD<br>**[g711UlawVbd]** | 10, 20 (default), 30, 40, 50, 60, 80, 100, 120 | 64 | 0 or Dynamic (default 110) | N/A |
| G.722<br>**[g722]** | 20 (default), 40, 60, 80, 100, 120 | 64 (default) | 9 | N/A |
| G.723.1<br>**[g7231]** | 30 (default), 60, 90, 120, 150 | ▪ **[0]** 5.3 (default)<br>▪ **[1]** 6.3 | 4 | ▪ **[0]** Disable<br>▪ **[1]** Enable |
| G.726<br>**[g726]** | 10, 20 (default), 30, 40, 50, 60, 80 | ▪ **[0]** 16<br>▪ **[1]** 24<br>▪ **[2]** 32 (default)<br>▪ **[3]** 40 | Dynamic (default 2) | ▪ **[0]** Disable<br>▪ **[1]** Enable |
| G.729<br>**[g729]** | 10, 20 (default), 30, 40, 50, 60, 80, 100 | 8 | 18 | ▪ **[0]** Disable<br>▪ **[1]** Enable<br>▪ **[2]** Enable w/o Adaptations |

| Coder Name | Packetization Time (msec) | Rate (kbps) | Payload Type | Silence Suppression |
|---|---|---|---|---|
| AMR<br>[Amr] | 20 (default) | ▪ [0] 4.75<br>▪ [1] 5.15<br>▪ [2] 5.90<br>▪ [3] 6.70<br>▪ [4] 7.40<br>▪ [5] 7.95<br>▪ [6] 10.2<br>▪ [7] 12.2 (default) | Dynamic | ▪ [0] Disable<br>▪ [1] Enable |
| iLBC<br>[iLBC] | 20 (default), 40, 60, 80, 100, 120 | 15 (default) | Dynamic (default 65) | ▪ [0] Disable<br>▪ [1] Enable |
| | 30 (default), 60, 90, 120 | 13 | | |
| silk-nb<br>[Silk-8Khz] | 20 (default), 40, 60, 80, and 100 | 8 | Dynamic (default 76) | N/A |
| silk-wb<br>[Silk-16Khz] | 20 (default), 40, 60, 80, and 100 | 16 | Dynamic (default 77) | N/A |
| T.38<br>**[t38fax]** | N/A | N/A | N/A | N/A |
| T.38 Version 3<br>[t38fax] | - | - | - | - |
| T.38 Over RTP | N/A | N/A | Dynamic (default 106) | N/A |
| OPUS<br>[Opus] | 20 (default), 40, 60, 80, 120 | N/A | Dynamic (default 111) | N/A |

## 19.2 Configuring Coder Groups

The Coder Group Settings table lets you configure up to 11 *Coder Groups.* A Coder Group is a set of configured coders (coder type, packetization time, rate, payload type, and silence suppression). Each Coder Group can include up to 10 coders.

The first coder in the Coder Group has the highest priority and is used by the device whenever possible. If the remote side cannot use the first coder, the device attempts to use the next coder in the Coder Group, and so on.

To define coders for specific calls, you can configure a Coder Group with the necessary coders and then assign the Coder Group to the calls using Tel Profiles (see Configuring Tel Profiles on page 362) or IP Profiles (see "Configuring IP Profiles" on page 366).

**Notes:**

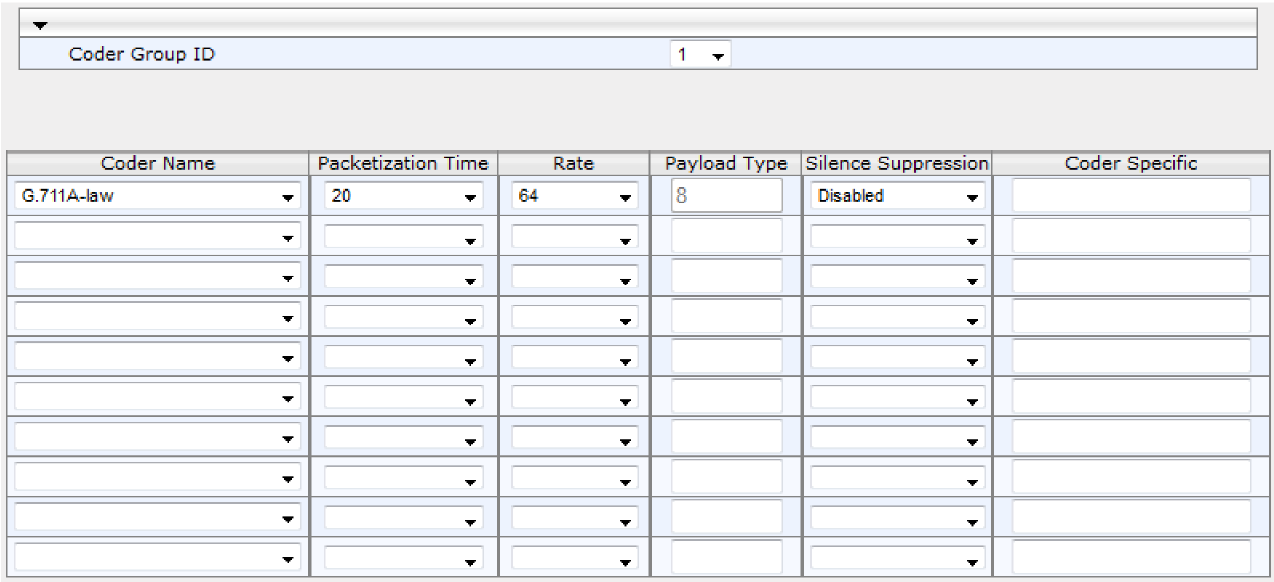
- To define coders for calls that are not assigned a specific Coder Group using Tel Profiles or IP Profiles, see "Configuring Default Coders" on page 357. This group of coders is termed the *Default Coder Group*.
- For a list of supported coders, see "Configuring Default Coders" on page 357.

The following procedure describes how to configure the Coders table through the Web interface. You can also configure it through ini file (CodersGroup) or CLI (configure voip > coders-and-profiles coders-group).

### ➢ To configure a Coder Group:

1. Open the Coder Group Settings page (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **Coders Group Settings**).

**Figure 19-2: Coder Group Settings Page**




2. Configure the Coder Group according to the parameters described in the table below.
3. Click **Add**, and then reset the device with a save ("burn") to flash memory.

**Table 19-3: Coder Group Settings Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Coder Group ID<br>[CodersGroupX_Index] | Defines an ID for the Coder Group. |
| Coder Name<br>`name`<br>[CodersGroupX_Name] | Defines the coder type.<br>**Note:** Each coder type (e.g., G.729) can be configured only once in the table. |
| Packetization Time<br>`p-time`<br>[CodersGroupX_pTime] | Defines the packetization time (in msec) for the coder. The packetization time determines how many coder payloads are combined into a single RTP packet. |
| Rate<br>`rate`<br>[CodersGroupX_rate] | Defines the bit rate (in kbps) for the coder. |
| Payload Type<br>`payload-type`<br>[CodersGroupX_PayloadType] | Defines the payload type if the payload type (i.e., format of the RTP payload) for the coder is dynamic. |

| Parameter | Description |
|---|---|
| Silence Suppression<br>`silence-suppression`<br>[CodersGroupX_Sce] | Enables silence suppression for the coder.<br>▪ **[0]** Disable (Default)<br>▪ **[1]** Enable<br>▪ **[2]** Enable w/o Adaptation =Applicable only to G.729.<br>▪ |
| Coder Specific<br>`coder-specific`<br>[CodersGroupX_CoderSpecific] | Defines additional settings specific to the coder.<br>Currently, the parameter is applicable only to the AMR coder and is used to configure the payload format type.<br>▪ [0] 0 = Bandwidth Efficient<br>▪ [1] 1 = Octet Aligned (default)<br>**Note:** The AMR payload type can be configured globally using the AmrOctetAlignedEnable parameter. However, the Coder Group configuration overrides the global parameter. |

## 19.3 Configuring Tel Profile

The Tel Profile Settings table lets you configure up to nine *Tel Profiles*. A Tel Profile is a set of parameters with specific settings which can be assigned to specific calls. The Tel Profile Settings table includes a wide range of parameters for configuring the Tel Profile. Each of these parameters has a corresponding "global" parameter, which when configured applies to all calls. The main difference, if any, between the Tel Profile parameters and their corresponding global parameters are their default values.

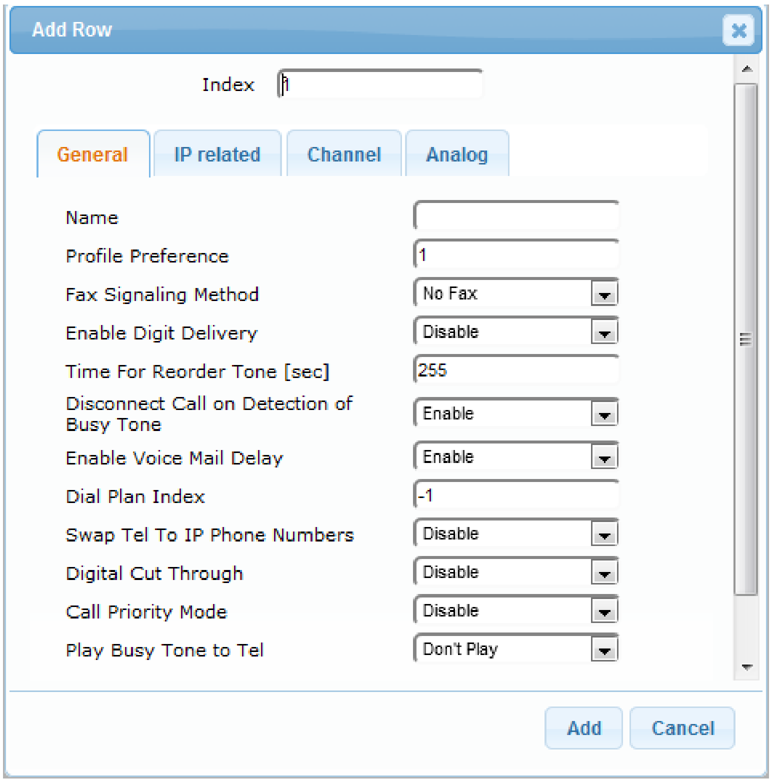
Tel Profiles provide high-level adaptation when the device interworks between different equipment and protocols (at both the Tel and IP sides), each of which may require different handling by the device. For example, if specific channels require the use of the G.711 coder, you can configure a Tel Profile with this coder and assign it to these channels.

To use your Tel Profile for specific calls, you need to assign it to specific channels in the Hunt Group table (see Configuring Hunt Groups on page 385)).

The following procedure describes how to configure Tel Profiles through the Web interface. You can also configure it through ini file (TelProfile) or CLI (configure voip/coders-and-profiles tel-profile).

➢ **To configure a Tel Profile:**

1. Open the Tel Profile Settings page (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **Tel Profile Settings**).
2. Click **Add**; the following dialog box appears:




3. Configure a Tel Profile according to the parameters described in the table below. For a description of each parameter, refer to the corresponding "global" parameter.
4. Click **Add**.

**Table 19-4: Tel Profile Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| **Common** | |
| Index<br>[TelProfile_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Profile Name<br>`profile-name`<br>[TelProfile_ProfileName] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters. |
| Profile Preference<br>`tel-preference`<br>[TelProfile_TelPreference] | Defines the priority of the Tel Profile, where **1** is the lowest priority and **20** the highest priority.<br>**Notes:**<br>1. If both the IP Profile and Tel Profile apply to the same call, the coders and common parameters of the Preferred profile are applied to the call.<br>2. If the Preference of the Tel Profile and IP Profile are identical, the Tel Profile parameters are applied.<br>3. If the coder lists of both the IP Profile and Tel Profile apply to the same call, only the coders common to both are used. The order of the coders is determined by the preference. |
| Fax Signaling Method<br>`fax-sig-method`<br>[TelProfile_IsFaxUsed] | IsFaxUsed |
| Enable Digit Delivery<br>`digit-delivery`<br>[TelProfile_EnableDigitDelivery] | EnableDigitDelivery |

| Parameter | Description |
|---|---|
| Time For Reorder Tone<br>`time-for-reorder-tone`<br>[TelProfile_TimeForReorderTone] | TimeForReorderTone |
| Disconnect Call on Detection of Busy Tone<br>`disconnect-on-busy-tone`<br>[TelProfile_DisconnectOnBusyTone] | DisconnectOnBusyTone |
| Enable Voice Mail Delay<br>`enable-voice-mail-delay`<br>[TelProfile_EnableVoiceMailDelay] | VoiceMailInterface<br>This is useful for disabling voice mail services per Hunt Group to eliminate the phenomenon of call delay on channels that do not implement voice mail when voice mail is enabled using the global parameter. |
| Dial Plan Index<br>`dial-plan-index`<br>[TelProfile_DialPlanIndex] | DialPlanIndex |
| Swap Tel To IP Phone Numbers<br>`swap-teltoip-phone-numbers`<br>[TelProfile_SwapTelToIpPhoneNumbers] | SwapTEl2IPCalled&CallingNumbers |
| Digital Cut Through<br>`digital-cut-through`<br>[TelProfile_DigitalCutThrough] | DigitalCutThrough |
| Call Priority Mode<br>`call-priority-mode`<br>[TelProfile_CallPriorityMode] | CallPriorityMode |
| **IP Related** | |
| Coders Group ID<br>`coders-group-id`<br>[TelProfile_CodersGroupID] | CodersGroup0 |
| RTP IP DiffServ<br>`rtp-ip-diffserv`<br>[TelProfile_IPDiffServ] | PremiumServiceClassMediaDiffServ |
| Signaling DiffServ<br>`signaling-diffserv`<br>[TelProfile_SigIPDiffServ] | PremiumServiceClassControlDiffServ |
| Enable Early Media<br>`early-media`<br>[TelProfile_EnableEarlyMedia] | EnableEarlyMedia |
| Progress Indicator to IP<br>`prog-ind-to-ip`<br>[TelProfile_ProgressIndicator2IP] | ProgressIndicator2IP |
| **Channel** | |
| Dynamic Jitter Buffer Minimum Delay<br>`jitter-buffer-minimum-delay`<br>[TelProfile_JitterBufMinDelay] | DJBufMinDelay |
| Dynamic Jitter Buffer Optimization Factor<br>`jitter-buffer-optimization-factor`<br>[TelProfile_JitterBufOptFactor] | DJBufOptFactor |

| Parameter | Description |
|---|---|
| DTMF Volume<br>`dtmf-volume`<br>[TelProfile_DtmfVolume] | DTMFVolume |
| Input Gain<br>`input-gain`<br>[TelProfile_InputGain] | InputGain |
| Voice Volume<br>`voice-volume`<br>[TelProfile_VoiceVolume] | VoiceVolume |
| Echo Canceler<br>`echo-canceller`<br>[TelProfile_EnableEC] | EnableEchoCanceller |
| Enable AGC<br>`enable-agc`<br>[TelProfile_EnableAGC] | EnableAGC |
| EC NLP Mode<br>`echo-canceller-nlp-mode`<br>[TelProfile_ECNlpMode] | ECNLPMode |
| Analog | |
| Enable Polarity Reversal<br>`polarity-rvrsl`<br>[TelProfile_EnableReversePolarity] | EnableReversalPolarity |
| Enable Current Disconnect<br>`current-disconnect`<br>[TelProfile_EnableCurrentDisconnect] | EnableCurrentDisconnect |
| MWI Analog Lamp<br>`mwi-analog-lamp`<br>[TelProfile_MWIAnalog] | MWIAnalogLamp |
| MWI Display<br>`mwi-display`<br>[TelProfile_MWIDisplay] | MWIDisplay |
| Flash Hook Period<br>`flash-hook-period`<br>[TelProfile_FlashHookPeriod] | FlashHookPeriod |
| DID Wink<br>`enable-did-wink`<br>[TelProfile_EnableDIDWink] | EnableDIDWink |
| Two Stage Dialing<br>`is-two-stage-dial`<br>[TelProfile_IsTwoStageDial] | IsTwoStageDial |
| Enable 911 PSAP<br>`enable-911-psap`<br>[TelProfile_Enable911PSAP] | Enable911PSAP |

## 19.4 Configuring IP Profiles

The IP Profile Settings table lets you configure up to 20 IP Profiles. An IP Profile is a set of parameters with user-defined settings relating to signaling (e.g., SIP message terminations such as REFER) and media (e.g., coder type). An IP Profile can later be assigned to specific IP calls (inbound and/or outbound). Thus, IP Profiles provide high-level adaptation when the device interworks between different IP entities, each of which may require different handling by the device. This can include, for example, transcoding or even transrating (of packetization time). For example, if a specific IP entity uses the G.711 coder only, you can configure an IP Profile with G.711 for this IP entity.

To use your IP Profile for specific calls, you need to assign it to any of the following:

- IP Groups - see "Configuring IP Groups" on page 323
- Tel-to-IP routing rules - see Configuring Tel-to-IP Routing Rules on page 409
- IP-to-Tel routing rules - see Configuring IP-to-Hunt Group Routing Rules on page 421

The device selects the IP Profile as follows:

- If you assign different IP Profiles (not default) to the same specific calls in all of the above-mentioned tables, the device uses the IP Profile that has the highest preference level (as set in the 'Profile Preference' parameter). If these IP Profiles have the same preference level, the device uses the IP Profile that you assigned in the IP Group table.
- If you assign different IP Profiles to all of the above-mentioned tables and one table is set to the default IP Profile, the device uses the IP Profile that is not the default.

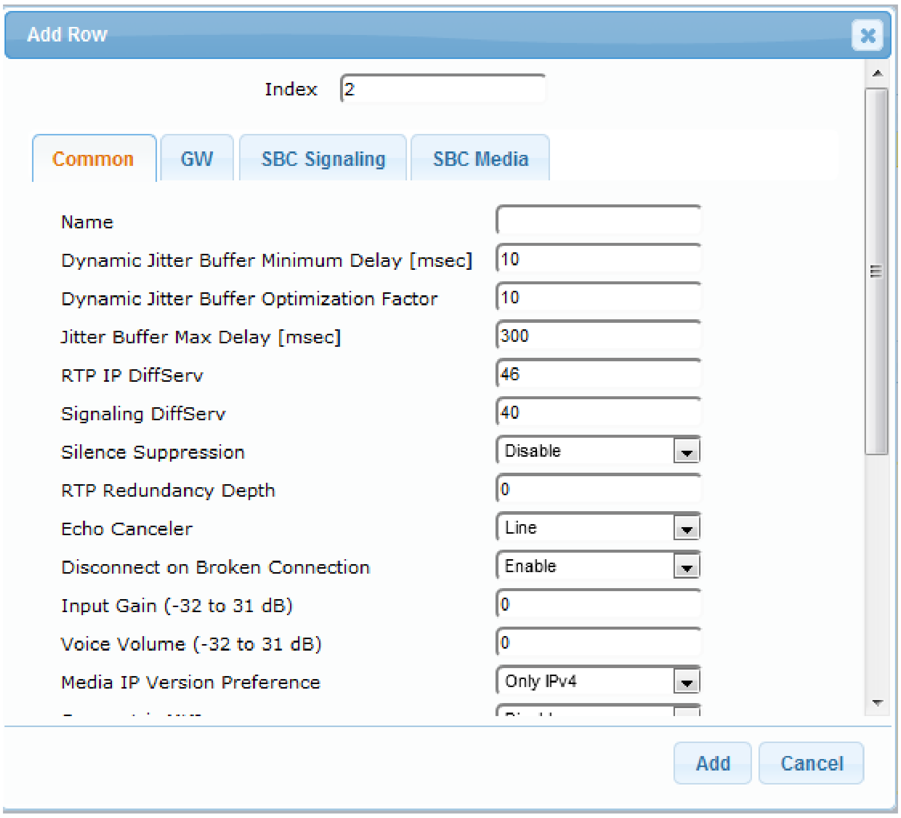
Many of the parameters in the IP Profile table have a corresponding "global" parameter. For calls that are not associated with any IP Profile, the settings of the "global" parameters are applied.

**Note:** IP Profiles can also be implemented when using a Proxy server (when the AlwaysUseRouteTable parameter is set to 1).

The following procedure describes how to configure IP Profiles through the Web interface. You can also configure it through ini file (IPProfile) or CLI (configure voip > coders-and-profiles ip-profile).

➢ **To configure an IP Profile:**

1. Open the IP Profile Settings table (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **IP Profile Settings**).
2. Click **Add**; the following dialog box appears:




3. Configure an IP Profile according to the parameters described in the table below.
4. Click **Add**.

**Table 19-5: IP Profile Settings Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| **Common** | |
| Index<br>[IpProfile_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>`profile-name`<br>[IpProfile_ProfileName] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters. |
| Dynamic Jitter Buffer Minimum Delay<br>`jitter-buffer-minimum-delay`<br>[IpProfile_JitterBufMinDelay] | Defines the minimum delay (in msec) of the device's dynamic Jitter Buffer.<br>The valid range is 0 to 150. The default delay is 10.<br>For more information on Jitter Buffer, see Dynamic Jitter Buffer Operation on page 193.<br>**Note:** The corresponding global parameter is DJBufMinDelay. |
| Dynamic Jitter Buffer Optimization Factor<br>`jitter-buffer-optimization-factor`<br>[IpProfile_JitterBufOptFactor] | Defines the Dynamic Jitter Buffer frame error/delay optimization factor.<br>The valid range is 0 to 12. The default factor is 10.<br>For more information on Jitter Buffer, see Dynamic Jitter Buffer Operation on page 193.<br>**Notes:**<br>▪ For data (fax and modem) calls, set the parameter to 12. |

| Parameter | Description |
|---|---|
| | ▪ The corresponding global parameter is DJBufOptFactor. |
| Jitter Buffer Max Delay<br>`jitter-buffer-max-delay`<br>[IpProfile_JitterBufMaxDelay] | Defines the maximum delay and length (in msec) of the Jitter Buffer.<br>The valid range is 150 to 2,000. The default is 250. |
| RTP IP DiffServ<br>`rtp-ip-diffserv`<br>[IpProfile_IPDiffServ] | Defines the DiffServ value for Premium Media class of service (CoS) content.<br>The valid range is 0 to 63. The default is 46.<br>**Note:** The corresponding global parameter is PremiumServiceClassMediaDiffServ. |
| Signaling DiffServ<br>`signaling-diffserv`<br>[IpProfile_SigIPDiffServ] | Defines the DiffServ value for Premium Control CoS content (Call Control applications).<br>The valid range is 0 to 63. The default is 40.<br>**Note:** The corresponding global parameter is PremiumServiceClassControlDiffServ. |
| RTP Redundancy Depth<br>`rtp-redundancy-depth`<br>[IpProfile_RTPRedundancyDepth] | Enables the device to generate RFC 2198 redundant packets. This can be used for packet loss where the missing information (audio) can be reconstructed at the receiver's end from the redundant data that arrives in subsequent packets. This is required, for example, in wireless networks where a high percentage (up to 50%) of packet loss can be experienced.<br>▪ **[0]** 0 = (Default) Disable.<br>▪ **[1]** 1 = Enable - previous voice payload packet is added to current packet.<br>**Notes:**<br>▪ When enabled, you can configure the payload type, using the RFC2198PayloadType parameter.<br>▪ For the Gateway application only: The RTP redundancy dynamic payload type can be included in the SDP, by using the EnableRTPRedundancyNegotiation parameter.<br>▪ The corresponding global parameter is RTPRedundancyDepth. |
| Echo Canceler<br>`echo-canceller`<br>[IpProfile_EnableEchoCanceller] | Enables the device's Echo Cancellation feature (i.e., echo from voice calls is removed).<br>▪ [0] Disable<br>▪ [1] Line (default)<br>▪ [2] Acoustic<br>For a detailed description of the Echo Cancellation feature, see Configuring Echo Cancellation on page 178.<br>**Note:** The corresponding global parameter is EnableEchoCanceller. |
| Broken Connection Mode<br>`disconnect-on-broken-connection`<br>[IpProfile_DisconnectOnBrokenConnection] | Defines the device's handling of calls when RTP packets (media) are not received within a user-defined timeout (configured by the BrokenConnectionEventTimeout parameter).<br>▪ [0] Ignore = The call is maintained despite no media and is released when signaling ends the call (i.e., SIP BYE).<br>▪ [1] Disconnect = (Default) The device ends the call.<br>**Note:**<br>▪ The device can only detect a broken RTP connection if silence compression is disabled for the RTP session. |

| Parameter | Description |
|---|---|
| | ▪ If during a call the source IP address (from where the RTP packets are received by the device) is changed without notifying the device, the device rejects these RTP packets. To overcome this, configure the DisconnectOnBrokenConnection parameter to 0. By this configuration, the device doesn't detect RTP packets arriving from the original source IP address and switches (after 300 msec) to the RTP packets arriving from the new source IP address.<br>▪ The corresponding global parameter is DisconnectOnBrokenConnection. |
| Input Gain<br>`input-gain`<br>[IpProfile_InputGain] | Defines the pulse-code modulation (PCM) input gain control (in decibels). Defines the level of the received signal for Tel-to-IP calls.<br>The valid range is -32 to 31 dB. The default is 0 dB.<br>**Note:** The corresponding global parameter is InputGain. |
| Voice Volume<br>`voice-volume`<br>[IpProfile_VoiceVolume] | Defines the voice gain control (in decibels). Defines the level of the transmitted signal for IP-to-Tel calls.<br>The valid range is -32 to 31 dB. The default is 0 dB.<br>**Note:** The corresponding global parameter is VoiceVolume. |
| Media IP Version Preference<br>media-ip-version-preference<br>[IpProfile_MediaIPVersionPreference] | Defines the preferred RTP media IP addressing version for outgoing SIP calls (according to RFC 4091 and RFC 4092). The RFCs concern Alternative Network Address Types (ANAT) semantics in the SDP to offer groups of network addresses (IPv4 and IPv6) and the IP address version preference to establish the media stream. The IP address is indicated in the "c=" field (Connection) of the SDP.<br>▪ [0] Only IPv4 = (Default) SDP offer includes only IPv4 media IP addresses.<br>▪ [1] Only IPv6 = SDP offer includes only IPv6 media IP addresses.<br>▪ [2] Prefer IPv4 = SDP offer includes IPv4 and IPv6 media IP addresses, but the first (preferred) media is IPv4.<br>▪ [3] Prefer IPv6 = SDP offer includes IPv4 and IPv6 media IP addresses, but the first (preferred) media is IPv6.<br>To indicate ANAT support, the device uses the SIP Allow header or to enforce ANAT it uses the Require header:<br>Require: sdp-anat<br>In the outgoing SDP, each 'm=' field is associated with an ANAT group. This is done using the 'a=mid:' and 'a=group:ANAT' fields. Each 'm=' field appears under a unique 'a=mid:' number, for example:<br>a=mid:1<br>m=audio 63288 RTP/AVP 0 8 18 101<br>c=IN IP6 3000::290:8fff:fe40:3e21<br>The 'a=group:ANAT' field shows the 'm=' fields belonging to it, using the number of the 'a=mid:' field. In addition, the ANAT group with the preferred 'm=' fields appears first. For example, the preferred group includes 'm=' fields under 'a=mid:1' and 'a=mid3':<br>a=group:ANAT 1 3 |

| Parameter | Description |
|---|---|
| | `a=group:ANAT 2 4`<br>If you configure the parameter to a "prefer" option, the outgoing SDP offer contains two medias which are the same except for the "c=" field. The first media is the preferred address type (and this type is also on the session level "c=" field), while the second media has its "c=" field with the other address type. Both medias are grouped by ANAT. For example, if the incoming SDP contains two medias, one secured and the other non-secured, the device sends the outgoing SDP with four medias:<br>▪ Two secured medias grouped in the first ANAT group, one with IPv4 and the other with IPv6. The first is the preferred type.<br>▪ Two non-secured medias grouped in the second ANAT group, one with IPv4 and the other with IPv6. The first is the preferred type.<br>**Note:**<br>▪ The parameter is applicable only when the device offers an SDP.<br>▪ The IP addressing version is determined according to the first SDP "m=" field.<br>▪ The feature is applicable to any type of media (e.g., audio and video) that has an IP address.<br>▪ The corresponding global parameter is MediaIPVersionPreference. |
| Symmetric MKI<br>`enable-symmetric-mki`<br>[IpProfile_EnableSymmetricMKI] | Enables symmetric MKI negotiation.<br>▪ **[0]** Disable = (Default) The device includes the MKI in its SIP 200 OK response according to the SRTPTxPacketMKISize parameter (if set to 0, it is not included; if set to any other value, it is included with this value).<br>▪ **[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:<br>`a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:TAaxNnQt8/qLQMnDuG4vxYfWl6K7eBK/ufk04pR4 \|2^31\|1:1`<br>`a=crypto:3 AES_CM_128_HMAC_SHA1_80 inline:bnuYZnMxSfUiGitviWJZmzr7OF3AiRO0l5Vnh0kH \|2^31`<br>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:<br>`a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:R1VyA1xV/qwBjkEklu4kSJyl3wCtYeZLq1/QFuxw \|2^31\|1:1`<br>If the device selects a crypto line that does not contain the MKI parameter, then the MKI parameter is not included in the crypto line in the SDP answer (even if the SRTPTxPacketMKISize parameter is set to any value other than 0). |

| Parameter | Description |
|---|---|
| | **Note:** The corresponding global parameter is EnableSymmetricMKI. |
| MKI Size<br>`mki-size`<br>[IpProfile_MKISize] | Defines the size (in bytes) of the Master Key Identifier (MKI) in SRTP Tx packets.<br>The valid value is 0 to 4. The default is 0 (i.e., new keys are generated without MKI).<br>**Notes:**<br>▪ The device only initiates the MKI size.<br>▪ The corresponding global parameter is SRTPTxPacketMKISize. |
| Reset SRTP Upon Re-key<br>`reset-srtp-upon-re-key`<br>[IpProfile_ResetSRTPStateUponRekey] | Enables synchronization of the SRTP state between the device and a server when a new SRTP key is generated upon a SIP session expire. This feature ensures that the roll-over counter (ROC), one of the parameters used in the SRTP encryption/decryption process of the SRTP packets is synchronized on both sides for transmit and receive packets.<br>▪ **[0]** Disable = (Default) ROC is not reset on the device side.<br>▪ **[1]** Enable = If the session expires causing a session refresh through a re-INVITE, the device or server generates a new key and the device resets the ROC index (and other SRTP fields) as done by the server, resulting in a synchronized SRTP.<br>**Notes:**<br>▪ If this feature is disabled and the server resets the ROC upon a re-key generation, one-way voice may occur.<br>▪ The corresponding global parameter is ResetSRTPStateUponRekey. |
| Generate SRTP Keys Mode<br>generate-srtp-keys<br>**[IpProfile_GenerateSRTPKeys]** | Enables the device to generate a new SRTP key upon receipt of a re-INVITE with the SIP entity associated with the IP Profile.<br>▪ **[0]** Only If Required= (Default) The device generates an SRTP key only if necessary.<br>▪ **[1]** Always = The device always generates a new SRTP key. |
| AMD Sensitivity Parameter Suite<br>`amd-sensitivity-parameter-suit`<br>[IpProfile_AMDSensitivityParameterSuit] | Defines the AMD Parameter Suite to use for the Answering Machine Detection (AMD) feature.<br>▪ [0] 0 = (Default) Parameter Suite 0 based on North American English with standard detection sensitivity resolution (8 sensitivity levels, from 0 to 7). This AMD Parameter Suite is provided by the AMD Sensitivity file, which is shipped pre-installed on the device.<br>▪ [1] 1 = Parameter Suite based 1 on North American English with high detection sensitivity resolution (16 sensitivity levels, from 0 to 15). This AMD Parameter Suite is provided by the AMD Sensitivity file, which is shipped pre-installed on the device.<br>▪ [2] 2 to [7]7 = Optional Parameter Suites that you can create based on any language (16 sensitivity levels, from 0 to 15). This requires a customized AMD Sensitivity file that needs to be installed on the device. For more information, contact your AudioCodes sales representative.<br>**Notes:** |

| Parameter | Description |
|---|---|
| | ▪ To configure the detection sensitivity level, use the 'AMD Sensitivity Level' parameter.<br>▪ For more information on the AMD feature, see Answering Machine Detection (AMD) on page 203.<br>▪ The corresponding global parameter is AMDSensitivityParameterSuit. |
| AMD Sensitivity Level<br>`amd-sensitivity-level`<br>[IpProfile_AMDSensitivityLevel] | Defines the AMD detection sensitivity level of the selected AMD Parameter Suite (using the 'AMD Sensitivity Parameter Suite' parameter).<br>For Parameter Suite 0, the valid range is 0 to 7, where 0 is for best detection of an answering machine and 7 for best detection of a live call. For any Parameter Suite other than 0, the valid range is 0 to 15, where 0 is for best detection of an answering machine and 15 for best detection of a live call.<br>**Note:** The corresponding global parameter is AMDSensitivityLevel. |
| AMD Max Greeting Time<br>`amd-max-greeting-time`<br>[IpProfile_AMDMaxGreetingTime] | Defines the maximum duration (in 5-msec units) that the device can take to detect a greeting message.<br>The valid range value is 0 to 51132767. The default is 300.<br>**Note:** The corresponding global parameter is AMDMaxGreetingTime. |
| AMD Max Post Silence Greeting Time<br>`amd-max-post-silence-greeting-time`<br>[IpProfile_AMDMaxPostSilenceGreetingTime] | Defines the maximum duration of silence from after the greeting time is over (configured by AMDMaxGreetingTime) until the device's AMD decision.<br>**Note:** The corresponding global parameter is AMDMaxPostGreetingSilenceTime. |
| GW (Gateway Calls) | |
| Profile Preference<br>`ip-preference`<br>[IpProfile_IpPreference] | Defines the priority of the IP Profile, where 20 is the highest priority and 1 the lowest priority.<br>**Notes:**<br>▪ If an IP Profile and a Tel Profile apply to the same call, the coders and other common parameters of the profile with the highest preference are applied to the call. If the preference of the profiles is identical, the Tel Profile parameters are applied.<br>▪ If the coder lists of both an IP Profile and a Tel Profile apply to the same call, only the coders common to both are used. The order of the coders is determined by the preference.<br>▪ The parameter is applicable only to the Gateway application. |
| Coders<br>`coders-group-id`<br>[IpProfile_CodersGroupID] | Assigns a Coders Group, which defines coders supported by the SIP entity associated with the IP Profile.<br>The value, Default Coders Group represents the coders configured in the Coders table (see Configuring Coders on page 357). All other optional values (e.g., Coders Group 1), represent the coders defined for the specific Coder Group configured in the Coder Group Settings table (see Configuring Coder Groups on page 360). |
| Media Security Mode | Defines the handling of SRTP for the SIP entity associated with the IP Profile. |

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td><code>media-security-behaviour</code><br>[IpProfile_MediaSecurityBehaviour]</td><td><ul><li>[-1] Not Configured = Applies the settings of the corresponding global parameter, MediaSecurityBehaviour.</li><li>[0] Preferable = (Default) The device initiates encrypted calls to this SIP entity. However, if negotiation of the cipher suite fails, an unencrypted call is established. The device accepts incoming calls received from the SIP entity that don't include encryption information.</li><li>[1] Mandatory = The device initiates encrypted calls to this SIP entity, but if negotiation of the cipher suite fails, the call is terminated. The device rejects incoming calls received from the SIP entity that don't include encryption information.</li><li>[2] Disable = This SIP entity does not support encrypted calls (i.e., SRTP).</li><li>[3] Preferable - Single Media = The device sends SDP with a single media ('m=') line only (e.g., m=audio 6000 RTP/AVP 4 0 70 96) with RTP/AVP and crypto keys. The SIP entity can respond with SRTP or RTP parameters:<ul><li>✓ If the SIP entity does not support SRTP, it uses RTP and ignores the crypto lines.</li><li>✓ If the device receives an SDP offer with a single media (as shown above) from the SIP entity, it responds with SRTP (RTP/SAVP) if the EnableMediaSecurity parameter is set to 1. If SRTP is not supported (i.e., EnableMediaSecurity is set to 0), it responds with RTP.</li><li>✓ If two 'm=' lines are received in the SDP offer, the device prefers the SAVP (secure audio video profile), regardless of the order in the SDP.</li></ul></li></ul>Notes:<ul><li>The parameter is applicable only when the EnableMediaSecurity parameter is set to 1.</li><li>The corresponding global parameter is MediaSecurityBehaviour.</li></ul></td></tr>
<tr><td>Is DTMF Used<br>[IpProfile_IsDTMFUsed]</td><td>Enables DTMF signaling.<ul><li>[0] Disable = (Default)</li><li>[1] Enable</li></ul></td></tr>
<tr><td>First Tx DTMF Option<br><code>first-tx-dtmf-option</code><br>[IpProfile_FirstTxDtmfOption]</td><td>Defines the first preferred transmit DTMF negotiation method.<ul><li>[0] Not Supported = No negotiation - DTMF digits are sent according to the parameters DTMFTransportType and RFC2833PayloadType (for transmit and receive).</li><li>[1] INFO (Nortel) = Sends DTMF digits according to IETF Internet-Draft draft-choudhuri-sip-info-digit-00.</li><li>[2] NOTIFY = Sends DTMF digits according to IETF Internet-Draft draft-mahy-sipping-signaled-digits-01.</li><li>[3] INFO (Cisco) = Sends DTMF digits according to the Cisco format.</li><li>[4] RFC 2833 = (Default) The device:<ul><li>✓ negotiates RFC 2833 payload type using local and remote SDPs.</li><li>✓ sends DTMF packets using RFC 2833 payload type according to the payload type in the received SDP.</li></ul></li></ul></td></tr>
</table>

| Parameter | Description |
|---|---|
| | ✓ expects to receive RFC 2833 packets with the same payload type as configured by the parameter RFC2833PayloadType.<br>✓ removes DTMF digits in transparent mode (as part of the voice stream).<br>▪ [5] INFO (Korea) = Sends DTMF digits according to the Korea Telecom format.<br>**Notes:**<br>▪ When out-of-band DTMF transfer is used ([1], [2], [3], or [5]), the DTMFTransportType parameter is automatically set to 0 (DTMF digits are removed from the RTP stream).<br>▪ The corresponding global parameter is FirstTxDTMFOption. |
| Second Tx DTMF Option<br>`second-tx-dtmf-option`<br>[IpProfile_SecondTxDtmfOption] | Defines the second preferred transmit DTMF negotiation method. For a description, see IpProfile_FirstTxDtmfOption (above).<br>**Note:** The corresponding global parameter is SecondTxDTMFOption. |
| Rx DTMF Option<br>`rx-dtmf-option`<br>[IpProfile_RxDTMFOption] | Enables the device to declare the RFC 2833 'telephony-event' parameter in the SDP.<br>▪ [0] Not Supported<br>▪ [3] Supported (default)<br>The device is always receptive to RFC 2833 DTMF relay packets. Thus, it is always correct to include the 'telephony-event' parameter by default in the SDP. However, some devices use the absence of the 'telephony-event' in the SDP to decide to send DTMF digits in-band using G.711 coder. If this is the case, set the parameter to 0.<br>**Note:** The corresponding global parameter is RxDTMFOption. |
| Fax Signaling Method<br>`fax-sig-method`<br>[IpProfile_IsFaxUsed] | Defines the SIP signaling method for establishing and transmitting a fax session when the device detects a fax.<br>▪ [0] No Fax = (Default) No fax negotiation using SIP signaling. The fax transport method is according to the FaxTransportMode parameter.<br>▪ [1] T.38 Relay = Initiates T.38 fax relay.<br>▪ [2] G.711 Transport = Initiates fax/modem using the coder G.711 A-law/Mu-law with adaptations (see Note below).<br>▪ [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/Mu-law with adaptations (see the Note below).<br>**Notes:**<br>▪ Fax adaptations (for options 2 and 3):<br>✓ Echo Canceller = On<br>✓ Silence Compression = Off<br>✓ Echo Canceller Non-Linear Processor Mode = Off<br>✓ Dynamic Jitter Buffer Minimum Delay = 40<br>✓ Dynamic Jitter Buffer Optimization Factor = 13<br>▪ If the device initiates a fax session using G.711 (option 2 or 3), a 'gpmd' attribute is added to the SDP in the following format:<br>✓ For A-law: 'a=gpmd:8 vbd=yes;ecan=on'<br>✓ For Mu-law: 'a=gpmd:0 vbd=yes;ecan=on' |

| Parameter | Description |
|---|---|
| | ▪ When the parameter is set to 1, 2, or 3, the parameter FaxTransportMode is ignored.<br>▪ When the 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.<br>▪ For more information on fax transport methods, see Fax/Modem Transport Modes on page 180.<br>▪ The corresponding global parameter is IsFaxUsed. |
| CNG Detector Mode<br>`cng-mode`<br>[IpProfile_CNGmode] | Enables the detection of the fax calling tone (CNG) and defines the detection method.<br>▪ [0] Disable = (Default) The originating fax does not detect CNG; the device passes the CNG signal transparently to the remote side.<br>▪ [1] Relay = The originating fax detects CNG. The device sends CNG packets to the remote side according to T.38 (if IsFaxUsed is set to 1) and the fax session is started. A SIP Re-INVITE message is not sent and the fax session starts by the terminating fax. This option is useful, for example, when the originating fax is located behind a firewall that blocks incoming T.38 packets on ports that have not yet received T.38 packets from the internal network (i.e., originating fax). To also send a Re-INVITE message upon detection of a fax CNG tone in this mode, set the parameter FaxCNGMode to 1 or 2.<br>▪ [2] Event Only = The originating fax detects CNG and a fax session is started by the originating fax, using the Re-INVITE message. Typically, T.38 fax session starts when the preamble signal is detected by the answering fax. Some SIP devices do not support the detection of this fax signal on the answering fax and thus, in these cases, it is possible to configure the device to start the T.38 fax session when the CNG tone is detected by the originating fax. However, this mode is not recommended.<br>**Note:** The corresponding global parameter is CNGDetectorMode. |
| Vxx Modem Transport Type<br>`vxx-transport-type`<br>[IpProfile_VxxTransportType] | Defines the modem transport type.<br>▪ [-1] = (Not Configured) The settings of the global parameters are used:<br>✓ V21ModemTransportType<br>✓ V22ModemTransportType<br>✓ V23ModemTransportType<br>✓ V32ModemTransportType<br>✓ V34ModemTransportType<br>▪ [0] Disable = Transparent.<br>▪ [2] Enable Bypass (Default)<br>▪ [3] Events Only = Transparent with Events.<br>For a detailed description of the parameter per modem type, see the relevant global parameter (listed above). |
| NSE Mode<br>`nse-mode`<br>[IpProfile_NSEMode] | Enables Cisco's compatible fax and modem bypass mode, Named Signaling Event (NSE) packets.<br>▪ [0] Disable (Default)<br>▪ [1] Enable |

| Parameter | Description |
|---|---|
| | In NSE bypass mode, the device starts using G.711 A-Law (default) or G.711-Law, according to the FaxModemBypassCoderType parameter. The payload type for these G.711 coders is a standard one (8 for G.711 A-Law and 0 for G.711 -Law). The parameters defining payload type for the 'old' Bypass mode FaxBypassPayloadType and ModemBypassPayloadType are not used with NSE Bypass. The bypass packet interval is configured according to the FaxModemBypassBasicRtpPacketInterval parameter.<br>The SDP contains the following line:<br>'a=rtpmap:100 X-NSE/8000'.<br>**Notes:**<br>▪ When enabled, the following conditions must also be met:<br>✓ The Cisco gateway must include the following definition: 'modem passthrough nse payload-type 100 codec g711alaw'.<br>✓ Set the Modem transport type to Bypass mode (VxxModemTransportType is set to 2) for all modems.<br>✓ Set the NSEPayloadType parameter to 100.<br>▪ The corresponding global parameter is NSEMode. |
| Play RB Tone to IP<br>`play-rbt-to-ip`<br>[IpProfile_PlayRBTone2IP] | Enables the device to play a ringback tone to the IP side for IP-to-Tel calls.<br>▪ [0] Disable (Default)<br>▪ [1] Enable = Plays a ringback tone after a SIP 183 session progress response is sent.<br>**Notes:**<br>▪<br>▪ To enable the device to send a 183/180+SDP responses, set the EnableEarlyMedia parameter to 1.<br>▪ 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.<br>▪ The corresponding global parameter is PlayRBTone2IP. |
| Early Media<br>`early-media`<br>[IpProfile_EnableEarlyMedia] | Enables the Early Media feature for sending media (e.g., ringing) before the call is established.<br>▪ [0] Disable (default)<br>▪ [1] Enable<br>✓ Analog: The device sends a SIP 183 Session Progress response with SDP instead of a 180 Ringing, allowing the media stream to be established before the call is answered.<br>**Notes:**<br>▪ See also the IgnoreAlertAfterEarlyMedia parameter. The parameter allows, for example, to interwork Alert with PI to SIP 183 with SDP instead of 180 with SDP.<br>▪ You can also configure early SIP 183 response immediately upon the receipt of an INVITE, using the EnableEarly183 parameter.<br>▪ Analog: To send a 183 response, you must also set the ProgressIndicator2IP parameter to 1. If set to 0, a 180 Ringing response is sent. |

| Parameter | Description |
|---|---|
| | ▪ The corresponding global parameter is EnableEarlyMedia. |
| Progress Indicator to IP<br>`prog-ind-to-ip`<br>[IpProfile_ProgressIndicator2IP] | Defines the progress indicator (PI) sent to the IP.<br>▪ [-1] = (Default) Not configured:<br>✓ Analog: Default values are used (0 for FXS interfaces).<br>✓ Analog: For IP-to-Tel calls, the device sends a 180 Ringing response to IP after placing a call to a phone (FXS).<br>▪ [1] PI = 1:<br>✓ Analog: For IP-to-Tel calls, if the EnableEarlyMedia parameter is set to 1, the device sends a 183 Session Progress message with SDP immediately after a call is placed to a phone/PBX. This is used to cut-through the voice path before the remote party answers the call. This allows the originating party to listen to network call progress tones such as ringback tone or other network announcements.<br>▪ [8] PI = 8: same as PI = 1.<br>**Note:** The corresponding global parameter is ProgressIndicator2IP. |
| Hold<br>`enable-hold`<br>[IpProfile_EnableHold] | Enables the Call Hold feature (analog interfaces). For analog: The Call Hold feature allows users, connected to the device, to place a call on hold (or remove from hold), using the phone's Hook Flash button.<br>▪ [0] Disable<br>▪ [1] Enable (default)<br>**Notes:**<br>▪ Analog interfaces: To use the call hold service, the devices at both ends must support this option.<br>▪ The corresponding global parameter is EnableHold. |
| Copy Destination Number to Redirect Number<br>`copy-dst-to-redirect-number`<br>[IpProfile_CopyDest2RedirectNumber] | Enables the device to copy the called number, received in the SIP INVITE message, to the redirect number in the outgoing Q.931 Setup message, for IP-to-Tel calls. Thus, even if there is no SIP Diversion or History header in the incoming INVITE message, the outgoing Q.931 Setup message will contain a redirect number.<br>▪ [0] Disable (default).<br>▪ [1] After Manipulation = Copies the called number after manipulation. The device first performs IP-to-Tel destination phone number manipulation, and only then copies the manipulated called number to the redirect number sent in the Q.931 Setup message to the Tel. Thus, the called and redirect numbers are the same.<br>▪ [2] Before Manipulation = Copies the called number before manipulation. The device first copies the original called number to the SIP Diversion header, and then performs IP-to-Tel destination phone number manipulation. Thus, the called (i.e., SIP To header) and redirect (i.e., SIP Diversion header) numbers are different.<br>**Note:** The corresponding global parameter is CopyDest2RedirectNumber. |
| Number of Calls Limit | Defines the maximum number of concurrent calls (incoming and outgoing) for the SIP entity associated with the IP Profile. If the |

| Parameter | Description |
|---|---|
| `call-limit`<br>[IpProfile_CallLimit] | number of concurrent calls reaches this limit, the device rejects any new incoming and outgoing calls belonging to this IP Profile.<br>The parameter can also be set to the following:<br>▪ [-1] = (Default) No limitation on calls.<br>▪ [0] = All calls are rejected. |

# Part V

## Gateway Application

# 20 Introduction

This section describes configuration of the Gateway application. The Gateway application refers to IP-to-Tel and Tel-to-IP call routing. For analog interfaces, Tel refers to FXS.

**Notes:**

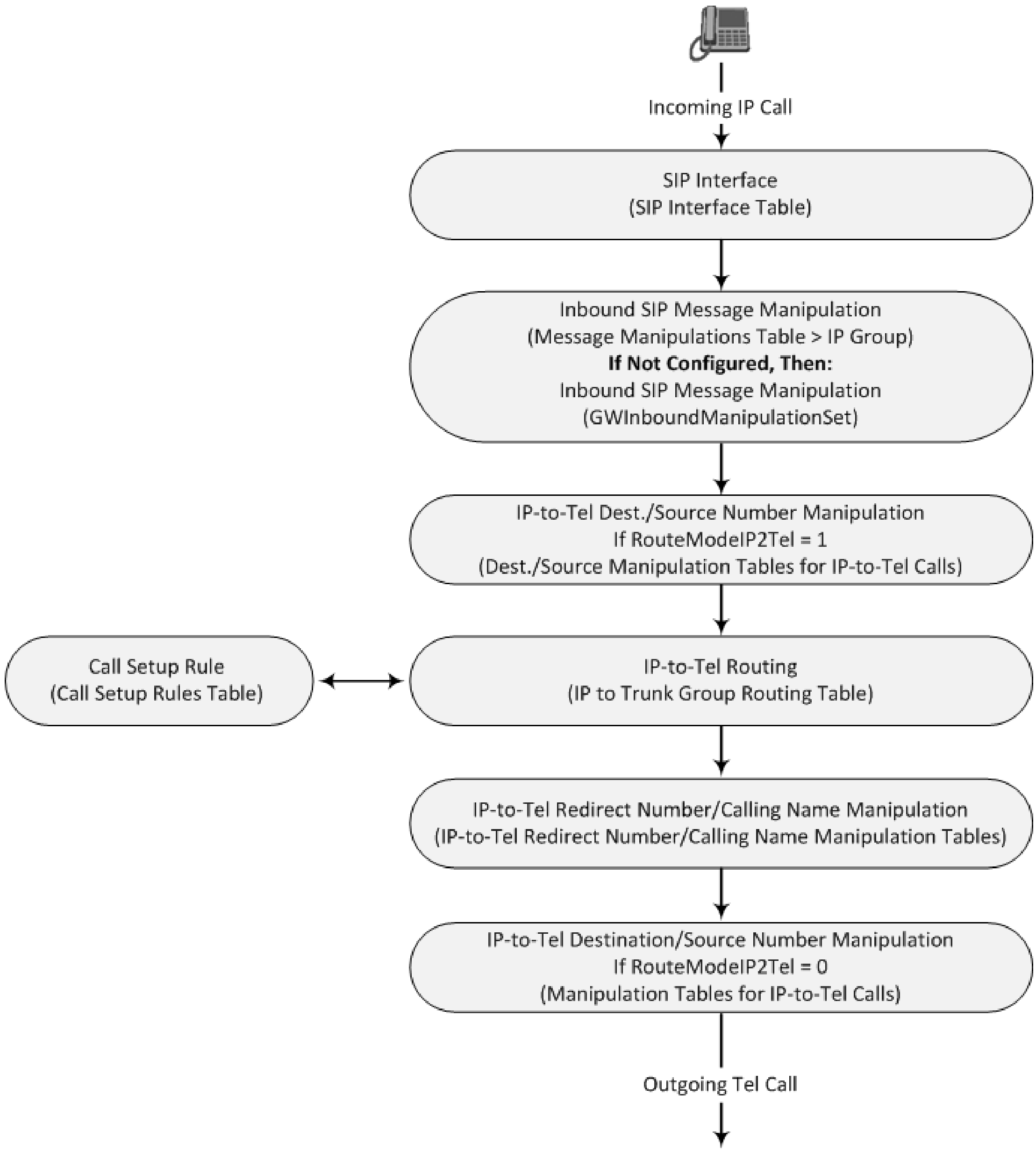
- In some areas of the Web interface, the term "GW" refers to the Gateway application.
- The terms *IP-to-Tel* and *Tel-to-IP* refer to the direction of the call relative to the device. IP-to-Tel refers to calls received from the IP network and destined to the PSTN/PBX (i.e., telephone connected directly or indirectly to the device); Tel-to-IP refers to calls received from telephones connected directly to the device's FXS ports or from the PSTN/PBX, and destined for the IP network.
- 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.

## 20.1 Call Processing Summary

The device's call processing for Gateway calls is shown in the following flowcharts.

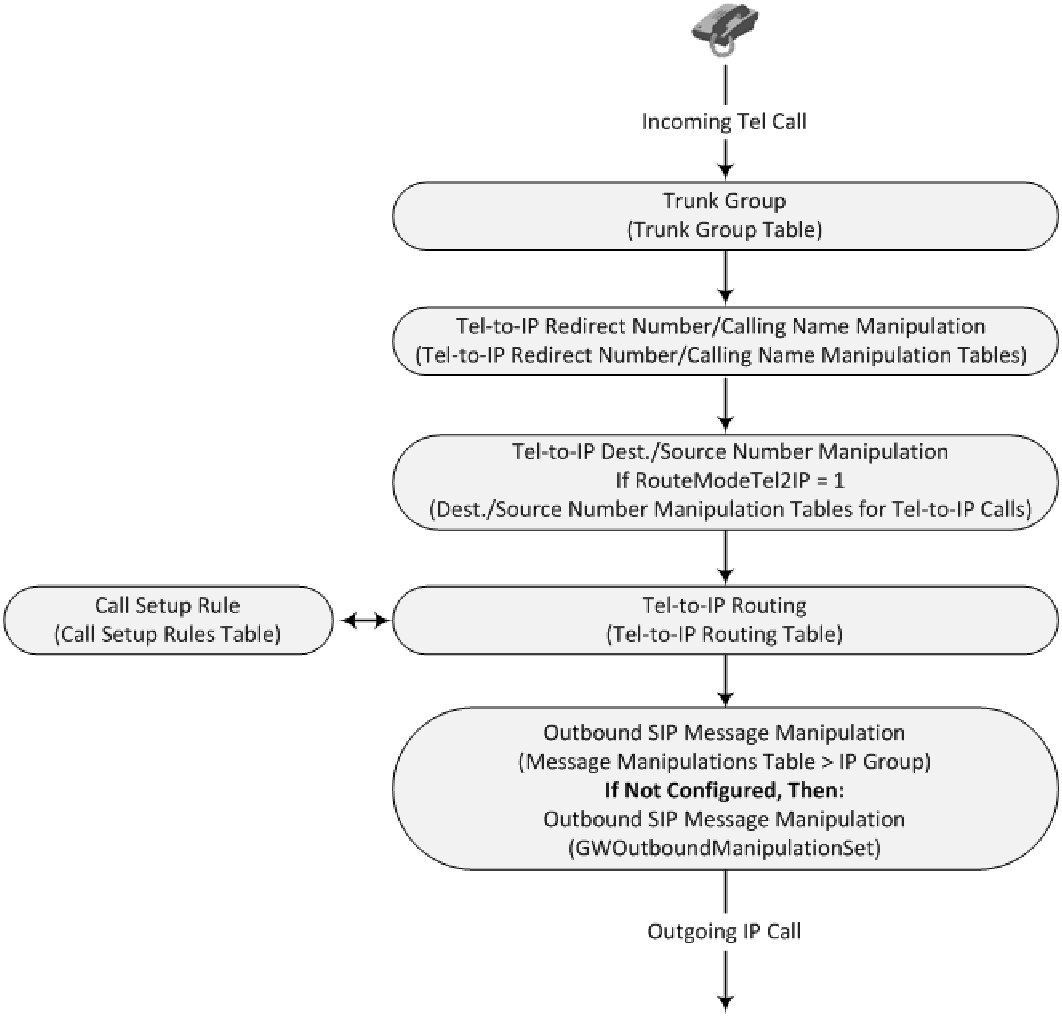
- **IP-to-Tel Call:**

**Figure 20-1: IP-to-Tel Call Processing Flowchart**



- **Tel-to-IP Call:**

**Figure 20-2: Tel-to-IP Call Processing Flowchart**



**This page is intentionally left blank.**

# 21 Hunt Groups

This section describes the configuration of the device's channels, which includes assigning them to Hunt Groups.

## 21.1 Configuring Hunt Groups

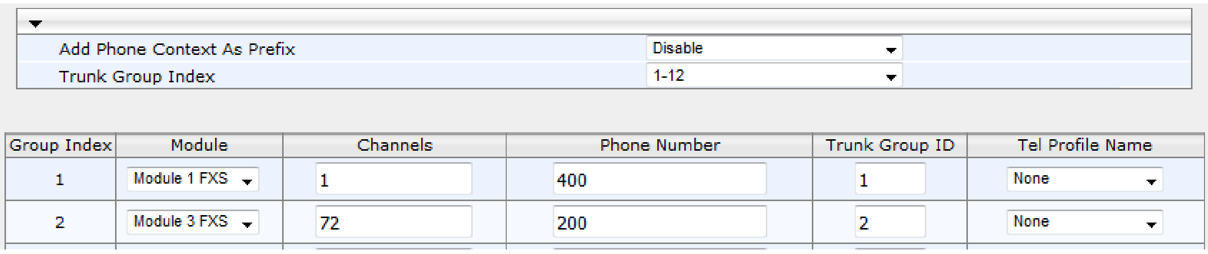
The Hunt Group table lets you configure up to 120 Hunt Groups. A Hunt Group is a logical group of physical channels. A Hunt Group can include ranges of channels. To enable and activate the channels of the device, Hunt Groups need to be configured and assigned with telephone numbers. Channels that are not configured in this table are disabled.

Once you have configured your Hunt Groups, you need to use them for routing incoming IP calls to the Tel side, which is represented by a specific Hunt Group (ID). For configuring IP-to-Tel routing rules, see "Configuring IP-to-Hunt Group Routing Rules" on page 421. You can also use Hunt Groups for routing Tel calls to the IP side. For configuring Tel-to-IP routing rules, see "Configuring Tel-to-IP Routing Rules" on page 409.

The following procedure describes how to configure Hunt Groups through the Web interface. You can also configure it through ini file (TrunkGroup_x) or CLI (configure voip > gw hunt-or-trunk-group trunk-group).

➢ **To configure a Hunt Group:**

1. Open the Hunt Group table (**Configuration** tab > **VoIP** menu > **Gateway** > **Hunt Group** > **Hunt Group**).




2. Configure a Hunt Group according to the parameters described in the table below.
3. Click **Submit**.

You can also register all your Hunt Groups. The registration method per Hunt Group is configured by the 'Registration Mode' parameter in the Hunt Group Settings page (see "Configuring Hunt Group Settings" on page 386).

- To register the Hunt Groups, click the **Register** button, located below the Hunt Group table.
- To unregister the Hunt Groups, click the **Unregister** button, located below the Hunt Group table.

**Table 21-1: Hunt Group Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Module<br>`module`<br>[TrunkGroup_Module] | Defines the telephony interface module (i.e., FXS blade) for which you want to define the Hunt Group. |
| Channels<br>`first-b-channel`<br>[TrunkGroup_FirstBChannel] | Defines the device's channels/ports (analog module). To enable channels, enter the channel numbers.<br>You can enter a range of channels by using the syntax *n-m*, where *n* represents the lower channel number and *m* the higher |

| Parameter | Description |
|---|---|
| `last-b-channel`<br>[TrunkGroup_LastBChannel] | channel number. For example, "1-4" specifies channels 1 through 4. |
| Phone Number<br>`first-phone-number`<br>[TrunkGroup_FirstPhoneNumber] | Defines the telephone number(s) of the channels.<br>The valid value can be up to 50 characters.<br>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.<br>These 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**.<br>**Notes:**<br>▪ 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').<br>▪ This field is optional. The logical numbers defined in this field are used when an incoming Tel call doesn't contain the calling number or called number (the latter being determined by the ReplaceEmptyDstWithPortNumber parameter). These numbers are used to replace them.<br>▪ This field is ignored if routing of IP-to-Tel calls is done according to the Supplementary Services table, where multiple line extension numbers are configured per port (see "Configuring Multi-Line Extensions and Supplementary Services" on page 460). For this routing method, the 'Channel Select Mode' must be set to **Select Trunk By Supplementary Services Table** in the Hunt Group Settings table (see "Configuring Hunt Group Settings" on page 386). |
| Trunk Group ID<br>`trunk-group-id`<br>[TrunkGroup_TrunkGroupNum] | Defines the Hunt Group ID for the specified channels. The same Hunt Group ID can be assigned to more than one group of channels. If an IP-to-Tel call is assigned to a Hunt Group, the IP call is routed to the channel(s) pertaining to that Hunt Group ID.<br>The valid value can be 0 to 119. |
| Tel Profile Name<br>`tel-profile-id`<br>[TrunkGroup_ProfileName] | Assigns a Tel Profile to the Hunt Group.<br>For configuring Tel Profiles, see "Configuring Tel Profiles" on page 362. |

## 21.2 Configuring Hunt Group Settings Table

The Hunt Group Settings table lets you configure various settings per Hunt Group, which are configured in the Hunt Group table. The main configuration includes the following:

- Channel select method, which defines how the device allocates IP-to-Tel calls to the channels of a Hunt Group.
- Registration method for registering Hunt Groups to remote IP servers (*Serving IP Group*).

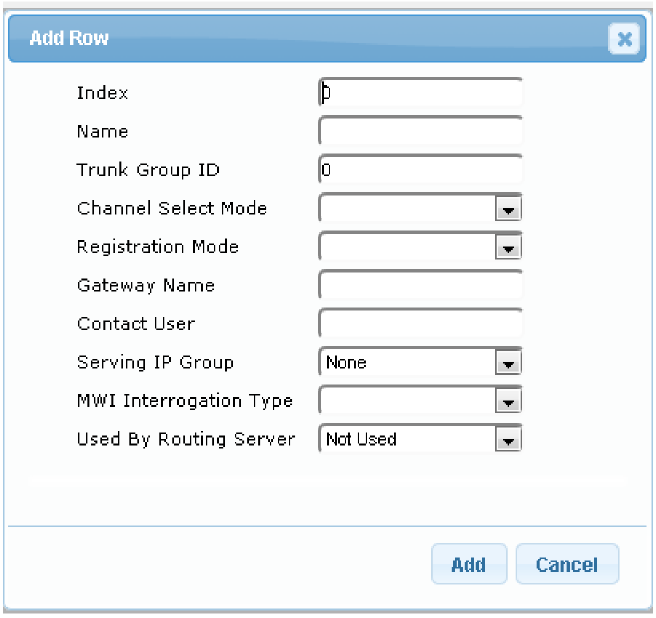
The Hunt Group Settings table also provides an **Action** drop-down button with commands that let you perform various actions per configured Hunt Group:

- **Lock / Unlock:** Locks (blocks) a Hunt Group in order to take its channels out-of-service. For more information, see 'Locking and Unlocking Hunt Groups' on page 491.
- **Register / Un-Register:** Initiates a registration request for the Hunt Group with a Serving IP Group. For more information, see the description of the 'Registration Mode' parameter of the Hunt Group Settings table in this section.

The following procedure describes how to configure settings for Hunt Groups through the Web interface. You can also configure it through ini file (TrunkGroupSettings) or CLI (configure voip/gw hunt-or-trunk-group trunk-group-setting).

### ➢ To configure Hunt Group settings:

1. Open the Hunt Group Settings table (**Configuration** tab > **VoIP** menu > **Gateway** > **Hunt Group** > **Hunt Group Settings**).
2. Click **Add**; the following dialog box appears:




3. Configure a Hunt Group according to the parameters described in the table below.
4. Click **Add**.

**Table 21-2: Hunt Group Settings Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[TrunkGroupSettings_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>`trunk-group-name`<br>[TrunkGroupSettings_TrunkGroupName] | Defines an arbitrary name to easily identify the row. The name represents the Hunt Group in the SIP 'tgrp' parameter in outgoing INVITE messages (according to RFC 4904). For example:<br>`sip:+16305550100;tgrp=`**`TG-1`**`;trunk-context=+1-630@isp.example.net;user=phone`<br>The valid value can be a string of up to 40 characters. By default, no name is configured.<br>**Notes:**<br>▪ Each row must be configured with a unique name. |

| Parameter | Description |
|---|---|
| | ▪ If the parameter is not configured, the Hunt Group decimal number is used in the SIP 'tgrp' parameter.<br>▪ The feature is enabled by any of the following parameters:<br>✓ UseSIPtgrp<br>✓ UseBroadsoftDTG |
| Trunk Group ID<br>trunk-group-id<br>[TrunkGroupSettings_TrunkGroupId] | Defines the Hunt Group ID that you want to configure. |
| Channel Select Mode<br>`channel-select-mode`<br>[TrunkGroupSettings_ChannelSelectMode] | Defines the method by which IP-to-Tel calls are assigned to the channels of the Hunt Group.<br>▪ **[0]** By Dest Phone Number = The channel is selected according to the called (destination) number. If the number is not located, the call is released. If the channel is unavailable (e.g., busy), the call is put on call waiting (if call waiting is enabled and no other call is on call waiting); otherwise, the call is released.<br>▪ **[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.<br>▪ **[2]** Ascending = The lowest available channel in the Hunt Group is selected, and if unavailable, the next higher channel is selected.<br>▪ **[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.<br>▪ **[4]** Descending = The highest available channel in the Hunt Group is selected, and if unavailable, the next lower channel is selected.<br>▪ **[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.<br>**Note:** If the called number is located, but the port associated with the number is busy, the call is released.<br>▪ **[6]** By Source Phone Number = The channel is selected according to the calling number.<br>▪ [9] Ring to Hunt Group = The device allocates IP-to-Tel calls to all the FXS ports (channels) in the Hunt Group (i.e., a ringing 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.<br>▪ [10] Select Hunt by Supplementary Services Table = The FXS port is selected according to the settings in the Supplementary |

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>Services table (see Configuring Multi-Line Extensions and Supplementary Services on page 460), allowing the routing of IP-to-Tel calls to specific endpoints according to extension number. This option is applicable only to FXS interfaces.
<ul>
<li>[11] Dest Number + Ascending = The device allocates a channels to incoming IP-to-Tel calls as follows:
<ol type="a">
<li>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.</li>
<li>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 Hunt Group. If located, the call is sent to that channel.</li>
<li>If all the channels are unavailable, the call is released.</li>
</ol></li>
</ul>
Note: If the parameter is not configured, the Hunt Group's channel select method is according to the global parameter, ChannelSelectMode.</td></tr>
<tr><td>Registration Mode<br><code>registration-mode</code><br>[TrunkGroupSettings_RegistrationMode]</td><td>Defines the registration method of the Hunt Group:
<ul>
<li>[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.</li>
<li>[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.</li>
<li>[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'.</li>
<li>[5] Per Account = Registrations are sent (or not) to an IP Group, according to the settings in the Account table (see "Configuring Registration Accounts" on page 341).</li>
</ul>
An example is shown below of a REGISTER message for registering endpoint "101" using the registration Per Endpoint mode:
<pre>REGISTER sip:SipGroupName SIP/2.0
Via: SIP/2.0/UDP
10.33.37.78;branch=z9hG4bKac862428454
From: <sip:101@GatewayName>;tag=1c862422082
To: <sip:101@GatewayName>
Call-ID: 9907977062512000232825@10.33.37.78
CSeq: 3 REGISTER
Contact: <sip:101@10.33.37.78>;expires=3600
Expires: 3600
User-Agent: Sip-Gateway/v.6.80A.227.005
Content-Length: 0</pre></td></tr>
</table>

| Parameter | Description |
|---|---|
| | The "SipGroupName" in the Request-URI is configured in the IP Group table (see "Configuring IP Groups" on page 323).<br>**Notes:**<br>▪ If the parameter is not configured, the registration is performed according to the global registration parameter, ChannelSelectMode.<br>▪ To enable Hunt Group registration, set the global parameter, IsRegisterNeeded to 1. This is unnecessary for 'Per Account' registration mode.<br>▪ If the device is configured globally to register Per Endpoint and an channel group includes four channels to register Per Gateway, the device registers all channels except the first four channels. The group of these four channels sends a single registration request. |
| Gateway Name<br>`gateway-name`<br>[TrunkGroupSettings_GatewayName] | Defines the host name for the SIP From header in INVITE messages, and the From and To headers in REGISTER requests.<br>**Note:** If the parameter is not configured, the global parameter, SIPGatewayName is used. |
| Contact User<br>`contact-user`<br>[TrunkGroupSettings_ContactUser] | Defines the user part for the SIP Contact URI in INVITE messages, and the From, To, and Contact headers in REGISTER requests.<br>**Notes:**<br>▪ The parameter is applicable only if the 'Registration Mode' parameter is set to 'Per Account' and registration through the Account table is successful.<br>▪ If registration fails, the user part in the INVITE Contact header is set to the source party number.<br>▪ The 'Contact User' parameter in the Account table overrides this parameter (see "Configuring Registration Accounts" on page 341). |

| Parameter | Description |
|---|---|
| Serving IP Group<br>`serving-ip-group`<br>[TrunkGroupSettings_ServingIPGroupName] | Assigns an IP Group to where the device sends INVITE messages for calls received from the Hunt Group. The actual destination to where the INVITE messages are sent is according to the Proxy Set 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 323).<br>**Notes:**<br>▪ If the parameter is not configured, the INVITE messages are sent to the default Proxy or according to the Tel-to-IP Routing table (see "Configuring Tel-to-IP Routing Rules" on page 409).<br>▪ If the PreferRouteTable parameter is set to 1 (see "Configuring Proxy and Registration Parameters" on page 346), the routing rules in the Tel-to-IP Routing table take precedence over the selected Serving IP Group ID. |
| MWI Interrogation Type<br>`mwi-interrogation-type`<br>[TrunkGroupSettings_MWIInterrogationType] | Defines message waiting indication (MWI) QSIG-to-IP interworking for interrogating MWI supplementary services:<br>▪ [255] Not Configured<br>▪ [0] None = Disables the feature.<br>▪ [1] Use Activate Only = MWI Interrogation messages are not sent and only "passively" responds to MWI Activate requests from the PBX.<br>▪ [2] Result Not Used = MWI Interrogation messages are sent, but the result is not used. Instead, the device waits for MWI Activate requests from the PBX.<br>▪ [3] Use Result = MWI Interrogation messages are sent, its results are used, and the MWI Activate requests are used. MWI Activate requests are interworked to SIP NOTIFY MWI messages. The SIP NOTIFY messages are sent to the IP Group defined by the NotificationIPGroupID parameter.<br>**Note:** The parameter appears in the table only if the VoiceMailInterface parameter is set to 3 (QSIG) (see Configuring Voice Mail on page 465). |
| Used By Routing Server<br>`used-by-routing-server`<br>[TrunkGroupSettings_UsedByRoutingServer] | Enables the use of the Hunt Group by a routing server for routing decisions.<br>▪ [0] Not Used (default)<br>▪ [1] Used<br>For more information, see Centralized Third-Party Routing Server on page 272. |
| Admin State | (Read-only) Displays the administrators state:<br>▪ "Locked": The **Lock** command has been chosen from the **Action** drop-down button.<br>▪ "Unlocked": The **Unlock** command has been chosen from the **Action** drop-down button. |

| Parameter | Description |
|---|---|
| Status | (Read-only) Displays the current status of the channels in the Hunt Group:<br>▪ "In Service": Indicates that all channels in the Hunt Group are in service, for example, when the Hunt Group is unlocked or Busy Out state cleared (see the EnableBusyOut parameter for more information).<br>▪ "Going Out Of Service": Appears as soon as you choose the **Lock** command and indicates that the device is starting to lock the Hunt Group and take channels out of service.<br>▪ "Going Out Of Service (<duration remaining of graceful period> sec / <number of calls still active> calls)": Appears when the device is locking the Hunt Group and indicates the number of buys channels and the time remaining until the graceful period ends, after which the device locks the channels regardless of whether the call has ended or not.<br>▪ "Out Of Service": All fully configured channels in the Hunt Group are out of service, for example, when the Hunt Group is locked or in Busy Out state (see the EnableBusyOut parameter). |

# 22 Manipulation

This section describes the configuration of various manipulation processes.

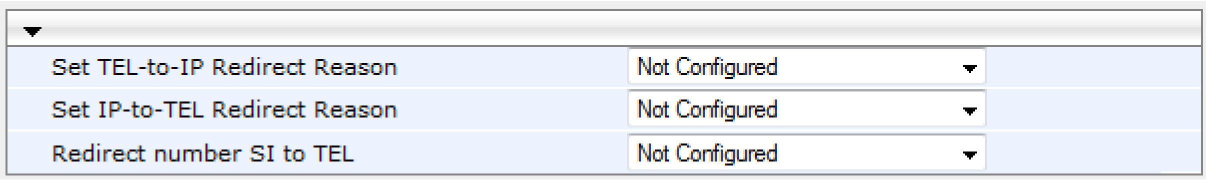
## 22.1 Configuring General Settings

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 643.

➢ **To configure the general manipulation parameters:**

1. Open the General Settings page (**Configuration** tab > **VoIP** menu > **Gateway** > **Manipulations** >**General Settings**).

**Figure 22-1: General Settings Page**




2. Configure the parameters as required.
3. Click **Submit**.

## 22.2 Configuring Source/Destination Number Manipulation Rules

The number manipulation tables let you configure rules for manipulating source and destination telephone numbers for IP-to-Tel and Tel-to-IP calls. The number manipulation tables include the following:

- **Tel-to-IP calls:**
  - Source Phone Number Manipulation Table for Tel-to-IP Calls (up to 120 entries)
  - Destination Phone Number Manipulation Table for Tel-to-IP Calls (up to 120 entries)
- **IP-to-Tel calls:**
  - Source Phone Number Manipulation Table for IP-to-Tel Calls (up to 120 entries)
  - Destination Phone Number Manipulation Table for IP-to-Tel Calls (up to 120 entries)

Configuration of number manipulation rules includes two areas:

- **Rule:** Defines the matching characteristics of the incoming call (e.g., prefix of destination number).
- **Action:** Defines the action that is done if the incoming call matches the characteristics of the rule (e.g., removes a user-defined number of digits from the left of the number).

The device searches the table from top to bottom for the first rule that matches the characteristics of the incoming call. If it finds a matching rule, it applies the manipulation configured for that rule. 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 configure the source prefix number as "551" for rule index 1 and "55" for rule index 2, the device uses rule index 1 for numbers that start with 551 and uses rule index 2 for numbers that start with 550, 552, 553, and so on until 559. However, if you configure the source prefix number as "55" for rule index 1 and "551" for rule index 2, the device applies rule index 1 to all numbers that start with 55, including numbers that start with 551. If the device doesn't find a matching rule, no manipulation is done on the call.You can perform a second "round" (additional) of source and destination number manipulations for IP-to-Tel calls, on an already manipulated number. The initial and additional number manipulation rules are both configured in the number manipulation tables for IP-to-Tel calls. The additional manipulation is performed on the initially manipulated number. Thus, for complex number manipulation schemes, you only need to configure relatively few manipulation rules in these tables (that would otherwise require many rules). To enable this additional manipulation, use the following parameters:

- Source number manipulation - PerformAdditionalIP2TELSourceManipulation
- Destination number manipulation - PerformAdditionalIP2TELDestinationManipulation

Telephone number manipulation can be useful, for example, for the following:

- Stripping or adding dialing plan digits from or to the number, respectively. For example, a user may need to first dial 9 before dialing the phone number to indicate an external line. This number 9 can then be removed by number manipulation before the call is setup.
- Allowing or blocking Caller ID information according to destination or source prefixes. For more information on Caller ID, see Configuring Caller Display Information on page 476.

**Notes:**

- Number manipulation can be performed 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, use the 'IP to Tel Routing Mode' parameter (RouteModeIP2Tel) and 'Tel to IP Routing Mode' parameter (RouteModeTel2IP).
- The device manipulates the number in the following order: 1) strips digits from the left of the number, 2) strips digits from the right of the number, 3) retains the defined number of digits, 4) adds the defined prefix, and then 5) adds the defined suffix.

The following procedure describes how to configure number manipulation rules through the Web interface. You can also configure this using the following management tools:

- **Destination Phone Number Manipulation Table for IP-to-Tel Calls table:** ini file table parameter, NumberMapIP2Tel or CLI command, configure voip/gw manipulations dst-number-map-ip2tel
- **Destination Phone Number Manipulation Table for Tel-to-IP Calls table:** ini file table parameter, NumberMapTel2IP or CLI command, configure voip/gw manipulations dst-number-map-tel2ip
- **Source Phone Number Manipulation Table for IP-to-Tel Calls table:** ini file table parameter, SourceNumberMapIP2Tel or CLI command, configure voip/gw manipulations src-number-map-ip2tel
- **Source Phone Number Manipulation Table for Tel-to-IP Calls table:** ini file table parameter, SourceNumberMapTel2IP or CLI command, configure voip/gw manipulations src-number-map-tel2ip

➢ **To configure a number manipulation rule:**

1. Open the required Number Manipulation page (**Configuration** tab > **VoIP** menu > **Gateway** > **Manipulations** > **Dest Number IP->Tel**, **Dest Number Tel->IP**, **Source Number IP->Tel**, or **Source Number Tel->IP**); the relevant Manipulation table page is

displayed.

2. Click **Add**; the following dialog box appears:

**Figure 22-2: Number Manipulation Table (Example) - Add Row Dialog Box**


3. Configure a number manipulation rule according to the parameters described in the table below.
4. Click **Add**.

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 |
|---|---|---|---|---|---|
| **Destination Prefix** | 03 | | * | * | [6,7,8] |
| **Source Prefix** | 201 | 1001 | 123451001# | [30-40]x | 2001 |
| **Stripped Digits from Left** | - | 4 | - | - | 5 |
| **Stripped Digits from Right** | - | - | - | 1 | - |
| **Prefix to Add** | 971 | 5 | - | 2 | 3 |

| Parameter | Rule 1 | Rule 2 | Rule 3 | Rule 4 | Rule 5 |
|---|---|---|---|---|---|
| **Suffix to Add** | - | 23 | 8 | - | - |
| **Number of Digits to Leave** | - | - | 4 | - | - |
| **Presentation** | Allowed | Restricted | - | - | - |

**Table 22-1: Number Manipulation Tables Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>[_ManipulationName] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters. By default, no value is defined. |
| **Rule (Matching Characteristics)** | |
| Source IP Address<br>`src-ip-address`<br>[_SourceAddress] | Defines the source IP address of the caller. This is obtained from the Contact header in the INVITE message.<br>The default is the asterisk (*) wildcard (i.e., any address).<br>**Notes:**<br>▪ The parameter is applicable only to the number manipulation tables for IP-to-Tel calls.<br>▪ 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.<br>▪ 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. |
| Destination IP Group<br>`dst-ip-group-id`<br>[_DestIPGroupID] | Defines the IP Group to where the call is sent.<br>The default is Any (i.e., any IP Group).<br>**Note:** The parameter is applicable only to the Destination Phone Number Manipulation Table for Tel -> IP Calls. |
| Source Trunk Group<br>`src-trunk-group-id`<br>[_SrcTrunkGroupID] | Defines the source Hunt Group ID for Tel-to-IP calls.<br>The default is -1 (i.e., any Hunt Group).<br>**Note:** The parameter is applicable only to the number manipulation tables for Tel-to-IP calls. |
| Source Prefix<br>`src-prefix`<br>[_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 639.<br>The default is the asterisk (*) wildcard (i.e., any prefix). |

| Parameter | Description |
|---|---|
| Source Host Prefix<br>`src-host-prefix`<br>[_SrcHost] | Defines the URI host name prefix of the incoming SIP INVITE message in the From header.<br>The default is the asterisk (*) wildcard (i.e., any prefix).<br>**Notes:**<br>▪ The parameter is applicable only to the number manipulation tables for IP-to-Tel calls.<br>▪ If the P-Asserted-Identity header is present in the incoming INVITE message, then the value of the parameter is compared to the P-Asserted-Identity URI host name (instead of the From header). |
| Destination Prefix<br>`dst-prefix`<br>[_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 639.<br>The default is the asterisk (*) wildcard (i.e., any prefix). |
| Destination Host Prefix<br>`dst-host-prefix`<br>[_DestHost] | Defines the Request-URI host name prefix of the incoming SIP INVITE message.<br>The default is the asterisk (*) wildcard (i.e., any prefix).<br>**Note:** The parameter is applicable only to the number manipulation tables for IP-to-Tel calls. |
| Source IP Group<br>`src-ip-group-id`<br>[_SrcIPGroupID] | Defines the IP Group from where the IP call originated. Typically, the IP Group of an incoming INVITE is determined by the Inbound IP Routing table.<br>The default is Any (i.e., any IP Group).<br>**Note:** The parameter is applicable only to the number manipulation tables for IP-to-Tel calls. |
| **Operation (Action)** | |
| Stripped Digits From Left<br>`remove-from-left`<br>[_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. |
| Stripped Digits From Right<br>`remove-from-right`<br>[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. |
| Number of Digits to Leave<br>`num-of-digits-to-leave`<br>[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. |
| Prefix to Add<br>`prefix-to-add`<br>**[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. |
| Suffix to Add<br>`suffix-to-add`<br>[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. |

| Parameter | Description |
|---|---|
| TON<br>`ton`<br>[NumberType] | Defines the Type of Number (TON).<br>▪ If you selected 'Unknown' for the NPI, you can select Unknown [0].<br>▪ If you selected 'Private' for the NPI, you can select Unknown [0], Level 2 Regional [1], Level 1 Regional [2], PISN Specific [3] or Level 0 Regional (Local) [4].<br>▪ If you selected 'E.164 Public' for the NPI, you can select Unknown [0], International [1], National [2], Network Specific [3], Subscriber [4] or Abbreviated [6].<br>The default is 'Unknown'.<br>**Notes:**<br>▪ The parameter is applicable only to number manipulation tables for IP-to-Tel calls.<br>▪ TON can be used in the SIP Remote-Party-ID header by using the EnableRPIHeader and AddTON2RPI parameters.<br>▪ For more information on available NPI/TON values, see Numbering Plans and Type of Number on page 408. |
| NPI<br>`npi`<br>[NumberPlan] | Defines the Numbering Plan Indicator (NPI).<br>▪ [0] Unknown (default)<br>▪ [9] Private<br>▪ [1] E.164 Public<br>▪ [-1] Not Configured = value received from PSTN/IP is used<br>**Notes:**<br>▪ The parameter is applicable only to number manipulation tables for IP-to-Tel calls.<br>▪ NPI can be used in the SIP Remote-Party-ID header by using the EnableRPIHeader and AddTON2RPI parameters.<br>▪ For more information on available NPI/TON values, see Numbering Plans and Type of Number on page 408. |
| Presentation<br>`is-presentation-restricted`<br>[IsPresentationRestricted] | Enables caller ID.<br>▪ Not Configured = Privacy is determined according to the Caller ID table (see Configuring Caller Display Information on page 476).<br>▪ **[0]** Allowed = Sends Caller ID information when a call is made using these destination/source prefixes.<br>▪ **[1]** Restricted = Restricts Caller ID information for these prefixes.<br>**Notes:**<br>▪ This field is applicable only to number manipulation tables for source phone number manipulation.<br>▪ 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. |

## 22.3 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 393): 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 22-2: 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.

## 22.4 SIP Calling Name Manipulations

The calling name manipulation tables lets you configure up to 120 manipulation rules for manipulating the calling name (i.e., caller ID) in SIP messages, for IP-to-Tel and Tel-to-IP calls. Manipulation includes modifying or removing the calling name. The calling name manipulation tables include the following:

- Calling Name Manipulation Table for IP-to-Tel Calls table
- Calling Name Manipulation Table for Tel-to-IP Calls table

For example, assume that an incoming SIP INVITE message includes the following header:

```
P-Asserted-Identity: "company:john" sip:6666@78.97.79.104
```

Using the Calling Name Manipulation Table for IP-to-Tel table, the text "company" can be changed to "worker" in the outgoing INVITE, as shown below:

```
P-Asserted-Identity: "worker:john" sip:996666@10.13.83.10
```

Configuration of calling name manipulation rules includes two areas:

- **Rule:** Defines the matching characteristics of an incoming call (e.g., prefix of destination number).
- **Action:** Defines the action that is done if the incoming call matches the characteristics of the rule (e.g., removes a user-defined number of digits from the left of the calling name).

The device searches the table from top to bottom for the first rule that matches the characteristics of the incoming call. If it finds a matching rule, it applies the manipulation configured for that rule.

**Note:** For using the Calling Name Manipulation Table for Tel-to-IP Calls table for retrieving the calling name (display name) from an Active Directory using LDAP queries, see Querying the AD for Calling Name on page 259.

The following procedure describes how to configure calling name manipulation rules through the Web interface. You can also configure these rules using the the following management tools:

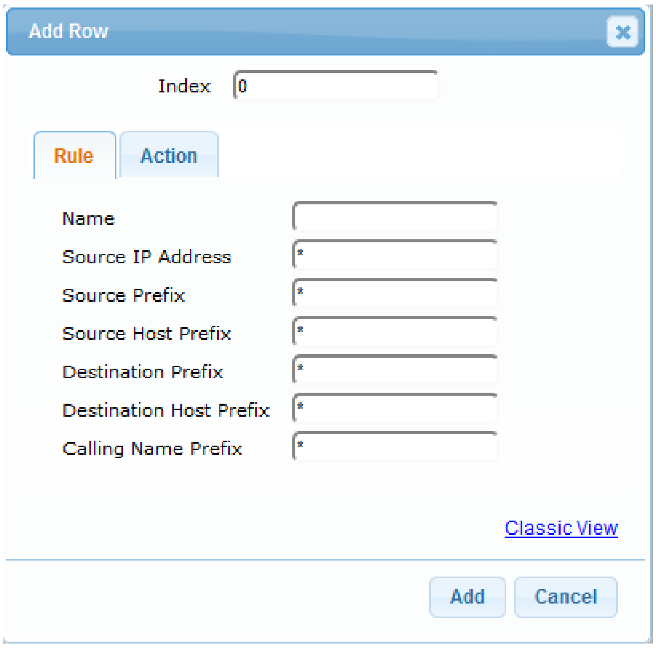
- Calling Name Manipulation Table for Tel-to-IP Calls table: *ini* file (CallingNameMapTel2Ip) or CLI (configure voip/gw manipulations calling-name-map-tel2ip)
- Calling Name Manipulation Table for IP-to-Tel Calls table: *ini* file (CallingNameMapIp2Tel) or CLI (configure voip/gw manipulations calling-name-map-ip2tel)

**➢ To configure calling name manipulation rules:**

1. Open the required calling name manipulations page (**Configuration** tab > **VoIP** menu > **Gateway** > **Manipulations** > **Calling Name IP->Tel** or **Calling Name Tel->IP**).

2. Click **Add**; the following dialog box appears:

**Figure 22-3: Calling Name Manipulation Table (Example) - Add Row Dialog Box**




3. Configure a manipulation rule according to the parameters described in the table below.
4. Click **Add**.

**Table 22-3: Calling Name Manipulation Tables Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>`manipulation-name`<br>[_ManipulationName] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters. |
| **Rule (Matching Characteristics)** | |
| Destination Prefix<br>`dst-prefix`<br>[_DestinationPrefix] | Defines the destination (called) telephone number prefix and/or suffix.<br>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 639.<br>The default value is the asterisk (*) symbol (i.e., any destination prefix). |
| Source Prefix<br>`src-prefix`<br>[_SourcePrefix] | Defines the source (calling) telephone number prefix and/or suffix.<br>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 639. |

| Parameter | Description |
|---|---|
| | The default value is the asterisk (*) symbol (i.e., any source prefix). |
| Calling Name Prefix<br>`calling-name-prefix`<br>[_CallingNamePrefix] | Defines the caller name (i.e., caller ID) prefix.<br>You can use special notations for denoting the prefix. For example, to denote calls without a calling name, use the $ sign. For a description of available notations, see "Dialing Plan Notation for Routing and Manipulation Tables" on page 639.<br>The default value is the asterisk (*) symbol (i.e., any calling name prefix). |
| Source Trunk Group ID<br>`src-trunk-group-id`<br>[_SrcTrunkGroupID] | Defines the source Hunt Group ID from where the Tel-to-IP call was received.<br>The default value is -1, which denotes any Hunt Group.<br>**Note:** The parameter is applicable only to the Calling Name Manipulation Table for Tel-to-IP Calls table. |
| Source IP Address<br>`src-ip-address`<br>[_SourceAddress] | Defines the source IP address of the caller, for IP-to-Tel calls. The source IP address appears in the SIP Contact header in the INVITE message.<br>The default value is the asterisk (*) symbol (i.e., any IP address). The source IP address can include the following wildcards:<br>▪ "x" wildcard: represents single digits. For example, 10.8.8.xx represents all IP addresses between 10.8.8.10 to 10.8.8.99.<br>▪ "*" (asterisk) wildcard: represents any number between 0 and 255. For example, 10.8.8.* represents all IP addresses between 10.8.8.0 and 10.8.8.255.<br>**Note:** The parameter is applicable only to the Calling Name Manipulation Table for IP-to-Tel Calls table. |
| Source Host Prefix<br>`src-host-prefix`<br>[_SrcHost] | Defines the URI host name prefix of the incoming SIP INVITE message in the From header.<br>The default value is the asterisk (*) symbol (i.e., any source host prefix).<br>**Notes:**<br>▪ The parameter is applicable only to the Calling Name Manipulation Table for IP-to-Tel Calls table.<br>▪ If the P-Asserted-Identity header is present in the incoming INVITE message, the value of the parameter is compared to the P-Asserted-Identity URI host name (instead of the From header). |
| Destination Host Prefix<br>`dst-host-prefix`<br>[_DestHost] | Defines the Request-URI host name prefix of the incoming SIP INVITE message.<br>The default value is the asterisk (*) symbol (i.e., any destination host prefix).<br>**Note:** The parameter is applicable only to the Calling Name Manipulation Table for IP-to-Tel Calls table. |
| **Operation (Action)** | |
| Stripped Characters From Left<br>`remove-from-left`<br>[_RemoveFromLeft] | 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". |

| Parameter | Description |
|---|---|
| Stripped Characters From Right<br>`remove-from-right`<br>[_RemoveFromRight] | Defines the number of characters to remove from the right of the calling name. For example, if you enter 3 and the calling name is "company:name", the new name is "company:n". |
| Number of Characters to Leave<br>`num-of-digits-to-leave`<br>[LeaveFromRight] | Defines the number of characters that you want to keep from the right of the calling name. For example, if you enter 4 and the calling name is "company:name", the new name is "name". |
| Prefix to Add<br>`prefix-to-add`<br>[_Prefix2Add] | Defines the number or string to add at the front of the calling name. For example, if you enter ITSP and the calling name is "company:name", the new name is ITSPcompany:john". |
| Suffix to Add<br>`suffix-to-add`<br>[_Suffix2Add] | Defines the number or string to add at the end of the calling name. For example, if you enter 00 and calling name is "company:name", the new name is "company:name00". |

## 22.5 Configuring Redirect Number IP to Tel

The redirect number manipulation tables let you configure rules for manipulating the redirect number received in SIP messages. The redirect number manipulation tables include:

- Redirect Number Tel to IP table: This table defines Tel-to-IP redirect number manipulation. You can manipulate the prefix of the redirect number, received from the Tel side, in the outgoing SIP Diversion, Resource-Priority, or History-Info headers sent to the IP side. This is configured in the Redirect Number Tel-to-IP table.

Configuration of redirect number manipulation rules includes two areas:

- **Rule:** Defines the matching characteristics of an incoming call (e.g., prefix of redirect number).
- **Action:** Defines the action that is done if the incoming call matches the characteristics of the rule (e.g., removes a user-defined number of digits from the left of the redirect number).
- The device searches the table from top to bottom for the first rule that matches the characteristics of the incoming call. If it finds a matching rule, it applies the manipulation configured for that rule.

**Notes:**

- If the device copies the received destination number to the outgoing SIP redirect number (enabled by the CopyDest2RedirectNumber parameter), no redirect number Tel-to-IP manipulation is done.
- The manipulation rules are done in the following order: Stripped Digits From Left, Stripped Digits From Right, Number of Digits to Leave, Prefix to Add, and then Suffix to Add.
- The device uses the 'Redirect Prefix' parameter before it manipulates the prefix.

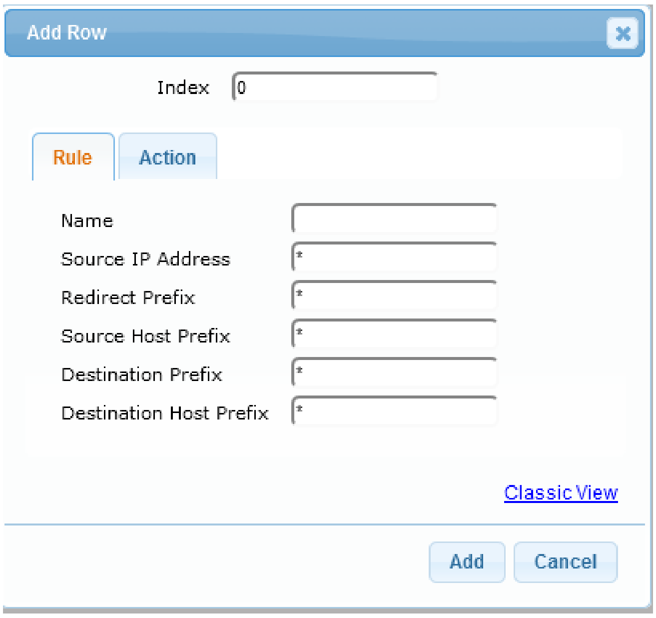
The following procedure describes how to configure redirect number manipulation rules through the Web interface. You can also configure these rules using the following management tools:

- Redirect Number Tel to IP table: ini file (RedirectNumberMapTel2Ip) or CLI (configure voip/gw manipulations redirect-number-map-tel2ip)

**➢ To configure a redirect number manipulation rule:**

1. Open the redirect number manipulation table (**Configuration** tab > **VoIP** menu > **Gateway** > **Manipulations** > **Redirect Number Tel > IP**).
2. Click **Add**; the following dialog box appears (e.g., Redirect Number Tel-to-IP table):

**Figure 22-4: Redirect Number Manipulation Table (Example) - Add Row Dialog Box**




3. Configure a manipulation rule according to the parameters described in the table below.
4. Click **Add**.

**Table 22-4: Redirect Number Manipulation Tables Parameter Description**

| Parameter | Description |
|---|---|
| Index<br>[_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>`manipulation-name`<br>[_ManipulationName] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters. |
| **Rule (Matching Characteristics)** | |
| Destination Prefix<br>`dst-prefix`<br>[_DestinationPrefix] | Defines the destination (called) telephone number prefix.<br>The default value is the asterisk (*) symbol (i.e., any number).<br>For manipulating the diverting and redirected numbers for call diversion, you can use the strings "DN" and "RN" to denote the destination prefix of these numbers. For more information, see Manipulating Redirected and Diverted Numbers for Call Diversion on page 406. |
| Redirect Prefix | Defines the redirect telephone number prefix. |

| Parameter | Description |
|---|---|
| `redirect-prefix`<br>[_RedirectPrefix] | The default value is the asterisk (*) symbol (i.e., any number prefix). |
| Source Trunk Group ID<br>`src-trunk-group-id`<br>[_SrcTrunkGroupID] | Defines the Hunt Group from where the Tel call is received.<br>To denote any Hunt Group, leave this field empty. The value -1 indicates that this field is ignored in the rule.<br>**Note:** The parameter is applicable only to the Redirect Number Tel-to-IP table. |
| Source IP Address<br>`src-ip-address`<br>[_SourceAddress] | Defines the IP address of the caller. The IP address appears in the SIP Contact header of the incoming INVITE message.<br>The default value is the asterisk (*) symbol (i.e., any IP address). The value can include the following wildcards:<br>▪ "x": represents single digits, for example, 10.8.8.xx denotes all addresses between 10.8.8.10 and 10.8.8.99.<br>▪ "*": 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.<br>**Note:** The parameter is applicable only to the Redirect Number IP-to-Tel table. |
| Source Host Prefix<br>`src-host-prefix`<br>[_SrcHost] | Defines the URI host name prefix of the caller. The host name appears in the SIP From header of the incoming SIP INVITE message.<br>The default value is the asterisk (*) symbol (i.e., any host name prefix).<br>**Notes:**<br>▪ The parameter is applicable only to the Redirect Number IP-to-Tel table.<br>▪ If the P-Asserted-Identity header is present in the incoming INVITE message, the value of the parameter is compared to the P-Asserted-Identity URI host name (instead of to the From header). |
| Destination Host Prefix<br>`dst-host-prefix`<br>[_DestHost] | Defines the Request-URI host name prefix, which appears in the incoming SIP INVITE message.<br>The default value is the asterisk (*) symbol (i.e., any prefix).<br>**Note:** The parameter is applicable only to the Redirect Number IP-to-Tel table. |
| **Operation (Action)** | |
| Stripped Digits From Left<br>`remove-from-left`<br>[_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. |
| Stripped Digits From Right<br>`remove-from-right`<br>[_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. |
| Number of Digits to Leave<br>`num-of-digits-to-leave`<br>[_LeaveFromRight] | Defines the number of digits that you want to retain from the right of the redirect number. |

| Parameter | Description |
|---|---|
| Prefix to Add<br>`prefix-to-add`<br>[_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. |
| Suffix to Add<br>`suffix-to-add`<br>[_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. |
| TON<br>`ton`<br>[_NumberType] | Defines the Type of Number (TON).<br>The default is Not Configured [-1].<br>▪ If NPI is set to Unknown, you can set TON to Unknown [0].<br>▪ If NPI is set to Private, you can set TON to Unknown [0], International [1], National [2], Network Specific [3] or Subscriber [4].<br>▪ If NPI is set to E.164 Public, you can set TON to Unknown [0], International [1], National [2], Network Specific [3], Subscriber [4] or Abbreviated [6].<br>For more information on available NPI/TON values, see Numbering Plans and Type of Number on page 408. |
| NPI<br>`npi`<br>[_NumberPlan] | Defines the Numbering Plan Indicator (NPI).<br>▪ [-1] Not Configured = (Default) Value received from PSTN/IP is used<br>▪ [0] Unknown<br>▪ [1] E.164 Public<br>▪ [9] Private<br>For more information on available NPI/TON values, see Numbering Plans and Type of Number on page 408. |
| Presentation<br>`is-presentation-restricted`<br>[_IsPresentationRestricted] | Enables caller ID.<br>▪ Not Configured = Privacy is determined according to the Caller ID table (see Configuring Caller Display Information on page 476).<br>▪ **[0]** Allowed = Sends Caller ID information when a call is made using these destination / source prefixes.<br>▪ **[1]** Restricted = Restricts Caller ID information for these prefixes.<br>**Note:** If the parameter is set to **Restricted** and the 'AssertedIdMode' parameter is set to **Add P-Asserted-Identity,** the From header in the INVITE message includes the following:<br>`From: 'anonymous' <sip:anonymous@anonymous.invalid> and 'privacy: id' header.` |

## 22.6 Mapping NPI/TON to SIP Phone-Context

The Phone Context table lets you configure rules for mapping the Numbering Plan Indication (NPI) and Type of Number (TON) to the SIP 'phone-context' parameter, and vice versa. The 'phone-context' parameter appears in the standard SIP headers where a phone number is used (i.e., Request-URI, To, From, and Diversion). When a call is received from the Tel side, the NPI and TON are compared against the table and the matching 'phone-context' value is used in the outgoing SIP INVITE message. The same mapping occurs when an INVITE with a 'phone-context' parameter is received.

For example, for a Tel-to-IP call with NPI/TON set as E164 National (values 1/2), the device can send the following SIP INVITE URI:

```
sip:12365432;phone-context= na.e.164.nt.com
```

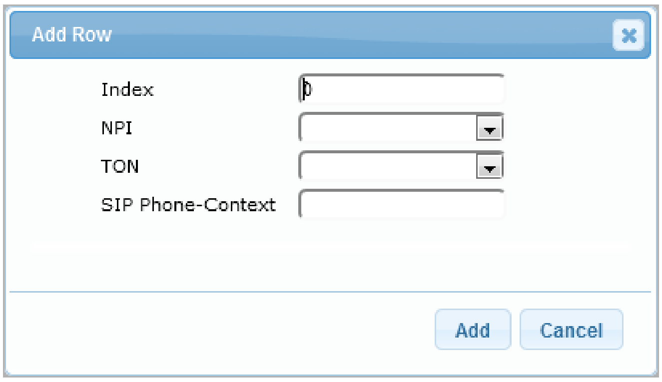
For an IP-to-Tel call, if the incoming INVITE contains this 'phone-context' (e.g. "phone-context= na.e.164.nt.com"), the NPI/TON of the called number in the outgoing Setup message is changed to E164 National.

The following procedure describes how to configure NPI/TON-SIP phone-context mapping rules through the Web interface. You can also configure it through ini file (PhoneContext) or CLI (configure voip > gw manipulations phone-context-table).

➢ **To configure NPI/TON-SIP phone-context mapping rules:**

1. Open the Phone Context table (**Configuration** tab > **VoIP** menu > **Gateway** > **Manipulations** > **Phone Context**).
2. Click **Add**; the following dialog box appears:

**Figure 22-5: Phone Context Table - Add Row Dialog Box**




3. Configure a mapping rule according to the parameters described in the table below.
4. To add the incoming SIP 'phone-context' parameter as a prefix to the outgoing called and calling numbers, from the 'Add Phone Context As Prefix' drop-down list (AddPhoneContextAsPrefix), select **Enable**.
5. Click **Add**.

**Note:** You can configure multiple rows with the same NPI/TON or same SIP 'phone-context'. In such a configuration, a Tel-to-IP call uses the first matching rule in the table.

**Table 22-5: Phone Context Table Parameter Description**

| Parameter | Description |
|---|---|
| Index<br>[PhoneContext_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| NPI<br>`npi`<br>[PhoneContext_Npi] | Defines the Number Plan Indicator (NPI).<br>▪ **[0]** Unknown (default)<br>▪ **[1]** E.164 Public<br>▪ **[9]** Private<br>For a detailed list of the available NPI/TON values, see Numbering Plans and Type of Number on page 408. |

| Parameter | Description |
|---|---|
| TON<br>`ton`<br>[PhoneContext_Ton] | Defines the Type of Number (TON).<br>▪ If you selected Unknown as the NPI, you can select Unknown **[0]**.<br>▪ If you selected Private as the NPI, you can select one of the following:<br>✓ **[0]** Unknown<br>✓ **[1]** Level 2 Regional<br>✓ **[2]** Level 1 Regional<br>✓ **[3]** PSTN Specific<br>✓ **[4]** Level 0 Regional (Local)<br>▪ If you selected E.164 Public as the NPI, you can select one of the following:<br>✓ **[0]** Unknown<br>✓ **[1]** International<br>✓ **[2]** National<br>✓ **[3]** Network Specific<br>✓ **[4]** Subscriber<br>✓ **[6]** Abbreviated |
| SIP Phone Context<br>`context`<br>[PhoneContext_Context] | Defines the SIP 'phone-context' URI parameter. |

# 23 Routing

This section describes the configuration of call routing rules.

## 23.1 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 643.

➢ **To configure general routing parameters:**

1. Open the Routing General Parameters page (**Configuration** tab > **VoIP** menu > **Gateway** > **Routing** > **General Parameters**).

| General Parameters | |
|---|---|
| Add Hunt Group ID as Prefix | No |
| Add Trunk ID as Prefix | No |
| Replace Empty Destination with B-channel Phone Number | No |
| Add NPI and TON to Called Number | No |
| Add NPI and TON to Calling Number | No |
| IP to Tel Remove Routing Table Prefix | No |
| Source IP Address Input | SIP Contact Header |
| Enable Alt Routing Tel to IP | Disable |
| Alt Routing Tel to IP Mode | Both |
| Alt Routing Tel to IP Connectivity Method | ICMP Ping |
| Alt Routing Tel to IP Keep Alive Time | 60 |
| Alternative Routing Tone Duration [ms] | 0 |
| Source Manipulation Mode | FROM & PAI (after manipulation) |
| Max Allowed Packet Loss for Alt Routing [%] | 20 |
| Max Allowed Delay for Alt Routing [msec] | 250 |

2. Configure the parameters as required.
3. Click **Submit**.

## 23.2 Configuring Tel-to-IP Routing Rules

The Tel-to-IP Routing table lets you configure up to 180 Tel-to-IP routing rules. Tel-to-IP routing rules are used to route calls from the Tel side to an IP destination.

Configuration of Tel-to-IP routing rules includes two areas:

- **Rule:** Defines the characteristics of the incoming Tel call (e.g., Hunt Group on which the call is received).
- **Action:** Defines the action that is done if the incoming call matches the characteristics of the rule (i.e., routes the call to the specified IP destination).

The device searches the table from top to bottom for the first rule that matches the characteristics of the incoming call. If it finds a matching rule, it sends the call to the IP destination configured for that rule. If it doesn't find a matching rule, it rejects the call.

You can configure the routing rule with one or more of the following incoming Tel characteristics:

- Source Hunt Group (from where the call is received)
- Source (calling) and destination (called) telephone number prefix and suffix

You can configure the IP destination to one of the following:

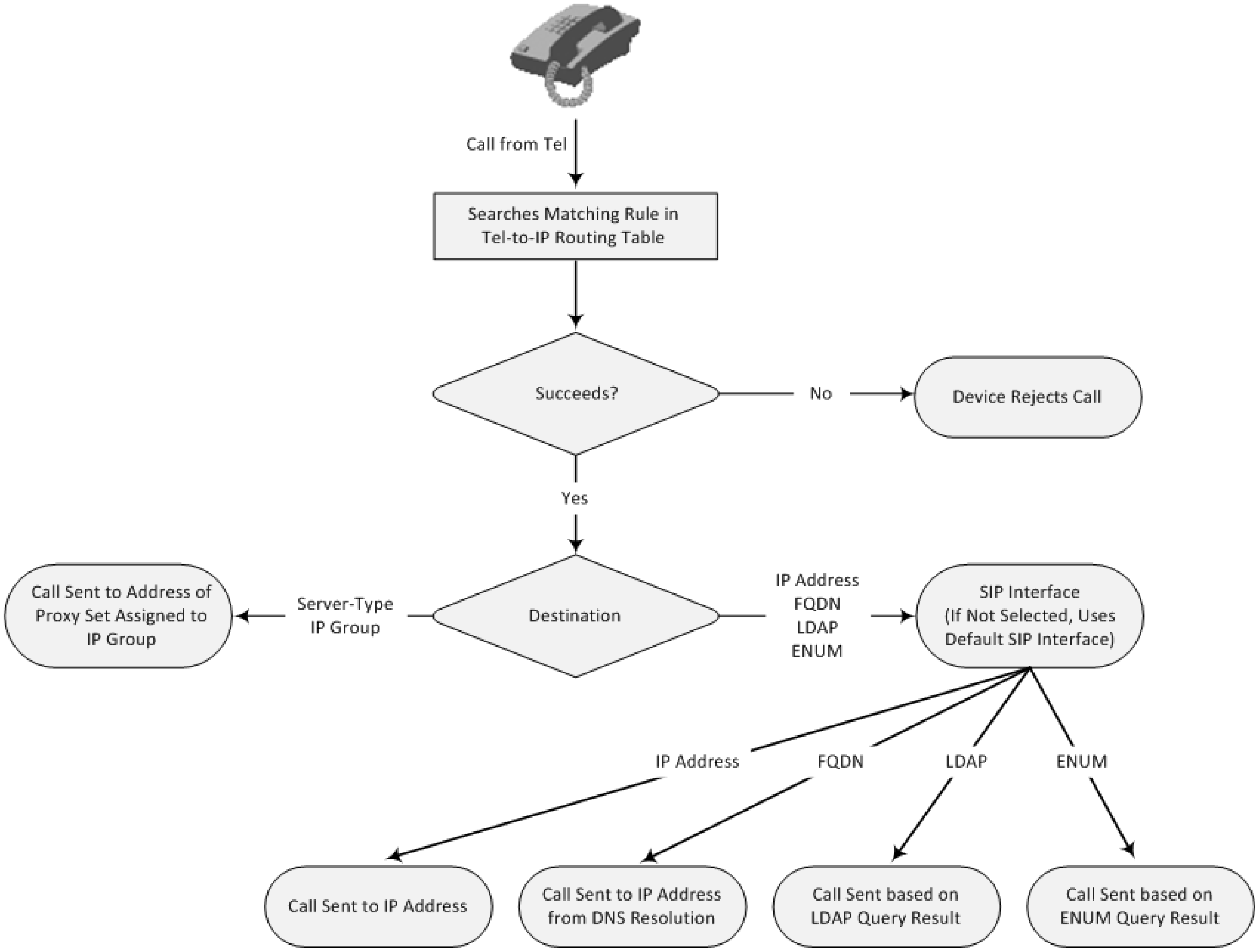
- IP address.
- FQDN.
- E.164 Telephone Number Mapping (ENUM service).
- Lightweight Directory Access Protocol (LDAP). For more information, see LDAP-based Management and SIP Services on page 232 and Active Directory-based Routing for Microsoft Lync on page 255.
- IP Group. When an IP Group is selected, the device sends the call to the IP address configured for the Proxy Set that is associated with the IP Group (configured in "Configuring IP Groups" on page 323). The SRD associated with the IP Group determines the:
  - SIP Interface (SIP port and control network interface) - important when using multiple SIP control VLANs
  - Media Realm (port and network interface for media / RTP voice)
  - SRD-related features on which the call is routed

If you configure the routing rule to send the call to any destination other than an IP Group (e.g., an IP address), you need to select a SIP Interface for the call. If no SIP Interface is selected, the device uses the SIP Interface associated with the default SRD (Index 0). If you have deleted this SRD or SIP Interface, for whatever reason, the device drops the call. The SIP Interface determines many attributes for the destination:

- Device's logical SIP port and network interface through which the call signaling is sent
- Device's logical RTP port and network interface through which the media is sent (Media Realm)
- Other features that can be configured for the SIP Interface
- SRD. As one of the attributes of a SIP Interface is an SRD and as you can configure multiple SIP Interfaces per SRD, the specific SIP Interface not only determines the

above-mentioned attributes, but also the SRD for routing the call.

**Figure 23-1: Locating SRD**




**Note:** When using a proxy server, it is unnecessary to configure routing rules in the Tel-to-IP Routing table, unless you require one of the following:

- Fallback (alternative) routing when communication with the proxy server fails.
- IP security, whereby the device routes only received calls whose source IP addresses are configured in the table. IP security is enabled using the SecureCallsFromIP parameter.
- Filter Calls to IP feature: the device checks the table before a call is routed to the proxy server. However, if the number is not allowed, i.e., the number is not specified in the table or a Call Restriction (see below) routing rule is configured, the call is rejected.
- Obtain different SIP URI host names (per called number).
- Assign IP Profiles to calls.
- For the table to take precedence over a proxy server for routing calls, you need to configure the PreferRouteTable parameter to 1. The device checks the 'Destination IP Address' field in the table for a match with the outgoing call; a proxy is used only if a match is not found.

In addition to normal Tel-to-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 261. If two routing rules have identical costs, 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 optional, default LCR settings configured by the Gateway Routing Policy (see "Configuring a Gateway Routing Policy Rule" on page 426).
- **Call Forking:** If the Tel-to-IP Call Forking feature is enabled, the device can send a Tel call to multiple IP destinations. An incoming Tel call with multiple matched routing rules (e.g., all with the same source prefix numbers) can be sent (forked) to multiple IP destinations if all these rules are configured with a Forking Group. The call is established with the first IP destination that answers the call.
- **Call Restriction:** Calls whose matching routing rule is configured with the destination IP address of 0.0.0.0 are rejected.
- **Always Use Routing Table:** Even if a proxy server is used, the SIP Request-URI host name in the outgoing INVITE message is obtained from this table. Using this feature, you can assign a different SIP URI host name for different called and/or calling numbers. This feature is enabled using the AlwaysUseRouteTable parameter.
- **IP Profiles:** IP Profiles can be assigned to destination addresses (also when a proxy is used).
- **Alternative Routing (when a proxy isn't used):** An alternative IP destination (alternative routing rule) can be configured for specific calls ("main" routing rule). When the "main" route fails (e.g., busy), the device can send the call to the alternative route. You must configure the alternative routing rules in table rows (indices) that are located anywhere **below** the "main" routing rule. For example, if you configure a "main" routing rule in Index 4, the alternative routing rule can be configured in Index 6. In addition, you must configure the alternative routing rules with identical matching characteristics (e.g., destination prefix number) as the "main" routing rule, but assigned with different destination IP addresses. Instead of an IP address, you can use an FQDN to resolve into two IP addresses. For more information on alternative routing, see "Alternative Routing for Tel-to-IP Calls" on page 429.

- **Advice of Charge (AOC):** AOC is a pre-billing feature that tasks the rating engine with calculating the cost of using a service (Tel-to-IP call) and relaying that information to the customer. AOC, which is configured in the Charge Codes table, can be applied per Tel-to-IP routing rule.

**Notes:**

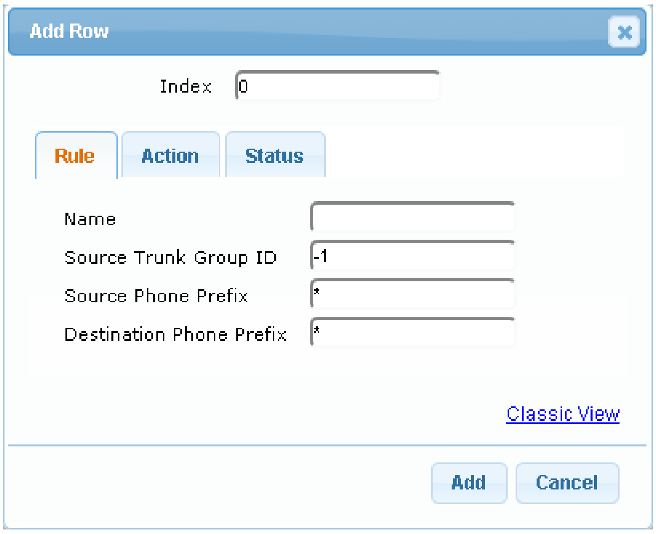
- Instead of using the table for Tel-to-IP routing, you can employ a third-party Routing server to handle the routing decisions. For more information, see Centralized Third-Party Routing Server on page 272.
- You can configure up to three alternative routing rules per "main" routing rule in the Tel-to-IP Routing table.
- By default, the device applies telephone number manipulation (if configured) only after processing the routing rule. You can change this and apply number manipulation before processing the routing rule (see the RouteModeTel2IP parameter).

The following procedure describes how to configure Tel-to-IP routing rules through the Web interface. You can also configure it through ini file (Prefix) or CLI (configure voip > gw routing tel2ip-routing).

➢ **To configure Tel-to-IP  routing rules:**

1. Open the Tel-to-IP Routing table (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Routing** > **Tel to IP Routing**).
2. Click **Add**; the following dialog box appears:

**Figure 23-2: Tel-to-IP Routing Table - Add Row Dialog Box**




3. Configure a routing rule according to the parameters described in the table below.
4. Click **Add**.

The following table shows configuration examples of Tel-to-IP routing rules, where:

- **Rules 1 and 2 (Least Cost Routing):** 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 "ABC". 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):** For all callers (*), if the called phone number prefix is 20, the call is sent to IP Group "ITSP-ZA".
- **Rule 4 (domain name destination):** For called phone number prefixes 5, 7, 8, or 9, and the caller belongs to Hunt Group ID 4, the call is sent to the domain "itsp.com".
- **Rule 5 (block):** For all callers (*), if the called phone number prefix is 00, the call is rejected (IP address 0.0.0.0).
- **Rule 6, 7, and 8 (Forking Group):** 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 23-1: Example of Tel-to-IP Routing Rules**

| Parameter | Rule 1 | Rule 2 | Rule 3 | Rule 4 | Rule 5 | Rule 6 | Rule 7 | Rule 8 |
|---|---|---|---|---|---|---|---|---|
| **Matching Characteristics of Incoming Call** | | | | | | | | |
| Source Trunk Group ID | | | | 4 | | * | * | * |
| Source Phone Prefix | 100 | 100 | * | * | * | * | * | * |
| Destination Phone Prefix | 10 | 10 | 20 | [5,7-9] | 00 | 100 | 100 | 100 |
| **Action** | | | | | | | | |
| Destination IP Group | | | ITSP-ZA | | | | | |
| Destination IP Address | 10.33.45.63 | 10.33.45.50 | | itsp.com | 0.0.0.0 | 10.33.45.68 | 10.33.45.67 | domain.com |
| IP Profile | ABC | ABC | | | | | | |
| Forking Group | | | | | | 1 | 2 | 1 |
| Cost Group ID | Weekend-Low | Weekend_High | | | | | | |

**Table 23-2: Tel-to-IP Routing table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[PREFIX_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>`route-name`<br>[PREFIX_RouteName] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters. By default, no value is defined.<br>**Note:** Each row must be configured with a unique name. |
| **Rule Tab - Matching Call Characteristics** | |

| Parameter | Description |
|---|---|
| Source Trunk Group ID<br>`src-trunk-group-id`<br>[PREFIX_SrcTrunkGroupID] | Defines the Hunt Group from where the call is received.<br>To denote any Hunt Group, use the asterisk (*) symbol. By default, no Hunt Group is defined (-1). |
| Source Phone Prefix<br>`src-phone-prefix`<br>[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 (default) 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 639.<br>The number can include up to 50 digits. |
| Destination Phone Prefix<br>`dst-phone-prefix`<br>[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 (default) 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 639.<br>The number can include up to 50 digits.<br>**Note:**<br>▪ For LDAP-based routing, enter the LDAP query keyword as the prefix number to denote the IP domain:<br>✓ "PRIVATE" = Private number<br>✓ "OCS" = Lync / OCS client number<br>✓ "PBX" = PBX / IP PBX number<br>✓ "MOBILE" = Mobile number<br>✓ "LDAP_ERR" = LDAP query failure<br>For more information, see Active Directory-based Routing for Microsoft Lync on page 255. |
| **Action Tab - IP Destination** | |
| Destination IP Group<br>`dst-ip-group-id`<br>[PREFIX_DestIPGroupName] | Defines the IP Group to where you want to route the call. The SIP INVITE message is sent to the IP address configured for the Proxy Set that is associated with the IP Group.<br>**Notes:**<br>▪ 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.<br>▪ If the destination is a User-type IP Group, the device searches for a match of the Request-URI in the received INVITE to an AOR registration record in the device's database. The INVITE is then sent to the IP address of the registered contact.<br>▪ If the AlwaysUseRouteTable parameter is set to 1 (see "Configuring IP Groups" on page 323), the Request-URI host name in the INVITE message is set to the value configured for the 'Destination IP Address' parameter (in this table); |

| Parameter | Description |
|---|---|
| | otherwise, if no IP address is defined, it is set to the value of the 'SIP Group Name' parameter (configured in the IP Group table).<br>▪ The parameter is used as the 'Serving IP Group' in the Account table for acquiring authentication username/password for this call (see "Configuring Registration Accounts" on page 341).<br>▪ For defining Proxy Sets, see "Configuring Proxy Sets" on page 329. |
| Destination SIP Interface<br>`dest-sip-interface-name`<br>[PREFIX_DestSIPInterfaceName] | Assigns a SIP Interface to the routing rule. The call is sent to its' destination through this SIP interface.<br>For configuring SIP Interfaces, see "Configuring SIP Interfaces" on page 319.<br>**Note:** If a SIP Interface is not assigned, the device uses the SIP Interface associated with the default SRD (Index 0). If, for whatever reason, you have deleted the default SRD and there are no SRDs, the call is rejected. |
| Destination IP Address<br>`dst-ip-address`<br>[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.<br>The IP address can include the following wildcards:<br>▪ "x": represents single digits. For example, 10.8.8.xx denotes all addresses between 10.8.8.10 and 10.8.8.99.<br>▪ "*": 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.<br>For ENUM-based routing, enter the string "ENUM". The device sends an ENUM query containing the destination phone number to an external DNS server, configured in the Interface table. The ENUM reply includes a SIP URI which is used as the Request-URI in the subsequent outgoing INVITE and for routing (if a proxy is not used). To configure the type of ENUM service (e.g., e164.arpa), see the EnumService parameter.<br>For LDAP-based routing, enter the string value, "LDAP" for denoting the IP address of the LDAP server. For more information, see Active Directory-based Routing for Microsoft Lync on page 255.<br>**Notes:**<br>▪ The parameter is ignored if you have configured a destination IP Group in the 'Destination IP Group' field (in this table).<br>▪ To reject calls, enter the IP address 0.0.0.0. For example, if you want to prohibit international calls, then in the 'Destination Phone Prefix' field, enter 00 and in the 'Destination IP Address' field, enter 0.0.0.0.<br>▪ For routing calls between phones connected to the device (i.e., local routing), enter the device's IP address. If the device's IP address is unknown (e.g., when DHCP is used), enter IP address 127.0.0.1.<br>▪ When using domain names, enter the DNS server's IP address or alternatively, configure these names in the |

| Parameter | Description |
|---|---|
| | Internal DNS table (see "Configuring the Internal DNS Table" on page 149). |
| Destination Port<br>`dst-port`<br>[PREFIX_DestPort] | Defines the destination port to where you want to route the call. |
| Transport Type<br>`transport-type`<br>[PREFIX_TransportType] | Defines the transport layer type used for routing the call:<br>▪ **[-1]** = (Default) Not defined - transport type is according to the settings of the global parameter, SIPTransportType.<br>▪ **[0]** UDP<br>▪ **[1]** TCP<br>▪ **[2]** TLS |
| IP Profile<br>`ip-profile-id`<br>[PREFIX_ProfileName] | Assigns an IP Profile to the routing rule in the outgoing direction. The IP Profile allows you to assign various configuration attributes (e.g., voice coder) per routing rule. To configure IP Profiles, see "Configuring IP Profiles" on page 366. |
| Call Setup Rules Set ID<br>`call-setup-rules-set-id`<br>[PREFIX_CallSetupRulesSetId] | Assigns a Call Setup Rule Set ID to the routing rule. The device performs the Call Setup rules of this Set ID if the incoming call matches the characteristics of the routing rule. The device routes the call to the destination according to the routing rule's configured action only after it has performed the Call Setup rules.<br>By default, no value is defined.<br>For configuring Call Setup rules, see "Configuring Call Setup Rules" on page 283. |
| Forking Group<br>`forking-group`<br>[PREFIX_ForkingGroup] | Defines a Forking Group number for the routing rule. This enables forking of incoming Tel calls to multiple IP destinations. The device sends simultaneous INVITE messages and handles multiple SIP dialogs until one of the calls is answered. When a call is answered, the other calls are dropped.<br>Each Forking Group can contain up to 10 members. In other words, up to 10 routing rules can be configured with the same Forking Group number.<br>By default, no value is defined.<br>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. If matched routing rules belong to different Forking Groups, the device sends the call to the |

| Parameter | Description |
|---|---|
| | Forking Group of the first matched routing rule. If the call cannot be established with any of the destinations associated with the 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:<br>▪ 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.<br>▪ 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.<br>▪ 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.<br>▪ 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.<br>**Notes:**<br>▪ To enable Tel-to-IP call forking, set the 'Tel2IP Call Forking Mode' (*Tel2IPCallForkingMode*) parameter to **Enable**.<br>▪ You can configure the device to immediately send the INVITE message to the first member of the Forking Group (as in normal operation) and then only after a user-defined interval, send the INVITE messages simultaneously to the other members. If the device receives a SIP 4xx or 5xx in response to the first INVITE, it immediately sends INVITEs to all the other members, regardless of the interval. To configure this feature, see the ForkingDelayTimeForInvite ini file parameter.<br>▪ You can implement Forking Groups when the destination is an LDAP server or a domain name using DNS. In such scenarios, the INVITE is sent to all the queried LDAP or resolved IP addresses, respectively. You can also use LDAP routing rules with standard routing rules for Forking Groups.<br>▪ When the UseDifferentRTPportAfterHold parameter is enabled, every forked call is sent with a different RTP port. |

| Parameter | Description |
|---|---|
| | Thus, ensure that the device has sufficient available RTP ports for these forked calls. |
| Cost Group<br>`cost-group-id`<br>[PREFIX_CostGroup] | Assigns a Cost Group to the routing rule for determining the cost of the call (i.e., Least Cost Routing or LCR).<br>By default, no value is defined (**None**).<br>To configure Cost Groups, see "Configuring Cost Groups" on page 263.<br>**Note:** To implement LCR and its Cost Groups, you must enable LCR<br>▪ To implement LCR and its Cost Groups, the Gateway Routing Policy must be enabled for LCR (see "Configuring a Gateway Routing Policy Rule" on page 426). If LCR is disabled, the device ignores the parameter.<br>▪ The Routing Policy also determines whether matched routing rules that are **not** assigned Cost Groups are considered as a higher or lower cost route compared to matching routing rules that are assigned Cost Groups. For example, if the 'Default Call Cost' parameter in the Routing Policy is configured to **Lowest Cost**, even if the device locates matching routing rules that are assigned Cost Groups, the first-matched routing rule without an assigned Cost Group is considered as the lowest cost route and thus, chosen as the preferred route. |
| Charge Code<br>`charge-code`<br>[PREFIX_MeteringCode] | Assigns a Charge Code to the routing rule for generating metering pulses (Advice of Charge).<br>By default, no value is defined (**None**).<br>To configure Charge Codes, see "Configuring Charge Codes" on page 464.<br>**Note:** The parameter is applicable only to FXS interfaces. |
| **Status Tab** | |
| Connectivity Status | (Read-only field) Displays the connectivity status of the routing rule's destination. The destination can be an IP address or an IP Group, as configured in the 'Destination IP Address' and 'Destination IP Group' fields respectively.<br>For IP Groups, the status indicates the connectivity with the SIP proxy server's address configured for the Proxy Set that is associated with the IP Group. For the status to be displayed, the Proxy Keep-Alive feature, which monitors the connectivity with proxy servers per Proxy Set, must be enabled for the Proxy Set (see "Configuring Proxy Sets" on page 329). If a Proxy Set is configured with multiple proxies for redundancy, the status may change according to the proxy server with which the device attempts to verify connectivity. For example, if there is no response from the first configured proxy address, the status displays "No Connectivity". However, if there is a response from the next proxy server in the list, the status changes to "OK".<br>If there is connectivity with the destination, the field displays "OK" and the device uses the routing rule if required. The routing rule is not used if any of the following is displayed:<br>▪ "n/a" = IP Group is unavailable. |

| Parameter | Description |
| --- | --- |
| | ▪ "No Connectivity" = No connection with the destination (no response to the SIP OPTIONS).<br>▪ "QoS Low" = Poor Quality of Service (QoS) of the destination.<br>▪ "DNS Error" = No DNS resolution. This status is applicable only when a domain name is used (instead of an IP address).<br>▪ "Not Available" = Destination is unreachable due to networking issues. |

## 23.3 Configuring IP-to-Hunt Group Routing Rules

The IP to Hunt Group Routing table lets you configure up to 120 IP-to-Hunt Group routing rules. IP-to-Hunt Group routing rules are 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 configured per Hunt Group (see "Configuring Hunt Group Settings" on page 386) or for all Hunt Groups using the global parameter ChannelSelectMode.

Configuration of IP-to-Hunt Group routing rules includes two areas:

- **Rule:** Defines the characteristics of the incoming IP call (e.g., source IP address from which the call is received).
- **Action:** Defines the action that is done if the incoming call matches the characteristics of the rule (i.e., routes the call to the specified Tel/Hunt Group destination).

The device searches the table from top to bottom for the first rule that matches the characteristics of the incoming call. If it finds a matching rule, it sends the call to the Tel destination configured for that rule. If it doesn't find a matching rule, it rejects the call..

If an IP-to-Tel call cannot be routed to the Hunt Group, the device can route it to an alternative destination:

- **Routing to an Alternative Hunt Group:** If the device sends the IP call to the Tel destination and a subsequent call release reason (cause) code (e.g., 17 for User Busy) is received from the Tel side, and you have configured this release reason code in the Reasons for IP-to-Tel Alternative Routing table, the device re-routes the call to an alternative Hunt Group if an alternative routing rule has been configured in the table. The alternative routing rules must be configured in table rows (indices) that are located anywhere below the "main" routing rule. For example, if you configure a "main" routing rule in Index 4, the alternative routing rule can be configured in Index 6. In addition, you must configure the alternative routing rules with identical matching characteristics (e.g., destination prefix number) to the "main" routing rule, but assigned with different destinations (i.e., Hunt Groups). For more information on IP-to-Tel alternative routing and for configuring call release reasons for alternative routing, see "Alternative Routing to Hunt upon Q.931 Call Release Cause Code" on page 435.
- **Routing to an IP Destination (i.e., Call Redirection):** The device can re-route the IP-to-Tel call to an alternative IP destination, using SIP 3xx responses. For more information, see "Alternative Routing to IP Destinations upon Busy Hunt" on page 436.

**Notes:**

- Instead of using the table for IP-to-Tel routing, you can employ a third-party Routing server to handle the routing decisions. For more information, see Centralized Third-Party Routing Server on page 272.
- You can configure up to three alternative routing rules per "main" routing rule in the table.
- If your deployment includes calls of many different called (source) and/or calling (destination) numbers that need to be routed to the same destination, you can employ user-defined prefix tags to represent these numbers. Thus, instead of configuring many routing rules, you need to configure only one routing rule using the prefix tag as the source and destination number matching characteristics, and a destination for the calls. For more information on prefix tags, see ''Dial Plan Prefix Tags for IP-to-Tel Routing'' on page 503.
- By default, the device applies destination telephone number manipulation (if configured) only after processing the routing rule. You can change this and apply number manipulation before processing the routing rule (see the RouteModeIP2Tel parameter). For configuring number manipulation, see ''Configuring Source/Destination Number Manipulation'' on page 393.

The following procedure describes how to configure Inbound IP Routing rules through the Web interface. You can also configure it through ini file (PSTNPrefix) or CLI (configure voip > gw routing ip2tel-routing).

### ➢ To configure IP-to-Tel routing rules:

1. Open the IP to Hunt Group Routing table (**Configuration** tab > **VoIP** menu > **Gateway** > **Routing** > **IP to Hunt Group Routing**).
2. Click **Add**; the following dialog box appears:

**Figure 23-3: IP to Hunt Group Table - Add Row Dialog Box**


3. Configure a routing rule according to the parameters described in the table below.
4. Click **Add**.

The following table shows configuration examples of Tel-to-IP routing rules, where:

- **Rule 1:** If the incoming IP call destination phone prefix is between 10 and 19, the call is assigned settings configured for IP Profile "ITSP-A" and routed to Hunt Group ID 3.
- **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 "ITSP-B" and routed to Hunt Group ID 2.
- **Rule 3:** If the incoming IP call has a From URI host prefix as abcd.com, the call is routed to Hunt Group ID 4.

**Table 23-3: Example of IP-to-Hunt Group Routing Rules**

| Parameter | Rule 1 | Rule 2 | Rule 3 |
|---|---|---|---|
| Source Host Prefix | | | abcd.domain |
| Destination Phone Prefix | 1x | [501-502] | |
| Source Phone Prefix | | 101 | |
| Source IP Address | | | |
| Trunk Group ID | 3 | 2 | 4 |
| IP Profile | ITSP-A | ITSP-B | |

**Table 23-4: IP to Hunt Group Table Parameter Description**

| Parameter | Description |
|---|---|
| Index<br>[PstnPrefix_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Name<br>`route-name`<br>[PstnPrefix_RouteName] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters. By default, no value is defined.<br>**Note:** Each row must be configured with a unique name. |
| **Rule (Matching Characteristics)** | |
| Source SIP Interface<br>`src-sip-interface-name`<br>[PstnPrefix_SrcSIPInterfaceName] | Defines the SIP Interface on which the incoming IP call is received.<br>The default is Any (i.e., any SIP Interface).<br>For configuring SIP Interfaces, see Configuring SIP Interfaces on page 319.<br>**Note:** If the incoming INVITE is received on the specified SIP Interface and the SIP Interface associated with the specified IP Group in the 'Source IP Group' parameter (in this table) is different, the incoming SIP call is rejected. If the 'Source IP Group' parameter is not defined, the SIP Interface associated with the default SRD (Index 0) is used. If there is no valid source IP Group, the call is rejected. |
| Source IP Address<br>`src-ip-address`<br>[PstnPrefix_SourceAddress] | Defines the source IP address of the incoming IP call.<br>The IP address must be configured in dotted-decimal notation (e.g., 10.8.8.5); not as an FQDN. By default, no value is defined.<br>**Notes:**<br>▪ The source IP address is obtained from the Contact header in the INVITE message.<br>▪ You can configure from where the source IP address is obtained, using the SourceIPAddressInput parameter.<br>▪ The source IP address can include the following wildcards:<br>✓ "x": denotes single digits. For example, 10.8.8.xx represents all the addresses between 10.8.8.10 and 10.8.8.99.<br>✓ "*": 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. |
| Source Phone Prefix<br>`src-phone-prefix`<br>[PstnPrefix_SourcePrefix] | Defines the prefix or suffix of the calling (source) telephone number.<br>The prefix can include up to 49 digits. You can use special notations for denoting the prefix. For example, [100-199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. To denote any prefix, use the asterisk (*) symbol. To denote calls without a calling number, use the $ sign. For a description of available notations, see "Dialing Plan Notation for Routing and Manipulation Tables" on page 639.<br>By default, no value is defined.<br>**Note:** If the P-Asserted-Identity header is present in the incoming INVITE message, the value of the parameter is |

| Parameter | Description |
|---|---|
| | compared to the P-Asserted-Identity URI host name (and not the From header). |
| Source Host Prefix<br>`src-host-prefix`<br>[PstnPrefix_SrcHostPrefix] | Defines the prefix of the URI host name in the From header of the incoming INVITE message.<br>By default, no value is defined. To denote any prefix, use the asterisk (*) wildcard.<br>**Note:** If the P-Asserted-Identity header is present in the incoming INVITE message, the value of the parameter is compared to the P-Asserted-Identity URI host name (and not the From header). |
| Destination Phone Prefix<br>`dst-host-prefix`<br>[PstnPrefix_DestHostPrefix] | Defines the prefix or suffix of the called (destined) telephone number. You can use special notations for denoting the prefix. For example, [100-199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. To denote any prefix, use the asterisk (*) symbol or to denote calls without a called number, use the $ sign. For a description of available notations, see "Dialing Plan Notation for Routing and Manipulation Tables" on page 639.<br>By default, no value is defined.<br>The prefix can include up to 49 digits. |
| Destination Host Prefix<br>`dst-phone-prefix`<br>[PstnPrefix_DestPrefix] | Defines the Request-URI host name prefix of the incoming INVITE message.<br>By default, no value is defined. To denote any prefix, use the asterisk (*) wildcard. |
| **Action Tab (Tel Destination)** | |
| Destination Type<br>`dst-type`<br>[PstnPrefix_DestType] | Defines the type of Tel destination:<br>▪ [0] Trunk Group (default)<br>▪ [1] Trunk |
| Trunk Group ID<br>`trunk-group-id`<br>[PstnPrefix_TrunkGroupId] | Defines the Hunt Group ID to where the incoming SIP call is sent. |
| Trunk ID<br>`trunk-id`<br>[PstnPrefix_TrunkId] | Defines the Hunt to where the incoming SIP call is sent.<br>**Notes:**<br>▪ If both 'Trunk Group ID' and 'Trunk ID' parameters are configured in the table, the routing is done according to the 'Trunk Group ID' parameter.<br>▪ To configure the method for selecting the Hunt Group's channel to which the IP call is sent, see the global parameter, ChannelSelectMode. |
| Source IP Group<br>`src-ip-group-id`<br>[PstnPrefix_SrcIPGroupName] | Defines the IP Group associated with the incoming IP call. This is the IP Group from where the SIP message (INVITE) is received.<br>By default, no value is defined.<br>The IP Group can later be used as the 'Serving IP Group' in the Account table for obtaining authentication username/password for this call. For configuring registration accounts, see "Configuring Account Table" on page 341. |

| Parameter | Description |
|---|---|
| IP Profile<br>`ip-profile-id`<br>[PstnPrefix_ProfileName] | Assigns an IP Profile (configured in "Configuring IP Profiles" on page 366) to the call. |
| Call Setup Rules Set ID<br>`call-setup-rules-set-id`<br>[PstnPrefix_CallSetupRulesSetId] | Assigns a Call Setup Rule Set ID to the routing rule. The device performs the Call Setup rules of this Set ID if the incoming call matches the characteristics of the routing rule. The device routes the call to the destination according to the routing rule's configured action, only after it has performed the Call Setup rules.<br>For configuring Call Setup rules, see "Configuring Call Setup Rules" on page 283. |

## 23.4 Configuring a Gateway Routing Policy Rule

The Gateway Routing Policy table lets you edit the default Gateway Routing Policy rule. The Routing Policy is used for Gateway call routing and defines the following:

- LDAP server (LDAP Server Group) for LDAP-based call routing (LDAP and/or Call Setup Rules queries). LDAP-based routing is applicable to Tel-to-IP routing ("Configuring Tel-to-IP Routing Rules" on page 409) and IP-to-Tel routing ("Configuring IP-to-Hunt Group Routing Rules" on page 421).
- Enables Least Cost Routing (LCR), and defines default call cost (highest or lowest) and average call duration for Tel-to-IP routing rules that are not assigned LCR Cost Groups. The default call cost determines whether matched routing rules that are not assigned a Cost Group are considered as a higher or lower cost route compared to other matching routing rules that are assigned Cost Groups. If you disable LCR, the device ignores the Cost Groups assigned to Tel-to-IP routing rules in the Tel-to-IP Routing table. LCR is applicable only to Tel-to-IP routing.

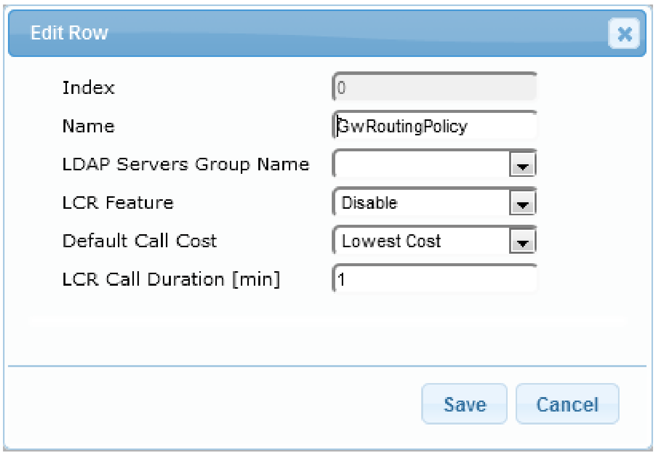
The following procedure describes how to configure Gateway Routing Policy rules through the Web interface. You can also configure it through ini file (GwRoutingPolicy) or CLI (configure voip > gw routing gw-routing-policy).

➢ **To edit the Gateway Routing Policy rule:**

1. Open the Gateway Routing Policy table (**Configuration** tab > **VoIP** menu > **Gateway** > **Routing** > **Gateway Routing Policy**).

2. Click **Add**; the following dialog box appears:

**Figure 23-4: Gateway Routing Policy Table - Edit Row Dialog Box**




3. Configure the Gateway Routing Policy rule according to the parameters described in the table below.
4. Click **Add**.

**Table 23-5: Gateway Routing Policy Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[GwRoutingPolicy_Index] | (Read-only) Displays the index number of the table row. |
| Name<br>`name`<br>[GWRoutingPolicy_Name] | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters. The default value is "GwRoutingPolicy". |
| LDAP Servers Group Name<br>`ldap-srv-group-name`<br>[GWRoutingPolicy_LdapServersGroupName] | Assigns an LDAP Server Group to the Routing Policy. IP-to-Tel and Tel-to-IP routing rules that require LDAP-based routing (and/or Call Setup Rules) use the LDAP server(s) assigned to the LDAP Server Group.<br>By default, no value is defined (**None**).<br>For more information on LDAP Server Groups, see "Configuring LDAP Server Groups" on page 234. |
| LCR Feature<br>`lcr-enable`<br>[GWRoutingPolicy_LCREnable] | Enables the Least Cost Routing (LCR) feature for the Routing Policy.<br>▪ [0] Disable (default)<br>▪ [1] Enable<br>For more information on LCR, see "Least Cost Routing" on page 261.<br>**Note:** LCR is applicable only to Tel-to-IP routing. |

| Parameter | Description |
|---|---|
| Default Call Cost<br>`lcr-default-cost`<br>[GWRoutingPolicy_LCRDefaultCost] | Defines whether routing rules in the Tel-to-IP Routing table that are not assigned a Cost Group are considered a higher cost or lower cost route compared to other matched routing rules that are assigned Cost Groups.<br>▪ [0] Lowest Cost = (Default) The device considers a matched routing rule that is not assigned a Cost Group as the lowest cost route. Therefore, it uses the routing rule.<br>▪ [1] Highest Cost = The device considers a matched routing rule that is not assigned a Cost Group as the highest cost route. Therefore, it is only used if the other matched routing rules that are assigned Cost Groups are unavailable. |
| LCR Call Duration<br>`lcr-call-length`<br>[GWRoutingPolicy_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:<br>cost = call connect cost + (minute cost * average call duration)<br>The valid value is 0-65533. The default is 1.<br>For example, assume the following Cost Groups:<br>▪ "Weekend A": call connection cost is 1 and charge per minute is 6. Therefore, a call of 1 minute cost 7 units.<br>▪ "Weekend B": call connection cost is 6 and charge per minute is 1. Therefore, a call of 1 minute cost 7 units.<br>Therefore, for calls under one minute, "Weekend A" carries the lower cost. However, if the average call duration is more than one minute, "Weekend B" carries the lower cost. |

## 23.5 IP Destinations Connectivity Feature

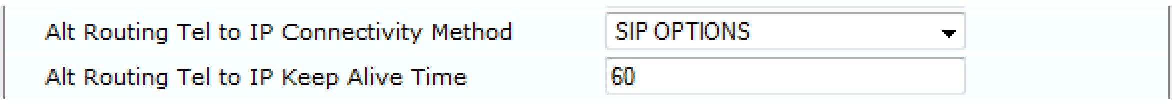
The device can be configured to check the integrity of the connectivity to IP destinations of Tel-to-IP routing rules in the Tel-to-IP Routing table. The IP Connectivity feature can be used for the Alternative Routing feature, whereby the device attempts to re-route calls from unavailable Tel-to-IP routing destinations to available ones (see "Alternative Routing Based on IP Connectivity" on page 429).

The device supports the following methods for checking the connectivity of IP destinations:

- **Network Connectivity:** The device checks the network connectivity of the IP destination configured by the 'Alt Routing Tel to IP Connectivity Method' parameter:
  - **SIP OPTIONS:** The device sends "keep-alive" SIP OPTIONS messages to the IP destination. If the device receives a SIP 200 OK in response, it considers the destination as available. If the destination does not respond to the OPTIONS message, then it is considered unavailable. You can configure the time interval for sending these OPTIONS messages, using the 'Alt Routing Tel to IP Keep Alive Time' parameter.

  These parameters are configured in the Routing General Parameters page (**Configuration** tab > **VoIP** menu > **Gateway** > **Routing** > **General Parameters**), as shown below:

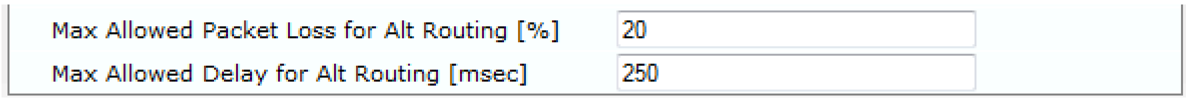
**Figure 23-5: IP Connectivity Method in Routing General Parameters Page**



- **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 23-6: IP QoS Thresholds in Routing General Parameters Page**




- **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:

- **Tel-to-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 Rules" on page 409.
- **IP Connectivity:** This page displays a more informative connectivity status of the IP destinations used in Tel-to-IP routing rules in the Tel-to-IP Routing table. For viewing this page, see "Viewing IP Connectivity" on page 568.

# 23.6 Alternative Routing for Tel-to-IP Calls

The device supports various alternative Tel-to-IP call routing methods, as described in this section.

## 23.6.1 Alternative Routing Based on IP Connectivity

You can configure the device to route Tel-to-IP calls to an alternative IP destination when the connectivity state of an IP destination is unavailable. The alternative routing rules are configured in the Tel-to-IP Routing table. These rules must be configured anywhere below the "main" routing rule and with identical matching characteristics (e.g., destination prefix number) to the "main" routing rule. The device uses the first alternative route that is available. For more information on configuring alternative Tel-to-IP routing rules in the Tel-to-IP Routing table, see "Configuring Tel-to-IP Routing Rules" on page 409.

**Note:**

- Alternative routing based on IP connectivity is applicable only when a proxy server is **not** used.
- You can also enable the Busy Out feature, whereby the device can take specified actions if all IP destinations of matching routing rules in the Tel-to-IP Routing table do not respond to connectivity checks. For more information, see the EnableBusyOut parameter.
- If the AltRoutingTel2IPEnable parameter is enabled, the Busy Out feature does not function with the Proxy Set keep-alive mechanism (see Alternative Routing Based on SIP Responses on page 431). To use the Busy Out feature with the Proxy Set keep-alive mechanism (for IP Groups), disable the AltRoutingTel2IPEnable parameter.

The device searches for an alternative routing rule (IP destination) when any of the following connectivity states are detected with the IP destination of the "main" routing rule:

- No response received from SIP OPTIONS messages. This depends on the chosen method for checking IP connectivity.
- Poor QoS according to the configured thresholds for packet loss and delay.
- No response from a DNS-resolved IP address, where the domain name (FQDN) is configured for the IP destination. If the device sends the INVITE message to the first IP address and receives no response, the device makes a user-defined number of attempts (configured by the HotSwapRtx parameter) to send it again (re-transmit). If there is still no response after all the attempts, it sends it to the next DNS-resolved IP address, and so on. For example, if you configure the parameter to "3" and the device receives no response from the first IP address, it attempts up to three times to send the INVITE to the first IP address and if unsuccessful, it attempts to send the call to the next DNS-resolved IP address, and so on.
- No response for in-dialog request from a DNS-resolved IP address, where the domain name is received in the Contact header of an incoming setup or target refresh SIP message (e.g., 200 OK). If no response is received from the first IP address, the device tries to send it again for up to a user-defined number of attempts (configured by the HotSwapRtx parameter). If there is still no response, it attempts to send the SIP request to the next DNS-resolved IP address, and so on.

The connectivity status of the IP destination is displayed in the 'Status' column of the Tel-to-IP Routing table per routing rule. If it displays a status other than "ok", 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 428.

The table below shows an example of alternative routing where the device uses an available alternative routing rule in the Tel-to-IP Routing table to re-route the initial Tel-to-IP call.

**Table 23-6: 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 Tel-to-IP Routing table, add alternative Tel-to-IP routing rules for specific calls.
2. In the Routing General Parameters page (**Configuration** tab > **VoIP** menu > **Gateway** > **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:
      - SIP OPTIONS failure
      - Poor QoS
      - SIP OPTIONS failure, poor QoS, or unresolved DNS
   c. The device plays a tone to the Tel endpoint (for analog interfaces) whenever an alternative route is used. This tone is played for a user-defined time configured by the 'Alternative Routing Tone Duration' parameter.

### 23.6.2 Alternative Routing Based on SIP Responses

The device can perform alternative routing based on the received SIP response code (i.e., 4xx, 5xx, 6xx, or 8xx). If you have configured this response code in the Reasons for Tel-to-IP Alternative Routing table, the device attempts to re-route the call to an alternative destination, if configured. You can configure up to 10 SIP response codes in the Reasons for Tel-to-IP Alternative Routing table.

Typically, the device performs alternative routing when there is no response at all to an INVITE message. This is done after a user-defined number of INVITE re-transmissions, configured by the SIPMaxRtx parameter. In such a scenario, the device issues itself the SIP response code 408 (Request Timeout). You can also configure the device to perform alternative routing for the following proprietary response codes that are issued by the device itself:

- **805 IP Profile Call Limit:** The device generates this response code when Call Admission Control (CAC) limits are exceeded for an IP Group. The CAC rules are configured in the IP Profile table (see "Configuring IP Profiles" on page 366). When this occurs, the device sends a SIP 480 (Temporarily Unavailable) response to the SIP entity.
- **806 Media Limits Exceeded:** The device generates this response code when the call is terminated due to crossed thresholds of QoE metrics such as MOS, packet delay, and packet loss (configured in the Quality of Experience Profile table) and/or media bandwidth (configured in the Bandwidth profile table). When this occurs, the device sends a SIP 480 (Temporarily Unavailable) response to the SIP entity. This is configured by 1) assigning an IP Group a QoE and/or Bandwidth profile that rejects calls if the threshold is crossed, 2) configuring 806 in the Reasons for Tel-to-IP Alternative Routing table and 3) configuring an alternative routing rule.

**Note:** The device also plays a tone to the endpoint whenever an alternative route is used. This tone is played for a user-defined time, configured by the AltRoutingToneDuration parameter.

Depending on configuration, the alternative routing is done using one of the following configuration entities:

- **Tel-to-IP Routing Rules:** You configure alternative routing rules for a specific routing rule in the Tel-to-IP Routing table. If the destination of the "main" routing rule is unavailable, the device searches the table for the next matching rule (e.g., destination phone number), and if available attempts to re-route the call to the IP destination configured for this alternative routing rule. For more information on configuring alternative Tel-to-IP routing rules, see ''Configuring Tel-to-IP Routing Rules'' on page 409. 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 23-7: Alternative Routing based on SIP Response Code Example**

| | Destination Phone Prefix | IP Destination | SIP Response | Rule Used? |
|---|---|---|---|---|
| **Main Route** | 40 | 10.33.45.68 | 408 Request Timeout | No |
| **Alternative Route #1** | 40 | 10.33.45.70 | 486 Busy Here | No |
| **Alternative Route #2** | 40 | 10.33.45.72 | 200 OK | Yes |

- **Proxy Sets:** Proxy Sets are used for Server-type IP Groups (e.g., an IP PBX or proxy), which define the address (IP address or FQDN) of the server (see ''Configuring Proxy Sets'' on page 329). As you can configure multiple IP destinations per Proxy Set, the device supports proxy redundancy, which works together with the alternative routing feature. If the destination of a routing rule in the Tel-to-IP Routing table is an IP Group, the device routes the call to the IP destination configured for the Proxy Set associated with the IP Group. If the first IP destination of the Proxy Set is unavailable, the device attempts to re-route the call to the next proxy destination, and so on until an available IP destination is located. To enable the Proxy Redundancy feature for a Proxy Set, set the IsProxyHotSwap parameter to 1 and the EnableProxyKeepAlive parameter to 1.

  When the Proxy Redundancy feature is enabled, the device continually monitors the connection with the proxies by using keep-alive messages (SIP OPTIONS). The device sends these messages every user-defined interval (ProxyKeepAliveTime parameter). If the first (primary) proxy in the list replies with a SIP response code that you have also configured by the 'Keep-Alive Failure Responses' parameter, the device considers the Proxy as down; otherwise, the device considers the proxy as "alive". If the proxy is still considered down after a user-defined number of re-transmissions (configured by the HotSwapRtx parameter), the device attempts to communicate (using the same INVITE) with the next configured (redundant) proxy in the list, and so on until an available redundant proxy is located. Once an available proxy is located, the device can operate in one of the following modes (configured by the ProxyRedundancyMode parameter):

  - **Parking mode:** The device continues operating with the redundant proxy (now active) until the next failure occurs, after which it switches to the next redundant proxy.
  - **Homing mode:** The device always attempts to operate with the primary proxy. In other words, it switches back to the primary proxy whenever it's available again.

  If none of the proxy servers respond, the device goes over the list again.

**Note:**

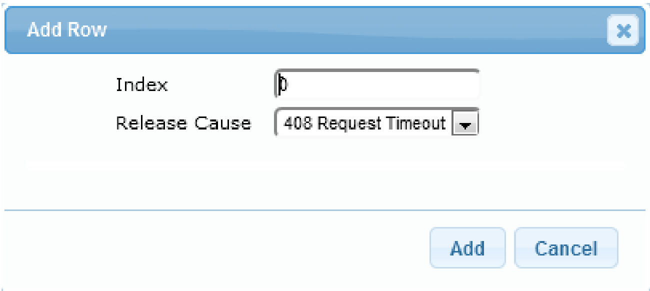
- The device assumes that all the proxy servers belonging to the Proxy Set are synchronized with regards to registered users. Thus, when the device locates an available proxy using the Hot Swap feature, it does not re-register the users; new registration (refresh) is done as normal.
- You can also enable the Busy Out feature, whereby the device can take specified actions if all Proxy Sets of associated destination IP Groups of matching routing rules in the Tel-to-IP Routing table do not respond to connectivity checks. For more information, see the EnableBusyOut parameter.
- If the AltRoutingTel2IPEnable parameter is enabled for the IP Connectivity feature (see Alternative Routing Based on IP Connectivity on page 429), the Busy Out feature does not function with the Proxy Set keep-alive mechanism (see below). To use the Busy Out feature with the Proxy Set keep-alive mechanism (for IP Groups), disable the AltRoutingTel2IPEnable parameter.

The steps for configuring alternative Tel-to-IP routing based on SIP response codes are summarized below.

➢ **To configure alternative Tel-to-IP routing based on SIP response codes:**

1. Configure SIP response codes (call failure reasons) that invoke alternative Tel-to-IP routing:
   a. Open the Reasons for Tel-to-IP Alternative Routing page (**Configuration** tab > **VoIP** menu > **Gateway** > **Routing** > **Alternative Reasons** > **Reasons for Tel-to-IP**).
   b. Click **Add**; the following dialog box appears:

**Figure 23-7: Reasons for Tel-to-IP Alternative Routing Table - Add Row Dialog Box**




   c. Configure a SIP response code for alternative routing according to the parameters described in the table below.
   d. Click **Add**.

**Table 23-8: Reasons for Tel-to-IP Alternative Routing Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[AltRouteCauseTel2Ip_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Release Cause<br>`rel-cause`<br>[AltRouteCauseTel2Ip_ReleaseCause] | Defines a SIP response code that if received, the device attempts to route the call to an alternative destination (if configured). |

2. Enable alternative routing based on SIP responses, by setting the 'Redundant Routing Mode' parameter in the Proxy & Registration page to one of the following:
   - **Routing Table:** Tel-to-IP Routing table is used for alternative routing.
   - **Proxy:** Proxy Set redundancy feature is used for alternative routing.
3. If you are using the Tel-to-IP Routing table, configure alternative routing rules with identical call matching characteristics, but with different IP destinations.
4. If you are using the Proxy Set, configure redundant proxies.

### 23.6.3 Alternative Routing upon SIP 3xx with Multiple Contacts

You can configure how the device handles received SIP 3xx responses that contain multiple alternative contacts. The 3xx response indicates that the original destination is unavailable (e.g., 301 Moved Permanently – user cannot be found) and that the call can be redirected to alternative destinations specified in the SIP Contact headers.

Configured by the '3xx Use Alt Route Reasons' parameter, the device can handle the receipt of 3xx responses using one of the following methods:

- The device tries each contact sequentially, listed in the Contact headers, until a successful destination is found. If a contact responds with a SIP 486 or 600, the device does not try to redirect the call to the next contact and drops the call.
- The device tries each contact sequentially, listed in the Contact headers. If a SIP 6xx Global Failure response is received during this process (e.g., 600 Busy Everywhere), the device does not try to redirect the call to the next contact and drops the call.
- The device redirects the call to the first contact listed in the Contact header. If the contact responds with a SIP response that is configured in the Reasons for Tel-to-IP Alternative Routing table (see "Alternative Routing Based on SIP Responses" on page 431), the device tries to redirect the call to the next contact, and so on. If a contact responds with a response that is not configured in the table, the device does not try to redirect the call to the next contact and drops the call.

**Note:** If a SIP 401 or 407 response is received from a contact, the device does not try to redirect the call to the next contact. Instead, the device continues with the regular authentication process, as indicated by these response types.

## 23.7 Alternative Routing for IP-to-Tel Calls

The device supports alternative IP-to-Tel call routing, as described in this section.

### 23.7.1 Alternative Routing to Hunt upon Q.931 Call Release Cause Code

You can configure up to 10 ISDN Q.931 release cause codes, which if received from the Tel side, the device routes the IP-to-Tel call to an alternative Hunt Group, if configured. Alternative IP-to-Tel routing rules are configured in the Inbound IP Routing table. These rules must be configured anywhere below the "main" routing rule and with identical matching characteristics (e.g., destination prefix number) to the "main" routing rule. The device uses the first alternative route that is available. For more information on configuring alternative IP-to-Tel routing rules in the Inbound IP Routing table, see "Configuring IP-to-Hunt Group Routing Rules" on page 421.

A release cause code indicates that the IP-to-Tel call has been rejected or disconnected on the Tel side. The release cause codes are configured in the Reasons for IP-to-Tel Alternative Routing table. For example, you can configure alternative IP-to-Tel routing for scenarios where the initial Tel destination is busy and a Q.931 Cause Code No. 17 is received (or for other call releases that issue the default Cause Code No. 3).

You can configure a default release cause code that the device issues itself upon the following scenarios:

- The device initiates a call release whose cause is unknown.
- No free channels (i.e., busy) in the Hunt Group.
- No appropriate routing rule located in the Inbound IP Routing table.
- Phone number is not located in the Inbound IP Routing table.

This default release code is set to Cause Code No. 3 (No Route to Destination).You can change the code number using the 'Default Release Cause' parameter, located on the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**).

**Notes:**

- If a Hunt Group is disconnected or not synchronized, the device issues itself the internal Cause Code No. 27. This cause code is mapped (by default) to SIP 502.
- The default release cause is described in the Q.931 notation and translated to corresponding SIP 40x or 50x values (e.g., Cause Code No. 3 to SIP 404, and Cause Code No. 34 to SIP 503).
- For analog interfaces: For information on mapping PSTN release causes to SIP responses, see PSTN Release Cause to SIP Response Mapping on page 408.
- For mapping SIP-to-Q.931 and Q.931-to-SIP release causes, see Configuring Release Cause Mapping on page 408.

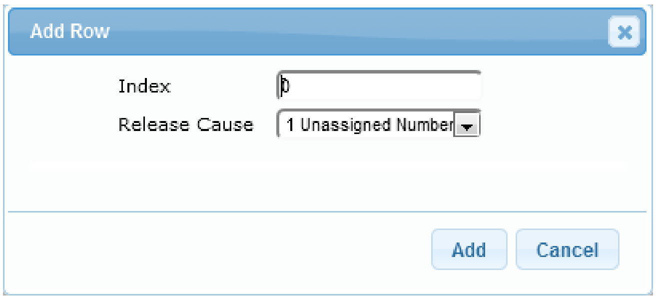
The following procedure describes how to configure alternative routing reasons for IP-to-Tel calls through the Web interface. You can also configure it through ini file (AltRouteCauseIP2Tel) or CLI (configure voip/gw routing alt-route-cause-ip2tel).

➢ **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 Q.931 cause codes that invoke alternative IP-to-Tel routing:
   a. Open the Reasons for IP-to-Tel Alternative Routing table (**Configuration** tab > **VoIP** menu > **Gateway** > **Routing** > **Alternative Routing Reasons** > **Reasons for IP-to-Tel**).
   b. Click **Add**; the following dialog box appears:

**Figure 23-8: Reasons for IP-to-Tel Alternative Routing Table - Add Row Dialog Box**




   c. Configure a Q.931 release cause code for alternative routing according to the parameters described in the table below.
   d. Click **Add**.

**Table 23-9: Reasons for IP-to-Tel Alternative Routing Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[AltRouteCauseIP2Tel_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Release Cause<br>`rel-cause`<br>[AltRouteCauseIP2Tel_ReleaseCause] | Defines a Q.931 release code that if received, the device attempts to route the call to an alternative destination (if configured). |

## 23.7.2 Alternative Routing to an IP Destination upon a Busy Hunt

The Forward on Busy Trunk Destination table lets you configure alternative routing rules for forwarding (i.e., call redirection) IP-to-Tel calls to an alternative IP destination using SIP 3xx responses. These rules are used upon the following:

- For analog interfaces: Unavailable FXS Hunt Group. This feature can be used, for example, to forward the call to another FXS device.

This feature is configured per Hunt Group. The alternative destination can be defined as a host name or as a SIP Request-URI user name and host part (i.e., user@host). For example, the below configuration forwards IP-to-Tel calls to destination user “112” at host IP address 10.13.4.12, port 5060, using transport protocol TCP, if Hunt Group ID 2 is unavailable:

```
ForwardOnBusyTrunkDest 1 = 2, 112@10.13.4.12:5060;transport=tcp;
```

When configured with user@host, the original destination number is replaced by the user part.

The device forwards calls using this table only if no alternative IP-to-Tel routing rule has been configured in the Inbound IP Routing table or alternative routing fails and the following reason(s) in the SIP Diversion header of 3xx messages exists:

- "unavailable":

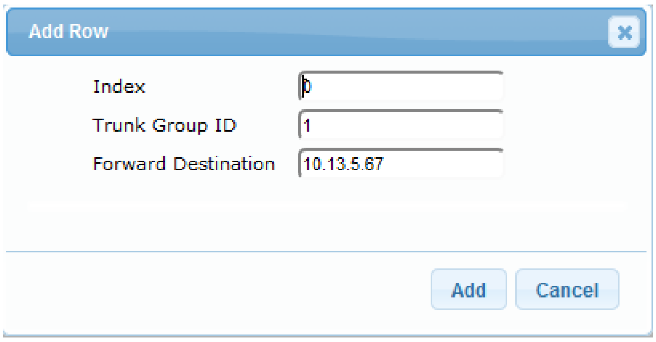
- For analog interfaces: All FXS lines pertaining to a Hunt Group are busy or unavailable

The following procedure describes how to configure Forward on Busy Hunts through the Web interface. You can also configure it through ini file (ForwardOnBusyTrunkDest) or CLI (configure voip/gw routing fwd-on-bsy-trk-dst).

### ➢ To configure a Forward on Busy Hunt Destination rule:

1. Open the Forward on Busy Hunt Destination table (**Configuration** tab > **VoIP** menu > **Gateway** > **Routing** > **Forward on Busy Hunt**).
2. Click **Add**; the following dialog box appears:

**Figure 23-9: Forward on Busy Hunt Destination Table - Add Row Dialog Box**




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.

3. Configure a rule according to the parameters described in the table below.
4. Click **Add**, and then reset the device with a burn-to-flash for your settings to take effect.

**Table 23-10: Forward on Busy Hunt Destination Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[ForwardOnBusyTrunkDest_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Trunk Group ID<br>`trunk-group-id`<br>[ForwardOnBusyTrunkDest_TrunkGroupId] | Defines the Hunt Group ID to which the IP call is destined to. |
| Forward Destination<br>`forward-dst`<br>[ForwardOnBusyTrunkDest_ForwardDestination] | Defines the alternative IP destination for the call used if the Hunt Group is busy or unavailable.<br>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).<br>**Note:** When configured with a user@host, the original destination number is replaced by the user part. |

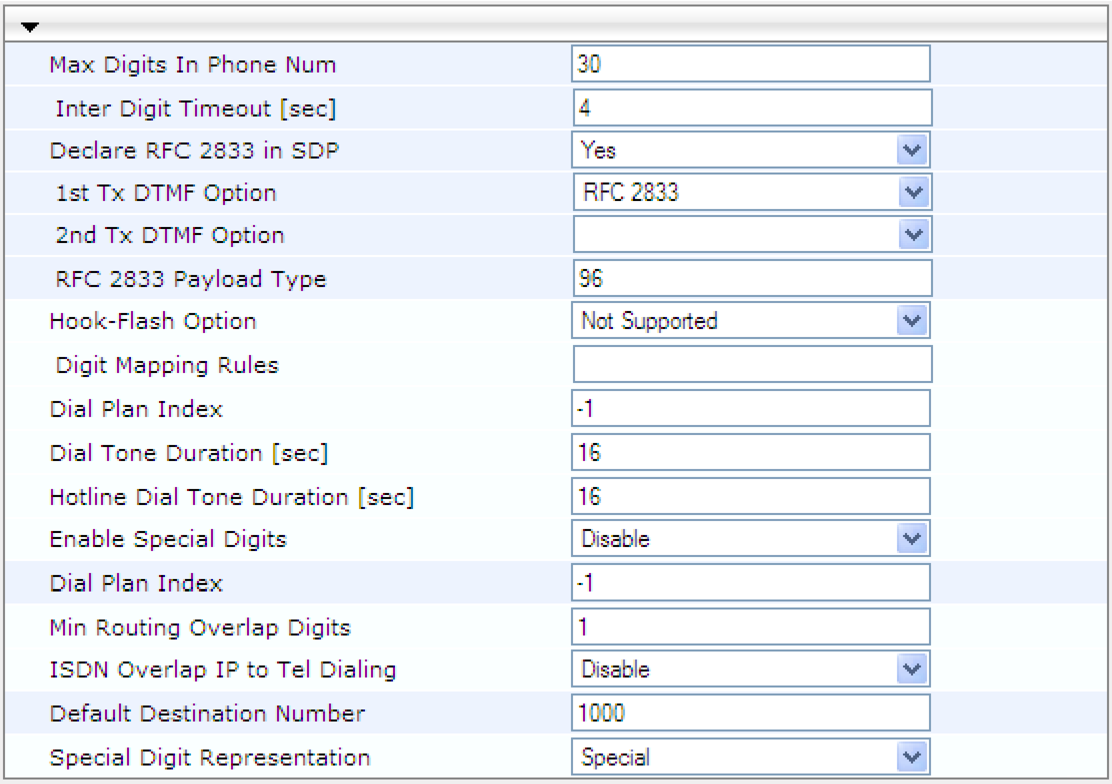
**This page is intentionally left blank**

# 24 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 643.

➢ **To configure the DTMF and dialing parameters:**

1. Open the DTMF & Dialing page (**Configuration** tab > **VoIP** menu > **Gateway** > **DTMF & Supplementary** > **DTMF & Dialing**).




2. Configure the parameters as required.
3. Click **Submit**.
4. To save the changes to flash memory, see "Saving Configuration" on page 490.

## 24.1 Dialing Plan Features

This section describes various dialing plan features supported by the device.

### 24.1.1 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) upon any of the following scenarios:

- Maximum number of digits is received. You can define (using the MaxDigits parameter) the maximum number of collected destination number digits that can be received (i.e., dialed) from the Tel side by the device. When the number of collected digits reaches the maximum (or a digit map pattern is matched), the device uses these digits for the called destination number.
- Inter-digit timeout expires (e.g., for open numbering schemes). This is defined using the TimeBetweenDigits parameter. This is the time that the device waits between each received digit. When this inter-digit timeout expires, the device uses the collected digits to dial the called destination number.
- The phone's pound (#) key is pressed.
- Digit string (i.e., dialed number) matches one of the patterns defined in the digit map.

Digit map (pattern) rules are defined using the DigitMapping parameter. The digit map pattern can contain up to 52 options (rules), each separated by a vertical bar ("|"). The maximum length of the entire digit pattern is 152 characters. The available notations are described in the table below:

**Table 24-1: 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.<br>**Note:** This notation does not apply to some scenarios when using the star (*) or hash (#) key. For example, the key sequence of ** must be presented in the dial plan as *x.s (instead of xx). |
| **T** | Dial timeout (configured by the TimeBetweenDigits parameter). |
| **S** | Short timer (configured by the TimeBetweenDigits parameter; default is two seconds) that can be used when a specific rule is defined after a more general rule. For example, if the digit map is 99\|998, then the digit collection is terminated after the first two 9 digits are received. Therefore, the second rule of 998 can never be matched. But when the digit map is 99s\|998, then after dialing the first two 9 digits, the device waits another two seconds within which the caller can enter the digit 8. |

Below is an example of a digit map pattern containing eight rules:

```
DigitMapping = 11xS|00[1-
7]xxx|8xxxxxxx|#xxxxxxx|*xx|91xxxxxxxxxx|9011x|xx.T
```

In the example, the rule "00[1-7]xxx" denotes dialed numbers that begin with 00, and then any digit from 1 through 7, followed by three digits (of any number). Once the device receives these digits, it does not wait for additional digits, but starts sending the collected digits (dialed number) immediately.

**Notes:**

- If you want the device to accept/dial any number, ensure that the digit map contains the rule "xx.T"; otherwise, dialed numbers not defined in the digit map are rejected.
- If you are using an external Dial Plan file for dialing plans (see "Dialing Plans for Digit Collection" on page 501), 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.

### 24.1.2 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 501.

# 25 Configuring Supplementary Services

This section describes SIP supplementary services that can enhance your telephone service.

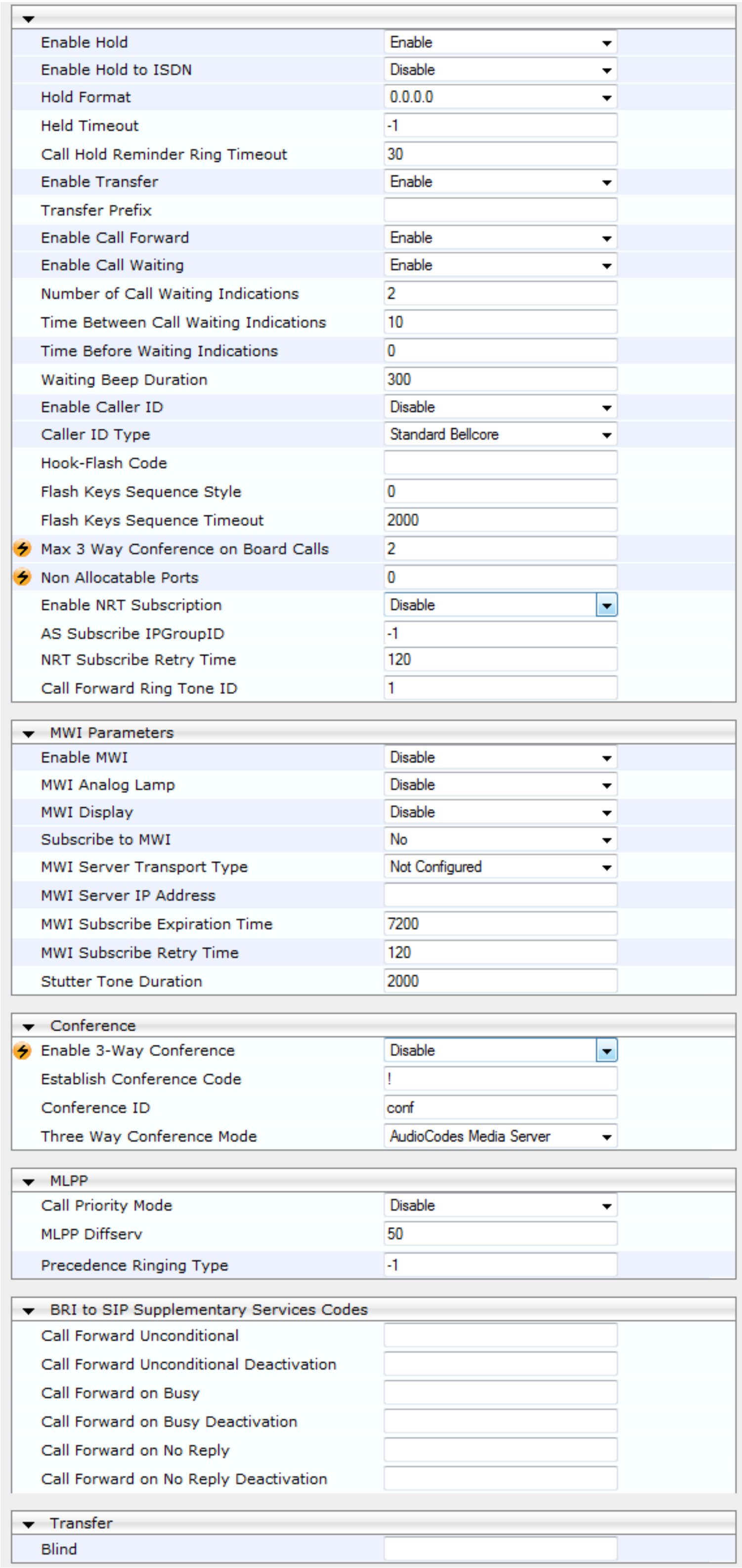
**Notes:**

- 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 643.

➢ **To configure supplementary services parameters:**

1. Open the Supplementary Services page (**Configuration** tab > **VoIP** menu > **Gateway** > **DTMF & Supplementary** > **Supplementary Services**).



| | |
|---|---|
| Enable Hold | Enable |
| Enable Hold to ISDN | Disable |
| Hold Format | 0.0.0.0 |
| Held Timeout | -1 |
| Call Hold Reminder Ring Timeout | 30 |
| Enable Transfer | Enable |
| Transfer Prefix | |
| Enable Call Forward | Enable |
| Enable Call Waiting | Enable |
| Number of Call Waiting Indications | 2 |
| Time Between Call Waiting Indications | 10 |
| Time Before Waiting Indications | 0 |
| Waiting Beep Duration | 300 |
| Enable Caller ID | Disable |
| Caller ID Type | Standard Bellcore |
| Hook-Flash Code | |
| Flash Keys Sequence Style | 0 |
| Flash Keys Sequence Timeout | 2000 |
| Max 3 Way Conference on Board Calls | 2 |
| Non Allocatable Ports | 0 |
| Enable NRT Subscription | Disable |
| AS Subscribe IPGroupID | -1 |
| NRT Subscribe Retry Time | 120 |
| Call Forward Ring Tone ID | 1 |
| **MWI Parameters** | |
| Enable MWI | Disable |
| MWI Analog Lamp | Disable |
| MWI Display | Disable |
| Subscribe to MWI | No |
| MWI Server Transport Type | Not Configured |
| MWI Server IP Address | |
| MWI Subscribe Expiration Time | 7200 |
| MWI Subscribe Retry Time | 120 |
| Stutter Tone Duration | 2000 |
| **Conference** | |
| Enable 3-Way Conference | Disable |
| Establish Conference Code | ! |
| Conference ID | conf |
| Three Way Conference Mode | AudioCodes Media Server |
| **MLPP** | |
| Call Priority Mode | Disable |
| MLPP Diffserv | 50 |
| Precedence Ringing Type | -1 |
| **BRI to SIP Supplementary Services Codes** | |
| Call Forward Unconditional | |
| Call Forward Unconditional Deactivation | |
| Call Forward on Busy | |
| Call Forward on Busy Deactivation | |
| Call Forward on No Reply | |
| Call Forward on No Reply Deactivation | |
| **Transfer** | |
| Blind | |

2. Configure the parameters as required.
3. Click **Submit**, or click the **Subscribe to MWI** or **Unsubscribe to MWI** buttons to save your changes and to subscribe / unsubscribe to the MWI server.
4. To save the changes to flash memory, see "Saving Configuration" on page 490.

## 25.1 Call Hold and Retrieve

Initiating Call Hold and Retrieve:

- Active calls can be put on-hold by pressing the phone's hook-flash button.
- The party that initiates the hold is called the *holding* party; the other party is called the *held* party.
- After a successful Hold, the holding party hears a dial tone (HELD_TONE defined in the device's Call Progress Tones file).
- Call retrieve can be performed only by the holding party while the call is held and active.
- The holding party performs the retrieve by pressing the telephone's hook-flash button.
- After a successful retrieve, the voice is connected again.
- Hold is performed by sending a re-INVITE message with IP address 0.0.0.0 or a=sendonly in the SDP, according to the HoldFormat parameter.
- The hold and retrieve functionalities are implemented by re-INVITE messages. The IP address 0.0.0.0 as the connection IP address or the string ‘a=inactive’ in the received re-INVITE SDP cause the device to enter Hold state and to play the held tone (configured in the device) to the PBX/PSTN. If the string ‘a=sendonly’ is received in the SDP message, the device stops sending RTP packets, but continues to listen to the incoming RTP packets. Usually, the remote party plays, in this scenario, Music on Hold (MOH) and the device forwards the MOH to the held party.

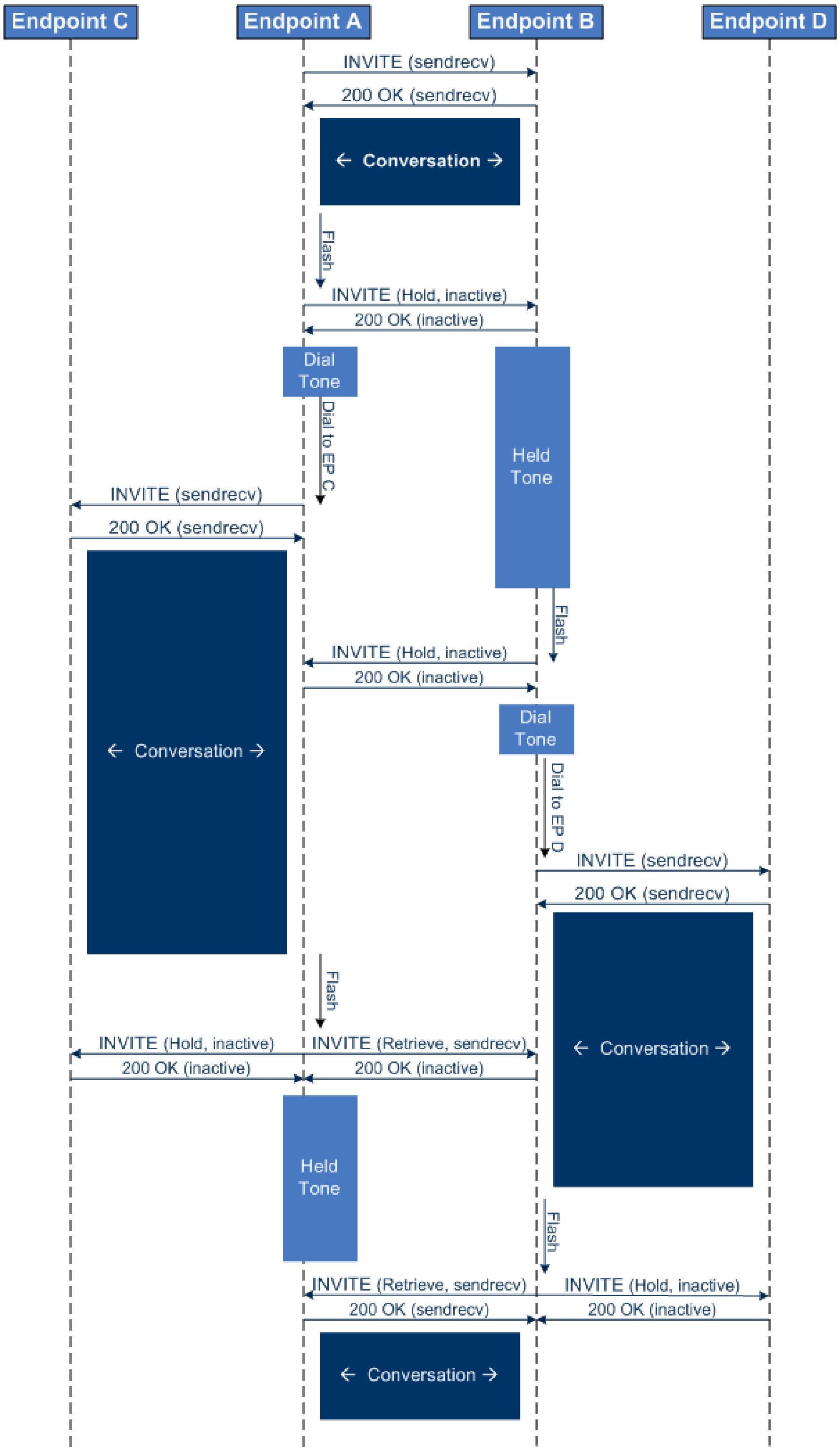
Receiving Hold/Retrieve:

- When an active call receives a re-INVITE message with IP address 0.0.0.0 or ‘inactive’ string in SDP, the device stops sending RTP and plays a local held tone.
- When an active call receives a re-INVITE message with the ‘sendonly’ string in SDP, the device stops sending RTP and listens to the remote party. In this mode, it is expected that music on-hold (or any other hold tone) is played (over IP) by the remote party.

You can also configure the device to keep a call on-hold for a user-defined time after which the call is disconnected, using the HeldTimeout parameter.

The device also supports "double call hold" for FXS interfaces where the called party, which has been placed on-hold by the calling party, can then place the calling party on hold as well and make a call to another destination. The flowchart below provides an example of this type of call hold:

**Figure 25-1: Double Hold SIP Call Flow**




The flowchart above describes the following "double" call-hold scenario:

1. A calls B and establishes a voice path.
2. A places B on hold; A hears a dial tone and B hears a held tone.
3. A calls C and establishes a voice path.
4. B places A on hold; B hears a dial tone.
5. B calls D and establishes a voice path.
6. A ends call with C; A hears a held tone.
7. B ends call with D.
8. B retrieves call with A.

**Notes:**

- If a party that is placed on hold (e.g., B in the above example) is called by another party (e.g., D), then the on-hold party receives a call waiting tone instead of the held tone.
- While in a Double Hold state, placing the phone on-hook disconnects both calls (i.e. call transfer is not performed).
- You can enable the device to handle incoming re-INVITE messages with "a=sendonly" in the SDP, in the same way as if "a=inactive" is received in the SDP. This is configured using the SIPHoldBehavior parameter. When enabled, the device plays a held tone to the Tel phone and responds with a 200 OK containing "a=recvonly" in the SDP.

## 25.2 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.

## 25.3 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.

# 25.4 Call Transfer

This section describes the device's support for call transfer types.

## 25.4.1 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

**Note:** For FXS interfaces, the device can also handle call transfers using SIP INVITE and re-INVITE messages, instead of REFER messages. This is useful when communicating with SIP UAs that do not support the receipt of REFER messages. This feature is applicable to FXS interfaces. To enable this support, use the EnableCallTransferUsingReinvites parameter.

## 25.4.2 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 (similar to consultation transfer, but skipping the consultation stage).

# 25.5 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 device).
- Diverted party: new destination of the forwarded call (FXS device).

The served party (FXS interface) can be configured through the Web interface (see Configuring Call Forward on page 477) or ini file to activate one of the call forward modes. These modes are configurable per endpoint.

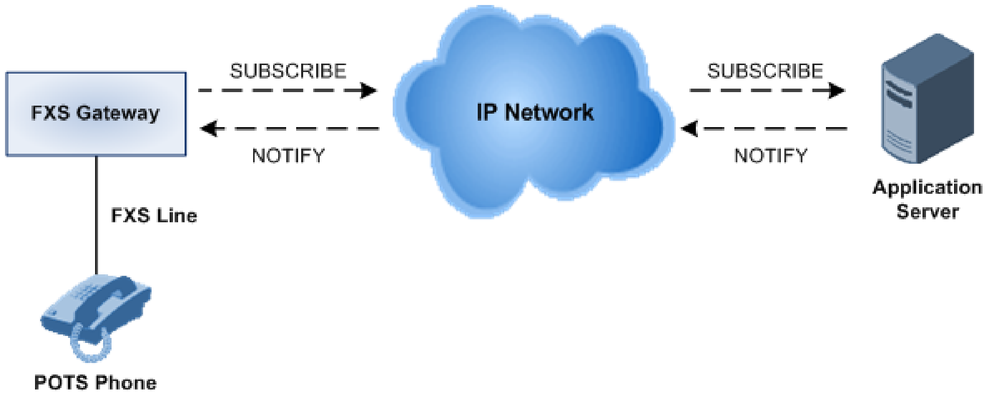
**Notes:**

- 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.

### 25.5.1 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 25-2: Call Forward Reminder with Application Server**




The device generates a Call Forward Reminder ring burst to the FXS endpoint each time it receives a SIP NOTIFY message with a "reminder ring" xml body. The NOTIFY request is sent from the Application Server to the device each time the Application Server forwards an incoming call. The service is cancelled when an UNSUBSCRIBE request is sent from the device, or when the Subscription time expires.

The reminder-ring tone can be defined by using the parameter CallForwardRingToneID, which points to a ring tone defined in the Call Progress Tone file.

The following parameters are used to configure this feature:

- EnableNRTSubscription
- ASSubscribeIPGroupID
- NRTSubscribeRetryTime

- CallForwardRingToneID

## 25.5.2 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.

## 25.5.3 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 =
Desvió-Inmediato
```

From this point on, the device plays a normal dial tone to the FXS phone when it goes off-hook.

## 25.6 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.

To indicate Call Waiting, the device sends a 182 Call Queued response. The device identifies Call Waiting when a 182 Call Queued response is received.

**Note:** The Call Waiting feature is applicable only to FXS interfaces.

## 25.7 Message Waiting Indication

The device supports Message Waiting Indication (MWI) according to IETF RFC 3842. The device also supports subscribing to an MWI server (using SIP SUBSCRIBE messages).

For analog interfaces: The FXS device can accept a SIP MWI NOTIFY message that indicates waiting messages or cleared messages. Users are informed of these messages by a stutter dial tone. You can define the stutter and confirmation tones in the CPT file. If the MWI display is configured, the number of waiting messages is also displayed. If the MWI lamp is configured, the phone's lamp (on a phone that is equipped with an MWI lamp) is lit. The device can subscribe to the MWI server per port (usually used on FXS).

> **Note:** For more information on configuring IP-based voice mail, refer to the *IP Voice Mail CPE Configuration Guide*.

To configure MWI, use the following parameters:

- EnableMWI
- MWIServerIP, or MWISubscribeIPGroupID and ProxySet
- MWIAnalogLamp
- MWIDisplay
- StutterToneDuration
- EnableMWISubscription
- MWIExpirationTime
- SubscribeRetryTime
- SubscriptionMode
- CallerIDType (determines the standard for detection of MWI signals)
- ETSIVMWITypeOneStandard
- BellcoreVMWITypeOneStandard
- VoiceMailInterface
- EnableVMURI

## 25.8 Caller ID

This section describes the device's Caller ID support.

**Note:** The Caller ID feature is applicable only to FXS interfaces.

### 25.8.1 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

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) 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).
- 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.

### 25.8.2 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.
- **FXS 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.

## 25.9 Three-Way Conferencing

The device supports three-way conference calls. Multiple, concurrent three-way conference calls are also supported. The device supports the following conference modes:

- **Conference Managed by External, AudioCodes Conferencing (Media) Server:** The conference-initiating INVITE sent by the device uses the ConferenceID concatenated with a unique identifier as the Request-URI. This same Request-URI is set as the Refer-To header value in the REFER messages that are sent to the two remote parties. This mode is configured by setting the 3WayConferenceMode parameter to 0 (default.)

  To join a conference, the Request-URI includes the Conference ID string preceded by the number of the participants in the conference and terminated by a unique number. INVITE messages with the same URI join the same conference. For example:

  ```
  INVITE sip:4conf1234@10.1.10.10
  ```

- **Conference Managed by External, Third-party Conferencing Server:** Two optional modes of operation:

  - The conference-initiating INVITE sent by the device uses only the ConferenceID as the Request-URI. The Conferencing server sets the Contact header of the 200 OK response to the actual unique identifier (Conference URI) to be used by the participants. This Conference URI is included (by the device) in the Refer-To header value in the REFER messages sent by the device to the remote parties. The remote parties join the conference by sending INVITE messages to the Conferencing server using this conference URI. This mode is configured by setting the 3WayConferenceMode parameter to 1.
  - The conference-initiating INVITE sent by the device uses only the ConferenceID as the Request-URI. The Conferencing server sets the Contact header of the 200 OK response to the actual unique identifier (Conference URI) to be used by the participants. The Conference URI is included in the URI of the REFER with a Replaces header sent by the device to the Conferencing server. The Conferencing server then sends an INVITE with a Replaces header to the remote participants. This mode is configured by setting the 3WayConferenceMode parameter to 3.

  To join a conference, the Request-URI includes the Conference ID string preceded by the number of the participants in the conference and terminated by a unique number. INVITE messages with the same URI join the same conference. For example:

  ```
  INVITE sip:4conf1234@10.1.10.10
  ```

- **Local, On-board Conferencing:** The conference is established on the device without the need for an external Conferencing server. This feature includes local mixing and transcoding of the 3-Way Call legs on the device, and even allowing multi-codec conference calls. The number of simultaneous, on-board conferences can be limited using the MaxInBoardConferenceCalls parameter. The device supports up to five simultaneous, on-board, three-way conference calls. This mode is configured by setting the 3WayConferenceMode parameter to 2.

**Notes:**

- Three-way conferencing using an external conference server is supported only by FXS interfaces.
- Instead of using the flash-hook button to establish a three-way conference call, you can dial a user-defined hook-flash code (e.g., "*1"), configured by the HookFlashCode parameter.
- Three-way conferencing is applicable only to FXS interfaces.

The following example demonstrates three-way conferencing using the device's local, on-board conferencing feature. In this example, telephone "A" connected to the device establishes a three-way conference call with two remote IP phones, "B" and "C":

1. A establishes a regular call with B.
2. A places B on hold, by pressing the telephone's flash-hook button and the number "1" key.
3. A hears a dial tone and then makes a call to C.
4. C answers the call.
5. A establishes a three-way conference call with B and C, by pressing the flash-hook button and the number "3" key.

To configure local, on-board three-way conferencing:

1. Open the Supplementary Services page.
2. Set 'Enable 3-Way Conference' to **Enable** (Enable3WayConference = 1).
3. Set '3-Way Conference Mode' to **On Board** (3WayConferenceMode = 2).
4. Set 'Flash Keys Sequence Style' to **Sequence 1** or **Sequence 2** (FlashKeysSequenceStyle = 1 or 2).

## 25.10 Emergency E911 Phone Number Services

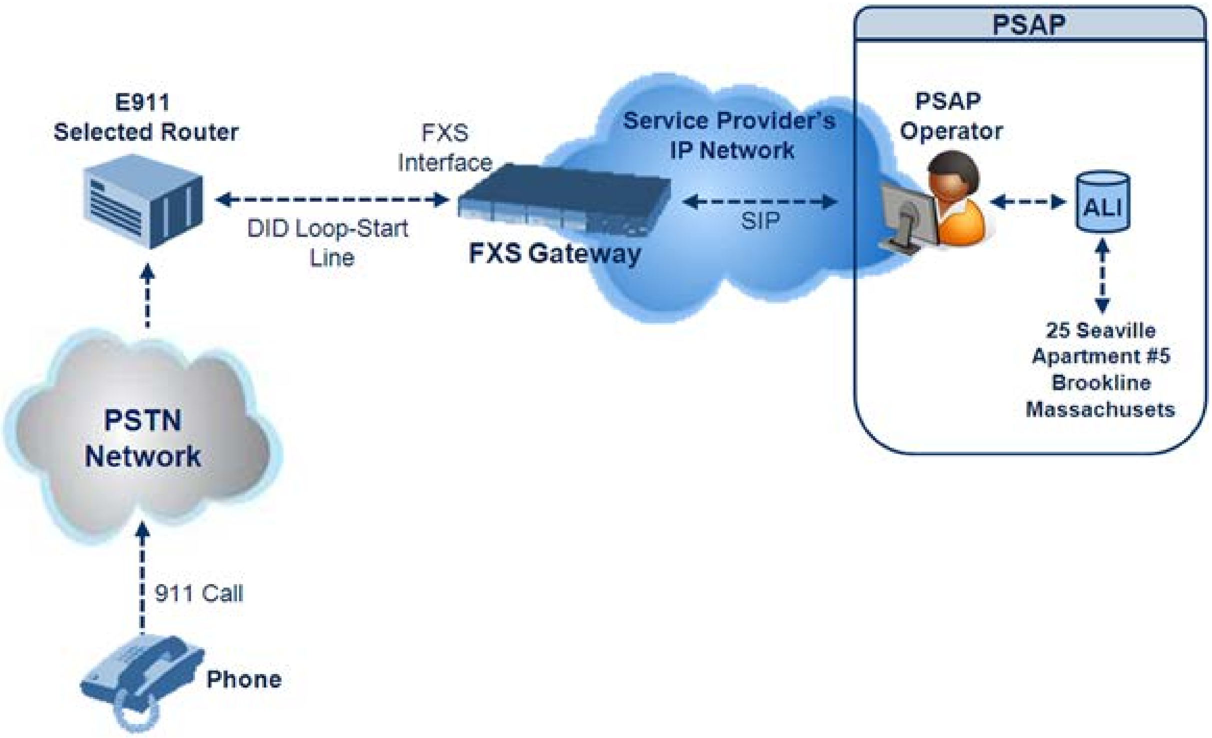
This section describes the device's support for emergency phone number services. The device supports the North American emergency telephone number system known as Enhanced 911 (E911), according to the TR-TSY-000350 and Bellcore's GR-350-Jun2003 standards. The E911 emergency system automatically associates a physical address with the calling party's telephone number, and routes the call to the most appropriate (closest) Public Safety Answering Point (PSAP), allowing the PSAP to quickly dispatch emergency response (e.g., police) to the caller's location.

Typically, the dialed emergency number is routed to the appropriate PSAP by the telephone company's switch, known as a 911 Selective Router (or E911 tandem switch). If the PSAP receives calls from the telephone company on old-style digital trunks, they are specially formatted Multi-Frequency (MF) trunks that pass only the calling party's number (known as Automatic Number Identification - ANI). Once the PSAP receives the call, it searches for the physical address that is associated with the calling party's telephone number (in the Automatic Location Identification database - ALI).

### 25.10.1 FXS Device Emulating PSAP using DID Loop-Start Lines

The device's FXS interface can be configured to emulate PSAP (using DID loop start lines), according to the Telcordia GR-350-CORE specification.

**Figure 25-3: FXS Device Emulating PSAP using DID Loop-Start Lines**




The call flow of an E911 call to the PSAP is as follows:

1. The E911 tandem switch seizes the line.
2. The FXS device detects the line seize, and then generates a wink signal (nominal 250 msec). The wink can be delayed by configuring the parameter DelayBeforeDIDWink to 200 (for 200 msec or a higher value).
3. The switch detects the wink and then sends the MF Spill digits with ANI and (optional) Pseudo-ANI (P ANI).
4. The FXS device collects the MF digits, and then sends a SIP INVITE message to the PSAP with all collected MF digits in the SIP From header as one string.

5. The FXS device generates a mid-call wink signal (two subsequent polarity reversals) toward the E911 tandem switch upon either detection of an RFC 2833 "hookflash" telephony event, or if a SIP INFO message with a "hooflash" body is received from the PSAP (see the example below). The duration of this "flashhook" wink signal is configured using the parameter FlashHookPeriod (usually 500 msec). Usually the wink signal is followed by DTMF digits sent by PSAP to perform call transfer. Another way to perform the call transfer is to use SIP REFER messages, as described below.
6. The FXS device supports call transfer initiated by the PSAP. If it receives a SIP REFER message with the Refer-To URI host part containing an IP address that is equal to the device's IP address, the FXS device generates a 500-msec wink signal (double polarity reversals), and then (after a user-defined interval configured by the parameter WaitForDialTime), plays DTMF digits according to the transfer number received in the SIP Refer-To header URI userpart.
7. When the call is answered by the PSAP operator, the PSAP sends a SIP 200 OK to the FXS device, and the FXS device then generates a polarity reversal signal to the E911 switch.
8. After the call is disconnected by the PSAP, the PSAP sends a SIP BYE to the FXS device, and the FXS device reverses the polarity of the line toward the tandem switch.

The following parameters need to be configured:

- EnableDIDWink = 1
- EnableReversalPolarity = 1
- PolarityReversalType = 1
- FlashHookPeriod = 500 (for 500 msec "hookflash" mid-call Wink)
- WinkTime = 250 (for 250 msec signalling Wink generated by the FXS device after it detects the line seizure)
- EnableTransfer = 1 (for call transfer)
- LineTransferMode = 1 (for call transfer)
- WaitforDialTime = 1000 (for call transfer)
- SwapTEl2IPCalled&CallingNumbers = 1
- DTMFDetectorEnable = 0
- MFR1DetectorEnable = 1
- DelayBeforeDIDWink = 200 (for 200 msec) - can be configured in the range from 0 (default) to 1000.

**Note:** Modification of the WinkTime parameter requires a device reset.

The outgoing SIP INVITE message contains the following headers:

```
INVITE sip:Line@DomainName
From: <sip:*81977820#@sipgw>;tag=1c143
To: <sip:Line@DomainName>
```

Where:

- *Line* = as configured in the Endpoint Phone Number Table
- *SipGtw* = configured by the SIPGatewayName parameter
- *From* header/user part = calling party number as received from the MF spill

The ANI and the pseudo-ANI numbers are sent to the PSAP either in the From and/or P-AssertedID SIP header.

Typically, the MF spills are sent from the E911 tandem switch to the PSAP, as shown in the table below:

**Table 25-1: Dialed MF Digits Sent to PSAP**

| Digits of Calling Number | Dialed MF Digits |
|---|---|
| 8 digits "nnnnnnnn" (ANI) | "KPnnnnnnnnST" |
| 12 digits "nnnnnnnnnnnn" (ANI) | "KPnnnnnnnnnnnnSTP" |
| 12 digits ANI and 10 digits PANI | "KPnnnnnnnnnnnnSTKPmmmmmmmmmmST" |
| two digits "nn" | "KPnnSTP" |

The MF KP, ST, and STP digits are mapped as follows:

- * for KP
- # for ST
- B for STP

For example, if ANI and PANI are received, the SIP INVITE contains the following From header:

```
From: <sip:*nnnnnnnnnnnn#*mmmmmmmmmm#@10.2.3.4>;tag=1c14
```

**Note:** It is possible to remove the * and # characters, using the device's number manipulation rules.

If the device receives the SIP INFO message below, it then generates a "hookflash" mid-call Wink signal:

```
INFO sip:4505656002@192.168.13.40:5060 SIP/2.0
Via:  SIP/2.0/UDP 192.168.13.2:5060
From: port1vega1 <sip:06@192.168.13.2:5060>
To: <sip:4505656002@192.168.13.40:5060>;tag=132878796-
1040067870294
Call-ID: 0010-0016-D69A7DA8-1@192.168.13.2
CSeq:2 INFO
Content-Type: application/broadsoft
Content-Length: 17
event flashhook
```

## 25.11 Configuring Multi-Line Extensions and Supplementary Services

The Supplementary Services table lets you configure up to 100 supplementary services for endpoints connected to the device. These endpoints include analog FXS phones.

The table can be used for the following functionalities:

- Configuring multiple phone line extension numbers per FXS port, supporting point-to-multipoint configuration of several phone numbers per FXS channel.
- Registration of each line extension (endpoint), using a user-defined user ID and password, to a third-party softswitch for authentication and/or billing. For each line extension, the device sends a SIP REGISTER to the softswitch, using the global number in the From/To headers. If authentication is necessary for registration, the device sends the endpoint’s user ID and password in the SIP MD5 Authorization

header. For viewing registration status, see "Viewing Registration Status" on page 566.

- Caller ID name per line extension, which is displayed to the called party (if enabled).
- Enabling receipt by the line extension of caller ID from incoming calls.
- Routing IP-to-Tel calls (including voice and fax) to specific endpoints based on called line extension number (local number). To enable this functionality, in the Hunt Group Settings table, set the 'Channel Select Mode' field to **Select Trunk by Supplementary Services Table** for the Hunt Group to which the FXS port belongs (see "Configuring Hunt Group Settings" on page 386).
- Mapping local numbers (line extension number) with global phone numbers (E.164). The endpoint can be configured with two numbers – *local* and *global*. The local number represents the endpoint's line extension number (e.g., PBX extension number); the global number represents the corresponding E.164 number used for the IP side in the SIP message:
  - IP-to-Tel calls: Maps the called global number in the user part of the SIP Request-URI in the incoming SIP message to the local number sent to the Tel side. For example, the device receives an incoming IP call with a destination (called) that is a global number 638002 and then sends the call to the Tel side with the destination number manipulated to the corresponding local number of 402.
  - Tel-to-IP Calls: Maps the calling (source) local number of the Tel side to the global number sent to the IP side (in the From and To headers of the outgoing SIP message). For example, if the device receives a Tel call from line extension local number 402, it changes this calling number to 638002 and then sends the call to the IP side with this calling number. This functionality in effect, validates the calling number.

> **Note:** If you have configured regular Tel-to-IP or IP-to-Tel manipulation rules (see "Configuring Source/Destination Number Manipulation Rules" on page 393), the device applies them before applying the local-global mapping rules configured in the table.

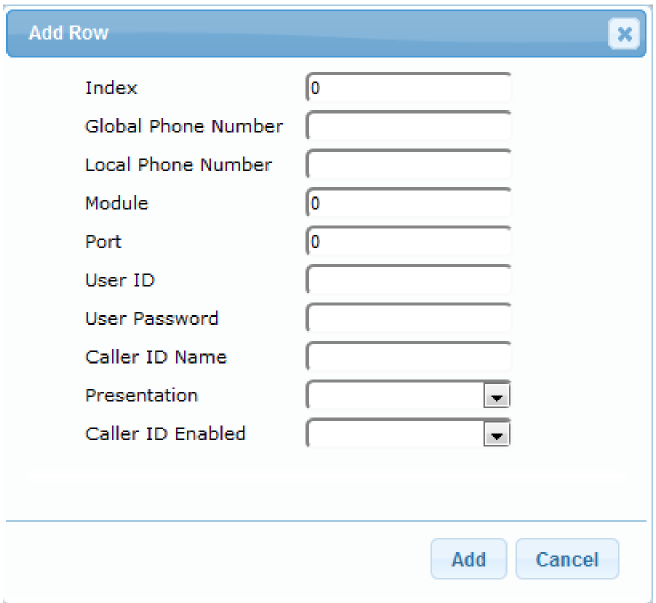
The following procedure describes how to configure the Supplementary Services table through the Web interface. You can also configure it through ini file (ISDNSuppServ) or CLI (configure voip > gw digitalgw isdn-supp-serv).

**➢ To configure endpoint supplementary services:**

1. Open the Supplementary Services table (**Configuration** tab > **VoIP** menu > **Gateway** > **DTMF and Supplementary** > **Supp Services Table**).

2. Click **Add**; the following dialog box appears:

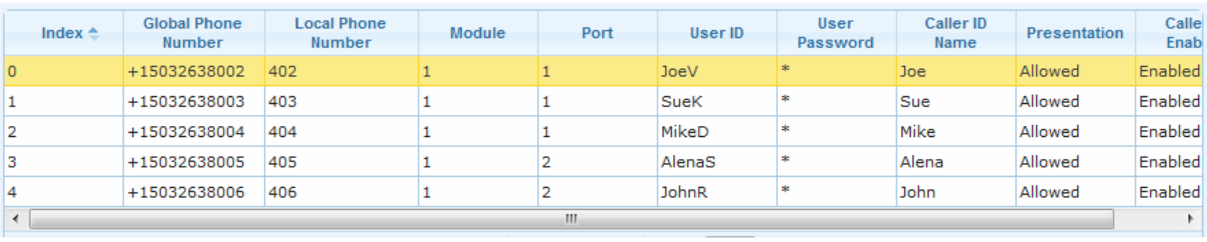
**Figure 25-4: Supplementary Services Table - Add Row Dialog Box**




3. Configure a supplementary service according to the parameters described in the table below.
4. Click **Add**.

The figure below displays an example of multiple-line extensions configured in the Supplementary Services table:

**Figure 25-5: Supplementary Services Table Page**




You can register and un-register an endpoint configured in the table. The registration method is according to the 'Registration Mode' parameter located in the Hunt Group Settings page (see "Configuring Hunt Group Settings" on page 386).

### ➢ To register or un-register an endpoint:

1. Select the required table row in which the endpoint is configured.
2. From the 'Action' drop-down list, select **Register**. To unregister the endpoint, select **Un-Register**.

**Table 25-2: Supplementary Services Table Parameter Description**

| Parameter | Description |
|---|---|
| Index<br>[ISDNSuppServ_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Global Phone Number<br>`phone-number` | Defines a global telephone extension number for the endpoint. The global number is used for the following functionalities: |

| Parameter | Description |
|---|---|
| [ISDNSuppServ_PhoneNumber] | ▪ Endpoint registration<br>▪ IP-to-Tel routing<br>▪ Mapping between local and global (E.164) numbers between Tel and IP sides respectively |
| Local Phone Number<br>`local-phone-number`<br>[ISDNSuppServ_LocalPhoneNumber] | Defines a local telephone extension number for the endpoint (e.g., the PBX extension number). The local number is used for the following functionalities:<br>▪ Validation of source (calling) number for Tel-to-IP calls<br>▪ Mapping between local and global (E.164) numbers between Tel and IP sides respectively |
| Module<br>`module`<br>[ISDNSuppServ_Module] | Defines the device's module number to which the endpoint is connected. |
| Port<br>`port`<br>[ISDNSuppServ_Port] | Defines the port number on the module to which the endpoint is connected. |
| User ID<br>`user-id`<br>[ISDNSuppServ_UserId] | Defines the User ID for registering the endpoint to a third-party softswitch for authentication and/or billing. |
| User Password<br>`user-password`<br>[ISDNSuppServ_UserPassword] | Defines the user password for registering the endpoint to a third-party softswitch for authentication and/or billing.<br>**Note:** For security, the password is displayed as an asterisk (*). |
| Caller ID Name<br>`caller-id-number`<br>[ISDNSuppServ_CallerID] | Defines the caller ID name of the endpoint (sent to the IP side).<br>The valid value is a string of up to 18 characters. |
| Presentation<br>`presentation-restricted`<br>[ISDNSuppServ_IsPresentationRestricted] | Determines whether the endpoint sends its Caller ID information to the IP when a call is made.<br>▪ **[0]** Allowed = The device sends the string defined in the 'Caller ID' field when this endpoint makes a Tel-to-IP call.<br>▪ **[1]** Restricted = The string defined in the 'Caller ID' field is not sent. |
| Caller ID Enabled<br>`caller-id-enable`<br>[ISDNSuppServ_IsCallerIDEnabled] | Enables the receipt of Caller ID.<br>▪ **[0]** Disabled = The device does not send Caller ID information to the endpoint.<br>▪ **[1]** Enabled = The device sends Caller ID information to the endpoint. |

## 25.12 Configuring Charge Codes

The Charge Codes table lets you configure metering tones that the device generates to the Tel side on its FXS interfaces. To enable the generation of metering tones, see Configuring Metering Tones on page 470.

You can configure up to 25 different Charge Codes, where each table row represents a Charge Code. Each Charge Code can include up to four different time periods in a day (24 hours). The device selects the time period by comparing the device's current time to the end time of each time period of the selected Charge Code. The device generates the number of pulses on answer once the call is connected, and from that point on, it generates a pulse for each pulse interval. If a call starts at a certain time period and crosses to the next, the information of the next time period is used.

To assign a Charge Code to an outgoing Tel-to-IP call, use the Tel-to-IP Routing table.

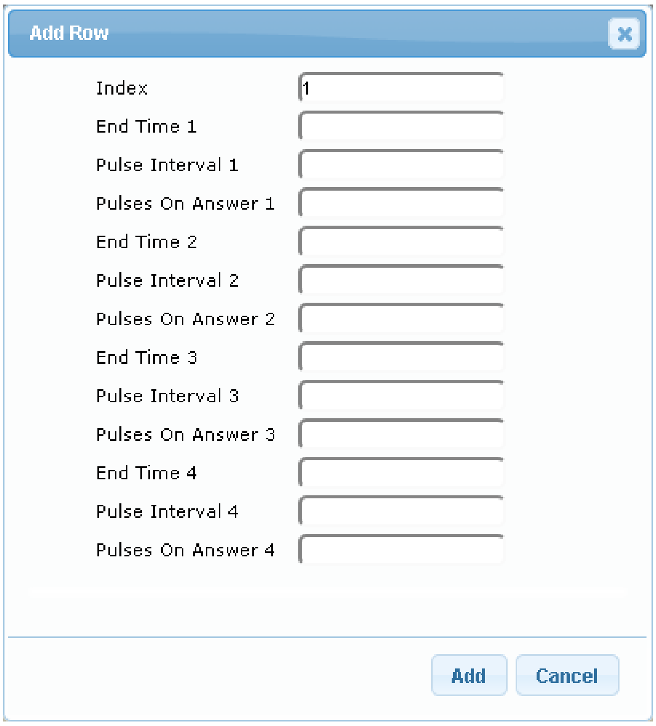
> **Note:** The Charge Codes table is applicable only to FXS interfaces.

The following procedure describes how to configure Charge Codes through the Web interface. You can also configure it through ini file (ChargeCode) or CLI (configure voip > gw analoggw charge-code).

➢ **To configure a Charge Code:**

1. Open the Charge Codes table (**Configuration** tab > **VoIP** menu > **Services** > **Least Cost Routing** > **Charge Codes**).
2. Click **Add**; the following dialog box appears:

**Figure 25-6: Charge Codes Table - Add Row Dialog Box**




3. Configure a Charge Code according to the parameters described in the table below.
4. Click **Add**.

**Table 25-3: Charge Codes Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[ChargeCode_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| End Time (1 - 4)<br>`end-time-<1-4>`<br>[ChargeCode_EndTime<1-4>] | Defines the end of the time period in a 24 hour format, *hh*. For example, "04" denotes 4 A.M.<br>**Notes:**<br>▪ The first time period always starts at midnight (00).<br>▪ 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 (1 - 4)<br>`pulse-interval-<1-4>`<br>[ChargeCode_PulseInterval<1-4>] | Defines the time interval between pulses (in tenths of a second). Once the call is established, the device generates a pulse for each pulse interval. |
| Pulses On Answer (1 - 4)<br>`pulses-on-answer-<1-4>`<br>[ChargeCode_PulsesOnAnswer<1-4>] | Defines the number of pulses that the device generates upon call answer. |

# 25.13 Converting Accented Characters from IP to Tel

The Char Conversion table lets you configure up to 40 Character Conversion rules. A Character Conversion rule maps (converts) accented characters (Unicode / UTF-8) received from the IP side into simple ASCII characters (ISO-8859) for sending to the Tel side. Typically, the device receives the caller ID and calling name in Unicode characters (in the SIP INVITE message). Unicode characters consist of two bytes, while ASCII characters consist of one byte. Accented characters are used in various languages such as German. An example of such a character is the umlaut (or diaeresis), which consists of two dots placed over a letter, as in ä. The importance of this conversion feature is that it allows PSTN entities that do not support accented characters, to receive ASCII characters. For example, the device can convert the Unicode character ä into the ASCII character "ae".

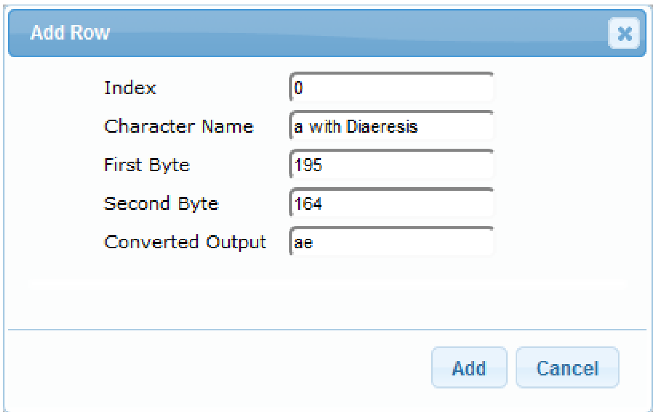
> **Note:** The table works in conjunction with the ISO8859CharacterSet parameter. When the parameter is set to [0] (Latin only), it converts accented characters into ASCII (e.g., ä to "a"). However, the table can be used to overwrite these "basic" conversions and customize them (e.g., ä to "ae" instead of the default "a").

The following procedure describes how to configure Character Conversion rules through the Web interface. You can also configure it through ini file (CharConversion) or CLI (configure voip > gw dtmf-and-suppl dtmf-and-dialing > char-conversion).

### ➢ To configure a Character Conversion rule:

1. Open the Char Conversion table (**Configuration** tab > **VoIP** menu > **Gateway** > **DTMF & Supplementary** > **Char Conversion**).
2. Click **Add**; the following dialog box appears:

**Figure 25-7: Char Conversion Table - Add Row Dialog Box**




The figure above shows a configuration example where ä is converted to ae.

3. Configure a Character Conversion rule according to the parameters described in the table below.
4. Click **Add**.

**Table 25-4: Char Conversion Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[CharConversion_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Character Name<br>`char-name` | Defines an arbitrary name to easily identify the row.<br>The valid value is a string of up to 40 characters. |

| Parameter | Description |
|---|---|
| [CharConversion_CharName] | **Note:** Each row must be configured with a unique name. |
| First Byte<br>`first-byte`<br>[CharConversion_FirstByte] | Defines the first byte of the Unicode character (e.g., 195).<br>The default is 194. |
| Second Byte<br>`second-byte`<br>[CharConversion_SecondByte] | Defines the second byte of the Unicode character (e.g., 164).<br>The default is 128. |
| Converted Output<br>`converted-output`<br>[CharConversion_ConvertedOutput] | Defines the ASCII character (e.g., "ae") to which the Unicode character must be converted.<br>The valid value is a string of up to four characters.<br>The valid value is up to four ASCII characters. This can include any ASCII character - alphanumerical (e.g., a, A, 6) and/or symbols (e.g., !, ?, _, &). |

**This page is intentionally left blank.**

# 26 Analog Gateway

This section describes configuration of analog settings.

## 26.1 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:**

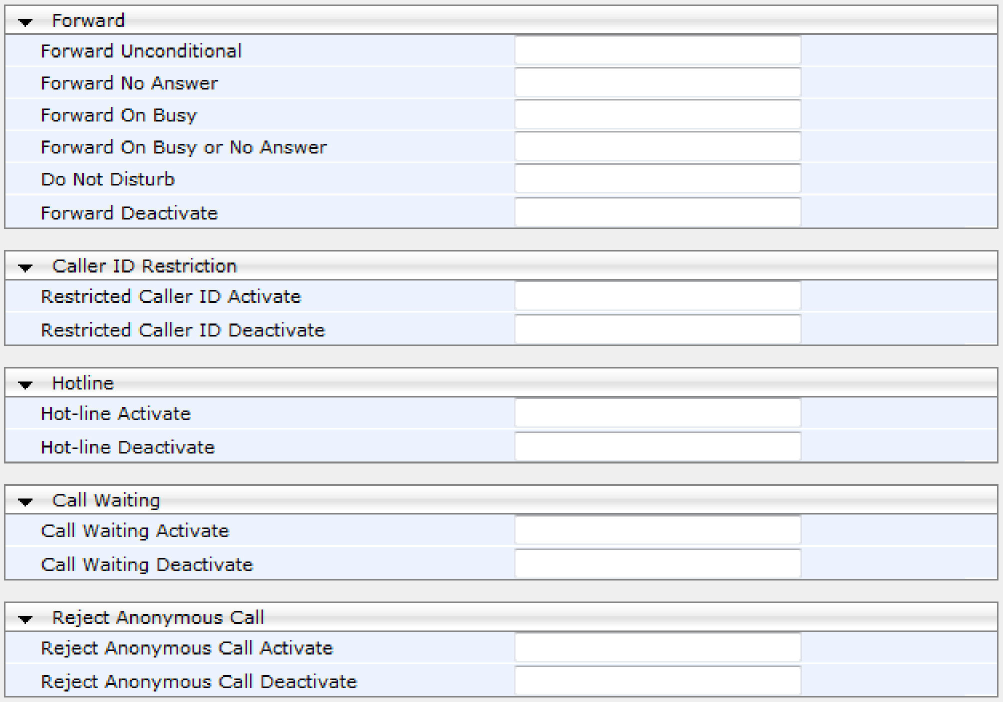
- The Keypad Features page is available only for FXS interfaces.
- The method used by the device to collect dialed numbers is identical to the method used during a regular call (i.e., max digits, interdigit timeout, digit map, etc.).
- The activation of each feature remains in effect until it is deactivated (i.e., not deactivated after a call).
- For a description of the keypad parameters, see "Telephone Keypad Sequence Parameters" on page 788.

➢ **To configure the keypad features**

1. Open the Keypad Features page (**Configuration** tab > **VoIP** menu > **Gateway** >

**Analog Gateway** > **Keypad Features**).

**Figure 26-1: Keypad Features Page**




2. Configure the keypad features as required.
3. Click **Submit**.

## 26.2 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 table (see "Configuring Tel-to-IP Routing Rules" on page 409). When a new call is established, the Tel-to-IP Routing table is searched for the destination IP address. Once a route is located, the Charge Code (configured for that route) is used to associate the route with an entry in the Charge Codes table.

➢ **To configure Metering tones:**

1. Open the Metering Tones page (**Configuration** tab > **VoIP** menu > **Gateway** > **Analog Gateway** > **Metering Tones**).

**Figure 26-2: Metering Tones Page**


2. Configure the Metering tones parameters as required. For a description of the parameters appearing on this page, see ''Configuration Parameters Reference'' on page 643.
3. Click **Submit**.
4. To save the changes to the flash memory, see ''Saving Configuration'' on page 490.

If you set the 'Generate Metering Tones' parameter to **Internal Table**, access the Charge Codes table by clicking the **Charge Codes Table** button. For more information on configuring the Charge Codes table, see ''Configuring Charge Codes'' on page 464.

## 26.3 Configuring Authentication

The Authentication table lets you configure an authentication username and password per device FXS port.

**Notes:**

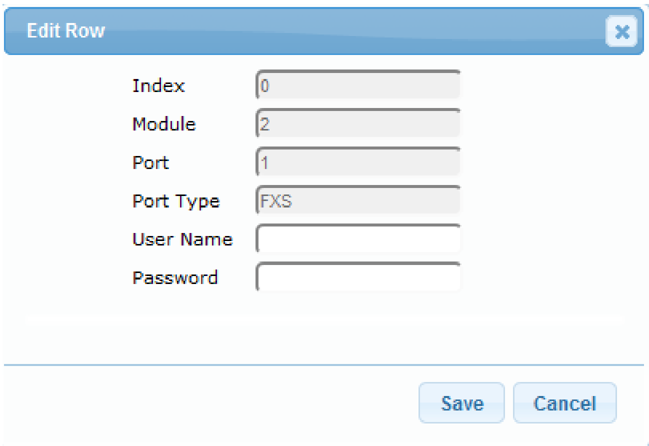
- For configuring whether authentication is done per port or for the entire device, use the parameter AuthenticationMode.
- If authentication is configured for the entire device, the configuration in this table is ignored.
- If the user name or password is not configured in this table, the port's phone number (configured in the Hunt Group table) and global password (configured by the global parameter, Password) are used instead for authentication of the port.
- After you click **Submit**, the password is displayed as an asterisk (*).

The following procedure describes how to configure authentication per port through the Web interface. You can also configure it through ini file (Authentication) or CLI (configure voip > gw analoggw authentication).

➢ **To configure authentication credentials per port:**

1. Set the parameter 'Registration Mode' (AuthenticationMode) to **Per Endpoint.** This can be configured in any of the following pages:
   - Proxy & Registration page (see "Configuring Proxy and Registration Parameters" on page 346).
   - Hunt Group Settings page (see "Configuring Hunt Group Settings" on page 386), where registration method is configured per Hunt Group.
2. Open the Authentication table (**Configuration** tab > **VoIP** menu > **Gateway** > **Analog Gateway** > **Authentication**).
3. Click **Add**; the following dialog box appears:

**Figure 26-3: Authentication Table - Edit Row Dialog Box**




4. Configure authentication per port according to the parameters described in the table below.
5. Click **Add**.

**Table 26-1: Authentication Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[Authentication_Index] | (Read-only) Displays the index number of the table row. |
| Module<br>`port-type`<br>[Authentication_Module] | (Read-only) Displays the module number on which the port is located. |
| Port<br>`port`<br>[Authentication_Port] | (Read-only) Displays the port number. |
| Port Type<br>[Authentication_PortType] | (Read-only) Displays the port type (FXS). |
| User Name<br>`user-name`<br>[Authentication_UserId] | Defines the user name used for authenticating the port. |
| Password<br>`password`<br>[Authentication_UserPassword] | Defines the password used for authenticating the port. |

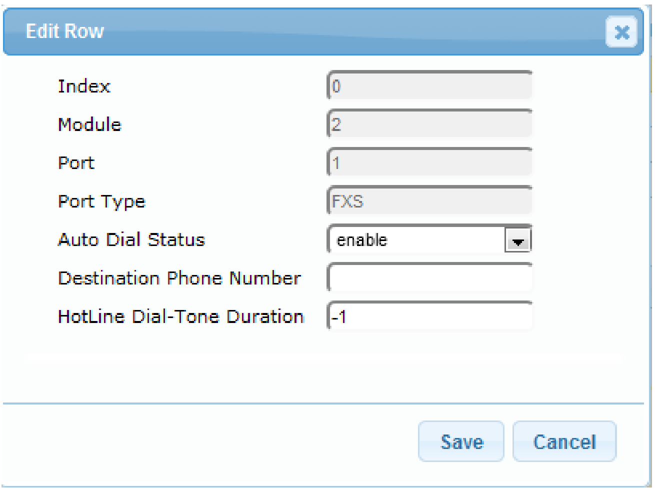
## 26.4 Configuring Automatic Dialing

The Automatic Dialing table lets you configure telephone numbers that are automatically dialed when FXS ports go off-hook. The dialing can be done immediately upon off-hook, or after a user-defined interval after off-hook referred to as *Hotline* dialing.

The following procedure describes how to configure automatic dialing upon off-hook through the Web interface. You can also configure it through ini file (TargetOfChannel) or CLI (configure voip > gw analoggw automatic-dialing).

➢ **To configure automatic dialing per port:**

1. Open the Automatic Dialing table (**Configuration** tab > **VoIP** menu > **Gateway** > **Analog Gateway** > **Automatic Dialing**).
2. Click **Add**; the following dialog box appears:




The figure above shows a configuration example where hotline-automatic dialing is enabled for an FXS port, whereby if the port is off-hooked for over 15 seconds, the device automatically dials 911.

3. Configure automatic dialing per port according to the parameters described in the table below.
4. Click **Add**.

**Table 26-2: Automatic Dialing Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[TargetOfChannel_Index] | (Read-only) Displays the index number of the table row. |
| Module<br>[TargetOfChannel_Module] | (Read-only) Displays the module number on which the port is located. |
| Port<br>`port`<br>[TargetOfChannel_Port] | (Read-only) Displays the port number. |
| Port Type<br>[TargetOfChannel_PortType] | (Read-only) Displays the port type (FXS). |
| Destination Phone Number<br>`dst-number`<br>[TargetOfChannel_Destination] | Defines the destination telephone number to automatically dial. |

| Parameter | Description |
|---|---|
| Auto Dial Status<br>`auto-dial-status`<br>[TargetOfChannel_Type] | Enables automatic dialing.<br>▪ **[0]** Disable = Automatic dialing for the specific port is disabled.<br>▪ **[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:<br>✓ FXS interfaces: The phone is off-hooked<br>▪ **[2]** Hotline = Automatic dialing is done after an interval configured by the 'Hotline Dial Tone Duration' parameter:<br>✓ 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. |
| Hotline Dial Tone Duration<br>`hotline-dia-ltone-duration`<br>[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**).<br>The valid value is 0 to 60. The default is 16.<br>**Note:** You can configure this Hotline interval for all ports, using the global parameter, HotLineToneDuration. |

## 26.5 Configuring Caller Display Information

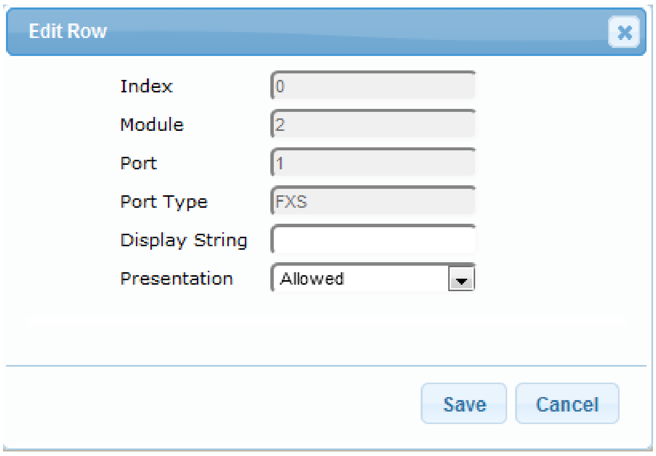
The Caller Display Information table lets you configure caller identification strings (Caller ID) per FXS port. this table also lets you enable the device to send the Caller ID to the IP when a call is made. The called party can use this information for caller identification. The device sends the configured caller ID in the outgoing INVITE message's From header. For information on Caller ID restriction according to destination/source prefixes, see "Configuring Source/Destination Number Manipulation" on page 393.

> **Note:** 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.

The following procedure describes how to configure caller ID through the Web interface. You can also configure it through ini file (CallerDisplayInfo) or CLI (configure voip > gw analoggw caller-display-info).

➢ **To configure Caller display:**

1. Open the Caller Display Information table (**Configuration** tab > **VoIP** menu > **Gateway** > **Analog Gateway** > **Caller Display Information**).
2. Click **Add**; the following dialog box appears:




3. Configure caller display per port according to the parameters described in the table below.
4. Click **Add**.

**Table 26-3: Caller Display Information Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[CallerDisplayInfo_Index] | (Read-only) Displays the index number of the table row. |
| Module<br>[CallerDisplayInfo_Module] | (Read-only) Displays the module number on which the port is located. |
| Port<br>[CallerDisplayInfo_Port] | (Read-only) Displays the port number. |
| Port Type<br>[CallerDisplayInfo_PortType] | (Read-only) Displays the port type (FXS). |

| Parameter | Description |
|---|---|
| Display String<br>`display-string`<br>[CallerDisplayInfo_DisplayString] | Defines the Caller ID string.<br>The valid value is a string of up to 18 characters.<br>**Note:** If you set the parameter to "Private" or "Anonymous", Caller ID is restricted and the settings of the 'Presentation' parameter is ignored. |
| Presentation<br>`presentation`<br>[CallerDisplayInfo_IsCidRestricted] | Enables the sending of the caller ID string.<br>▪ **[0]** Allowed = The caller ID string is sent when a Tel-to-IP call is made.<br>▪ **[1]** Restricted = The caller ID string is not sent. The Caller ID is sent to the remote side using only the SIP P-Asserted-Identity or P-Preferred-Identity headers, according to the AssertedIdMode parameter.<br>**Note:** The parameter is overridden by the 'Presentation' parameter in the Source Number Manipulation table (see "Configuring Source/Destination Number Manipulation" on page 393). |

## 26.6 Configuring Call Forward

The Call Forward table lets you configure call forwarding per FXS port, for IP-to-Tel calls. This redirects the call, using a SIP 302 response, initially destined to a specific port, to a different port on the device or to an IP destination.

**Note:** To enable call forwarding, set the 'Enable Call Forward' parameter to **Enable**. This is done in the Supplementary Services page (**Configuration** tab > **VoIP** menu > **Gateway** > **DTMF and Supplementary** > **Supplementary Services**).

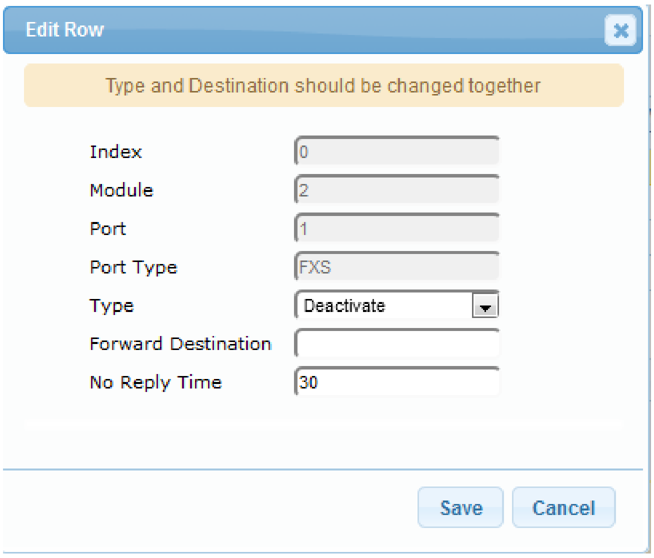
The following procedure describes how to configure call forwarding per port through the Web interface. You can also configure it through ini file (FwdInfo) or CLI (configure voip > gw analoggw call-forward).

➢ **To configure call forward per port:**

1. Open the Call Forward table (**Configuration** tab > **VoIP** menu > **Gateway** > **Analog Gateway** > **Call Forward**).

2. Click **Add**; the following dialog box appears:

**Figure 26-4: Call Forward Table - Edit Row Dialog Box**




3. Configure call forwarding per port according to the parameters described in the table below.
4. Click **Add**.

**Table 26-4: Call Forward Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[FwdInfo_Index] | (Read-only) Displays the index number of the table row. |
| Module<br>[FwdInfo_Module] | (Read-only) Displays the module number on which the port is located. |
| Port<br>[FwdInfo_Port] | (Read-only) Displays the port number. |
| Port Type<br>[FwdInfo_PortType] | (Read-only) Displays the port type (FXS). |
| Type<br>`type`<br>[FwdInfo_Type] | Defines the condition upon which the call is forwarded.<br>▪ **[0]** Deactivate = (Default) Don't forward incoming calls.<br>▪ **[1]** On Busy = Forward incoming calls when the port is busy.<br>▪ **[2]** Unconditional = Always forward incoming calls.<br>▪ **[3]** No Answer = Forward incoming calls that are not answered within the time specified in the 'No Reply Time' field.<br>▪ **[4]** On Busy or No Answer = Forward incoming calls when the port is busy or when calls are not answered within the time specified in the 'No Reply Time' field.<br>▪ **[5]** Don't Disturb = Immediately reject incoming calls. |
| Forward Destination<br>`destination`<br>[FwdInfo_Destination] | Defines the telephone number or URI (<number>@<IP address>) to where the call is forwarded.<br>**Note:** If the 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 table (see "Configuring Tel-to-IP Routing Rules" on page 409). |

| Parameter | Description |
|---|---|
| No Reply Time<br>`no-reply-time`<br>[FwdInfo_NoReplyTime] | If you have set the 'Type' parameter for this port to **No Answer** or **On Busy or No Answer**, then configure the number of seconds the device waits before forwarding the call to the specified phone number. |

## 26.7 Configuring Caller ID Permissions

The Caller ID Permissions table lets you enable per port, Caller ID generation for FXS interfaces.

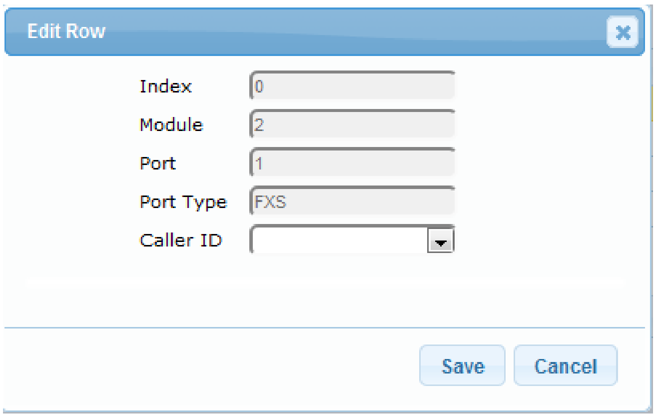
**Note:** If Caller ID permissions is not configured for a port in this table, its Caller ID generation / detection is determined according to the global parameter, 'Enable Call ID' in the Supplementary Services page (**Configuration** tab > **VoIP** menu > **Gateway** > **DTMF and Supplementary** > **Supplementary Services**).

The following procedure describes how to configure caller ID permissions through the Web interface. You can also configure it through ini file (EnableCallerID) or CLI (configure voip > gw analoggw enable-caller-id).

➢ **To configure caller ID permissions per port:**

1. Open the Caller ID Permissions table (**Configuration** tab > **VoIP** menu > **Gateway** > **Analog Gateway** > **Caller ID Permissions**).
2. Click **Add**; the following dialog box appears:

**Figure 26-5: Caller ID Permissions Table - Edit Row Dialog Box**




3. Configure a caller ID permission per port according to the parameters described in the table below.
4. Click **Add**.

**Table 26-5: Caller ID Permissions Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[EnableCallerId_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Module | (Read-only) Displays the module number on which the port is located. |

| Parameter | Description |
|---|---|
| [EnableCallerId_Module] | |
| Port<br>[EnableCallerId_Port] | (Read-only) Displays the port number. |
| Port Type<br>[EnableCallerId_PortType] | (Read-only) Displays the port type (e.g., FXS). |
| Caller ID<br>`caller-id`<br>[EnableCallerId_IsEnabled] | Enables Caller ID generation (FXS) per port.<br>▪ **[0]** Disable<br>▪ **[1]** Enable |

## 26.8 Configuring Call Waiting

The Call Waiting table lets you enable or disable call waiting per FXS port.

**Notes:**

- You can enable or disable call waiting for all the device's ports using the global parameter, 'Enable Call Waiting' in the Supplementary Services page (**Configuration** tab > **VoIP** menu > **Gateway** > **DTMF and Supplementary** > **Supplementary Services**).
- The CPT file installed on the device must include a 'call waiting Ringback' tone (caller side) and a 'call waiting' tone (called side, FXS interfaces only).
- The EnableHold parameter must be enabled on both the calling and the called sides.
- For additional call waiting configuration, see the following parameters: FirstCallWaitingToneID (in the CPT file), TimeBeforeWaitingIndication, WaitingBeepDuration, TimeBetweenWaitingIndications, and NumberOfWaitingIndications.

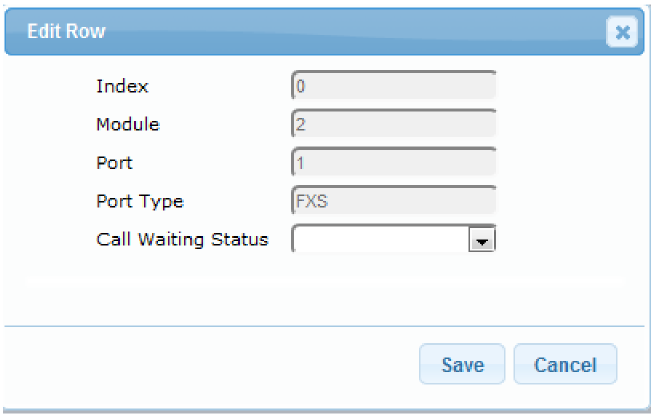
The following procedure describes how to configure call waiting per port through the Web interface. You can also configure it through ini file (CallWaitingPerPort) or CLI (configure voip > gw analoggw call-waiting).

➢ **To enable call waiting per port:**

1. Open the Call Waiting table (**Configuration** tab > **VoIP** menu > **Gateway** > **Analog Gateway** > **Call Waiting**).

2. Click **Add**; the following dialog box appears:

**Figure 26-6: Call Waiting Table - Edit Row Dialog Box**




3. Configure call waiting per port according to the parameters described in the table below.
4. Click **Add**.

**Table 26-6: Call Waiting Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[CallWaitingPerPort_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| Module<br>[CallWaitingPerPort_Module] | (Read-only) Displays the module number on which the port is located. |
| Port<br>[CallWaitingPerPort_Port] | (Read-only) Displays the port number. |
| Port Type<br>[CallWaitingPerPort_PortType] | (Read-only) Displays the port type (e.g., FXS). |
| Call Waiting Configuration<br>`enable-call-waiting`<br>[CallWaitingPerPort_IsEnabled] | Enables call waiting for the port.<br>▪ **[0]** Disable<br>▪ **[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. |

# 26.9 Rejecting Anonymous Calls

You can configure the device to reject anonymous calls received from the IP and destined for an FXS port. This can be configured using the ini file parameter, RejectAnonymousCallPerPort. If configured, when an FXS interface receives an anonymous call, the device rejects the call and responds with a SIP 433 (Anonymity Disallowed) response. For a description of the parameter see "Caller ID Parameters" on page 758.

## 26.10 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.

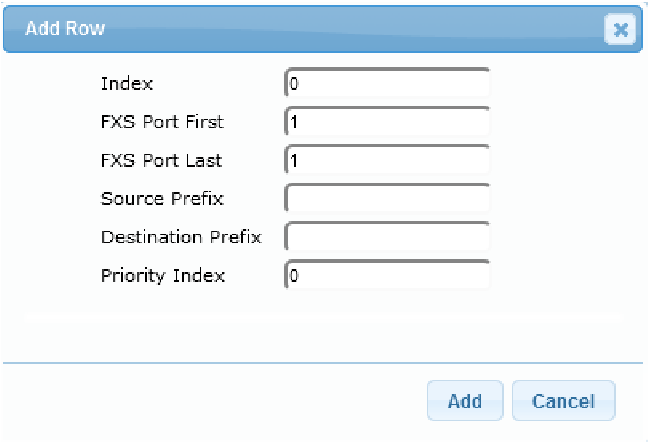
**Note:** The table is applicable only to FXS interfaces.

The following procedure describes how to configure tones per FXS through the Web interface. You can also configure it through ini file (ToneIndex) or CLI (configure voip > gw analoggw tone-index).

➢ **To configure distinctive ringing and call waiting per FXS port:**

1. Open the Tone Index table (**Configuration** tab > **VoIP** menu > **Gateway** > **Analog Gateway** > **Tone Index**).
2. Click **Add**; the following dialog box appears:

**Figure 26-7: Tone Index Table - Add Row Dialog Box**




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 **Add**.

**Table 26-7: Tone index Table Parameter Description**

| Parameter | Description |
|---|---|
| Index | Defines the table index entry. Up to 50 entries can be defined. |
| FXS Port First<br>`fxs-port-first`<br>[ToneIndex_FXSPort_First] | Defines the first port in the FXS port range. |
| FXS Port Last<br>`fxs-port-last`<br>[ToneIndex_FXSPort_Last] | Defines the last port in the FXS port range. |
| Source Prefix<br>`src-prefix`<br>[ToneIndex_SourcePrefix] | Defines the prefix of the calling number. |
| Destination Prefix<br>`dst-prefix`<br>[ToneIndex_DestinationPrefix] | Defines the prefix of the called number. |
| Priority Index<br>`priority`<br>[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.<br>The default is 0. |

## 26.11 FXS Coefficient Types

The FXS 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 = 3
- 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. The 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 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 26-8: FXS Coefficient Parameters in Analog Settings Page**


2. From the 'FXS Coefficient Type' drop-down list (FXSCountryCoefficients), select the required FXS Coefficient type.
3. Click **Submit**.
4. Save your settings to the flash memory ("burn") with a device reset.

# Part VI

## Maintenance

# 27 Basic Maintenance

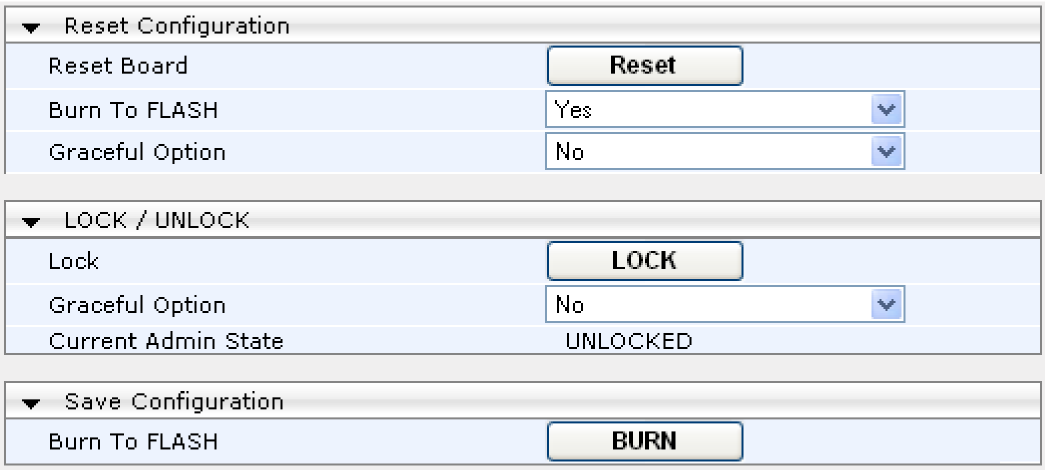
The Maintenance Actions page allows you to perform the following:

- Reset the device - see "Resetting the Device" on page 487
- Lock and unlock the device - see "Locking and Unlocking the Device" on page 489
- Save configuration to the device's flash memory - see "Saving Configuration" on page 490

**➢ 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 27-1: Maintenance Actions Page**




## 27.1 Resetting the Device

The Maintenance Actions page allows you to remotely reset the device. Before resetting the device, you can also choose the following options:

- "Burn" (save) the device's current configuration to the device's flash memory (non-volatile).
- Graceful Shutdown, whereby the device resets only after a user-defined time (i.e., timeout) or after there is no traffic currently processed by the device (the earliest thereof).

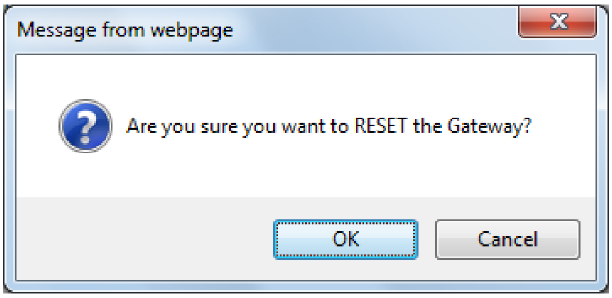
**Notes:**

- Throughout the Web interface, parameters displayed with a lightning symbol are not applied on-the-fly and require that you reset the device for them to take effect.
- When you modify parameters that require a device reset, once you click the **Submit** button in the relevant page, the toolbar displays "Reset" (see "Toolbar Description" on page 44) 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 45).

➢ **To reset the device:**

1. Open the Maintenance Actions page (see ''Basic Maintenance'' on page 487).
2. Under the 'Reset Configuration' group, from the 'Burn To FLASH' drop-down list, select one of the following options:
   - **Yes:** The device's current configuration is saved (*burned*) to the flash memory prior to reset (default).
   - **No:** Resets the device without saving the current configuration to flash (discards all unsaved modifications).
3. Under the 'Reset Configuration' group, from the 'Graceful Option' drop-down list, select one of the following options:
   - **Yes:** Reset starts only after the user-defined time in the 'Shutdown Timeout' field (see Step 4) expires or after no more active traffic exists (the earliest thereof). In addition, no new traffic is accepted.
   - **No:** Reset starts regardless of traffic, and any existing traffic is terminated at once.
4. Under the 'Reset Configuration' group, in the 'Shutdown Timeout' field (relevant only if the 'Graceful Option' in the previous step is set to **Yes**), enter the time after which the device resets. Note that if no traffic exists and the time has not yet expired, the device resets.
5. Click the **Reset** button; a confirmation message box appears, requesting you to confirm.

**Figure 27-2: 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.

## 27.2 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.

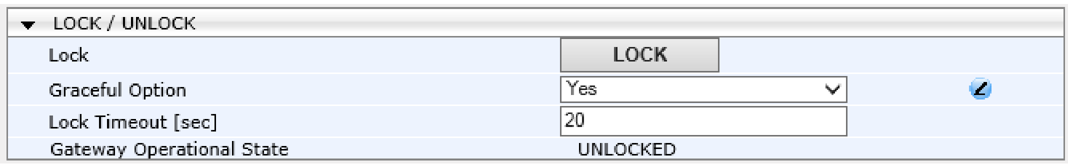
## 27.3 Locking and Unlocking the Device

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 487).
2. Scroll down to the 'LOCK / UNLOCK' group:

**Figure 27-3: Locking the Device**




3. From the 'Graceful Option' drop-down list, select one of the following options:
   - **Yes:** The device is locked only after the user-defined time in the 'Lock Timeout' field (see Step 4) expires or no more active traffic exists (the earliest thereof). In addition, no new traffic is accepted.
   - **No:** The device is locked regardless of traffic. Any existing traffic is terminated immediately.

   **Note:** These options are only available if the current status of the device is in UNLOCKED state.
4. If you set 'Graceful Option' to **Yes** (in the previous step), then in the 'Lock Timeout' field, enter the time (in seconds) after which the device locks. If no traffic exists and the time has not yet expired, the device locks immediately.

5. Click the **LOCK** button; a confirmation message box appears requesting you to confirm device lock.
6. Click **OK** to confirm device lock; if you set 'Graceful Option' to **Yes**, a lock icon is delayed and a window appears displaying the number of remaining calls and time. If you set 'Graceful Option' to **No**, the lock process begins immediately. The 'Gateway Operational State' field displays "LOCKED".

➢ **To unlock the device:**

- Under the 'LOCK / UNLOCK' group, click the **UNLOCK** button. Unlock starts immediately and the device accepts new incoming calls. The 'Gateway Operational State' field displays "UNLOCKED".

**Note:** The Home page's General Information pane displays whether the device is locked or unlocked (see "Viewing the Home Page" on page 63).

## 27.4 Saving Configuration

The Maintenance Actions page allows you to save (*burn*) the current parameter configuration (including loaded Auxiliary files) to the device's *non-volatile* memory (i.e., flash). The parameter modifications that you make throughout the Web interface's pages are temporarily saved (to the *volatile* memory - RAM) when you click the **Submit** or **Add** buttons on these pages. Parameter settings that are saved only to the device's RAM revert to their previous settings after a hardware/software reset (or power failure). Therefore, to ensure that your configuration changes are retained, you must save them to the device's flash memory using the burn option described below.

➢ **To save the changes to the non-volatile flash memory:**

1. Open the Maintenance Actions page (see "Basic Maintenance" on page 487).
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 489).
- 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 487).
- The Home page's General Information pane displays whether the device is currently "burning" the configuration (see "Viewing the Home Page" on page 63).

# 28 Channel Maintenance

This chapter describes various channel-related maintenance procedures.

## 28.1 Disconnecting Active Calls

You can forcibly disconnect all active (established) calls or disconnect specific calls based on their Session ID. This is done in the CLI using the following commands (from basic command mode):

- Disconnects all active calls:

```
# clear voip calls
```

- Disconnects active calls belonging to a specified Session ID:

```
# clear voip calls <Session ID>
```

## 28.2 Resetting an Analog Channel

You can inactivate (*reset*) an FXS analog channel through the Web interface, as described in Section 37.1 on page 557.

## 28.3 Disabling Analog Ports

You can disable an analog port (FXS) on the device. When disabled, the port cannot be used and no signaling is transmitted through the port. In addition, the LED of the analog port is lit red. By default, all the analog ports are enabled.

To disable an analog port, use the following CLI command:

```
(config-voip)# interface fxs-fxo
(fxs-fxo)# analog-port-enable <module>/<port> [on|off]
```

For example, to disable port 2 on module 1:

```
(fxs-fxo)# analog-port-enable 1/2 off
```

## 28.4 Locking and Unlocking Hunt Groups

You can lock a Hunt Group to take its channels out of service. When you initiate the lock process, the device rejects all new incoming calls for the Hunt Group and immediately terminates active calls (busy channels), eventually taking the entire Hunt Group out of service.

When you lock a Hunt Group, the method for taking channels out-of-service is determined by the FXSOOSBehavior parameter.

If you have configured registration for the Hunt Group (see the 'Registration Mode' parameter in the Hunt Group Settings table) and you lock the Hunt Group, it stops performing registration requests with the Serving IP Group with which you have configured it to register. When you unlock such a Hunt Group, it starts performing registration requests with the Serving IP Group once its channels return to service.

➢ **To lock or unlock a Hunt Group:**

1. Open the Hunt Group Settings table (**Configuration** tab > **VoIP** menu > **Gateway** > **Hunt Group** > **Hunt Group Settings**).
2. Select the table row of a Hunt Group that you want to lock or unlock.

3. From the **Action** drop-down list located on the table's toolbar, choose one of the following commands:
   - **Lock:** Locks the Hunt Group.
   - **Unlock**: Unlocks a locked Hunt Group.

The Hunt Group Settings table provides the following read-only fields related to locking and unlocking of a Hunt Group:

- 'Admin State': Displays the administrators state - "Locked" or "Unlocked"
- 'Status': Displays the current status of the channels in the Hunt Group:
  - "In Service": Indicates that all channels in the Hunt Group are in service, for example, when the Hunt Group is unlocked or Busy Out state cleared (see the EnableBusyOut parameter for more information).
  - "Going Out Of Service": Appears as soon as you choose the **Lock** button and indicates that the device is starting to lock the Hunt Group and take channels out of service.
  - "Going Out Of Service (<duration remaining of graceful period> sec / <number of calls still active> calls)": Appears when the device is locking the Hunt Group and indicates the number of buys channels and the time remaining until the graceful period ends, after which the device locks the channels regardless of whether the call has ended or not.
  - "Out Of Service": All fully configured channels in the Hunt Group are out of service, for example, when the Hunt Group is locked or in Busy Out state (see the EnableBusyOut parameter).

> **Note:** If the device is reset, a locked Hunt Group remains locked. If the device is reset while graceful lock is in progress, the Hunt Group is forced to lock immediately after the device finishes its reset.

# 29 Software Upgrade

This chapter describes various software update procedures.

## 29.1 Auxiliary Files

You can install various Auxiliary files on the device. Auxiliary files provide the device with additional configuration settings. The table below lists the different types of Auxiliary files.

**Table 29-1: Auxiliary Files**

| File | Description |
|---|---|
| INI | Configures the device. The Web interface enables practically full device provisioning. However, some features may only be configured by ini file or you may wish to configure your device using the ini file. For more information on the ini file, see "INI File-Based Management" on page 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 496. |
| 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 500. |
| 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 501. |
| 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 504. |
| AMD Sensitivity | Answer Machine Detector (AMD) Sensitivity file containing the AMD Sensitivity suites. For more information, see AMD Sensitivity File on page 509. |

### 29.1.1 Loading Auxiliary Files

You can load Auxiliary files to the device using one of the following methods:

- Web interface - see "Loading Auxiliary Files through Web Interface" on page 494
- CLI - see Loading Auxiliary Files through CLI on page 495
- TFTP - see "Loading Auxiliary Files through ini File using TFTP" on page 495

**Notes:**

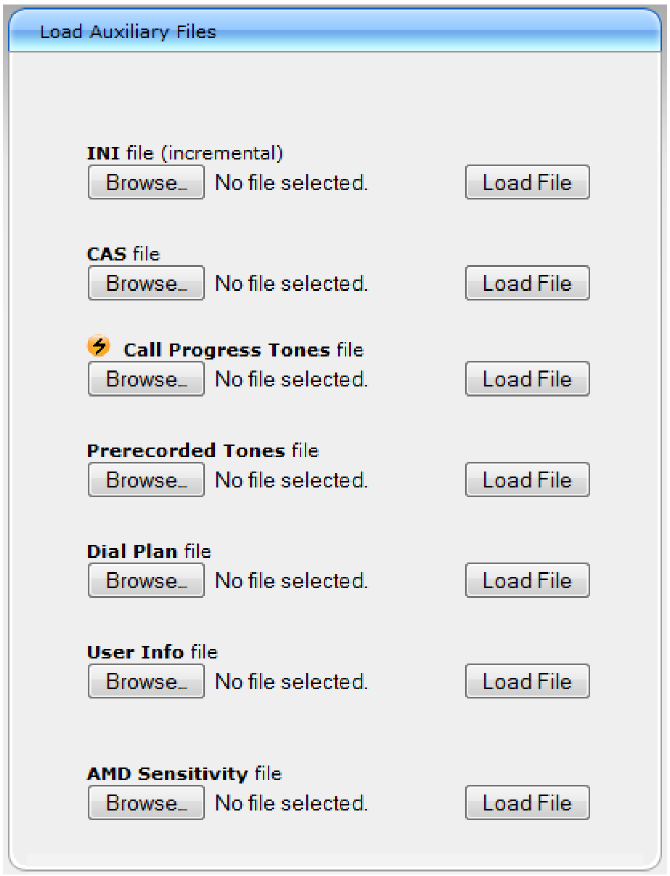
- You can schedule automatic loading of updated Auxiliary files using HTTP/HTTPS. For more information, see Automatic Update Mechanism.
- Saving Auxiliary files to flash memory may disrupt traffic on the device. To avoid this, disable all traffic on the device by performing a graceful lock as described in "Locking and Unlocking the Device" on page 489.
- To delete installed Auxiliary files, see "Viewing Device Information" on page 545.

#### 29.1.1.1 Loading Auxiliary Files through Web Interface

The following procedure describes how to load Auxiliary files through the Web interface.

➢ **To load Auxiliary files through the Web interface:**

1. Open the Load Auxiliary Files page (**Maintenance** tab > **Software Update** menu > **Load Auxiliary Files**).




> **Note:** The appearance of certain file load fields depends on the installed Software License Key.

2. Click the **Browse** button corresponding to the Auxiliary file type that you want to load, navigate to the folder in which the file is located, and then click **Open**; the name of the file appears in the field next to the **Browse** button.
3. Click the **Load File** button corresponding to the file you want to load.
4. Repeat steps 2 through 3 for each file you want to load.
5. Reset the device with a burn-to-flash for your settings to take effect (if you have loaded a Call Progress Tones file).

> **Note:** When loading an *ini* file using the Web interface, Auxiliary files that are already installed on the device are maintained if the ini file does not contain these Auxiliary files.

#### 29.1.1.2 Loading Auxiliary Files through CLI

You can load Auxiliary files from user-defined URLs, using the following CLI commands:

- **Single Auxiliary file:**

  ```
  # copy <file> from <URL>
  ```

  For example:

  ```
  # copy call_progress_tones from
  http://192.169.11.11:80/cpt_us.dat
  ```

- **Multiple (batch) Auxiliary files:** The Auxiliary files must be contained in a TAR (Tape ARchive) file (.tar). The TAR file can contain any number and type of Auxiliary files (e.g., Dial Plan file and CPT file).

  ```
  # copy aux-package from | to <URL with TAR file name>
  ```

  For example:

  ```
  # copy aux-package from http://192.169.11.11:80/aux_files.tar
  ```

For more information on CLI commands, refer to the *CLI Reference Guide*.

#### 29.1.1.3 Loading Auxiliary Files through ini File using TFTP

You can load Auxiliary files to the device through the ini file, using a TFTP server. For more information on Auxiliary ini file parameters, see "Auxiliary and Configuration File Name Parameters" on page 654.

➢ **To load Auxiliary files through ini file:**

1. Create an ini file that includes the names of the Auxiliary files that you want loaded, for example:

   ```
   CallProgressTonesFilename = 'usa_tones_13.dat'
   DialPlanFileName = 'dial-plan-us.dat'
   ```

2. Save the ini file and the Auxiliary files in the same folder on your TFTP server.
3. Reset the device (you can power off and then power on the device); the device loads the ini file and then the Auxiliary files as defined in the ini file, through TFTP.

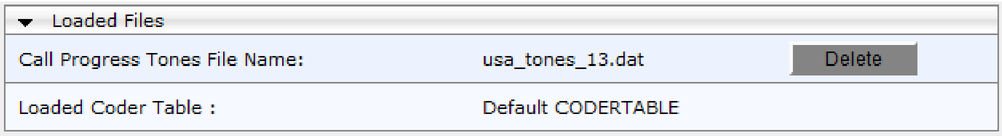
### 29.1.2 Deleting Auxiliary Files

You can delete loaded Auxiliary files through the Web interface, as described below.

➢ **To delete a loaded file:**

1. Open the Device Information page (**Status & Diagnostics** tab > **System Status** menu > **Device Information**); loaded files are listed under the Loaded Files group, as shown in the example below:

**Figure 29-1: Loaded Files Listed on Device Information Page**




2. Click the **Delete** button corresponding to the file that you want to delete; a confirmation message box appears.
3. Click **OK** to confirm deletion.
4. Reset the device with a burn-to-flash for your settings to take effect.

### 29.1.3 Call Progress Tones File

The Call Progress Tones (CPT) and Distinctive Ringing (for analog interfaces only) Auxiliary file includes the definitions of the CPT (levels and frequencies) that are detected / generated by the device.

The CPT for analog interfaces is comprised of two sections:

- The first section contains the definitions of the Call Progress Tones (levels and frequencies) that are detected/generated by the device.
- The second section contains the characteristics of the Distinctive Ringing signals that are generated by the device (see Distinctive Ringing on page 498).

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, the parameter defines the detection period. For burst tones, it defines the tone's duration.
- **First Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the first cadence on-off cycle (for cadence tones). For burst tones, the parameter defines the off time required after the burst tone ends and the tone detection is reported. For continuous tones, the parameter is ignored.
- **Second Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.
- **Second Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.
- **Third Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the third cadence on-off cycle. Can be omitted if there isn't a third cadence.
- **Third Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the third cadence on-off cycle. Can be omitted if there isn't a third cadence.
- **Fourth Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the fourth cadence on-off cycle. Can be omitted if there isn't a fourth cadence.
- **Fourth Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the fourth cadence on-off cycle. Can be omitted if there isn't a fourth cadence.
- **Carrier Freq [Hz]:** Frequency of the carrier signal for AM tones.
- **Modulation Freq [Hz]:** Frequency of the modulated signal for AM tones (valid range from 1 to 128 Hz).

- **Signal Level [-dBm]:** Level of the tone for AM tones.
- **AM Factor [steps of 0.02]:** Amplitude modulation factor (valid range from 1 to 50). Recommended values from 10 to 25.

**Notes:**

- When the same frequency is used for a continuous tone and a cadence tone, the 'Signal On Time' parameter of the continuous tone must have a value that is greater than the 'Signal On Time' parameter of the cadence tone. Otherwise, the continuous tone is detected instead of the cadence tone.
- The tones frequency must differ by at least 40 Hz between defined tones.

For example, to configure the dial tone to 440 Hz only, enter the following text:

```
[NUMBER OF CALL PROGRESS TONES]
Number of Call Progress Tones=1
#Dial Tone
[CALL PROGRESS TONE #0]
Tone Type=1
Tone Form =1 (continuous)
Low Freq [Hz]=440
High Freq [Hz]=0
Low Freq Level [-dBm]=10 (-10 dBm)
High Freq Level [-dBm]=32 (use 32 only if a single tone is
required)
First Signal On Time [10msec]=300; the dial tone is detected after
3 sec
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=0
Second Signal Off Time [10msec]=0
```

### 29.1.3.1 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
```

### 29.1.4 Prerecorded Tones File

The Prerecorded Tone (PRT) is a .dat file containing a set of prerecorded tones that can be played by the device. For example, it can be used to play music on hold (MoH) to a call party that has been put on hold. Up to 40 tones (totaling approximately 10 minutes) can be stored in a single PRT file on the device's flash memory.

The PRT file overcomes the limitations of the CPT file such as limited number of predefined tones and limited number of frequency integrations in one tone. If a specific prerecorded tone exists in the PRT file, it overrides the same tone that exists in the CPT file, and is played instead.

**Notes:**

- The PRT file only generates (plays) tones; detection of tones is according to the CPT file.
- The device does not require DSPs for playing tones from a PRT file if the coder defined for the tone is the same as that used by the current call. If the coders are different, the device uses DSPs.
- The device requires DSPs for local generation of tones.

The prerecorded tones can be created using standard third-party, recording utilities such as Adobe Audition, and then combined into a single file (PRT file) using AudioCodes DConvert utility (refer to the document, *DConvert Utility User's Guide* for more information).

The raw data files must be recorded with the following characteristics:

- Coders: G.711 A-law or G.711 µ-law (and other coders)
- Rate: 8 kHz
- Resolution: 8-bit
- Channels: mono

The prerecorded tones are played repeatedly. This allows you to record only part of the tone and then play the tone for the full duration. For example, if a tone has a cadence of 2 seconds on and 4 seconds off, the recorded file should contain only these 6 seconds. The device repeatedly plays this cadence for the configured duration. Similarly, a continuous tone can be played by repeating only part of it.

Once created, you need to install the PRT file on the device. This can be done using the Web interface (see "Loading Auxiliary Files" on page 493).

### 29.1.5 Dial Plan File

The Dial Plan file can be used for various digit mapping features, as described in this section.

#### 29.1.5.1 Creating a Dial Plan File

The Dial Plan file is a text-based file that can contain up to 8 Dial Plans (Dial Plan indices) and up to 8,000 rules (lines). The general syntax rules for the Dial Plan file are as follows (syntax specific to the feature is described in the respective section):

- Each Dial Plan index must begin with a Dial Plan name enclosed in square brackets "[...]" on a new line.
- Each line under the Dial Plan index defines a rule.
- Empty lines are ignored.
- Lines beginning with a semicolon ";" are ignored. The semicolon can be used for comments.

Creating a Dial Plan file is similar for all Dial Plan features. The main difference is the syntax used in the Dial Plan file and the method for selecting the Dial Plan index.

➢ **To create a Dial Plan file:**

1. Create a new file using a text-based editor (such as Notepad) and configure your Dial Plans as required.
2. Save the file with the *ini* file extension name (e.g., mydialplanfile.ini).
3. Convert the *ini* file to a *dat* binary file, using AudioCodes DConvert utility. For more information, refer to *DConvert Utility User's Guide*.
4. Load the converted file to the device, as described in ''Loading Auxiliary Files'' on page 493.
5. Select the Dial Plan index that you want to use. This depends on the feature and is described in the respective section.

#### 29.1.5.2 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 > **Gateway** > **DTMF and Supplementary** > **DTMF & Dialing**).

The Dial Plan is used for the following:

- FXS 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 8 Dial Plans (Dial Plan indices), with a total of up to 8,000 dialing rules (lines) of distinct prefixes (e.g. area codes, international telephone number patterns) for the PSTN to which the device is connected.

The Dial Plan file is created in a textual *ini* file with the following syntax:

```
<called number prefix>,<total digits to wait before sending>
```

- Each new Dial Plan index begins with a Dial Plan name enclosed in square brackets "[...]" on a new line.
- Each line under the Dial Plan index defines a dialing prefix and the number of digits

expected to follow that prefix. The prefix is separated by a comma "," from the number of additional digits.

- The prefix can include numerical ranges in the format [x-y], as well as multiple numerical ranges [n-m][x-y] (no comma between them).
- The prefix can include the asterisk "*" and number "#" signs.
- The number of additional digits can include a numerical range in the format x-y.
- Empty lines are ignored.
- Lines beginning with a semicolon ";" are ignored. The semicolon can be used for comments.

Below shows an example of a Dial Plan file (in *ini*-file format), containing two dial plans:

```
; Example of dial-plan configuration.
; This file contains two dial plans:
[ PLAN1 ]
; Destination cellular area codes 052, 054, and 050 with 8 digits.

052,8
054,8
050,8
; Defines International prefixes 00, 012, 014.
; The number following these prefixes may
; be 7 to 14 digits in length.
00,7-14
012,7-14
014,7-14
; Defines emergency number 911. No additional digits are expected.
911,0
[ PLAN2 ]
; Defines area codes 02, 03, 04.
; In these area codes, phone numbers have 7 digits.
0[2-4],7
; Operator services starting with a star: *41, *42, *43.
; No additional digits are expected.
*4[1-3],0
```

The following procedure provides a summary on how to create a Dial Plan file and select the required Dial Plan index.

### ➢ To create a Dial Plan file:

1. Create a new file using a text-based editor (such as Notepad) and configure your Dial Plans, as required.
2. Save the file with the *ini* file extension name (e.g., mydialplans.ini).
3. Convert the *ini* file to a *dat* binary file, using AudioCodes DConvert utility. For more information, refer to *DConvert Utility User's Guide*.
4. Install the converted file on the device, as described in "Loading Auxiliary Files" on page 493.
5. The required Dial Plan is selected using the 'Dial Plan Index' parameter. The parameter can be set to **0** through **7**, where **0** denotes PLAN1, **1** denotes PLAN2, and so on.

**Notes:**

- The Dial Plan file must not contain overlapping prefixes. Attempting to process an overlapping configuration by the DConvert utility results in an error message specifying the problematic line.
- The Dial Plan index can be selected globally for all calls (as described in the previous procedure), or per specific calls using Tel Profiles.
- It may be useful to configure both Dial Plan file and Digit Maps. For example, the Digit Map can be used for complex digit patterns (which are not supported by the Dial Plan file) and the Dial Plan can be used for long lists of relatively simple digit patterns. In addition, as timeout between digits is not supported by the Dial Plan, the Digit Map can be used to configure digit patterns that are shorter than those defined in the Dial Plan or left at default (MaxDigits parameter). For example, the "xx.T" digit map instructs the device to use the Dial Plan and if no matching digit pattern is found, it waits for two more digits and then after a timeout (TimeBetweenDigits parameter), it sends the collected digits. Therefore, this ensures that calls are not rejected as a result of their digit pattern not been completed in the Dial Plan.
- By default, if no matching digit pattern is found in both the Dial Plan and Digit Map, the device rejects the call. However, if you set the DisableStrictDialPlan parameter to 1, the device attempts to complete the call using the MaxDigits and TimeBetweenDigits parameters. In such a setup, it collects the number of digits configured by the MaxDigits parameters. If more digits are received, it ignores the settings of the parameter and collects the digits until the inter-digit timeout configured by the TimeBetweenDigits parameter is exceeded.

### 29.1.5.3 Dial Plan Prefix Tags for Routing

### 29.1.5.4 Obtaining IP Destination from Dial Plan File

You can use a Dial Plan index listed in a loaded Dial Plan file for determining the IP destination of Tel-to-IP calls. This enables the mapping of called numbers to IP addresses (in dotted-decimal notation) or FQDNs (up to 15 characters).

➢ **To configure routing to an IP destination based on Dial Plan:**

1. Create the Dial Plan file. The syntax of the Dial Plan index for this feature is as follows:

   ```
   <destination / called prefix number>,0,<IP destination>
   ```

   **Note:** The second parameter "0" is not used and ignored.

   An example of a configured Dial Plan (# 6) in the Dial Plan file is shown below:

   ```
   [ PLAN6 ]
   200,0,10.33.8.52    ; called prefix 200 is routed to
   10.33.8.52
   201,0,10.33.8.52
   300,0,itsp.com      ; called prefix 300 is routed to itsp.com
   ```

2. Convert the file to a loadable file and then load it to the device (see "Creating a Dial Plan File" on page 501).
3. Assign the Dial Plan index to the required routing rule:
   - Tel-to-IP Calls (Gateway application): In the Tel-to-IP Routing table, do the following:
     a. In the 'Destination Address' field, enter the required Dial Plan index using the following syntax:

DialPlan<index>

Where "DialPlan0" denotes [PLAN1] in the Dial Plan file, "DialPlan1" denotes [PLAN2], and so on.

**Note:** The "DialPlan" string is case-sensitive.

#### 29.1.5.5 Viewing Information of Installed Dial Plan File

You can view information about the Dial Plan file currently installed on the device, through the device's CLI:

- **Viewing Dial Plan file information:** You can view the file name of the installed Dial Plan file and the names of the Dial Plans defined in the Dial Plan file, by entering the following CLI command (in Enable mode):

```
# debug auxilary-files dial-plan info
```

  For example, the following shows the file name of the installed Dial Plan file and lists its Dial Plans:

```
# debug auxilary-files dial-plan info
  File Name: MyDialPlan.txt
  Plans:
  Plan #0 = PLAN1
  Plan #1 = PLAN2
```

  Note that the index number of the first Dial Plan is 0.

- **Searching a prefix number:** You can check whether a specific prefix number is defined in a specific Dial Plan (and view the corresponding tag if the Dial Plan implements tags), by entering the following CLI command (in Enable mode):

```
# debug auxilary-files dial-plan match-number <Dial Plan
number> <prefix number>
```

  For example, the following checks whether the called prefix number 2000 is defined in Dial Plan 1, which is used for obtaining the destination IP address (tag):

```
# debug auxilary-files dial-plan match-number PLAN1 2000
  Match found for 4 digits
  Matched prefix: 2000
  Tag: 10.33.45.92
```

### 29.1.6 User Information File

This section describes the User Info table and how to configure the table.

#### 29.1.6.1 Enabling the User Info Table

Before you can use the User Info table, you need to enable the User Info functionality as described in the following procedure.

➢ **To enable the User Info table:**

1. Open the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**).
2. Set the 'Enable User-Information Usage' parameter (EnableUserInfoUsage) to **Enable**.

3. Save this setting to the device with a reset for the setting to take effect.

#### 29.1.6.2 Gateway User Information for PBX Extensions and "Global" Numbers

The GW User Info table contains user information that can be used for the following Gateway-related features:

- **Mapping (Manipulating) PBX Extension Numbers with Global Phone Numbers:** maps PBX extension number, connected to the device, with any "global" phone number (alphanumerical) for the IP side. In this context, the "global" phone number serves as a routing identifier for calls in the "IP world" and the PBX extension uses this mapping to emulate the behavior of an IP phone. This feature is especially useful in scenarios where unique or non-consecutive number translation per PBX is needed. This number manipulation feature supports the following call directions:
  - IP-to-Tel Calls: Maps the called "global" number (in the Request-URI user part) to the PBX extension number. For example, if the device receives an IP call destined for "global" number 638002, it changes this called number to the PBX extension number 402, and then sends the call to the PBX extension on the Tel side.

> **Note:** If you have configured regular IP-to-Tel manipulation rules (see "Configuring Source/Destination Number Manipulation" on page 393), 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 393), the device applies these rules before applying the mapping rules of the User Info table.

- **Registering Users:** The device can register each PBX user configured in the User Info table. For each user, the device sends a SIP REGISTER to an external IP-based Registrar server, using the "global" number in the From/To headers. If authentication is necessary for registration, the device sends the user's username and password, configured in the User Info table, in the SIP MD5 Authorization header.

You can configure up to 500 mapping rules in the GW User Info table. These rules can be configured using any of the following methods:

- Web interface - see "Configuring GW User Info Table through Web Interface" on page 506
- CLI - see Configuring GW User Info Table through CLI on page 507
- Loadable User Info file - see "Configuring GW User Info Table in Loadable Text File" on page 508

**Notes:**

- To enable user registration, set the following parameters on the Proxy & Registration page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Proxy & Registration**) as shown:
  - √ 'Enable Registration': **Enable** (IsRegisterNeeded is set to 1).
  - √ 'Registration Mode': **Per Endpoint** (AuthenticationMode is set to 0).
- For FXS ports, when the device needs to send a new SIP request with the Authorization header (e.g., after receiving a SIP 401 response), it uses the username and password configured in the Authentication table (see Configuring Authentication per Port on page 472). To use the username and password configured in the User Info file, set the 'Password' parameter to any value other than its default value.

#### 29.1.6.2.1 Configuring GW User Info Table through Web Interface

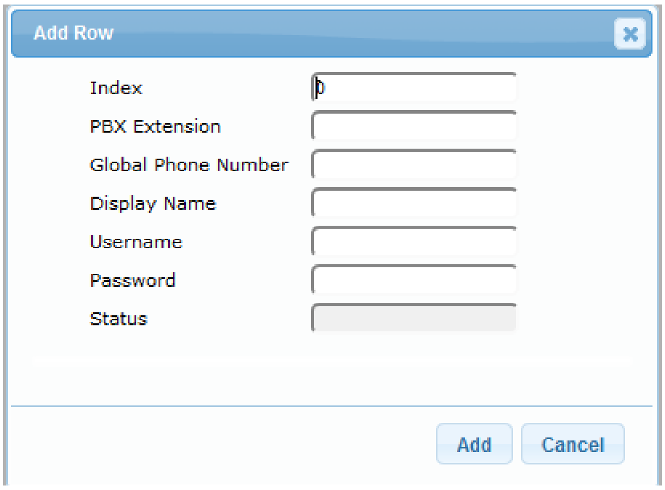
The following procedure describes how to configure and register users in the GW User Info table through the Web interface.

**Note:** If a User Info file is loaded to the device (as described in "Configuring GW User Info Table in Loadable Text File" on page 508), all previously configured entries are removed from the table in the Web interface and replaced with the entries from the loaded User Info file.

➢ **To configure the GW User Info table through the Web interface:**

1. Open the GW User Info table (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **User Information** > **GW User Info Table**).
2. Click **Add**; the following dialog box appears:

**Figure 29-2: GW User Info Table - Add Row Dialog Box**




3. Configure the GW User Info table parameters according to the table below.
4. Click **Add**.
5. To save the changes to flash memory, see "Saving Configuration" **on page 490.**

To register a user, select the user's table entry, and then from the **Action** button's drop-down list , choose **Register**. To un-register a user, select the user, and then from the **Action** button's drop-down list , choose **Un-Register**.

**Table 29-2: GW User Info Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[GWUserInfoTable_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| PBX Extension<br>[GWUserInfoTable_PBXExtension] | Defines the PBX extension number.<br>The valid value is a string of up to 10 characters.<br>**Note:** The parameter is mandatory. |
| Global Phone Number<br>[GWUserInfoTable_GlobalPhoneNumber] | Defines the "global" phone number for the IP side.<br>The valid value is a string of up to 20 characters.<br>**Note:** The parameter is mandatory. |
| Display Name<br>[GWUserInfoTable_DisplayName] | Defines the Caller ID of the PBX extension.<br>The valid value is a string of up to 30 characters. |
| Username<br>[GWUserInfoTable_Username] | Defines the username for registering the user when authentication is necessary.<br>The valid value is a string of up to 40 characters. |
| Password<br>[GWUserInfoTable_Password] | Defines the password for registering the user when authentication is necessary.<br>The valid value is a string of up to 20 characters. |
| Status | (Read-only field) Displays the status of the user - "Registered" or "Not Registered". |

### 29.1.6.2.2 Configuring GW User Info Table through CLI

The GW User Info table can be configured in the CLI using the following commands:

- To add and/or modify a user (example):

```
# configure voip
(config-voip)# sip-definition proxy-and-registration
(sip-def-proxy-and-reg)# user-info gw-user-info <index, e.g.,
1>
(gw-user-info-1)# username JohnDee
(gw-user-info-1)# <activate | exit>
```

- To delete a specific user, use the no command:

```
(sip-def-proxy-and-reg)# no user-info gw-user-info <index,
e.g., 1>
```

- To view all table entries:

```
(sip-def-proxy-and-reg)# user-info gw-user-info display
---- gw-user-info-0 ----
  pbx-ext (405)
  global-phone-num (405)
  display-name (Ext405)
  username (user405)
  password (0aGzoKfh5uI=)
  status (not-resgistered)
```

- To view a specific entry (example):

```
(sip-def-proxy-and-reg)# user-info gw-user-info <index, e.g.,
0>
```

```
(gw-user-info-0)# display
  pbx-ext (405)
  global-phone-num (405)
  display-name (Ext405)
  username (user405)
  password (0aGzoKfh5uI=)
  status (not-resgistered)
```

- To search a user by pbx-ext:

```
(sip-def-proxy-and-reg)# user-info find <pbx-ext e.g., 405>
405: Found at index 0 in GW user info table, not registered
```

#### 29.1.6.2.3 Configuring GW User Info Table in Loadable Text File

The GW User Info table can be configured as a User Info file using a text-based file (*.txt). This file can be created using any text-based program such as Notepad. You can load the User Info file using any of the following methods:

- Web interface - see "Loading Auxiliary Files" on page 493
- *ini* file, using the UserInfoFileName parameter - see "Auxiliary and Configuration File Name Parameters" on page 654
- Automatic Update mechanism, using the UserInfoFileURL parameter - see Automatic Update Mechanism

To add mapping rules to the User Info file, use the following syntax:

```
[ GW ]
FORMAT
PBXExtensionNum,GlobalPhoneNum,DisplayName,UserName,Password
```

Where:

- *[ GW ]* indicates that this part of the file is the GW User Info table
- *PBXExtensionNum* is the PBX extension number (up to 10 characters)
- *GlobalPhoneNum* is the "global" phone number (up to 20 characters) for the IP side
- *DisplayName* is the Caller ID (string of up to 30 characters) of the PBX extension
- *UserName* is the username (string of up to 40 characters) for registering the user when authentication is necessary
- *Password* is the password (string of up to 20 characters) for registering the user when authentication is necessary

Each line in the file represents a mapping rule of a single PBX extension user.

**Notes:**

- Make sure that the last line in the User Info file ends with a carriage return (i.e., by pressing the <Enter> key).
- To modify the GW User Info table using a User Info file, you need to load to the device a new User Info file containing your modifications.

Below is an example of a configured User Info file:

```
[ GW ]
FORMAT
PBXExtensionNum,GlobalPhoneNum,DisplayName,UserName,Password
401 , 638001 , Mike , miked , 1234
402 , 638002 , Lee , leem, 4321
403 , 638003 , Sue , suer, 8790
404 , 638004 , John , johnd, 7694
405 , 638005 , Pam , pame, 3928
```

```
406 , 638006 , Steve , steveg, 1119
407 , 638007 , Fred , frede, 8142
408 , 638008 , Maggie , maggiea , 9807
```

#### 29.1.6.3 Viewing the Installed User Info File Name

You can view the name of the User Info file currently installed on the device, through the device's CLI (in Enable mode):

```
# debug auxilary-files user-info info
```

For example:

```
# debug auxilary-files user-info info
  User Info File Name MyUsers.txt
```

### 29.1.7 AMD Sensitivity File

The device is shipped with a default, pre-installed *AMD Sensitivity* file for its Answering Machine Detection (AMD) feature. This file includes the detection algorithms for detecting whether a human or answering machine has answered the call, and is based on North American English. In most cases, the detection algorithms in this file suffice even when your deployment is in a region where a language other than English is spoken. However, if you wish to replace the default file with a different AMD Sensitivity file containing customized detection algorithms, please contact your AudioCodes sales representative for more information.

The AMD Sensitivity file is created in .xml format and then converted to a binary .dat file that can be installed on the device. The XML-to-binary format conversion can be done using AudioCodes DConvert utility. For more information on using this utility, refer to *DConvert Utility User's Guide*. Only one AMD Sensitivity file can be installed on the device. To install a new AMD Sensitivity file, use any of the following methods:

- Web interface: On the Load Auxiliary Files page - see "Loading Auxiliary Files" on page 493.
- TFTP during initialization: You need to configure the *ini* file parameter, AMDSensitivityFileName, and then copy the AMD Sensitivity file to the TFTP directory.
- Automatic Update feature: For more information, see Automatic Update Mechanism. For this method, the AMDSensitivityFileUrl parameter must be set through SNMP or *ini* file.

For more information on the AMD feature, see "Answering Machine Detection (AMD)" on page 203.

## 29.2 Software License Key

The License Key determines the device's supported features and call capacity, as ordered from your AudioCodes sales representative. You can upgrade or change your device's supported features and capacity, by purchasing and installing a new License Key that match your requirements.

**Note:** The availability of certain Web pages depends on the installed License Key.

### 29.2.1 Viewing the License Key

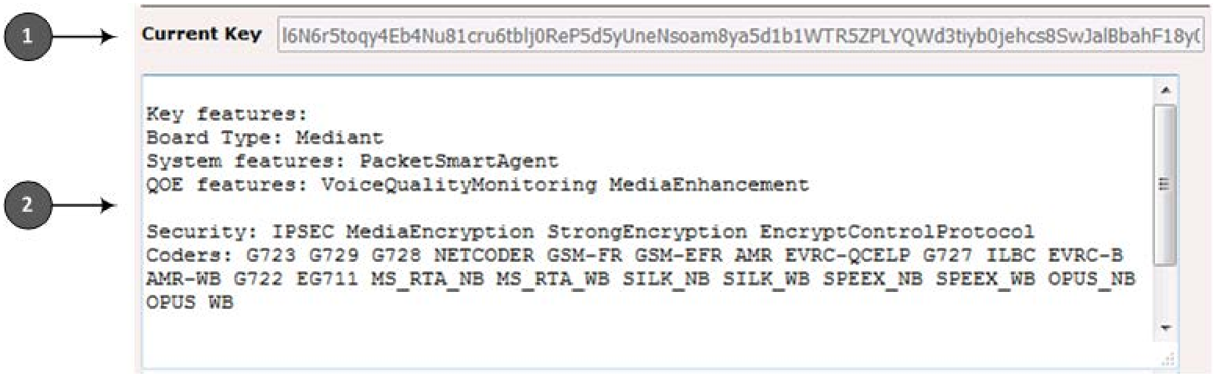
The following procedure describes how to view the device's License Key.

➢ **To view the License Key:**

- Open the Software Upgrade Key Status page (**Maintenance** tab > **Software Update** folder > **Software Upgrade Key**).

The License Key is displayed in encrypted-string format in the 'Current Key' field (1) and the main features provided by the License Key are displayed in the pane (2) below it, as shown in the example below:

**Figure 29-3: Viewing License Key (Example)**




### 29.2.2 Installing a New Software License Key

This section describes how to install a new License Key on the device.

**Note:** When you install a new License Key, it overwrites the previously installed License Key. Any license-based features that were included in the old License Key, but not included in the new License Key, will no longer be available.

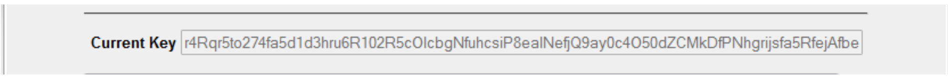
#### 29.2.2.1 Installing Software License Key through Web Interface

The following procedure describes how to install the Software License Key through the Web interface.

➢ **To install the Software License Key through the Web interface:**

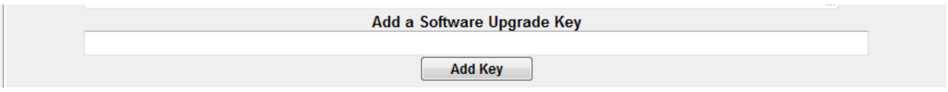
1. Open the Software Upgrade Key Status page (**Maintenance** tab > **Software Update** menu > **Software Upgrade Key**).
2. Back up the Software License Key currently installed on the device, as a precaution. If the new Software License Key does not comply with your requirements, you can re-load this backup to restore the device's original capabilities.
   a. In the 'Current Key' field, select the entire text string and copy it to any standard text file (e.g., Notepad):

   **Figure 29-4: Current Key Field**




   b. Save the text file with any file name and file extension (e.g., key.txt) to a folder on your computer.
3. Load the License Key as follows:
   a. Open the License Key file using a text-based program such as Notepad.
   b. Copy-and-paste the contents of the file into the 'Add a Software Upgrade Key' field.
   c. Click **Add Key**.

   **Figure 29-5: Add Key Button**




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.

#### 29.2.2.2 Installing Software License Key through CLI

To install the Software License Key through CLI, use the following commands:

- To install the Software License Key:

```
(config-system)# feature-key <"string enclosed in double
quotation marks">
```

- To view the Software License Key:

```
show system feature-key
```

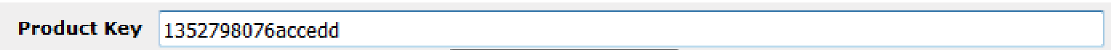
## 29.2.3 Viewing the Device's Product Key

The Product Key identifies a specific purchase of your device installation for the purpose of subsequent communication with AudioCodes (e.g., for support and software upgrades). The Product Key is your chassis' serial number--"S/N(Product Key)"--which also appears on the product label affixed to the chassis.

The Product Key is included in the License Key. Once the License Key is installed, you can view the Product Key in the following Web pages:

- Software Upgrade Key Status page (**Maintenance** tab > **Software Update** folder > **Software Upgrade Key**). The Product Key is displayed in the read-only 'Product Key' field, as shown in the example below:

**Figure 29-6: Viewing Product Key**




- Device Information page (see Viewing Device Information on page 545).

## 29.3 Software Upgrade Wizard

The Web interface's Software Upgrade Wizard lets you easily upgrade the device's software version (.cmp file). The wizard also provides you the option to load other files such as an *ini* file and Auxiliary files (e.g., Call Progress Tone / CPT file). However, loading a .cmp file is mandatory through the wizard and before you can load any other type of file, the .cmp file must be loaded.

**Notes:**

- You can obtain the latest software files from AudioCodes Web site at http://www.audiocodes.com/downloads.
- When you start the wizard, the rest of the Web interface is unavailable. After the files are successfully installed with a device reset, access to the full Web interface is restored.
- If you upgraded your firmware (.cmp file) and the "SW version mismatch" message appears in the Syslog or Web interface, your Software License Key does not support the new .cmp file version. If this occurs, contact AudioCodes support for assistance.
- Instead of manually upgrading the device, you can use the device's Automatic Update feature for automatic provisioning (see "Automatic Provisioning" on page 519).
- You can also upgrade the device's firmware by loading a .cmp file from an external USB hard drive connected to the device's USB port. For more information, see USB Storage Capabilities on page 541.

The following procedure describes how to load files using the Web interface's Software Upgrade Wizard. Alternatively, you can load files using the CLI:

- cmp file:

  ```
  copy firmware from <URL>
  ```

- ini or Auxiliary file:

  ```
  copy <ini file or auxiliary file> from <URL>
  ```

- CLI script file:

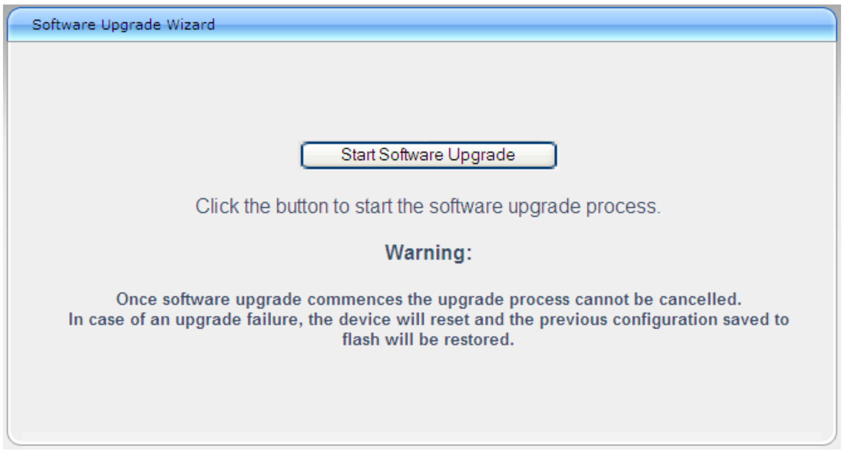
  ```
  copy cli-script from <URL>
  ```

➢ **To upgrade the device using the Software Upgrade Wizard:**

1. Make sure that you have installed a new Software License Key (see "Software License Key" on page 510) that is compatible with the software version to be installed.
2. It is recommended to enable the Graceful Lock feature (see "Locking and Unlocking the Device" on page 489). The wizard resets the device at the end of the upgrade process, thereby causing current calls to be untimely terminated. To minimize this traffic disruption, the Graceful Lock feature prevents the establishment of new calls.
3. It is recommended to save a copy of the device's configuration to your computer. If an upgrade failure occurs, you can restore your configuration settings by uploading the backup file to the device. For saving and restoring configuration, see "Backing Up and Loading Configuration File" on page 517.
4. Open the Software Upgrade wizard, by performing one of the following:
   - Select the **Maintenance** tab, click the **Software Update** menu, and then click **Software Upgrade Wizard**.

- On the toolbar, click **Device Actions**, and then choose **Software Upgrade Wizard**.

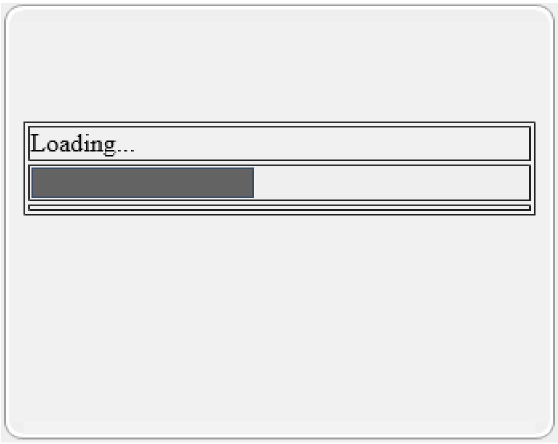
**Figure 29-7: Start Software Upgrade Wizard Screen**




5. Click **Start Software Upgrade**; the wizard starts, prompting you to load a .cmp file.

> **Note:** At this stage, you can quit the Software Upgrade Wizard without having to reset the device, by clicking **Cancel** . However, if you continue with the wizard and start loading the cmp file, the upgrade process must be completed with a device reset.

6. Click **Browse**, and then navigate to where the .cmp file is located on your computer. Select the file, and then click **Open**.
7. Click **Load File**; the device begins to install the .cmp file. A progress bar displays the status of the loading process and a message informs you when file load successfully completes:

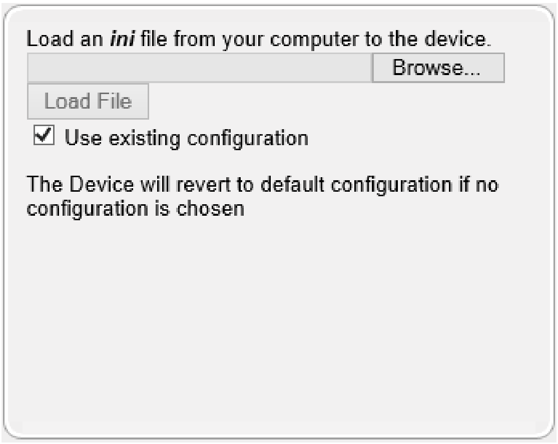


8. If you want to load additional files, skip this step and continue with the next step. If you **only** want to load a .cmp file, click **Reset** ; the device burns the .cmp file to its flash memory and then resets. The device uses the existing configuration (*ini*) and Auxiliary files.

**Note:** Device reset may take a few minutes (even up to 30 minutes), depending on cmp file version.

9. To load additional files, use the **Next** and **Back** buttons to navigate through the wizard to the desired file-load wizard page. Alternatively, you can navigate to the relevant file-load wizard page by clicking the respective file-name buttons listed in the left pane of the wizard pages.
10. The wizard page for loading an *ini* file provides you with the following options:
    - **Load a new ini file:** In the 'Load an ini file...' field, click **Browse**, and then navigate to where the ini file is located on your computer. Select the file, and then click **Load File**; the device loads the *ini* file.
    - **Retain the existing configuration (default):** Select the 'Use existing configuration' check box to use the current configuration (and do not select an ini file).
    - **Restore configuration to factory defaults:** Clear the 'Use existing configuration' check box (and do not select an ini file).

**Figure 29-8: Software Upgrade Wizard - Load INI File**

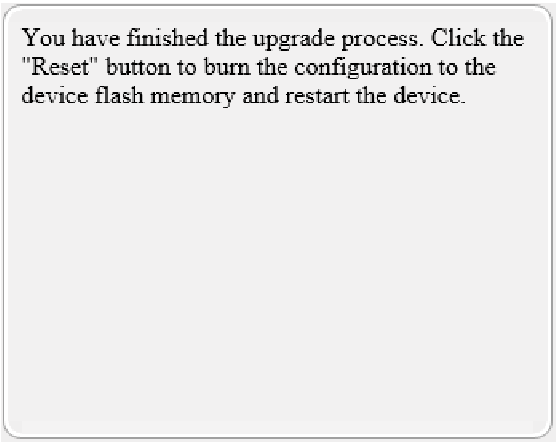



**Note:** If you use the wizard to load an *ini* file, parameters excluded from the *ini* file are assigned default values (according to the .cmp file running on the device) and thereby, overwrite values previously configured for these parameters.

11. When you have completed loading all the desired files, click **Next** until the last

wizard page appears (the **FINISH** button is highlighted in the left pane):

**Figure 29-9: Software Upgrade Wizard - Files Loaded**

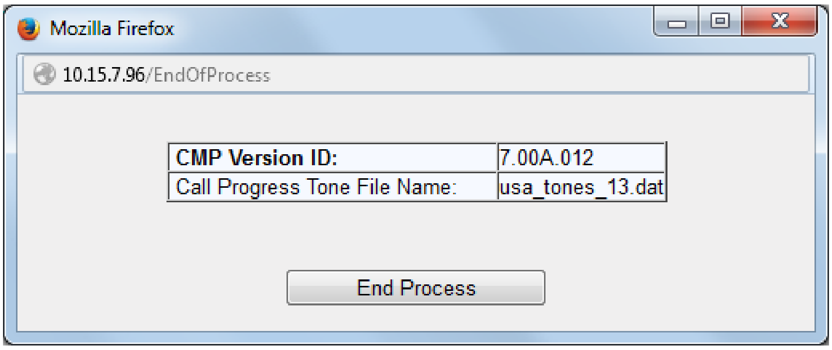



12. Click **Reset** to burn the files to the device's flash memory; the "Burn and reset in progress" message is displayed and the device 'burns' the newly loaded files to flash memory and then resets.

> **Note:** Device reset may take a few minutes (even up to 30 minutes), depending on .cmp file version.

When the device finishes the installation process and resets, the following wizard page is displayed, showing the installed software version and other files (ini file and Auxiliary files) that you may also have installed:

**Figure 29-10: Software Upgrade Process Completed Successfully (Example)**




13. Click **End Process** to close the wizard; the Web Login dialog box appears.
14. Enter your login username and password, and then click **Login**; a message box appears informing you of the new .cmp file version.
15. Click **OK**; the Web interface becomes active, reflecting the upgraded device.

## 29.4 Backing Up and Loading Configuration File

You can save a copy/backup of the device's current configuration settings as an *ini* file to a folder on your computer, using the Configuration File page. The saved file includes only parameters that were modified and parameters with other than default values. The Configuration File page also allows you to load an *ini* file to the device. If the device has "lost" its configuration, you can restore the device's configuration by loading the previously saved *ini* file or by simply loading a newly created *ini* file.

You can also save the current configuration to a remote server or USB and update configuration from an external USB hard drive connected to the device's USB port. For more information, see USB Storage Capabilities on page 541.

```
# copy cli-script to <URL of TFTP/HTTP/HTTPS server or USB>
```

For example:

- Remote server:

  ```
  # copy cli-script to tftp://192.168.0.3/config-device1.txt
  ```

- USB:

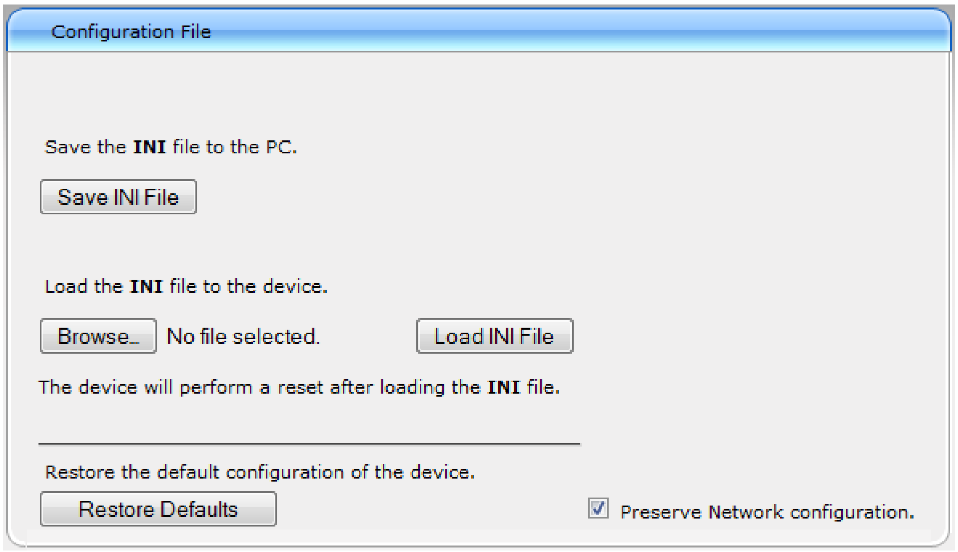
  ```
  # copy cli-script to usb://config-device1.txt
  ```

**Note:** When loading an *ini* file using the Configuration File page, parameters not included in the *ini* file are reset to default settings.

#### ➢ To save or load an ini file:

1. Open the Configuration File page by doing one of the following:
   - From the Navigation tree, click the **Maintenance** tab, click the **Software Update** menu, and then click **Configuration File**.
   - On the toolbar, click **Device Actions**, and then from the drop-down menu, choose **Load Configuration File** or **Save Configuration File**.

**Figure 29-11: Configuration File Page**




2. To save the *ini* file to a folder on your computer:

a. Click the **Save INI File** button; the File Download dialog box appears.
b. Click the **Save** button, navigate to the folder where you want to save the file, and then click **Save**.

3. To load the *ini* file to the device:
   a. Click the **Browse** button, navigate to the folder where the file is located, select the file, and then click **Open**; the name and path of the file appear in the field beside the **Browse** button.
   b. Click the **Load INI File** button, and then at the prompt, click **OK**; the device uploads the file and then resets. Once complete, the Web Login screen appears, requesting you to enter your user name and password.

# 30 Automatic Provisioning

This chapter describes the device's automatic provisioning mechanisms.

## 30.1 Automatic Configuration Methods

The table below summarizes the automatic provisioning methods supported by the device:

**Table 30-1: Automatic Provisioning Methods**

| BootP / TFTP | DHCP | | Automatic Update Methods | | | | SNMP (EMS) |
|---|---|---|---|---|---|---|---|
| | 67 | 66 | HTTP/S | TFTP | FTP | NFS | |
| No | Yes | Yes | Yes | Yes | Yes | No | Yes |

### 30.1.1 DHCP-based Provisioning

A third-party DHCP server can be configured to automatically provide each device, acting as a DHCP client, with a temporary IP address so that individual MAC addresses are not required. The DHCP server can provide additional networking parameters such as subnet mask, default gateway, primary and secondary DNS server, and two SIP server addresses. These network parameters have a time limit, after which the device must 'renew' its lease from the DHCP server.

The device can use a host name in the DHCP request. The host name is set to acl_nnnnn, where *nnnnn* denotes the device's serial number. The serial number is the last six digits of the MAC address converted to decimal representation. In networks that support this feature and if the DHCP server registers this host name to a DNS server, you can access the device (through a Web browser) using the URL, http://acl_<serial number> (instead of using the device's IP address). For example, if the device's MAC address is 00908f010280, the DNS name is acl_66176.

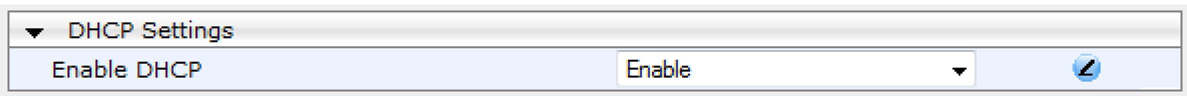
**Notes:**

- When using DHCP to acquire an IP address, the Interface table, VLANs and other advanced configuration options are disabled.
- For additional DHCP parameters, see "DHCP Parameters" on page 666.

➢ **To enable the device as a DHCP client:**

1. Open the Application Settings page (**Configuration** tab > **System** menu > **Application Settings**).

**Figure 30-1: 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 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.

#### 30.1.1.1 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;
    }
}
```

#### 30.1.1.2 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.

### 30.1.2 HTTP-based Provisioning

An HTTP or HTTPS server can be located in the network in which the device is deployed, storing configuration and software files for the device to download. This does not require additional servers and is NAT-safe.

For example, assume the core network HTTPS server is https://www.corp.com. A master configuration ini file can be stored on the server, e.g., https://www.corp.com/gateways/master.ini. This file could point to additional ini files, Auxiliary files (e.g., call progress tones), and software files (cmp), all on the same HTTP server or different HTTP servers in the network.

The main advantage of this method is that the device can be configured to periodically check the HTTP server for file updates. HTTP(S) is not sensitive to NAT devices, enabling configuration whenever needed without on-site intervention. For additional security, the URL may contain a different port, and username and password.

The only configuration required is to preconfigure the device(s) with the URL of the initial (master) ini file. This can be done using one of the following methods:

- DHCP as described in "DHCP-based Provisioning" on page 519 or 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 520).
- 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 523.

### 30.1.3 FTP- based Provisioning

Some networks block access to HTTP(S). The Automatic Update feature provides limited support for FTP/FTPS connectivity. Periodic polling for updates is not possible since these protocols do not support conditional fetching, i.e., updating files only if it is changed on the server.

The only difference between this method and those described in "HTTP-based Provisioning" on page 521 and Provisioning from HTTP Server using DHCP Option 67 on page 520 is that the protocol in the URL is "ftp" (instead of "http").

### 30.1.4 Provisioning using AudioCodes EMS

AudioCodes EMS server functions as a core-network provisioning server. The device's SNMP Manager should be configured with the IP address of the EMS server, using one of the methods detailed in the previous sections. As soon as a registered device contacts the EMS server through SNMP, the EMS server handles all required configuration automatically, upgrading software as needed. This alternative method doesn't require additional servers at the customer premises, and is NAT-safe.

## 30.2 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 655 or refer to the CLI Reference Guide.
- For additional security, use HTTPS or FTPS. The device supports HTTPS (RFC 2818) and FTPS using the AUTH TLS method <draft-murray-auth-ftp-ssl-16>.

### 30.2.1 Files Provisioned by Automatic Update

You can use the Automatic Update feature to update the device with any of the following files:

- Software file (*cmp)*
- Auxiliary files (e.g., Call Progress Tones, SSL Certificates, SSL Private Key)
- Configuration file:
    - ini File: Contains only ini file parameters and configures all the device's functionalities.
    - CLI Script File: Contains only CLI commands and configures all the device's functionalities (except commands such as show, debug or copy). The file updates the device's configuration only according to the configuration settings in the file. The device's existing configuration settings (not included in the file) are retained. The device does not undergo a reset and therefore, this file typically contains configuration settings that do not require a device reset. If a reset is required, for example, to apply certain settings, you must include the following CLI command (root level) at the end of the file:

```
# reload if-needed
```

To configure the URL of the server where the file is located, use the AUPDCliScriptURL ini file parameter or CLI command, configure system > automatic-update > cli-script <URL>.

### 30.2.2 File Location for Automatic Update

The files for updating the device can be stored on any standard Web (HTTP/S), FTP, or TFTP server. The files can be loaded periodically to the device using HTTP, HTTPS, FTP, or TFTP. This mechanism can be used even when the device is installed behind NAT and firewalls.

The Automatic Update feature is done per file and configured by specifying the file name and URL address of the provisioning server where the file is located. For a description of the parameters used to configure URLs per file, see "Automatic Update Parameters" on page 655. Below are examples for configuring the file names and their URLs for Automatic Update:

- ini File:

```
IniFileURL = 'http://www.corp.com/configuration.ini'
CptFileURL = 'http://www.corp.com/call_progress.dat'
FeatureKeyURL = 'https://www.company.com/License_Key.txt'
AutoCmpFileUrl = 'http://www.corp.com/SIP_F7.00A.008.cmp
```

- CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# cli-script https://company.com/cli/<MAC>
(automatic-update)# voice-configuration http://www.company.com/configuration.ini
(automatic-update)# feature-key
http://www.company.com/License_Key.txt
(automatic-update)# call-progress-tones http://www.company.com/call_progress.dat
(automatic-update)# auto-firmware http://www.company.com/SIP_F7.00A.008.cmp
```

**Note:** For configuration files (ini), the file name in the URL can automatically contain the device's MAC address for enabling the device to download a file unique to the device. For more information, see "MAC Address Placeholder in Configuration File Name" on page 529.

### 30.2.3 Triggers for Automatic Update

The Automatic Update feature can be triggered by the following:

- Upon device startup (reset or power up). To disable this trigger, run the following CLI command:

```
(config-system)# automatic-update
(automatic-update)# run-on-reboot off
```

- Periodically:
    - Specified time of day (e.g., 18:00), configured by the ini file parameter AutoUpdatePredefinedTime or CLI command configure system > automatic-update > predefined-time.
    - Interval between Automatic Updates (e.g., every 60 minutes), configured by the ini file parameter AutoUpdateFrequency or CLI command configure system > automatic-update > update-frequency.
- Centralized provisioning server request:
    - Upon receipt of an SNMP request from the provisioning server.

- Upon receipt of a special SIP NOTIFY message from the provisioning server. The NOTIFY message includes an Event header with the AudioCodes proprietary value, "check-sync;reboot=false", as shown in the example below:

```
NOTIFY sip:<user>@<dsthost> SIP/2.0
To: sip:<user>@<dsthost>
From: sip:sipsak@<srchost>
CSeq: 10 NOTIFY
Call-ID: 1234@<srchost>
Event: check-sync;reboot=false
```

**To enable this feature through the Web interface:**

a. Open the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**).
b. Under the **Misc Parameters** group, set the 'SIP Remote Reset' parameter to **Enable**.
c. Click **Submit**.

To enable through CLI: configure voip > sip-definition advanced-settings > sip-remote-reset.

### 30.2.4 Access Authentication with HTTP Server

You can configure the device to authenticate itself with the HTTP/S server. The device authenticates itself by providing the HTTP/S server with its authentication username and password. You can configure one of the following HTTP authentication schemes:

- **Basic Access Authentication:** The device provides its username and password to the HTTP server. The username and password is configured in the URL that you define for downloading the file:
  - ini file:

```
AutoCmpFileUrl = 'https://<username>:<password>@<IP address or domain name>/<file name>'
```

  - CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# auto-firmware https://<username>:<password>@<IP address or domain name>/<file name>
```

- **Digest Access Authentication:** The authentication username and password is negotiated between the device and HTTP/S server, using digest MD5 cryptographic hashing. This method is safer than basic access authentication. The digest authentication username and password are configured using the AUPDDigestUsername and AUPDDigestPassword parameters, respectively.

### 30.2.5 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 525). If authentication succeeds, Step 2 occurs.
2. The device establishes an HTTP/S connection with the URL host (provisioning server). If the connection is HTTPS, the device verifies the certificate of the provisioning server, and presents its own certificate if requested by the server.
3. The device queries the provisioning server for the requested file by sending an HTTP

Get request. This request contains the HTTP User-Agent Header, which identifies the device to the provisioning server. By default, the header includes the device's model name, MAC address, and currently installed software and configuration versions. Based on its own dynamic applications for logic decision making, the provisioning server uses this information to check if it has relevant files available for the device and determines which files must be downloaded (working in conjunction with the HTTP If-Modified-Since header, described further on in this section).

You can configure the information sent in the User-Agent header, using the AupdHttpUserAgent parameter or CLI command, configure system > http-user-agent. The information can include any user-defined string or the following supported string variable tags (case-sensitive):

- **<NAME>:** product name, according to the installed Software License Key
- **<MAC>:** device's MAC address
- **<VER>:** software version currently installed on the device, e.g., "7.00.200.001"
- **<CONF>:** configuration version, as configured by the ini file parameter, INIFileVersion or CLI command, configuration-version

The device automatically populates these tag variables with actual values in the sent header. By default, the device sends the following in the User-Agent header:

```
User-Agent: Mozilla/4.0 (compatible; AudioCodes;
<NAME>;<VER>;<MAC>;<CONF>)
```

For example, if you set AupdHttpUserAgent = MyWorld-<NAME>;<VER>(<MAC>), the device sends the following User-Agent header:

```
User-Agent: MyWorld-Mediant;7.00.200.001(00908F1DD0D3)
```

**Note:** If you configure the AupdHttpUserAgent parameter with the <CONF> variable tag, you must reset the device with a burn-to-flash for your settings to take effect.

4. If the provisioning server has relevant files available for the device, the following occurs, depending on file type and configuration:

   - **File Download upon each Automatic Update process:** This is applicable to software (.cmp), ini files. In the sent HTTP Get request, the device uses the HTTP If-Modified-Since header to determine whether to download these files. The header contains the date and time (timestamp) of when the device last downloaded the file from the specific URL. This date and time is regardless of whether the file was installed or not on the device. An example of an If-Modified-Since header is shown below:

     ```
     If-Modified-Since: Mon, 1 January 2014 19:43:31 GMT
     ```

     If the file on the provisioning server was unchanged (modified) since the date and time specified in the header, the server replies with an HTTP 304 response and the file is not downloaded. If the file was modified, the provisioning server sends an HTTP 200 OK response with the file in the body of the HTTP response. The device downloads the file and compares the version of the file with the currently installed version on its flash memory. If the downloaded file is of a later version, the device installs it after the device resets (which is only done after the device completes all file downloads); otherwise, the device does not reset and does not install the file.

     To enable the automatic software (.cmp) file download method based on this timestamp method, use the ini file parameter, AutoCmpFileUrl or CLI command, configure system > automatic-update > auto-firmware <URL>. The device uses the same configured URL to download the .cmp file for each subsequent Automatic Update process.

You can also enable the device to run a CRC on the downloaded configuration file (ini) to determine whether the file has changed in comparison to the previously downloaded file. Depending on the CRC result, the device can install or discard the downloaded file. For more information, see "Cyclic Redundancy Check on Downloaded Configuration Files" on page 529.

**Notes:**

- When this method is used, there is typically no need for the provisioning server to check the device's current firmware version using the HTTP-User-Agent header.
- The Automatic Update feature assumes that the Web server conforms to the HTTP standard. If the Web server ignores the If-Modified-Since header or doesn't provide the current date and time during the HTTP 200 OK response, the device may reset itself repeatedly. To overcome this problem, modify the update frequency, using the ini file parameter AutoUpdateFrequency or CLI command configure system > automatic update > update-frequency.

- **One-time File Download:** This is applicable to software (.cmp) and Auxiliary (e.g., License Key, CPT and Dial Plan) files. The device downloads these files only **once**, regardless of how many times the device may repeat the Automatic Update process. Once they are downloaded, the device discards their configured URLs. To update these files again, you need to configure their URL addresses and filenames again. Below is an example of how to configure URLs for some of these files:

  **Auxiliary Files:**

  - ini:

    ```
    CptFileURL =
    'https://www.company.com/call_progress.dat'
    FeatureKeyURL =
    'https://www.company.com/License_Key.txt'
    ```

  - CLI:

    ```
    (config-system)# automatic-update
    (automatic-update)# call-progress-tones
    http://www.company.com/call_progress.dat
    (automatic-update)# tls-root-cert https://company.com/root.pem
    ```

  **Software (.cmp) File:**

  - ini:

    ```
    CmpFileUrl =
    'https://www.company.com/device/v.6.80A.227.005.cmp'
    ```

  - CLI:

    ```
    (config-system)# automatic-update
    (automatic-update)# firmware
    https://www.company.com/device/v.6.80A.227.005.cmp
    ```

**Notes:**

- For one-time file download, the HTTP Get request sent by the device does not include the If-Modified-Since header. Instead, the HTTP-User-Agent header can be used in the HTTP Get request to determine whether firmware update is required.
- When downloading SSL certificates (Auxiliary file), it is recommended to use HTTPS with mutual authentication for secure transfer of the SSL Private Key.
- After the device downloads the License Key file (FeatureKeyURL), it checks that the serial number in the file (“S/N <serial number>") is the same as that of the device. If the serial number is the same and the license key is different to the one currently installed on the device, it applies the new License Key.

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.

### 30.2.6 File Download Sequence

Whenever the Automatic Update feature is triggered (see "Triggers for Automatic Update" on page 524), the device attempts to download each file from the configured URLs, in the following order:

1. ini file
2. CLI Script file
3. Periodic software file (.cmp) download
4. One-time software file (.cmp) download
5. Auxiliary file(s)

The following files automatically instruct the device to reset:

- Periodic software file (.cmp)
- One-time software file (.cmp)

When multiple files requiring a reset are downloaded, the device resets only **after** it has downloaded and installed **all** the files. However, you can explicitly instruct the device to immediately reset for the following files:

- ini file: Use the ResetNow in file parameter
- CLI Script file: Use the reload if-needed CLI command

**Warning:** If you use the ResetNow parameter in an ini file for periodic automatic provisioning with non-HTTP (e.g., TFTP) and without CRC, the device resets after every file download. Therefore, use the parameter with caution and only if necessary for your deployment requirements.

**Notes:**

- For ini file downloads, by default, parameters not included in the file are set to defaults. To retain the current settings of these parameters, set the SetDefaultOnINIFileProcess parameter to 0.
- If you have configured one-time software file (.cmp) download (configured by the ini file parameter CmpFileURL or CLI command configure system > automatic-update > firmware), the device will only apply the file if one-time software updates are enabled. This is disabled by default to prevent unintentional software upgrades. To enable one-time software upgrades, set the ini file parameter AutoUpdateCmpFile to 1 or CLI command, configure system > automatic-update > update-firmware on.
- If you need to update the device's software and configuration, it is recommended to first update the software. This is because the current ("old") software (before the upgrade) may not be compatible with the new configuration. However, if both files are available for download on the provisioning server(s), the device first downloads and applies the new configuration, and only then does it download and install the new software. Therefore, this is a very important issue to take into consideration.
- If more than one file needs to be updated - CLI Script and cmp: The device downloads and applies the CLI Script file on the currently ("old") installed software version. It then downloads and installs the cmp file with a reset. Therefore, the CLI Script file MUST have configuration compatible with the "old" software version.

### 30.2.7 Cyclic Redundancy Check on Downloaded Configuration Files

You can enable the device to perform cyclic redundancy checks (CRC) on downloaded configuration files (ini) during the Automatic Update process. The CRC checks whether the content (raw data) of the downloaded file is different to the content of the previously downloaded file from the previous Automatic Update process. The device compares the CRC check value (code) result with the check value of the previously downloaded file. If the check values are identical, it indicates that the file has no new configuration settings, and the device discards the file. If the check values are different, it indicates that the downloaded file is different (i.e., includes updates), and the device installs the downloaded file and applies the new configuration settings.

CRC is useful, for example, when the service provider replaces a file, on the provisioning server, with another file whose contents are the same. When the device sends an HTTP Get request during the Automatic Update process, the provisioning server sends the new file to the device. This occurs as the timestamp between the previously downloaded file and this new file is different (determined by the HTTP If-Modified-Since header in the Get request). Therefore, the CRC feature can be used to prevent the device from installing such files.

For enabling CRC, use the ini file parameter AUPDCheckIfIniChanged or CLI command, configure system > automatic-update > crc-check regular. By default, CRC is disabled. For more information on the parameter, see "Automatic Update Parameters" on page 655.

### 30.2.8 MAC Address Placeholder in Configuration File Name

You can configure the file name of the configuration file (ini) in the URL to automatically include the MAC address of the device. As described in "File Location for Automatic Update" on page 524, the file name is included in the configured URL of the provisioning server where the file is located.

Including the MAC address in the file name is useful if you want the device to download a file that is unique to the device. This feature is typically implemented in mass provisioning of devices where each device downloads a specific configuration file. In such a setup, the provisioning server stores configuration files per device, where each file includes the MAC address of a specific device in its file name.

To support this feature, you need to include the MAC address placeholder string, "<MAC>" anywhere in the configured file name of the URL, for example:

```
IniFileURL = 'https://www.company.com/config_<MAC>.ini'
(automatic-update)# cli-script
https://company.com/files/cli_script_<MAC>.txt
```

The device automatically replaces the string with its hardware MAC address, resulting in a file name request that contains the device's MAC address, for example, config_00908F033512.ini. Therefore, you can configure all the devices with the same URL and file name.

> **Note:** If you write the MAC address placeholder string in lower case (i.e., "<mac>"), the device adds the MAC address in lower case to the file name (e.g., config_<mac>.ini results in config_00908f053736e); if in upper case (i.e., "<MAC>"), the device adds the MAC address in upper case to the file name (e.g., config_<MAC>.ini results in config_00908F053736E).

### 30.2.9 File Template for Automatic Provisioning

To facilitate automatic provisioning setup, you can use a single template to define the files to download during automatic provisioning. The template uses special keywords to denote the different file types to download and in the URL address of the provisioning server it uses a placeholder for the file names which is replaced by hardcoded file names and extensions according to file type, as described in more detail below.

> **Note:**
>
> - Unlike the parameters that define specific URLs for Auxiliary files (e.g., CptFileURL), the file template feature always retains the URLs after each automatic update process. Therefore, with the file template the device always attempts to download the files upon each automatic update process.
> - If you configure a parameter that defines a URL for a specific file (e.g., CptFileURL), the settings of the file template (TemplateUrl parameter) is ignored for the specific file type (e.g., CPT file).
> - Additional placeholders can be used in the file name in the URL, for example, <MAC> for MAC address (see MAC Address Placeholder in Configuration File Name on page 529).

➢ **To use a file template for automatic provisioning:**

1. Define the file **types** to download by the file template, using the AupdFilesList parameter. Use the keywords listed in the table below to specify each file type. For example, to specify ini, Feature Key, and CPT files:
   - ini File:
     ```
     AupdFilesList = 'ini', 'fk', 'cpt'
     ```
   - CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# template-files-list ini,fk,cpt
```

2. Define the URL address of the provisioning server on which the files (specified in Step 1) are located, using the TemplateUrl parameter. When you configure the URL, you must include the file type placeholder, "<FILE>", which represents the file name. For each file type specified in Step 1, the device sends an HTTP request to the server, where the placeholder in the URL is replaced with the filename and extension, as listed in the below table. For example, if you configure the AupdFilesList parameter as in Step 1 and the TemplateUrl parameter to:
    - ini File:
      ```
      TemplateUrl = 'http://10.8.8.20/Site1_<FILE>'
      ```
    - CLI:
      ```
      # configure system
      (config-system)# automatic update
      (automatic-update)# template-url http://10.8.8.20/Site1_<FILE>
      ```

    The device sends HTTP requests to the following URLs:
    - http://10.8.8.20/Site1_**device.ini**
    - http://10.8.8.20/Site1_**fk.ini**
    - http://10.8.8.20/Site1_**cpt.data**
3. Place the files to download on the provisioning server. Make sure that their file names and extensions are based on the hardcoded string values specific to the file type for the <FILE> placeholder (e.g., "Site1_device.ini" for the ini file), as shown in the table below.

**File Template Keywords and Placeholder Values per File Type**

| File Type | Keywords for Template File | Value Replacing <FILE> Placeholder |
|---|---|---|
| ini file | ini | device.ini |
| CLI Script file | cli | cliScript.txt |
| CMP file based on timestamp | acmp | autoFirmware.cmp |
| User Info file | usrinf | userInfo.txt |
| CMP file | cmp | firmware.cmp |
| Feature Key file | fk | fk.ini |
| Call Progress Tone (CPT) file | cpt | cpt.dat |
| Prerecorded Tones (PRT) file | prt | prt.dat |
| Dial Plan file | dpln | dialPlan.dat |
| Answering Machine Detection (AMD) file | amd | amd.dat |
| SSL/TLS Private Key file | sslp | pkey.pem<br>pkey<ID>.pem (for multi-certificate system) |
| SSL/TLS Root Certificate file | sslr | root.pem<br>root<ID>.pem (for multi-certificate system) |
| SSL/TLS Certificate file | sslc | cert.pem<br>cert<ID>.pem (for multi-certificate system) |

## 30.2.10 Automatic Update Configuration Examples

This section provides a few examples on configuring the Automatic Update feature.

### 30.2.10.1 Automatic Update for Single Device

This simple example describes how to configure the Automatic Update feature for updating a single device. In this example, the device queries the provisioning server for software, configuration and Auxiliary files every 24 hours.

➢ **To set up Automatic Provisioning for single device (example):**

1. Set up an HTTP Web server (e.g., http://www.company.com) and place all the required configuration files on this server.
2. Configure the device with the IP address of the DNS server for resolving the domain name (e.g., http://www.company.com) that is used in the URL of the provisioning server. You configure this in the Interface table:
   - ini File:
     ```
     [ InterfaceTable ]
     FORMAT InterfaceTable_Index =
     ```

```
InterfaceTable_ApplicationTypes,
InterfaceTable_InterfaceMode, InterfaceTable_IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable_VlanID, InterfaceTable_InterfaceName,
InterfaceTable_PrimaryDNSServerIPAddress,
InterfaceTable_SecondaryDNSServerIPAddress,
InterfaceTable_UnderlyingDevice;
InterfaceTable 0 = 6, 10, 10.15.7.95, 16, 10.15.0.1, 1,
"Voice", 80.179.52.100, 0.0.0.0, "vlan 1";
[ \InterfaceTable ]
```

- CLI:

```
# configure voip
(config-voip)# interface network-if 0
(network-if-0)# primary-dns 80.179.52.100
```

3. Configure the device with the following Automatic Update settings:

   a. Automatic Update is done every 24 hours (1440 minutes):

      - ini File:

        ```
        AutoUpdateFrequency = 1440
        ```

      - CLI:

        ```
        # configure system
        (config-system)# automatic update
        (automatic-update)# update-frequency 1440
        ```

   b. Automatic Update of software file (.cmp):

      - ini File:

        ```
        AutoCmpFileUrl =  'https://www.company.com/sw.cmp'
        ```

      - CLI:

        ```
        # configure system
        (config-system)# automatic update
        (automatic-update)# auto-firmware 'http://www.company.com/sw.cmp'
        ```

   c. Automatic Update of Call Progress Tone file:

      - ini File:

        ```
        CptFileURL =
        'https://www.company.com/call_progress.dat'
        ```

      - CLI:

        ```
        # configure system
        (config-system)# automatic update
        (automatic-update)# call-progress-tones
        'http://www.company.com/call_progress.dat'
        ```

   d. Automatic Update of ini configuration file:

      - ini File:

        ```
        IniFileURL = 'https://www.company.com/config.ini'
        ```

      - CLI:

        ```
        # configure system
        (config-system)# automatic update
        (automatic-update)# voice-configuration
        'http://www.company.com/config.ini'
        ```

   e. Enable Cyclical Redundancy Check (CRC) on downloaded ini file:

      - ini File:

        ```
        AUPDCheckIfIniChanged = 1
        ```

      - CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# crc-check regular
```

4. Power down and then power up the device.

### 30.2.10.2 Automatic Update from Remote Servers

This example describes how to configure the Automatic Update feature where files are stored and downloaded from different file server types. The example scenario includes the following:

- 
- FTPS server at ftpserver.corp.com for storing the Voice Prompts (VP) file. The login credentials to the server are username "root" and password "wheel".
- HTTP server at www.company.com for storing the configuration file (ini).
- DNS server at 80.179.52.100 for resolving the domain names of the provisioning servers (FTPS and HTTP).

**➢ To set up Automatic Provisioning for files stored on different server types (example):**

1. **VP file:**
   a. Set up an FTPS server and copy the VP file to the server.
   b. Configure the device with the URL path of the VP file:
      - ini File:
        ```
        VPFileUrl =
        'ftps://root:wheel@ftpserver.corp.com/vp.dat'
        ```
      - CLI:
        ```
        # configure system
        (config-system)# automatic update
        (automatic-update)# voice-prompts
        'ftps://root:wheel@ftpserver.corp.com/vp.dat'
        ```
2. **Software (.cmp) and ini files:**
   a. Set up an HTTP Web server and copy the .cmp and configuration files to the server.
   b. Configure the device with the URL paths of the .cmp and ini files:
      - ini File:
        ```
        AutoCmpFileUrl =
        'http://www.company.com/device/sw.cmp'
        IniFileURL = 'http://www.company.com/device/inifile.ini'
        ```
      - CLI:
        ```
        # configure system
        (config-system)# automatic update
        (automatic-update)# auto-firmware 'http://www.company.com/sw.cmp'
        ```
3. Configure the device with the IP address of the DNS server for resolving the domain names of the FTPS and HTTP servers:
   ```
   [ InterfaceTable ]
   FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes,
   InterfaceTable_InterfaceMode, InterfaceTable_IPAddress,
   InterfaceTable_PrefixLength, InterfaceTable_Gateway,
   InterfaceTable_VlanID, InterfaceTable_InterfaceName,
   InterfaceTable_PrimaryDNSServerIPAddress,
   InterfaceTable_SecondaryDNSServerIPAddress,
   ```

```
InterfaceTable_UnderlyingDevice;
InterfaceTable 0 = 6, 10, 10.15.7.95, 16, 10.15.0.1, 1,
"Voice", 80.179.52.100, 0.0.0.0, "vlan 1";
[ \InterfaceTable ]
```

4. Configure the device to perform the Automatic Update process daily at 03:00 (3 a.m):
   - ini File:

```
AutoUpdatePredefinedTime = '03:00'
```

   - CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# predefined-time 03:00
```

### 30.2.10.3 Automatic Update for Mass Deployment

This example describes how to configure the Automatic Update feature for updating multiple devices (i.e., mass deployment) using an HTTP provisioning server. In this example, all the devices are configured to download the same "master" configuration file. This file serves as the configuration template and instructs the devices which files to download and how often to perform the Automatic Update process. In addition, the master file also instructs each device to download an ini configuration file whose file name contains the MAC address of the device.

The example scenario is as follows:

- All devices download a "master" configuration file that contains the following:
  - Common configuration shared by all device's.
  - Specific configuration that instructs each device to download a specific configuration file based on the device's MAC address, using the special string "<MAC>" in the URL, as described in "MAC Address Placeholder in Configuration File Name" on page 529.
- Device queries the provisioning server daily at 24:00 (midnight) for software, configuration and Auxiliary files.
- HTTP-based provisioning server at www.company.com for storing the files.
- DNS server at 80.179.52.100 for resolving the domain name of the provisioning server.

➢ **To set up automatic provisioning for mass provisioning (example):**

1. Create a "master" configuration file template named "master_configuration.ini" with the following settings:
   - Common configuration for all devices:
     - ini file:

```
AutoUpdatePredefinedTime = '24:00'
CptFileURL = 'https://www.company.com/call_progress.dat'
AutoCmpFileUrl = 'https://www.company.com/sw.cmp'
```

     - CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# update-frequency 24:00
(automatic-update)# call-progress-tones
https://www.company.com/call_progress.dat
(automatic-update)# auto-firmware https://www.company.com/sw.cmp
```

   - Configuration per device based on MAC address:
     - ini file:

```
IniFileURL = 'http://www.company.com/config_<MAC>.ini'
```

- CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# cli-script
https://company.com/files/cli_script_<MAC>.txt
(automatic-update)# voice-configuration
http://www.company.com/config_<MAC>.ini
```

2. Copy the master configuration file that you created in Step 1 as well as the CPT and .cmp files to the HTTP-based provisioning server.
3. Configure **each** device with the following:
   a. URL of the master configuration file:
      - ini File:

```
IniFileURL =
'http://www.company.com/master_configuration.ini'
```

      - CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# voice-configuration
http://www.company.com/master_configuration.ini
(automatic-update)# cli-script
https://company.com/files/master_startup.txt
```

   b. Configure the device with the IP address of the DNS server for resolving the domain name (e.g., http://www.company.com) that is used in the URL for the provisioning server. This is done in the Interface table:
      - ini File:

```
[ InterfaceTable ]
FORMAT InterfaceTable_Index =
InterfaceTable_ApplicationTypes,
InterfaceTable_InterfaceMode, InterfaceTable_IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable_VlanID, InterfaceTable_InterfaceName,
InterfaceTable_PrimaryDNSServerIPAddress,
InterfaceTable_SecondaryDNSServerIPAddress,
InterfaceTable_UnderlyingDevice;
InterfaceTable 0 = 6, 10, 10.15.7.95, 16, 10.15.0.1, 1,
"Voice", 80.179.52.100, 0.0.0.0, "vlan 1";
[ \InterfaceTable ]
```

      - CLI:

```
# configure voip
(config-voip)# interface network-if 0
(network-if-0)# primary-dns 80.179.52.100
```

4. Power down and then power up the device.

# 31 Restoring Factory Defaults

This section describes the different ways that you can restore the device's configuration to factory defaults.

## 31.1 Restoring Factory Defaults through CLI

You can restore the device to factory defaults through CLI, as described in the following procedure.

➢ **To restore factory defaults through CLI:**

1. Access the CLI:
   a. Connect the RS-232 serial port of the device to the communication port on your computer. For serial cabling, refer to the Hardware Installation Manual.
   b. Establish serial communication with the device using a serial communication program (such as HyperTerminalTM) with the following communication port settings:
      - Baud Rate: 115,200 bps
      - Data Bits: 8
      - Parity: None
      - Stop Bits: 1
      - Flow Control: None
2. At the CLI prompt, type the username (default is "Admin" - case sensitive), and then press Enter:
   ```
   # Username: Admin
   ```
3. At the prompt, type the password (default is "Admin" - case sensitive), and then press Enter:
   ```
   # Password: Admin
   ```
4. At the prompt, type the following, and then press Enter:
   ```
   # enable
   ```
5. At the prompt, type the password again, and then press Enter:
   ```
   # Password: Admin
   ```
6. At the prompt, type the following to reset the device to default settings, and then press Enter:
   ```
   # write factory
   ```

## 31.2 Restoring Factory Defaults through Web Interface

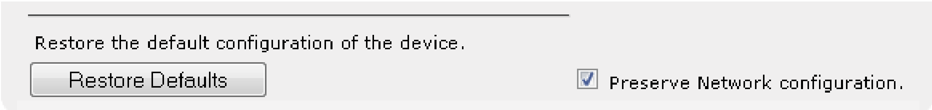
You can restore the device to factory defaults through the Web interface.

> **Note:** When restoring to factory defaults, you can preserve your IP network settings that are configured in the Interface table (see "Configuring IP Network Interfaces" on page 135), as described in the procedure below. This may be important, for example, to maintain connectivity with the device (through the OAMP interface) after factory defaults have been applied.

➢ **To restore factory defaults through Web interface:**

1. Open the Configuration File page:
   - Toolbar: From the **Device Actions** drop-down list, choose **Restore Defaults**
   - Navigation Tree: **Maintenance** tab > **Software Update** > **Configuration File**

**Figure 31-1: Restoring Factory Defaults through Web**




2. To keep your current IP network settings, select the **Preserve Network Configuration** check box. To overwrite all your IP network settings with the default IP network interface, clear the **Preserve Network Configuration** check box.
3. Click the **Restore Defaults** button; a message appears requesting you to confirm.
4. Click **OK** to confirm or **Cancel** to return to the page.
5. Once the device is restored to factory defaults, reset the device for the settings to take effect.

## 31.3 Restoring Defaults using Hardware Reset Button

You can restore the device to factory defaults by pressing the device's hardware reset pinhole button.

➢ **To restore default settings using the hardware reset pinhole button:**

- With a paper clip or any other similar pointed object, press and hold down the reset pinhole button located on the CPU module (rear panel) for at least 10 seconds (but no more than 30 seconds).

## 31.4 Restoring Defaults through ini File

You can restore the device to factory defaults by loading an empty *ini* file to the device. This is done using the Web interface's Configuration File page (see "Backing Up and Loading Configuration File" on page 517). If the *ini* file does include content (e.g., parameters), ensure that they are on lines beginning with comment signs (i.e., semicolons ";") so that the device ignores them.

**Note:** The only settings that are not restored to default are the management (OAMP) LAN IP address and the Web interface's login user name and password.

# 32 Automatic Archiving of Configuration File

You can configure the device to automatically save a configuration file each time you modify the device's configuration. The archived file can be saved to a user-defined URL of a remote server (TFTP or HTTP/S), or to a USB storage device attached to the device. The device first saves the configuration to its flash memory and then sends the file to the remote server or USB. The configuration in the archived file is based only on CLI commands. Archiving configuration can be useful, for example, when you need to revert to a previously backed-up configuration (for whatever reason).

To configure configuration-file archiving, use the following CLI command (root level):

- **Archiving to a remote server:**

  ```
  # write-and-backup to <URL path with file name>
  ```

- Archiving to a USB:

  ```
  # write-and-backup to usb:///<file name>
  ```

**This page is intentionally left blank.**

# 33 USB Storage Capabilities

The device supports USB storage using an external USB hard drive or flash disk (disk on key) connected to its USB port. The storage capabilities are configured using the CLI and include the following:

- To save network captures to the USB:
```
# debug capture data physical stop usb
```
- To update the device's firmware from the USB:
```
# copy firmware from usb:///<cmp file name>
```
- To update the device's configuration from the USB:
```
# copy voice-configuration from usb:///<ini configuration file
name>
```
- To save the current configuration to the USB:
```
# copy voice-configuration to usb:///<ini configuration file
name>
```

**Note:** Only a single USB storage (formatted to FAT/FAT32) operation is supported at any given time.

**This page is intentionally left blank.**

# Part VII

## Status, Performance Monitoring and Reporting

# 34 System Status

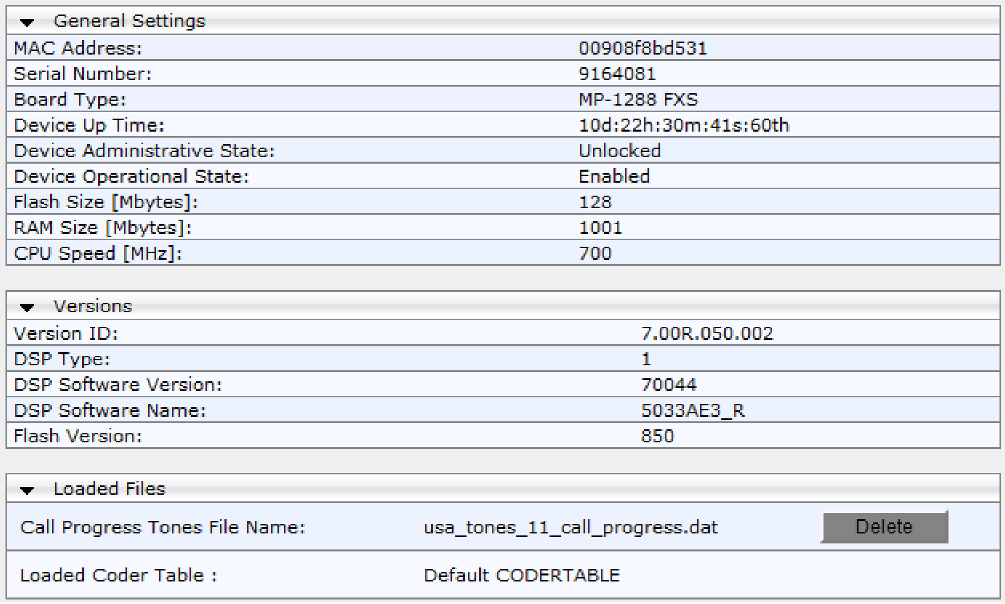
This section describes how to view various system statuses.

## 34.1 Viewing Device Information

The Device Information page displays hardware and software information about the device.

➢ **To access the Device Information page:**

- Open the Device Information page (**Status & Diagnostics** tab > **System Status** menu > **Device Information**).




**Device Information Description**

| Parameter | Description |
|---|---|
| **General Settings** | |
| **MAC Address** | Media access control (MAC) address. |
| **Serial Number** | Serial number of the CPU. This serial number also appears on the product label that is affixed to the chassis, as "CPU S/N". |
| **Product Key** | Product Key, which identifies the specific device purchase. The Product Key also appears on the product label that is affixed to the chassis, as "S/N(Product Key)". For more information, see Viewing the Device's Product Key on page 512. |
| **Board Type** | Product name of the device. |
| **Device Up Time** | Duration that the device has been up and running since the last reset. The duration is displayed in the following format: *dd:hh:mm:ss:100th of a second* |
| **Device Administrative State** | Administrative status ("Unlocked" or "Locked"), as performed in Locking and Unlocking the Device on page 489. |

| Parameter | Description |
|---|---|
| **Device Operational State** | Operational status:<br>▪ "Disabled"<br>▪ "Enabled"<br>▪ "Error"<br>▪ "Unknown" |
| **Flash Size [Mbytes]** | Size of the non-volatile storage memory (flash), measured in megabytes. |
| **RAM Size [Mbytes]** | Size of the random access memory (RAM), measured in megabytes. |
| **CPU Speed [MHz]** | Clock speed of the CPU, measured in megahertz (MHz). |
| **Versions** | |
| **Version ID** | Software version number. |
| **DSP Type** | Type of DSP. |
| **DSP Software Version** | DSP software version. |
| **DSP Software Name** | DSP software name. |
| **Flash Version** | Flash memory version number. |
| **Loaded Files:** Displays installed Auxiliary files. You can also delete a file, by clicking the corresponding **Delete** button, as described in Deleting Auxiliary Files on page 495. | |

## 34.2 Viewing Ethernet Port Information

The Ethernet Port Information page displays read-only information about the Ethernet Group connections.

➢ **To view Ethernet port information:**

- Open the Ethernet Port Information page:
  - Navigation menu tree: **Status & Diagnostics** tab > **System Status** menu > **Ethernet Port Info**
  - On the Home page, click any Ethernet port on the graphical display of the device (see "Viewing the Home Page" on page 63)

| | Port Name | Active | Speed | Duplex Mode | State | Group Member |
|---|---|---|---|---|---|---|
| 1 | GE_1 | Yes | 100 Mbps | Full Duplex | Forwarding | GROUP_1 |
| 2 | GE_2 | No | 10 Mbps | Half Duplex | Disabled | GROUP_1 |

**Table 34-1: Ethernet Port Information Parameters**

| Parameter | Description |
|---|---|
| Port Name | Displays the name of the port. |
| Active | Displays whether the port is active ("Yes") or not ("No"). |
| Speed | Displays the speed (in Mbps) of the Ethernet port. |
| Duplex Mode | Displays whether the port is half- or full-duplex. |
| State | Displays the state of the port:<br>▪ "Forwarding": Active port (data is being received and sent)<br>▪ "Disabled": Redundancy port |
| Group Member | Displays the port-pair group ID to which the port belongs. |

## 34.3 Viewing Hardware Components Status

The Components Status page provides read-only, real-time status of the device's chassis components such as slot occupants, fans, and power supply units.

> **Note:** You can also access this page from the Home page (see "Viewing the Home Page" on page 63).

➢ **To view the status of the device's hardware components:**

- Open the Components Status table, by doing one of the following:
  - Navigation tree: **Status & Diagnostics** tab > **System Status** menu > **Components Status**.
  - Home page: Click a power supply or fan tray icon**.**

**Figure 34-1: Components Status Page**

| CPU | |
|---|---|
| SW version | 7.00R.050.002 |
| CPLD version | 18 |
| LAN | Copper |

| Power Supply | |
|---|---|
| PS1 | No Alarm |
| PS2 | Failure |

| Fan Tray | |
|---|---|
| Fan | No Alarm |

| Slots | |
|---|---|
| Slot #1 | 72 FXS Ports; No Alarm; CPLD Version 19; |
| Slot #2 | 72 FXS Ports; No Alarm; CPLD Version 19; |
| Slot #3 | 72 FXS Ports; No Alarm; CPLD Version 19; |
| Slot #4 | 72 FXS Ports; No Alarm; CPLD Version 19; |

## 34.4 Reporting DSP Utilization through SNMP MIB

You can obtain information on the percentage of DSP resources utilized by the device, through the SNMP MIB table, acPMDSPUsage. You can also configure low and high DSP utilization thresholds for this MIB, that if crossed, the SNMP trap event, acPerformanceMonitoringThresholdCrossing is sent by the device. For more information on this MIB, refer to the *SNMP Reference Guide*.

**This page is intentionally left blank.**

# 35 Carrier-Grade Alarms

This section describes how to view SNMP alarms raised by the device.

## 35.1 Viewing Active Alarms

The Active Alarms table displays a list of currently active alarms that have been raised by the device. Once an alarm has been resolved (cleared), the device moves it into the History Alarms table (see "Viewing History Alarms" on page 551). For detailed information on SNMP alarms, refer to the *SNMP Reference Guide* document.

**Note:**

- The alarms in the table are deleted upon a device reset.
- To configure the maximum number of active alarms that can be displayed in the table, see the ini file parameter, ActiveAlarmTableMaxSize.
- For more information on SNMP alarms, refer to the *SNMP Reference Guide* document.

➢ **To view active alarms:**

- Open the Active Alarms table (**Status & Diagnostics** tab > **System Status** menu > **Carrier-Grade Alarms** > **Active Alarms).** You can also access the table from the Home page (see "Viewing the Home Page" on page 63).

| Sequential number | Severity | Source | Description | Date |
|---|---|---|---|---|
| 3 | Minor | Board#1/EthernetLink#2 | Ethernet link alarm. LAN port number 2 is down. | 2.3.2010 , 03:29:51 |

For each alarm, the following information is provided:

- **Sequential Number:** number of the alarm (sequential numbering of each alarm)
- **Severity:** severity level of the alarm:
  - Critical (red)
  - Major (orange)
  - Minor (yellow)
- **Source:** device component from which the alarm was raised
- **Description:** brief explanation of the reason of the alarm
- **Date:** date and time that the alarm was generated

You can view the next 20 alarms (if exist), by clicking the **Go to page** button.

## 35.2 Viewing History Alarms

The Alarms History table displays a list of alarms that have been cleared (resolved). You can configure the maximum number of alarms displayed in the table, using the AlarmHistoryTableMaxSize ini file parameter. If the maximum is reached and a new alarm is added to the table, the oldest alarm is removed from the table to accommodate the new alarm.

➢ **To view history alarms:**

- Open the Alarms History table (**Status & Diagnostics** tab > **System Status** menu > **Carrier-Grade Alarms** > **Alarms History**).

| Sequential number | Severity | Source | Description | Date |
|---|---|---|---|---|
| 1 | Major | Board#1 | Controller failure alarm Proxy Set 0: Proxy lost. looking for another proxy | 6.1.2010 , 14:1:26 |
| 2 | cleared | Board#1 | Alarm cleared: Controller failure alarm Proxy Set 0: Proxy lost. looking for another proxy | 6.1.2010 , 14:1:26 |
| 3 | Major | Board#1 | Controller failure alarm Proxy Set ID 0 | 6.1.2010 , 14:1:26 |
| 4 | Major | Board#1/WanLink#1 | WAN link alarm. FE interface 1 is down. | 6.1.2010 , 14:1:29 |
| 5 | Minor | Board#1/EthernetLink#2 | Ethernet link alarm. LAN port number 2 is down. | 6.1.2010 , 14:1:29 |
| 6 | Major | Board#1 | NTP server alarm. No connection to NTP server. | 6.1.2010 , 14:11:14 |

For each alarm, the following information is provided:

- **Severity:** severity level of the alarm:
  - Critical (red)
  - Major (range)
  - Minor (yellow)
  - Cleared (green)
- **Source:** unit from which the alarm was raised
- **Description:** brief explanation of the alarm
- **Date:** date and time that the alarm was generated

To view the next 20 alarms (if exist), click the **Go to page** button.

➢ **To delete all the alarms in the table:**

1. Click the **Delete History Table** button; a confirmation message box appears.
2. Click **OK** to confirm.

**This page is intentionally left blank.**

# 36 Performance Monitoring

This section describes how to view performance monitoring in the device's Web interface.

## 36.1 Viewing MOS per Media Realm

The MOS Per Media Realm page displays statistics on Media Realms (configured in "Configuring Media Realms" on page 303). This page provides two graphs:

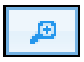
- Upper graph: displays the Mean Opinion Score (MOS) quality in RTCP data per selected Media Realm.
- Lower graph: displays the bandwidth of transmitted media (in Kbps) in RTCP data per Media Realm.

#### ➢ To view the MOS per Media Realm graph:

1. Open the MOS Per Media Realm page (**Status & Diagnostics** tab > **Performance Monitoring** menu > **MOS Per Media Realm**).

**Figure 36-1: MOS Per Media Realm Graph**


2. From the 'Media Realm' drop-down list, select the Media Realm for which you want to view.

Use the **Zoom In**  button to increase the displayed time resolution or the **Zoom Out**  button to decrease it. Instead of using these zoom buttons, you can use the slide ruler. As you increase the resolution, more data is displayed on the graph. The minimum resolution is about 30 seconds; the maximum resolution is about an hour.

To pause the graph, click the **Pause** button; click **Play** to resume.

# 37 VoIP Status

This section describes how to view VoIP status and statistics.

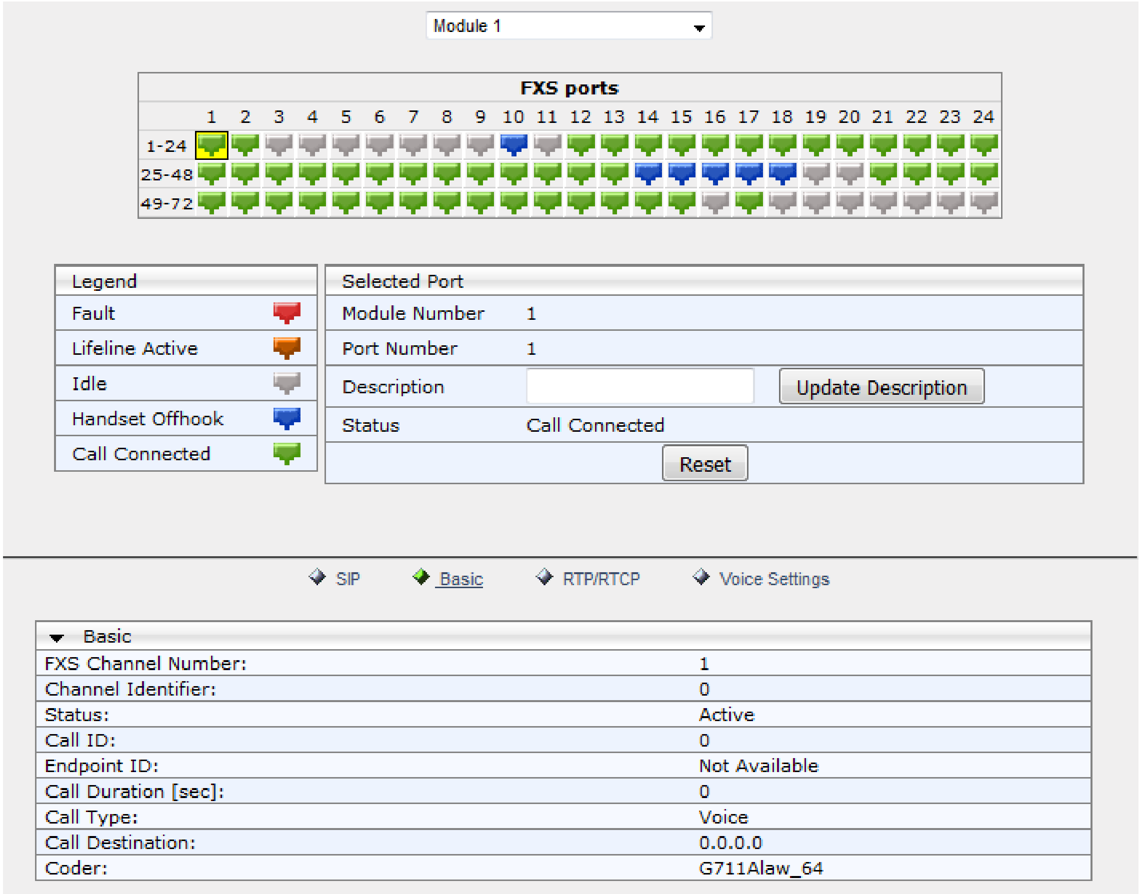
## 37.1 Viewing Port Status

You can view the status of FXS ports and channels of a specific FXS blade.

➢ **To view port and channel status:**

1. Open the Port Status page, by doing one of the following:
   - **Navigation tree: Status & Diagnostics** tab > **VoIP Status** menu > **Port Status**.
   - **Home page:** Click an FXS blade (see Viewing Device Status on Monitor Page).
2. From the drop-down list, located on the top of the page, select the required FXS blade; the blade's FXS ports are displayed:

**Figure 37-1: Viewing FXS Port Status**




Each row of ports depicts a specific 50-pin FXS connector on the specific FXS blade (rear panel), where the top row is the right-most connector (channels 1-24), the middle row the middle connector (channels 25-48) and the bottom row the left-most connector (channels 49-72).

3. Port status is indicated by the color of the port icons:
   - **Green:** Call is connected.
   - **Gray:** Idle port (telephone handset is in on-hook state).

- **Blue:** Telephone handset is in off-hook state.
- **Orange:** Port has been activated for Lifeline. The Lifeline telephone must be connected to Port 1 and the Lifeline interface (PSTN / PBX) to Port 25. Lifeline is activated upon a power outage.
- **Red:** Faulty port - hardware problem (such as SPI error or thermal shutdown) or due to DSP failure.

4. To view the status of a specific port, click its port icon. The Selected Port table displays the port's number in the 'Port Number' field and the port's status in the 'Status' read-only field, which corresponds to the port icon's color state (see Step 3 for descriptions). In addition, various information is displayed at the bottom of the page:

**Port Status Description**

| Tab | Field | Description |
|---|---|---|
| **SIP** | Endpoint Status | Status of endpoint:<br>▪ "IDLE": No call<br>▪ "ACTIVE": Active call |
| | Assigned Phone Number | Phone number of the port. |
| | Hunt Group | Hunt Group to which the port is assigned. To configure Hunt Groups, see |
| | MWI Information | Indicates if the phone has a voice-mail message (Message Waiting Indicator):<br>▪ "yes": Endpoint has a voice-mail message.<br>▪ "...": No voice mail for the endpoint. |
| | Call ID | Call ID number of the call (SIP Call-ID header). |
| | Call Originator | Caller:<br>▪ "TEL": Call made from Tel side (i.e., the port)<br>▪ "IP": Call made from IP side |
| | Source Tel Number | Telephone number of the caller. |
| | Destination Tel Number | Telephone number of the called party. |
| | Redirect Calling Number | Telephone number of the redirected number. |
| | Remote Signaling IP | IP address used for SIP on the IP side. |
| | Remote RTP (IP:Port) | IP address and port used for RTP on the IP side. |
| | Call Establishment Duration | Length of time (in seconds) it took to establish the call. |
| | Call Duration | Call duration (in seconds) from when call was established. |
| | Call State | Current state of the call:<br>▪ "IDLE": No call.<br>▪ "SETUP": Signaling to setup call (SIP INVITE).<br>▪ "ALERT": Ringing at remote end (SIP 180 Ringing).<br>▪ "RINGBACK": Ringback played to port.<br>▪ "SESSION": Call has been answered and is established. |

| Tab | Field | Description |
|---|---|---|
| | | ▪ "RELEASE": Call has been terminated (SIP 200 OK). |
| | Fax State | Currently, not in use. |
| | Coder + PTime | Coder and packetization time used for the call. |
| | Call Type | Type of call:<br>▪ "Voice": Voice call<br>▪ "Fax": Fax call |
| | Call Establishment Method | Mode of call:<br>▪ "Normal": No early media during SIP session establishment (before call accepted).<br>▪ "EarlyMedia": Media sent (e.g., announcements) before call accepted by called party. |
| | DTMF Selected Method for Tx/Rx | DTMF Transport method used for the call. For configuring the transport method, see Configuring DTMF Transport Types. |
| | Status | Status of port:<br>▪ "Inactive": No call<br>▪ "Active": Active call |
| | Call ID | See above. |
| | Endpoint ID | ID of endpoint:<br>▪ "Not Available" |
| | Call Duration | Call duration (in seconds) from when call was established. |
| | Call Type | Type of call:<br>▪ "Voice": Voice call<br>▪ "Fax": Fax call |
| | Call Destination | IP address of called party. |
| | Coder | Coder type used for the call. |
| **RTP/RTCP** | Channel Identifier | Channel identifier number. |
| | RTP Direction | Direction of RTP:<br>▪ "Tx & Rx": both directions (transmit and receive) |
| | Local UDP Port | Local UDP port on the device. |
| | Remote IP Addr | IP address of the remote IP side. |
| | Remote UDP Port | Port of the remote IP side. |
| | Rx Octet Count | Total number of received packets. |
| | Tx Octet Count | Total number of transmitted packets. |
| | Network Jitter | Network jitter (in msec). |

| Tab | Field | Description |
|---|---|---|
| | Roundtrip Delay | Round-trip delay time (in msec). |
| | Packet Loss | Packet loss (in %). |
| | Remote RTCP CName | |
| **Voice Settings** | Channel Identifier | Channel ID. |
| | Coder | See Configuring Coder Groups. |
| | Frame Duration | Frame duration (in msec). |
| | Echo Canceller | See Configuring Echo Cancellation. |
| | Silence Suppression | See Configuring Silence Suppression. |
| | Input Gain | See Configuring Voice Gain (Volume) Control. |
| | Voice Volume | See Configuring Voice Gain (Volume) Control. |
| | DTMF Transport Type | See Configuring DTMF Transport Types. |
| | Enabled Detectors | Detectors (e.g., AMD). |
| | Fax Transport Type | See Fax and Modem Capabilities. |

## 37.2 Viewing Active IP Interfaces

The IP Interface Status page displays the device's active IP interfaces that are listed in the Interface table (see "Configuring IP Network Interfaces" on page 135).

➢ **To view active IP network interfaces:**

- Open the IP Interface Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **IP Interface Status**).

| Index | Application Type | IP Address | Interface Mode | Prefix Length | Default Gateway | Interface Name | Primary DNS Server IP Address | Secondary DNS Server IP Address | Underlying Device | Address State |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | O+M+C | 10.8.128.88 | IPv4 Manual | 16 | 10.8.0.1 | O+M+C | 0.0.0.0 | 0.0.0.0 | vlan 1 | Permanent |
| NA | Internal | 169.253.254.254 | IPv4 Manual | 16 | 0.0.0.0 | InternalIf 2 | 0.0.0.0 | 0.0.0.0 | InternalIf 2 | Permanent |

## 37.3 Viewing Ethernet Device Status

The Ethernet Device Status page displays the configured Ethernet Devices that have been successfully applied to the device. For configuring Ethernet Devices, see "Configuring Underlying Ethernet Devices" on page 132.

➢ **To view the configured and applied Ethernet Devices:**

- Open the Ethernet Device Status page (**Status & Diagnostics** tab > **VoIP Status** menu >**Ethernet Device Status Table**).

| Index | VLAN ID | Underlying Interface | Name |
|---|---|---|---|
| 0 | 1 | GROUP_1 | vlan 1 |
| 1 | 400 | GROUP_1 | vlan 4 |

## 37.4 Viewing Static Routes Status

The IP Routing Status table displays the status of the static routes. These are routes configured in the Static Route table (see "Configuring Static IP Routing" on page 143) and routes through the Default Gateway.

The status of the static routes can be one of the following:

- "Active": Static route is used by the device.
- "Inactive": Static route is not used. When the destination IP address is not on the same segment with the next hop, or the interface does not exist, the route state changes to "Inactive".

➢ **To view the status of static IP routing:**

- Open the IP Routing Status table (**Status & Diagnostics** tab > **VoIP Status** menu >**Static Route Status**).

**Figure 37-2: IP Routing Status Table Page**

| Index | Destination IP Address | Prefix Length | Gateway IP Address | Metric | Device Name | Status | Description |
|---|---|---|---|---|---|---|---|
| NA | 169.254.254.252 | 30 | 0.0.0.0 | 0 | InternalIF 1 | Active | |
| NA | 10.8.0.0 | 16 | 0.0.0.0 | 0 | vlan 1 | Active | |
| NA | 0.0.0.0 | 0 | 10.8.0.1 | 1 | vlan 1 | Active | |
| NA | 0.0.0.0 | 0 | 169.254.254.253 | 2 | InternalIF 1 | Active | |
| 0 | 10.37.5.5 | 16 | 10.8.0.1 | 1 | Unknown | Inactive | |

## 37.5 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 37-3: 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.

## 37.6 Viewing CDR History

The CDR History table displays historical Call Detail Record (CDR) information of Gateway calls. CDR history information is stored on the device's memory. The CDR History table can contain up to 4,096 CDRs. When a new CDR is generated, the device adds it to the top of the table and all previous entries are shifted one down in the table. If the table has reached maximum capacity of entries and a new CDR is added, the last CDR entry is removed from the table.

**Note:** If the device is reset, all CDR history information is deleted from memory and subsequently, the CDR History table appears empty.

The following procedure describes how to view CDR history in the Web interface. You can also view CDR history using the following CLI commands:

- All CDR history:

```
# show voip calls history
```

- CDR history for a specific SIP session ID:

```
# show voip calls history <session ID>
```

➢ **To view CDR history:**

- Open the CDR History page (**Status & Diagnostics** tab > **VoIP Status** menu > **CDR**

**History**).

**Figure 37-4: CDR History Table**

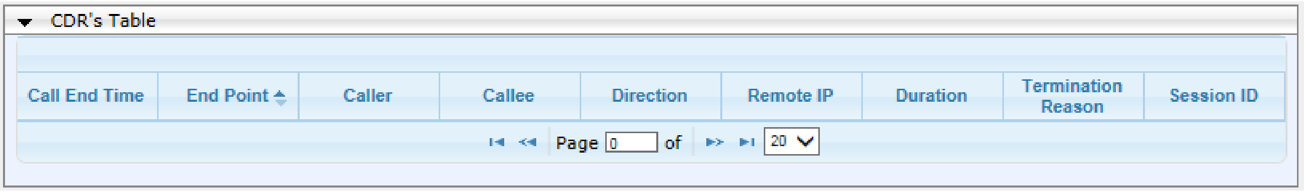



**Table 37-1: CDR History Table**

| Field | Description |
|---|---|
| **Call End Time** | Displays the time at which the call ended. The time is displayed in the format, hh:mm:ss, where *hh* is the hour, *mm* the minutes and *ss* the seconds (e.g., 15:06:36). |
| **End Point** | Displays the device's endpoint involved in the call, displayed in the format:<br>▪ Analog: <interface>-<module>/<port>. For example, "FXS-3/1" denotes FXS module 3, port 1. |
| **Caller** | Displays the phone number (source number) of the party who made the call. |
| **Callee** | Displays the phone number (destination number) of the party to whom the call was made. |
| **Direction** | Displays the direction of the call with regards to IP and Tel sides:<br>▪ "Incoming": IP-to-Tel call<br>▪ "Outgoing": Tel-to-IP call |
| **Remote IP** | Displays the IP address of the call party. For an "Incoming" call, this is the source IP address; for an "Outgoing" call, this is the destination IP address. |
| **Duration** | Displays the duration of the call, displayed in the format hh:mm:ss, where *hh* is hours, *mm* minutes and *ss* seconds. For example, 00:01:20 denotes 1 minute and 20 seconds. |
| **Termination Reason** | Displays the reason for the call being released (ended). For example, "NORMAL_CALL_CLEAR" indicates a normal off-hook (hang up) of the call party. |
| **Session ID** | Displays the SIP session ID of the call. |

## 37.7 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 37-5: 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 37-2: 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:<br>▪ GWAPP_REASON_NOT_RELEVANT (0)<br>▪ GWAPP_NORMAL_CALL_CLEAR (16)<br>▪ GWAPP_NORMAL_UNSPECIFIED (31)<br>And the internal reasons:<br>▪ RELEASE_BECAUSE_UNKNOWN_REASON<br>▪ RELEASE_BECAUSE_REMOTE_CANCEL_CALL<br>▪ RELEASE_BECAUSE_MANUAL_DISC<br>▪ RELEASE_BECAUSE_SILENCE_DISC<br>▪ RELEASE_BECAUSE_DISCONNECT_CODE<br>**Note:** When the duration of the call is zero, the release reason GWAPP_NORMAL_CALL_CLEAR increments the 'Number of Failed |

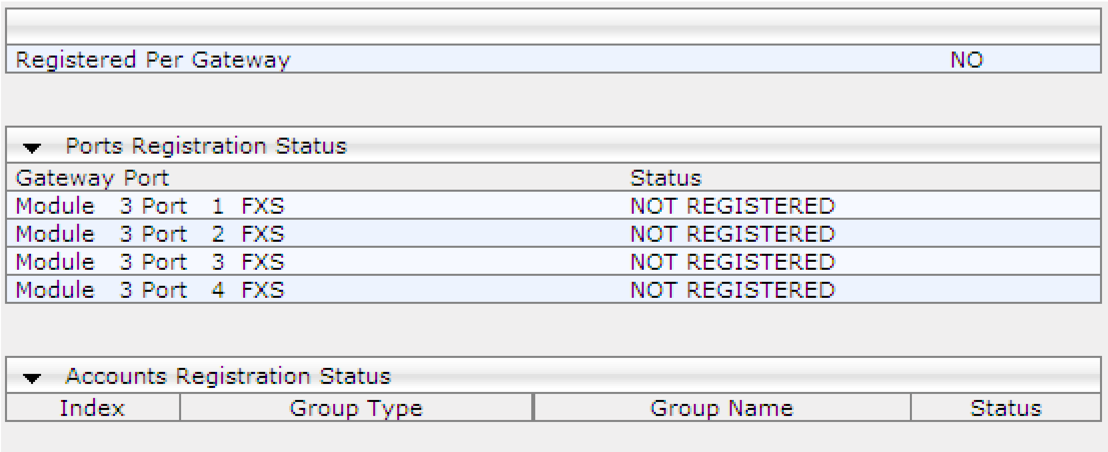
| Counter | Description |
|---|---|
| | Calls due to No Answer' counter. The rest of the release reasons increment the 'Number of Failed Calls due to Other Failures' counter. |
| **Percentage of Successful Calls (ASR)** | The percentage of established calls from attempted calls. |
| **Number of Calls 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:<br>▪ GWAPP_NO_USER_RESPONDING (18)<br>▪ GWAPP_NO_ANSWER_FROM_USER_ALERTED (19)<br>▪ 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:<br>▪ GWAPP_UNASSIGNED_NUMBER (1)<br>▪ GWAPP_NO_ROUTE_TO_DESTINATION (3) |
| **Number of Failed Calls due to No Matched Capabilities** | Indicates the number of calls that failed due to mismatched device capabilities. It is incremented as a result of an internal identification of capability mismatch. This mismatch is reflected to CDR via the value of the parameter DefaultReleaseReason (default is GWAPP_NO_ROUTE_TO_DESTINATION (3)) or by the GWAPP_SERVICE_NOT_IMPLEMENTED_UNSPECIFIED (79) reason. |
| **Number of Failed Calls due to No Resources** | Indicates the number of calls that failed due to unavailable resources or a device lock. The counter is incremented as a result of one of the following release reasons:<br>▪ GWAPP_RESOURCE_UNAVAILABLE_UNSPECIFIED<br>▪ 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. |

## 37.8 Viewing Registration Status

The Registration Status page displays the registration status of the device's endpoints and SIP Accounts, which are configured in the Accounts table (see "Configuring Registration Accounts" on page 341).

➢ **To view registration status:**

- Open the Registration Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **Registration Status**).




- Registered Per Gateway (applicable only to the Gateway application): Registration of device as one entity - "YES" or "NO"
- Ports Registration Status: "REGISTERED" or "NOT REGISTERED"
- **Accounts Registration Status:**
  - **Group Type:** served Hunt Group or IP Group
  - **Group Name:** name of served Hunt Group or IP Group, if applicable
  - **Status:** "Registered" or "Unregistered"

**Note:** The registration mode (i.e., per device, endpoint, account. or no registration) is configured in the Hunt Group Settings table (see Configuring Hunt Group Settings on page 386) or using the TrunkGroupSettings ini file parameter.

## 37.9 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 37-6: 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 (--) | -- |

**Table 37-3: Call Routing Status Parameters**

| Parameter | Description |
|---|---|
| **Call-Routing Method** | ▪ Proxy/GK = Proxy server (Proxy Set) is used to route calls. For configuring Proxy Sets, see "Configuring Proxy Sets" on page 329.<br>▪ Routing Table = Calls are routed using the routing table:<br>✓ Tel-to-IP Routing table (Configuring Tel-to-IP Routing Rules on page 409) |
| **IP Address** | ▪ Not Used = Proxy server isn't defined.<br>▪ IP address and FQDN (if exists) of the Proxy server with which the device currently operates. |
| **State** | ▪ N/A = Proxy server isn't defined.<br>▪ OK = Communication with the Proxy server is in order.<br>▪ Fail = No response from any of the defined Proxies. |

## 37.10 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 table (see "Configuring Tel-to-IP Routing Rules" on page 409).

**Note:** he information in columns 'Quality Status' and 'Quality Info' (per IP address) is reset if two minutes elapse without a call to that destination.

➢ **To view IP connectivity information:**

1. In the Routing General Parameters page, set the 'Enable Alt Routing Tel to IP' parameter (AltRoutingTel2IPMode) to **Enable** or **Status Only** (see "Configuring General Routing Parameters" on page 409).
2. Open the IP Connectivity page (**Status & Diagnostics** tab > **VoIP Status** menu > **IP Connectivity**).

**Figure 37-7: 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 37-4: IP Connectivity Parameters**

| Column Name | Description |
|---|---|
| **IP Address** | The IP address can be one of the following:<br>▪ IP address defined as the destination IP address in the Tel-to-IP Routing table.<br>▪ IP address resolved from the host name defined as the destination IP address in the Tel-to-IP Routing table. |
| **Host Name** | Host name (or IP address) as configured in the Tel-to-IP Routing table. |
| **Connectivity Method** | The method according to which the destination IP address is queried periodically (SIP OPTIONS request). |
| **Connectivity Status** | The status of the IP address' connectivity according to the method in the 'Connectivity Method' field.<br>▪ OK = Remote side responds to periodic connectivity queries.<br>▪ Lost = Remote side didn't respond for a short period.<br>▪ Fail = Remote side doesn't respond.<br>▪ Init = Connectivity queries not started (e.g., IP address not resolved). |

| Column Name | Description |
|---|---|
| | ▪ Disable = The connectivity option is disabled, i.e., parameter 'Alt Routing Tel to IP Mode' (AltRoutingTel2IPMode *ini*) is set to 'None' or 'QoS'. |
| **Quality Status** | Determines the QoS (according to packet loss and delay) of the IP address.<br>▪ Unknown = Recent quality information isn't available.<br>▪ OK<br>▪ Poor<br>**Notes:**<br>▪ The parameter is applicable only if the parameter 'Alt Routing Tel to IP Mode' is set to 'QoS' or 'Both' (AltRoutingTel2IPMode = 2 or 3).<br>▪ The 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.<br>**Notes:**<br>▪ The parameter is applicable only if the parameter 'Alt Routing Tel to IP Mode' is set to 'QoS' or 'Both' (AltRoutingTel2IPMode = 2 or 3).<br>▪ The parameter is reset if no QoS information is received for 2 minutes. |
| **DNS Status** | DNS status can be one of the following:<br>▪ DNS Disable<br>▪ DNS Resolved<br>▪ DNS Unresolved |

**This page is intentionally left blank.**

# 38 Reporting Information to External Party

This section describes features for reporting various information to an external party.

## 38.1 Configuring RTCP XR

RTP Control Protocol Extended Reports (RTCP XR) is a VoIP management control that defines a set of metrics containing information for assessing VoIP call quality and for diagnosing problems. RTCP XR (RFC 3611) extends the RTCP reports defined in RFC 3550 by providing additional VoIP metrics (Quality of Experience). RTCP XR information publishing is implemented in the device according to RFC 6035. This draft defines how a SIP User Agent (UA) publishes the detailed information to a defined collector. RTCP XR measures VoIP call quality such as packet loss, delay, signal / noise / echo levels, estimated R-factor, and mean opinion score (MOS). RTCP XR measures these parameters using metrics as listed in the table below.

**Notes:**

- The RTCP XR feature is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see "Software License Key" on page 510.
- If the RTCP XR feature is unavailable (not licensed or disabled), the R-factor VoIP metrics are not provided in CDRs (CDR fields, Local R Factor and Remote R Factor) generated by the device. Instead, these CDR fields are sent with the value 127, meaning that information is unavailable.

RTCP XR messages containing key call-quality-related metrics are exchanged periodically (user-defined) between the device and the SIP UA. This allows an analyzer to monitor these metrics midstream, or a device to retrieve them using SNMP.

You can configure the device to send RTCP XR to a specific 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 Group name to where the RTCP XR is sent
- Event: "vq-rtcpxr"
- Content-Type: "application/vq-rtcpxr"

You can configure the device to send RTXP XR at the end of each call and according to a user-defined interval between consecutive reports.

**Table 38-1: RTCP XR Published VoIP Metrics**

| Group | Metric Name |
|---|---|
| **General** | Start Timestamp |
| | Stop Timestamp |
| | Call-ID |
| | Local Address (IP, Port & SSRC) |
| | Remote Address (IP, Port & SSRC) |
| **Session Description** | Payload Type |

| Group | Metric Name |
|---|---|
| | 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 |
| | RCQ Est. Algorithm |
| | External R In |
| | Ext. R In Est. Algorithm |
| | External R Out |

| Group | Metric Name |
|---|---|
| | 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:

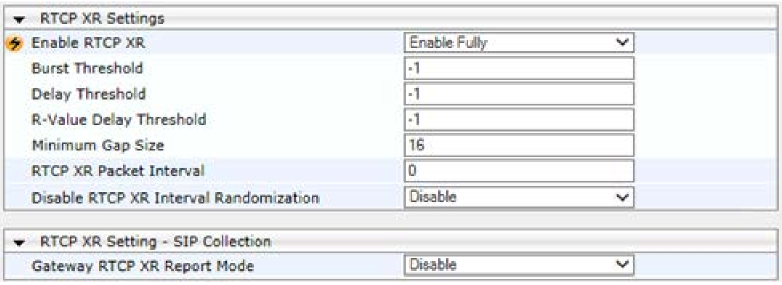
```
PUBLISH sip:172.17.116.201 SIP/2.0
Via: SIP/2.0/UDP 172.17.116.201:5060;branch=z9hG4bKac2055925925
Max-Forwards: 70
From: <sip:172.17.116.201>;tag=1c2055916574
To: <sip:172.17.116.201>
Call-ID: 2055916072161220152095 2@172.17.116.201
CSeq: 1 PUBLISH
Contact: <sip:172.17.116.201:5060>
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUB
SCRIBE,UPDATE
Event: vq-rtcpxr
Expires: 3600
User-Agent: device/<swver>
Content-Type: application/vq-rtcpxr
Content-Length: 1066
VQSessionReport
CallID=2032863474161220152094 3@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:

**Figure 38-1: RTCP XR Parameters in RTP/RTCP Settings Page**




**2.** Under the RTCP XR Settings group, configure the following:

- 'Enable RTCP XR' (*VQMonEnable*) - enables voice quality monitoring and RTCP XR.
- 'Burst Threshold' (*VQMonBurstHR*) - defines the voice quality monitoring excessive burst alert threshold.
- 'Delay Threshold' (*VQMonDelayTHR*) - defines the voice quality monitoring excessive delay alert threshold.
- 'R-Value Delay Threshold' (*VQMonEOCRValTHR*) - defines the voice quality monitoring end of call low quality alert threshold.
- 'Minimum Gap Size' (*VQMonGMin*) - defines the voice quality monitoring minimum gap size (number of frames).
- 'RTCP XR Packet Interval' (*RTCPInterval*) - defines the time interval between adjacent RTCP reports.
- 'Disable RTCP XR Interval Randomization' (*DisableRTCPRandomize*) - determines whether RTCP report intervals are randomized or whether each report interval accords exactly to the parameter RTCPInterval.

**3.** Under the RTCP XR Setting - SIP Collection group, configure the following:

- (Gateway Application Only) 'Gateway RTCP XR Report Mode' (RTCPXRReportMode) - enables RTCP XR reports and defines the interval at which they are sent.

**4.** Using the PublicationIPGroupID ini file parameter, define the IP Group to where you want to send the RTCP XR.

**5.** Click **Submit**, and then reset the device with a save ("burn") for your settings to take effect.

## 38.2 Generating Call Detail Records

The Call Detail Record (CDR) contains vital statistic information on calls made from the device. The device can be configured to generate and report CDRs for various stages of the call, including SIP messages and/or media. You can configure when CDRs for a call are generated, for example, only at the end of the call or only at the start and end of the call.

Once generated, the device can send the CDRs to any of the following:

- Syslog server. The CDR Syslog message complies with RFC 3164 and is identified by Facility 17 (local1) and Severity 6 (Informational).
- RADIUS server. For CDR in RADIUS format, see "Configuring RADIUS Accounting" on page 588. For configuring RADIUS servers for CDR reporting, see "Configuring RADIUS Servers" on page 224.

**Note:**

- You can view the latest CDRs, which are stored on the device's memory, in the CDR History table. For more information, see Viewing CDR History on page 562.
-

### 38.2.1 CDR Field Description

This section describes the default CDR fields that are generated by the device.

**Note:** You can customize the default CDR fields if desired. For customizing Gateway-related CDRs, see Customizing CDRs for Gateway Calls on page 581.

#### 38.2.1.1 CDR Fields for Gateway Application

The default CDR fields for the Gateway calls are listed in the table below.

**Table 38-2: Default CDR Fields for Gateway Calls**

| Field Name | Description |
|---|---|
| **GWReportType** | Report type:<br>▪ CALL_START<br>▪ CALL_CONNECT<br>▪ CALL_END |
| **Cid** | Port number |
| **SessionId** | SIP session identifier |
| **ConId** | SIP conference ID |
| **TG** | Hunt Group ID |
| **EPTyp** | Endpoint type:<br>▪ FXS |

| Field Name | Description |
|---|---|
| **Orig** | Call originator:<br>▪ LCL (Tel side)<br>▪ RMT (IP side) |
| **SourceIp** | Source IP address |
| **DestIp** | Destination IP address |
| **TON** | Source phone number type |
| **NPI** | Source phone number plan |
| **SrcPhoneNum** | Source phone number |
| **SrcNumBeforeMap** | Source number before manipulation |
| **TON** | Destination phone number type |
| **NPI** | Destination phone number plan |
| **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 for Gateway Application" on page 579) |
| **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<br>**Note:** To configure the time zone string (e.g., "UTC" - default, "GMT+1", and "EST"), use the TimeZoneFormat parameter. |
| **ConnectTime** | Call connect time<br>**Note:** To configure the time zone string (e.g., "UTC" - default, "GMT+1", and "EST"), use the TimeZoneFormat parameter. |
| **ReleaseTime** | Call release time<br>**Note:** To configure the time zone string (e.g., "UTC" - default, "GMT+1", and "EST"), use the TimeZoneFormat parameter. |
| **RTPdelay** | RTP delay |
| **RTPjitter** | RTP jitter |

| Field Name | Description |
|---|---|
| **RTPssrc** | Local RTP SSRC |
| **RemoteRTPssrc** | Remote RTP SSRC |
| **RedirectReason** | Redirect reason |
| **TON** | Redirection phone number type |
| **NPI** | Redirection phone number plan |
| **RedirectPhonNum** | Redirection phone number |
| **MeteringPulses** | Number of generated metering pulses |
| **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<br>Data is stored per call and sent in the syslog as follows:<br>▪ currency-type: amount multiplier for currency charge (euro or usd)<br>▪ recorded-units: for unit charge (1-999999) |
| **Mult** | 0,001-1000 (in steps of 10)<br>(See explanation above.) |
| **TrmReasonCategory** | Termination reason category:<br>▪ Calls with duration 0 (i.e., not connected):<br>✓ **NO_ANSWER -** GWAPP_NORMAL_CALL_CLEAR, GWAPP_NO_USER_RESPONDING, GWAPP_NO_ANSWER_FROM_USER_ALERTED<br>✓ **BUSY -** GWAPP_USER_BUSY<br>✓ **NO_RESOURCES -** GWAPP_RESOUUCE_UNAVAILABLE_UNSPECIFIED, RELEASE_BECAUSE_NO_CONFERENCE_RESOURCES_LEFT, RESOURCE_BECAUSE_NO_TRANSCODING_RESOURCES_LEFT, RELEASE_BECAUSE_GW_LOCKED<br>✓ **NO_MATCH -** RELEASE_BECAUSE_UNMATCHED_CAPABILITIES<br>✓ **FORWARDED -** RELEASE_BECAUSE_FORWARD<br>✓ **GENERAL_FAILED -** any other reason<br>▪ Calls with duration:<br>✓ **NORMAL_CALL_CLEAR -** GWAPP_NORMAL_CALL_CLEAR<br>✓ **ABNORMALLY_TERMINATED -** Anything else<br>▪ **N/A -** Reasons not belonging to above categories |
| **RedirectNumBeforeMap** | Redirect number before manipulation |

| Field Name | Description |
|---|---|
| **SrdId** | SRD ID name |
| **SIPInterfaceId** | SIP interface ID |
| **ProxySetId** | Proxy Set ID |
| **IpProfileId** | IP Profile name |
| **MediaRealmId** | Media Realm name |
| **SigTransportType** | SIP signaling transport type (UDP, TCP, or TLS) |
| **TxRTPIPDiffServ** | Media IP DiffServ |
| **TxSigIPDiffServ** | Signaling IP DiffServ |
| **LocalRFactor** | Local R-factor<br>**Note:** If the RTCP XR feature is unavailable (not licensed or disabled), this R-factor VoIP metric is not provided. Instead, the device sends the CDR field with the value 127, meaning that information is unavailable. |
| **RemoteRFactor** | Remote R-factor<br>**Note:** If the RTCP XR feature is unavailable (not licensed or disabled), this R-factor VoIP metric is not provided. Instead, the device sends the CDR field with the value 127, meaning that information is unavailable. |
| **LocalMosCQ** | Local MOS for conversation quality |
| **RemoteMosCQ** | Remote MOS for conversation quality |
| **SigSourcePort** | SIP source port |
| **SigDestPort** | SIP destination port |
| **MediaType** | Media type - audio, video, or text |
| AMD | Information relating to the Automatic Machine Detection (AMD) feature:<br>▪ V - voice<br>▪ A - answer machine<br>▪ S - silence<br>▪ U - unknown |
| % | Information relating to AMD that shows the success that the answering type (probability) was correctly detected |
| **SIPTrmReason** | SIP call termination reason (BYE, CANCEL, or SIP error codes, e.g., 404) |
| **SipTermDesc** | Description of SIP termination reason:<br>▪ SIP Reason header, if exists, for example: SIP ;cause=200 ;text="Call completed elsewhere".<br>▪ 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".<br>▪ 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). |

| Field Name | Description |
|---|---|
| **LatchedRtpIp** | Remote IP address of the incoming RTP stream that the device "latched" on to as a result of the RTP latching mechanism for NAT traversal. |
| **LatchedRtpPort** | Remote RTP port of the incoming RTP stream that the device "latched" on to as a result of the RTP latching mechanism for NAT traversal. |
| **LatchedT38Ip** | Latching of a new T.38 stream - new IP address |
| **LatchedT38Port** | Latching of a new T.38 stream - new port |

##### 38.2.1.1.1Release Reasons in CDR for Gateway Application

The possible reasons for call termination for the Gateway 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"

### 38.2.2 Customizing CDRs for Gateway Calls

The Gateway CDR Format table lets you configure CDR customization rules for Gateway-related CDRs that are sent in Syslog messages and/or RADIUS accounting request messages. The table lets you configure up to 128 CDR customization rules for Syslog CDRs and up to 40 rules for RADIUS-accounting CDRs. If you do not configure a CDR customization rule for a specific CDR, the device generates the CDR in a predefined default CDR format (see CDR Field Description on page 575).

For RADIUS accounting, you can customize CDRs for standard RADIUS Attributes and vendor-specific RADIUS Attributes (VSA). You can customize the RADIUS Attribute's prefix name (*Column Type*) and ID. For example, instead of the default VSA name, "h323-connect-time" with ID 28, you can change the name to "Call-Connect-Time" with ID 29.

**Notes:**

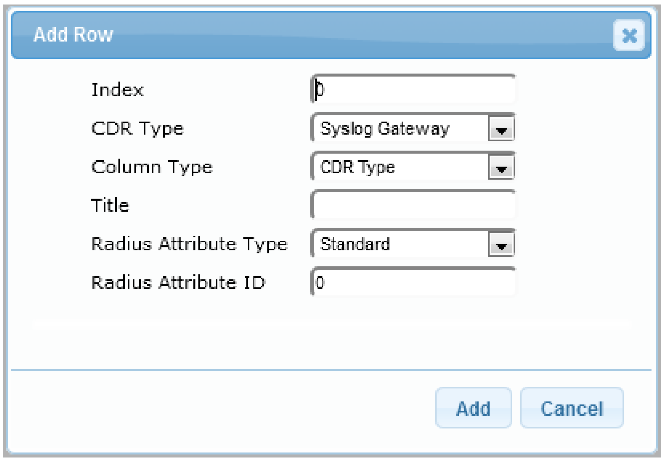
- The following standard RADIUS Attributes cannot be customized: 1 through 6, 18 through 20, 22, 23, 27 through 29, 32, 34 through 39, 41, 44, 52, 53, 55, 60 through 85, 88, 90, and 91.
- If the RTCP XR feature is unavailable (not licensed or disabled), the R-factor VoIP metrics are not provided in CDRs (CDR fields, Local R Factor and Remote R Factor) generated by the device. Instead, these CDR fields are sent with the value 127, meaning that information is unavailable.

The following procedure describes how to customize Gateway CDRs through the Web interface. You can also configure it through ini file (GWCDRFormat) or CLI (configure voip > services cdr > cdr-format gw-cdr-format).

➢ **To customize Gateway CDRs:**

1. Open the Gateway CDR Format table (**Configuration** tab > **System** menu > **Call Detail Record** > **Gateway CDR Format**).
2. Click **Add**; the following dialog box appears:

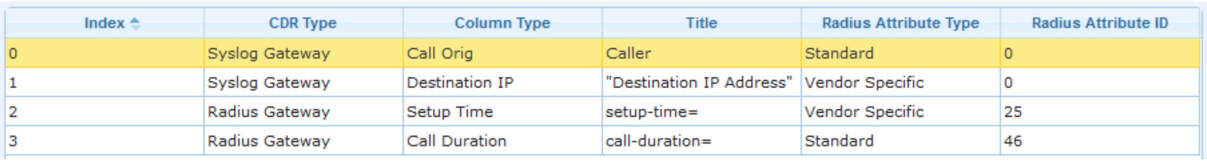
**Figure 38-2: Gateway CDR Format Table - Add Row Dialog Box**




3. Configure the CDR according to the parameters described in the table below.
4. Click **Add**.

An example of CDR customization rules configured in the table is shown below:

**Figure 38-3: Example of CDR Customization Rules for Gateway Calls**



| Index | CDR Type | Column Type | Title | Radius Attribute Type | Radius Attribute ID |
|---|---|---|---|---|---|
| 0 | Syslog Gateway | Call Orig | Caller | Standard | 0 |
| 1 | Syslog Gateway | Destination IP | "Destination IP Address" | Vendor Specific | 0 |
| 2 | Radius Gateway | Setup Time | setup-time= | Vendor Specific | 25 |
| 3 | Radius Gateway | Call Duration | call-duration= | Standard | 46 |

**Gateway CDR Format Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| Index<br>[GWCDRFormat_Index] | Defines an index number for the new table row.<br>**Note:** Each row must be configured with a unique index. |
| CDR Type<br>`cdr-type`<br>[GWCDRFormat_CDRType] | Defines the application type for which you want to customize CDRs.<br>▪ [0] Syslog Gateway = (Default) Customizes CDR field names for CDRs sent in Syslog messages.<br>▪ [1] RADIUS Gateway = Customizes CDR field names (RADIUS Attribute prefix names) for CDRs sent in RADIUS accounting requests. |
| Column Type<br>`col-type`<br>[GWCDRFormat_ColumnType] | Defines the CDR field (column) that you want to customize.<br>[0] CDR Type (**default**); [1] Call ID; [2] Session ID; [3] Report Type; [4] Media Type; [5] Accounting Status Type; [6] H323 ID; [7] Radius Call ID; [8] Blank; [10] Endpoint Type; [11] Call Orig; [12] Source IP; [13] Destination IP; [14] Remote IP; [15] Source Port; [16] Dest Port; [17] Remote Port; [18] Call Duration; [19] Termination Side; [20] Termination Reason; [21] Setup Time; [22] Connect Time; [23] Release Time; [24] Redirect Reason; [25] Was Call Started; [26] IP Group ID; [27] IP Group Name; [28] SRD ID; [29] SRD Name; [30] SIP Interface ID; [31] Transport Type; [32] Signaling IP DiffServ; [33] Termination Reason |

| Parameter | Description |
|---|---|
| | Category; [34] Proxy Set ID; [35] IP Profile ID; [36] IP Profile Name; [37] Media Realm ID; [38] Media Realm Name; [39] SIP Termination Reason; [40] SIP Termination Description; [41] Caller Display ID; [42] Callee Display ID; [43] SIPInterface Name; [44] Call Orig Radius; [45] Termination Side Radius; [46] Termination Side Yes No; [47] Termination Reason Value; [48] ProxySet Name; [100] Trunk ID; [101] B-Channel; [102] Conn ID; [103] Trunk Group ID; [104] Metering Pulses Generated; [105] Fax On Call; [106] Source Number Before Manipulation; [107] Source Number; [108] Source Number Type; [109] Source Number Plan; [110] Destination Number Before Manipulation; [111] Destination Number; [112] Destination Number Type; [113] Destination Number Plan; [114] Redirect Number Before Manipulation; [115] Redirect Number; [116] Source Host Name Before Manipulation; [117] Source Host Name; [118] Destination Host Name Before Manipulation; [119] Destination Host Name; [120] PSTN Termination Reason; [121] Module And Port; [122] AOC Currency; [123] AOC Amount; [124] AOC Multiplier; [125] ISDN Line Type; [150] Channel ID; [151] Coder Type;[152] Packet Interval; [153] Payload Type; [154] Local Input Packets; [155] Local Output Packets; [156] Local Input Octets; [157] Local Output Octets; [158] Local Packet Loss; [159] Local Round Trip Delay; [160] Local Jitter; [161] Local SSRC Sender; [162] Remote Input Packets; [163] Remote Output Packets; [164] Remote Input Octets; [165] Remote Output Octets; [166] Remote Packet Loss; [167] Remote Round Trip Delay; [168] Remote Jitter; [169] Remote SSRC Sender; [170] Local RTP IP; [171] Local RTP Port; [172] Remote RTP IP; [173] Remote RTP Port; [174] RTP IP DiffServ; [175] Local R Factor; [176] Remote R Factor; [177] Local MOS CQ; [178] Remote MOS CQ; [179] AMD Decision; [180] AMD Decision Probability; [181] Latched RTP IP; [182]Latched RTP Port; [183] Latched T38 IP; [184] Latched T38 Port. |
| Title<br>`title`<br>[GWCDRFormat_Title] | Defines a new name for the CDR field (for Syslog) or for the RADIUS Attribute prefix name (for RADIUS accounting) that you selected in the 'Column Type' parameter.<br>The valid value is a string of up to 31 characters.<br>You can configure the name to be enclosed by apostrophes (single or double). For example, if you want the CDR field name to appear as 'Phone Duration', you must configure the parameter to 'Phone Duration'. You can also configure the CDR field name with an equals (=) sign, for example "call-connect-time=".<br>**Notes:**<br>▪ For RADIUS Attributes that do not require a prefix name, leave the parameter undefined.<br>▪ The parameter's value is case-sensitive. For example, if you want the CDR field name to be Phone-Duration, you must configure the parameter to "Phone-Duration" (i.e., upper case "P" and "D"). |
| RADIUS Attribute Type<br>`radius-type`<br>[GWCDRFormat_RadiusType] | Defines whether the RADIUS Attribute of the CDR field is a standard or vendor-specific attribute.<br>▪ [0] Standard = (Default) For standard RADIUS Attributes.<br>▪ [1] Vendor Specific = For vendor-specific RADIUS Attributes (VSA).<br>**Note:** The parameter is applicable only for RADIUS accounting (i.e., 'CDR Type' parameter configured to **RADIUS Gateway**). |

| Parameter | Description |
|---|---|
| RADIUS Attribute ID<br>`radius-id`<br>[GWCDRFormat_RadiusID] | Defines an ID for the RADIUS Attribute. For vendor-specific Attributes, this represents the VSA ID; for standard attributes, this represents the Attribute ID (first byte of the Attribute).<br>The valid value is 0 to 255 (one byte). The default is 0.<br>**Notes:**<br>▪ The parameter is applicable only for RADIUS accounting (i.e., 'CDR Type' parameter configured to **RADIUS Gateway**).<br>▪ For VSA's (i.e., 'RADIUS Attribute Type' parameter configured to **Vendor Specific**), the parameter must be configured to any value other than 0.<br>▪ For standard RADIUS Attributes (i.e., 'RADIUS Attribute Type' parameter configured to **Standard**), the value **must** be a "known" RADIUS ID (per RFC for RADIUS). However, if you configure the ID to 0 (default) for any of the RADIUS Attributes (configured in the 'Column Type' parameter) listed below and then apply your rule (click **Add**), the device automatically replaces the value with the RADIUS Attribute's ID according to the RFC:<br>✓ **Destination Number:** 30<br>✓ **Source Number:** 31<br>✓ **Accounting Status Type:** 40<br>✓ **Local Input Octets:** 42<br>✓ **Local Output Octets:** 43<br>✓ **Call Duration:** 46<br>✓ **Local Input Packets:** 47<br>✓ **Local Output Packets:** 48<br>If you configure the value to 0 and the RADIUS Attribute is not any of the ones listed above, the configuration is invalid. |

### 38.2.3 Configuring CDR Reporting

The following procedure describes how to configure CDR reporting.

➢ **To configure CDR reporting:**

1. Enable the Syslog feature for sending log messages generated by the device to a collecting log message server. For more information, see "Enabling Syslog" on page 612.
2. Open the Call Detail Record Settings page (**Configuration** tab > **System** menu > **Call Detail Record** > **Call Detail Record Settings**).

**Figure 38-4: CDR Parameters in Call Detail Record Settings Page**

| CDR and Debug | |
|---|---|
| CDR Server IP Address | 10.8.6.55 |
| CDR Report Level | Start & End Call |
| Media CDR Report Level | End Media |
| CDR Syslog Sequence Number | Enable |

3. Configure the parameters as required. For a description of the parameters, see "Syslog, CDR and Debug Parameters" on page 670.
4. Click **Submit**.

> **Note:**
>
> - If the CDR server IP address is not configured, the CDRs are sent to the Syslog server configured in "Enabling Syslog" on page 612.
> - To configure the time zone string (e.g., GMT+1) that is displayed with the timestamp in CDRs ("Connect Time", "Release Time", and "Setup Time" CDR fields), use the TimeZoneFormat parameter.

### 38.2.4 Storing CDRs on the Device

The CDRs of Gateway calls generated by the device can also be stored locally on the device (RAM).

> **Note:** When the device is reset or powered off, stored CDRs are deleted.

You can specify the calls (configuration entities) for which you wish to create and store CDRs locally. This is done using Logging Filter rules in the Logging Filters table. For example, you can configure a rule to create CDRs for traffic belonging only to IP Group 2 and store the CDRs locally.

The CDRs are saved in a comma-separated values file (*.csv), where each CDR is shown on a dedicated row. An example of a CSV file with two CDRs are shown below:

- CSV file viewed in Excel:

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | 3b463e:215:1 | CALL_END | 4 | 14:34:40.000 UTC Wed Dec 16 2015 | 14:34:35.000 UTC Wed Dec 16 2015 | 14:34:33.000 UTC Wed Dec 16 2015 | RMT | GWAPP_NORMAL_ |
| 2 | 3b463e:215:1 | CALL_END | 4 | 14:34:40.000 UTC Wed Dec 16 2015 | 14:34:35.000 UTC Wed Dec 16 2015 | 14:34:33.000 UTC Wed Dec 16 2015 | LCL | GWAPP_NORMAL_ |
| 3 | | | | | | | | |

- CSV file viewed in a text editor (Notepad):

```
3b463e:215:1,CALL_END,4,14:34:40.000  UTC Wed Dec 16 2015,14:34:35.000  UTC Wed Dec 16 2015,14:34:33.000  UTC Wed Dec 16 2015,RMT,GWAPP_NORMAL_
3b463e:215:1,CALL_END,4,14:34:40.000  UTC Wed Dec 16 2015,14:34:35.000  UTC Wed Dec 16 2015,14:34:33.000  UTC Wed Dec 16 2015,LCL,GWAPP_NORMAL_
```

To view the CDR column headers corresponding to the CDR data in the CSV file, run the following CLI command:

- Gateway CDRs:

```
(config-system)# cdr
(cdr)# cdr-format show-title local-storage-gw
```

> **Note:** Each CDR can contain up to 1023 characters. If it contains more than this, the device removes the extra characters.

You can do the following with locally saved CDR files (*.csv), through the CLI (root menu):

- View stored CDR files:
  - View all stored CDR files:
    ```
    show storage-history
    ```
  - View all stored, unused CDR files:
    ```
    show storage-history unused
    ```
- Delete stored CDR files:
  - Delete all stored files:
    ```
    clear storage-history cdr-storage-history all
    ```
  - Delete all stored, unused CDR files:
    ```
    clear storage-history cdr-storage-history unused
    ```
- Save stored CDR files to an external destination:
  ```
  copy storage-history cdr-storage-history <filename> to
  <protocol://destination>
  ```
  Where:
  - *filename:* name you want to assign the file. Any file extension name can be used, but as the file content is in CSV format, it is recommended to use the .csv file extension.
  - *protocol:* protocol over which the file is sent (tftp, http, or https).

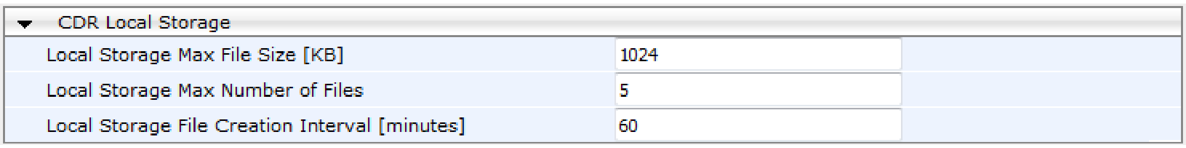
  For example:
  ```
  copy storage-history cdr-storage-history my_cdrs.csv to
  tftp://company.com/cdrs
  ```

The following procedure describes how to configure local CDR storage through the Web interface.

### ➢ To configure local CDR storage:

1. Open the Logging Settings page (**Configuration** tab > **System** menu > **Logging** > **Logging Settings**), and then scroll down to the Local Storage group:

**Figure 38-5: CDR Local Storage on Logging Settings Page**




2. Configure the following parameters:
   - 'Local Storage Max File Size' (CDRLocalMaxFileSize): Enter the maximum size (in kilobytes) of the CDR file. Once the file size is reached, the device creates a new file for subsequent CDRs, and so on.

- 'Local Storage Max Number of Files' (CDRLocalMaxNomOfFiles): Enter the maximum number of CDR files. Once the maximum is reached, a subsequent CDR file replaces the oldest created file.
- 'Local Storage File Creation Interval' (CDRLocalInterval): Enter the time in minutes for how often the device creates a new CDR file. For example, if configured to 60, it creates a new file every hour even if the maximum file size has not been reached.

For a detailed description of each parameter, see Syslog, CDR and Debug Parameters on page 670.

3. Open the Logging Filters table (**Configuration** tab > **System** menu > **Logging** > **Logging Filters Table**), and then configure a log filtering rule with the following parameter settings:
   - 'Filter Type' and 'Value': (as desired)
   - 'Log Destination': Local Storage
   - 'Log Type': CDR Only
   - 'Mode': Enable

   For more information on the Logging Filters table, see Configuring Log Filter Rules on page 597.

**Note:** If you have enabled the CDR storage feature and you later decide to change the maximum number of files (CDRLocalMaxNomOfFiles) to a lower value (e.g., from 50 to 10), the device stores the remaining files (e.g., 40) in its memory (i.e., unused files).

## 38.3 Configuring RADIUS Accounting

The device can send accounting data of SIP calls as call detail records (CDR) to a RADIUS Accounting server. CDR-based accounting messages can be sent upon call release, call connection and release, or call setup and release. For a list of the CDR attributes for RADIUS accounting, see the table following the procedure below.

RADIUS CDR attributes have the following format:

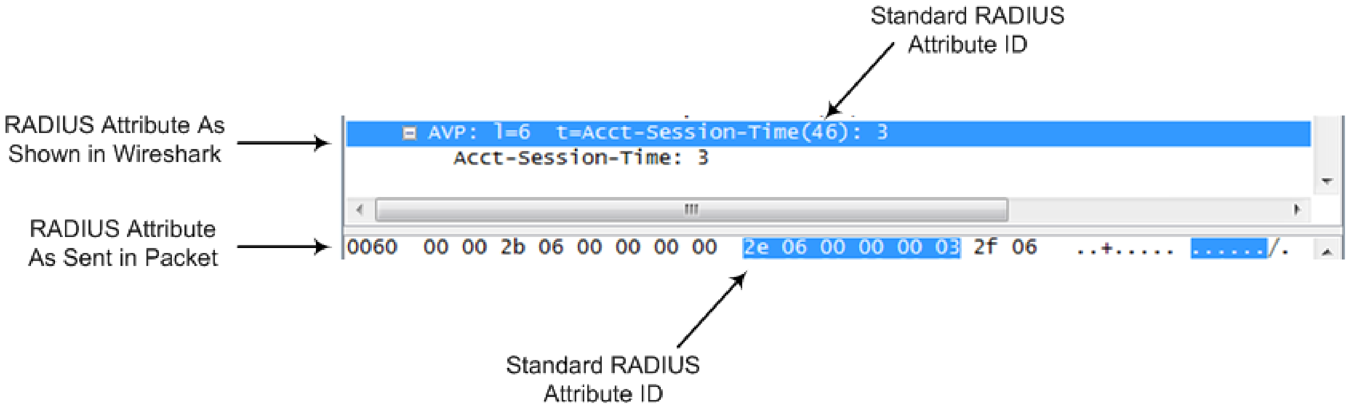
- **Standard RADIUS attributes (per RFC):** A typical standard RADIUS attribute is shown below. The RADIUS attribute ID depends on the attribute.

  **Figure 38-6: Typical Standard RADIUS Attribute**

```
2e 06 00 00 00 03 --- Data
|  |
|  Length (including header)
RADIUS ID
```

  The following figure shows a standard RADIUS attribute collected by Wireshark. The bottom pane shows the RADIUS attribute information as sent in the packet; the upper pane is Wireshark's interpretation of the RADIUS information in a more readable format. The example shows the attribute in numeric format (32-bit number in 4 bytes).

**Figure 38-7: Example of Standard RADIUS Attribute Collected by Wireshark**



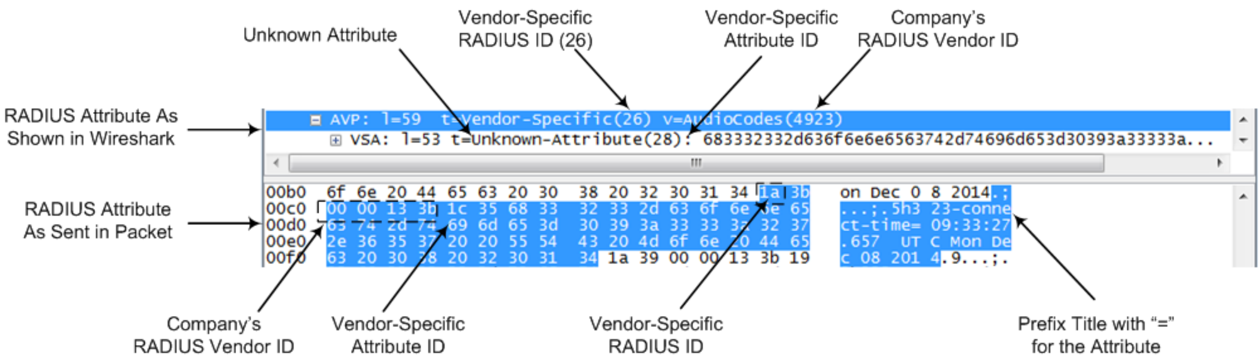

- **Vendor-specific RADIUS attributes:** RADIUS attributes that are specific to the device (company) are referred to as Vendor-specific attributes (VSA). The CDR of VSAs are sent with a general RADIUS ID of 26 to indicate that they are vendor-specific (non-standard). In addition, the company's registered vendor ID (as registered with the Internet Assigned Numbers Authority or IANA) is also included in the packet. The device's default vendor ID is 5003, which can be changed by using the RadiusVSAVendorID parameter. The VSA ID is also included in the packet.

**Figure 38-8: Example of a Vendor-Specific Attribute**

```
1a 13 00 00 13 8b 21 0d 68 33 32 33 2d 67 77 2d 69 64 3d --- Data
|  |  |  |  |  |  |  |
|  |  Vendor ID   |  Vendor part length
|  |  (5003)      Vendor-Specific Attribute (VSA) ID
|  Length (including header)
RADIUS ID indicating vendor-specific (26)
```

The following figure shows a vendor-specific RADIUS attribute collected by Wireshark. The bottom pane shows the RADIUS attribute information as sent in the packet; the upper pane is Wireshark's interpretation of the RADIUS information in a more readable format. The example shows the attribute in string-of-characters format.

**Figure 38-9: Example of Vendor-Specific RADIUS Attribute Collected by Wireshark**



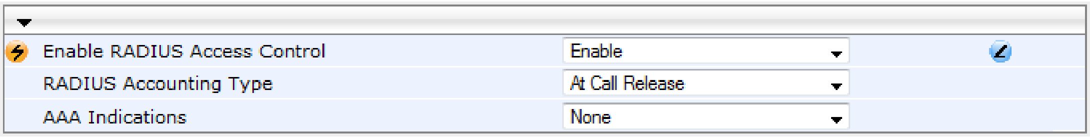
**Notes:**

- You can customize the prefix title of the RADIUS attribute name and the ID. For more information, see Customizing CDRs for Gateway Calls on page 581.
- To configure the address of the RADIUS Accounting server, see "Configuring RADIUS Servers" on page 224.

### ➢ To configure RADIUS accounting:

1. Open the Call Detail Record Settings page (**Configuration** tab > **VoIP** menu > **Services** > **Call Detailed Record** > **Call Detail Record Settings**).

**Figure 38-10: RADIUS Parameters**




2. Set the 'Enable RADIUS Access Control' parameter to **Enable**.
3. Configure the remaining parameters as required. For a description of these parameters, see "RADIUS Parameters" on page 814.
4. Click **Submit**.
5. For your settings to take effect, reset the device with a flash burn.

The table below lists the RADIUS Accounting CDR attributes included in the communication packets transmitted between the device and a RADIUS server.

**Table 38-3: Supported RADIUS Accounting CDR Attributes**

| Attribute ID | Attribute Name | Vendor-Specific Attribute (VSA) ID | Description | Value Format | Example | AAA |
|---|---|---|---|---|---|---|
| **Request Attributes** | | | | | | |
| 1 | user-name | (Standard) | Account number or calling party number or blank | String up to 15 digits long | 5421385747 | Start Acc Stop Acc |
| 4 | nas-ip-address | (Standard) | IP address of the requesting device | Numeric | 192.168.14.43 | Start Acc Stop Acc |
| 6 | service-type | (Standard) | Type of service requested | Numeric | 1: login | Start Acc Stop Acc |
| 26 | h323-incoming-conf-id | 1 | SIP call identifier | Up to 32 octets | h323-incoming-conf-id=38393530 | Start Acc Stop Acc |
| 26 | h323-remote-address | 23 | IP address of the remote gateway | Numeric | - | Stop Acc |

| Attribute ID | Attribute Name | Vendor-Specific Attribute (VSA) ID | Description | Value Format | Example | AAA |
|---|---|---|---|---|---|---|
| 26 | h323-conf-id | 24 | H.323/SIP call identifier | Up to 32 octets | | Start Acc Stop Acc |
| 26 | h323-setup-time | 25 | Setup time in NTP format 1 | String | h323-setup-time=09:33:26.621 Mon Dec 2014 | Start Acc Stop Acc |
| 26 | h323-call-origin | 26 | Originator of call:<br>▪ "answer": Call originated from the IP side<br>▪ "originate": Call originated from the Tel side | String | h323-call-origin=answer | Start Acc Stop Acc |
| 26 | h323-call-type | 27 | Protocol type or family used on this leg of the call | String | h323-call-type=VOIP | Start Acc Stop Acc |
| 26 | h323-connect-time | 28 | Connect time in NTP format | String | h323-connect-time=09:33:37.657 UTC Mon Dec 08 2015 | Stop Acc |
| 26 | h323-disconnect-time | 29 | Disconnect time in NTP format | String | - | Stop Acc |
| 26 | h323-disconnect-cause | 30 | Disconnect cause code (Q.850) | Numeric | h323-disconnect-cause=16 | Stop Acc |
| 26 | h323-gw-id | 33 | Name of the gateway | String | h323-gw-id=<SIP ID string> | Start Acc Stop Acc |
| 26 | sip-call-id | 34 | SIP Call ID | String | sip-call-id=abcde@ac.com | Start Acc Stop Acc |
| 26 | call-terminator | 35 | Terminator of the call:<br>▪ "yes": Call terminated by the Tel side<br>▪ "no": Call terminated by the IP side | String | call-terminator=yes | Stop Acc |

| Attribute ID | Attribute Name | Vendor-Specific Attribute (VSA) ID | Description | Value Format | Example | AAA |
|---|---|---|---|---|---|---|
| 26 | terminator | 37 | Terminator of the call:<br>▪ "answer": Call originated from the IP side<br>▪ "originate": Call originated from the Tel side | String | terminator=originate | Stop Acc |
| 30 | called-station-id | (Standard) | Destination phone number | String | 8004567145 | Start Acc |
| 31 | calling-station-id | (Standard) | Calling Party Number (ANI) | String | 5135672127 | Start Acc Stop Acc |
| 40 | acct-status-type | (Standard) | Account Request Type - start (1) or stop (2)<br>**Note:** 'start' isn't supported on the Calling Card application. | Numeric | 1 | Start Acc Stop Acc |
| 41 | acct-delay-time | (Standard) | No. of seconds tried in sending a particular record | Numeric | 5 | Start Acc Stop Acc |
| 42 | acct-input-octets | (Standard) | Number of octets received for that call duration | Numeric | - | Stop Acc |
| 43 | acct-output-octets | (Standard) | Number of octets sent for that call duration | Numeric | - | Stop Acc |
| 44 | acct-session-id | (Standard) | A unique accounting identifier - match start & stop | String | 34832 | Start Acc Stop Acc |
| 46 | acct-session-time | (Standard) | For how many seconds the user received the service | Numeric | - | Stop Acc |
| 47 | acct-input-packets | (Standard) | Number of packets received during the call | Numeric | - | Stop Acc |
| 48 | acct-oputput-packets | (Standard) | Number of packets sent during the call | Numeric | - | Stop Acc |

| Attribute ID | Attribute Name | Vendor-Specific Attribute (VSA) ID | Description | Value Format | Example | AAA |
|---|---|---|---|---|---|---|
| 61 | nas-port-type | (Standard) | Physical port type of device on which the call is active | String | 0: Asynchronous | Start Acc Stop Acc |
| **Response Attributes** | | | | | | |
| 26 | h323-return-code | 103 | The reason for failing authentication (0 = ok, other number failed) | Numeric | 0 Request accepted | Stop Acc |
| 44 | acct-session-id | (Standard) | A unique accounting identifier – match start & stop | String | - | Stop Acc |

Below is an example of RADIUS Accounting, where non-standard parameters are preceded with brackets:

```
Accounting-Request (4)
user-name = 111
acct-session-id = 1
nas-ip-address = 212.179.22.213
nas-port-type = 0
acct-status-type = 2
acct-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
```

# 38.4 Querying Device Channel Resources using SIP OPTIONS

The device reports its maximum and available channel resources in SIP 200 OK responses upon receipt of SIP OPTIONS messages. The device sends this information in the SIP X-Resources header with the following parameters:

- **telchs:** Specifies the total telephone channels and the number of free (available) telephone channels.
- **mediachs:** Not applicable.

Below is an example of the X-Resources:

```
X-Resources: telchs= 12/4;mediachs=0/0
```

In the example above, "telchs" specifies the number of available channels and the number of occupied channels (4 channels are occupied and 12 channels are available).

**Note:** This feature is applicable only to the Gateway application.

# Part VIII

## Diagnostics

# 39 Syslog and Debug Recording

For debugging and troubleshooting, you can use the device's Syslog and/or Debug Recording capabilities:

- **Syslog:** Syslog is an event notification protocol that enables a device to send event notification messages across IP networks to event message collectors, also known as Syslog servers. The device contains an embedded Syslog client, which sends error reports / events that it generates to a remote Syslog server using the IP / UDP protocol. This information is a collection of error, warning, and system messages that records every internal operation of the device.
- **Debug Recording:** The device can send debug recording packets to a debug capturing server. When the debug recording is activated, the device duplicates all messages that are sent and/or received by it and then sends them to an external server defined by IP address. The debug recording can be done for different types of traffic such as RTP/RTCP, T.38, and SIP. Debug recording is used for advanced debugging when you need to analyze internal messages and signals. Debug recording is also useful for recording network traffic in environments where hub or port mirroring is unavailable and for recording internal traffic between two endpoints on the same device.

**Note:** You can include Syslog messages in debug recording (see "Configuring Log Filter Rules" on page 602).

## 39.1 Configuring Log Filter Rules

The Logging Filters table lets you configure up to 60 rules for filtering debug recording (DR) packets, Syslog messages, and Call Detail Records (CDR). The log filter determines the calls for which you want to generate DR packets, Syslog messages or CDRs. For example, you can add a rule to generate Syslog messages only for calls belonging to IP Groups 2 and 4, or for calls belonging to all IP Groups except for IP Group 3. You can also configure log filters for generating CDRs only and saving them on the device (local storage). DR log filters can include signaling information such as SIP messages, Syslog messages, CDRs, media (RTP, RTCP, and T.38), and pulse-code modulation (PCM) of voice signals from and to the TDM.

If you don't configure any rules in the Logging Filters table and you have enabled DR, Syslog, and/or CDR generation (done by simply configuring an IP address for the relevant servers - see Note below), logs are generated for all calls. Thus, the benefit of log filtering is that it allows you to create logs per specific calls, eliminating the need for additional device resources (CPU consumption), otherwise required when logs are generated for all calls.

You can enable and disable configured Logging Filter rules. Enabling a rule activates the rule, whereby the device starts generating the DR packets, Syslog messages, or CDRs. Disabling a rule is useful, for example, if you no longer require the rule, but may need it in the future. Thus, instead of deleting the rule entirely, you can simply disable it.

**Notes:**

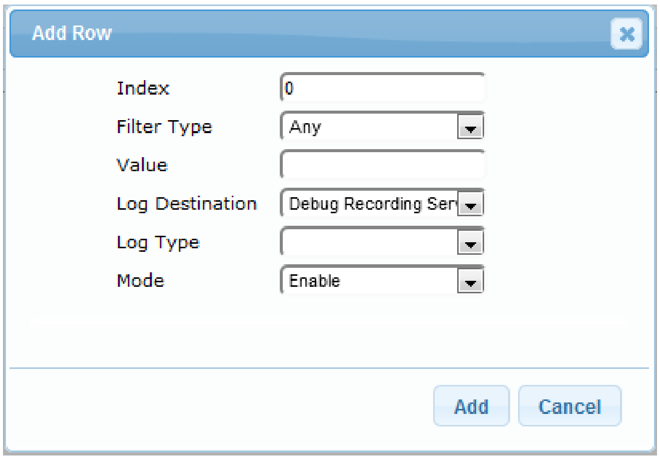
- If you want to configure a Logging Filter rule that logs Syslog messages to a Syslog server (i.e., not to a Debug Recording server), you must enable Syslog functionality, using the 'Enable Syslog' (EnableSyslog) parameter (see "Enabling Syslog" on page 612). Enabling Syslog functionality is not required for rules that include Syslog messages in the DR sent to a Debug Recording server.
- To configure the Syslog server's address, see "Configuring Address of Syslog Server" on page 612. To configure additional, global Syslog settings, see Configuring Syslog on page 604.
- To configure the Debug Recording server's address, see "Configuring Address of Debug Recording Server" on page 615.
- To configure additional, global CDR settings such as at what stage of the call the CDR is generated (e.g., start and end of call), see Configuring CDR Reporting on page 586.

The following procedure describes how to configure Logging Filter rules through the Web interface. You can also configure it through ini file (LoggingFilters) or CLI (configure system > logging > logging-filters).

➢ **To configure a logging filtering rule:**

1. Open the Logging Filters table (**Configuration** tab > **System** menu > **Logging** > **Logging Filters Table**).
2. Click **Add**; the following dialog box appears:

**Figure 39-1: Logging Filters Table - Add Row Dialog Box**




3. Configure a logging filter according to the parameters described in the table below.
4. Click **Add**.

**Logging Filters Table Parameter Descriptions**

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Index<br>[LoggingFilters_Index]</td><td>Defines an index number for the new table row.<br>Note: Each row must be configured with a unique index.</td></tr>
<tr><td>Filter Type<br>filter-type<br>[LoggingFilters_FilterType]</td><td>Defines the filter type criteria.<br>▪ [1] Any (default)<br>▪ [2] Trunk ID = Filters log according to Trunk ID. Note: Applicable only to the Gateway application.<br>▪ [3] Trunk Group ID = Filters log according to Hunt Group ID. For configuring Hunt Groups, see Configuring Hunt Groups on page 385. Note: Applicable only to the Gateway application.<br>▪ [5] FXS or FXO = Filters log according to FXS port.<br>▪ [6] Tel-to-IP = Filters log according to a Tel-to-IP routing rule. For configuring Tel-to-IP routing rules, see Configuring Tel-to-IP Routing Rules on page 409. Note: Applicable only to the Gateway application.<br>▪ [7] IP-to-Tel = Filters log according to an IP-to-Tel routing rule. For configuring IP-to-Tel routing rules, see Configuring IP-to-Hunt Group Routing Rules on page 421. Note: Applicable only to the Gateway application.<br>▪ [8] IP Group = Filters log according to an IP Group. For configuring IP Groups, see "Configuring IP Groups" on page 323.<br>▪ [9] SRD = Filters log according to an SRD. For configuring SRDs, see Configuring SRDs on page 311.<br>▪ [12] User = Filters log according to a user. The user is defined by username or username@hostname in the Request-URI of the SIP Request-Line. For example, "2222@10.33.45.201", which represents the following INVITE:<br><code>INVITE sip:2222@10.33.45.201;user=phone SIP/2.0</code><br>▪ [13] IP Trace = Filters log according to an IP network trace, Wireshark-like expression. For more information on configuring IP traces, see "Filtering IP Network Traces" on page 602.<br>▪ [14] SIP Interface = Filters log according to SIP Interface. For configuring SIP Interfaces, see Configuring SIP Interfaces on page 319.</td></tr>
</table>

| Parameter | Description |
|---|---|
| Value<br>value<br>[LoggingFilters_Value] | Defines the value for the selected filtering type in the 'Filter Type' parameter. The value can include the following:<br>▪ A single value.<br>▪ A range, using a hyphen "-" between the two values. For example, to specify IP Groups 1, 2 and 3, configure the parameter to "1-3" (without apostrophes).<br>▪ Multiple, non-contiguous values, using commas "," between each value. For example, to specify IP Groups 1, 3 and 9, configure the parameter to "1,3,9" (without apostrophes).<br>▪ FXS pertaining to a module, using the syntax module number/port or port, for example:<br>✓ "1/2" (without apostrophes), means module 1, port 2<br>✓ "1/[2-4]" (without apostrophes), means module 1, ports 2 through 4<br>▪ The exclamation (**!**) wildcard character can be used for excluding a specific configuration entity from the filter. For example, to include all IP Groups in the filter except IP Group ID 2, configure the 'Filter Type' parameter to **IP Group** and the 'Value' parameter to "!2" (without apostrophes).<br>▪ **Any** to indicate all.<br>**Notes:**<br>▪ You can use the index number or string name to specify the configuration entity for the following 'Filter Types': **Tel-to-IP**, **IP-to-Tel**, **IP Group**, **SRD**, or **SIP Interface**. For example, to specify IP Group at Index 2 with the name "SIP Trunk", configure the parameter to either "2" or "SIP Trunk" (without apostrophes).<br>▪ For IP trace expressions, see "Filtering IP Network Traces" on page 602. |

| Parameter | Description |
|---|---|
| Log Destination<br>log-dest<br>[LoggingFilters_Log Destination] | Defines where the device sends the log file.<br>▪ [0] Syslog Server = The device generates Syslog messages based on the configured log filter and sends them to a user-defined Syslog server. The Syslog messages can contain one of the following types of information, depending on the settings of the 'Log Type' parameter (described later in the table):<br>✓ Not configured (default): The Syslog messages contain the regular syslog information.<br>✓ CDR Only: The Syslog messages contain only CDRs (no system information and alerts).<br>▪ [1] Debug Recording Server = (Default) The device generates DR packets based on the configured log filter and sends them to a user-defined Debug Recording server..<br>▪ [2] Local Storage = The device generates CDRs based on the configured log filter and stores them locally on the device. For more information on local CDR storage, see Storing CDRs on the Device on page 586.<br>**Notes:**<br>▪ If the 'Filter Type' parameter is configured to **IP Trace**, you must configure the parameter to **Debug Recording Server**.<br>▪ If you configure the parameter to **Local Storage**, you must configure the 'Log Type' parameter to **CDR Only**.<br>▪ If you configure the parameter to **Syslog Server** and the debug level (GwDebugLevel) is configured to **No Debug** (see "Configuring Syslog Debug Level" on page 611), the Syslog messages include only system Warnings and Errors. |
| Log Type<br>log-type<br>[LoggingFilters_CaptureType] | Defines the type of messages to include in the log file.<br>▪ [0] = (Default) Not configured. The option is applicable only for sending Syslog messages to a Syslog server (i.e., 'Log Destination' parameter is configured to **Syslog Server**).<br>▪ [1] Signaling = The option is applicable only for DR (i.e., 'Log Destination' parameter is configured to **Debug Recording Server**). The DR includes signaling information such as SIP signaling messages, Syslog messages, CDRs, and the device's internal processing messages.<br>▪ [2] Signaling & Media = The option is applicable only for DR (i.e., 'Log Destination' parameter is configured to **Debug Recording Server**). The DR includes signaling, Syslog messages, and media (RTP/RTCP/T.38).<br>▪ [3] Signaling & Media & PCM = The option is applicable only for DR (i.e., 'Log Destination' parameter is configured to D**ebug Recording Server**). The DR includes signaling, Syslog messages, media, and PCM (voice signals from and to TDM).<br>▪ [5] CDR Only = Only CDRs are generated. The option is applicable only if the 'Log Destination' parameter is configured to **Syslog Server** or **Local Storage**. When configured to **Syslog Server**, only CDRs are included in the Syslog messages (excluding all system logs and alerts) sent to the Syslog server.<br>**Notes:**<br>▪ If you configure the 'Log Destination' parameter to **Local Storage**, the 'Log Type' parameter must be configured to **CDR Only**. |

| Parameter | Description |
|---|---|
| | ▪ The parameter is not applicable when the 'Filter Type' parameter is configured to **IP Trace**.<br>▪ To include Syslog messages in DR, it is unnecessary to enable Syslog functionality.. |
| Mode<br>`mode`<br>[LoggingFilters_Mode] | Enables and disables the rule.<br>▪ [0] Disable<br>▪ [1] Enable (default) |

## 39.1.1 Filtering IP Network Traces

You can filter Syslog and debug recording messages for IP network traces, by setting the 'Filter Type' parameter to **IP Trace** in the Logging Filters table. IP traces are used to record any IP stream, according to destination and/or source IP address, or port and Layer-4 protocol (UDP, TCP or any other IP type as defined by http://www.iana.com). Network traces are typically used to record HTTP.

When the **IP Trace** option is selected, only the ‘Value’ parameter is applicable; the 'Syslog' and 'Capture Type' parameters are not relevant. The 'Value' parameter configures the Wireshark-like filtering expressions for your IP trace. The following Wireshark-like expressions are supported:

**Table 39-1: Supported Wireshark-like Expressions for 'Value' Parameter**

| Expression | Description |
|---|---|
| ip.src, ip.dst | Source and destination IP address |
| ip.addr | IP address - up to two IP addresses can be entered |
| ip.proto | IP protocol type (PDU) entered as an enumeration value (e.g., 1 is ICMP, 6 is TCP, 17 is UDP) |
| udp, tcp, icmp, sip, ldap, http, https | Single expressions for protocol type |
| udp.port, tcp.port | Transport layer |
| udp.srcport, tcp.srcport | Transport layer for source port |
| udp.dstport, tcp.dstport | Transport layer for destination port |
| and, &&, ==, <, > | Between expressions |

Below are examples of configured expressions for the 'Value' parameter:

- udp && ip.addr==10.8.6.55
- ip.src==10.8.6.55 && udp.port>=5000 and udp.port<6000
- ip.dst==10.8.0.1/16
- ip.addr==10.8.6.40

For conditions requiring the "or" / "||" expression, add multiple table rows. For example, the Wireshark condition "(ip.src == 1.1.1.1 or ip.src == 2.2.2.2) and ip.dst == 3.3.3.3" can be configured using the following two table row entries:

1. ip.src == 1.1.1.1 and ip.dst == 3.3.3.3
2. ip.src == 2.2.2.2 and ip.dst == 3.3.3.3

**Notes:**

- If the 'Value' field is undefined, the device records all IP traffic types.
- You cannot use ip.addr or udp/tcp.port together with ip.src/dst or udp/tcp.srcport/dstport. For example, "ip.addr==1.1.1.1 and ip.src==2.2.2.2" is an invalid configuration value.

# 39.2 Configuring Syslog

This section describes the Syslog message format, how to configure and enable Syslog, and how to view the generated Syslog messages. For filtering Syslog messages for specific calls, see "Configuring Log Filter Rules" on page 602.

## 39.2.1 Syslog Message Format

The Syslog message is sent from the device to a Syslog server as an ASCII (American Standard Code for Information Interchange) message. Syslog uses UDP as its underlying transport layer mechanism. By default, UDP port 514 is assigned to Syslog, but this can be changed (see "Enabling Syslog" on page 612).

Syslog includes two types of log messages:

- SIP call session logs: Logs relating to call sessions (e.g., call established). These logs are identified by a session ID ("SID"), described in detail in the table below. The following is an example of a SIP-session related Syslog message:

```
13:10:57.811 : 10.13.4.12 : NOTICE  : [S=235][SID:2ed1c8:96:5]
(lgr_flow)(63)  UdpTransportObject#0- Adding socket event for
address 10.33.2.42:5060 [Time: 04-19-2012@18:29:39]
```

- Board logs: Logs relating to the operation of the device (infrastructure) that are non-call session related (e.g., device reset or Web login). These logs are identified by a board ID ("BID"), described in detail in the table below. The following is an example of a board Syslog message:

```
10:21:28.037 : 10.15.7.95 : NOTICE  : [S=872] [BID=3aad56:32]
Activity Log: WEB: Successful login at 10.15.7.95:80. User:
Admin. Session: HTTP (10.13.22.54)
```

The format of the Syslog message is described in the following table below:

**Table 39-2: Syslog Message Format Description**

| Message Item | Description |
|---|---|
| **Message Types** | Syslog generates the following types of messages:<br>▪ ERROR: Indicates that a problem has been identified that requires immediate handling.<br>▪ WARNING: Indicates an error that might occur if measures are not taken to prevent it.<br>▪ NOTICE: Indicates that an unusual event has occurred.<br>▪ INFO: Indicates an operational message.<br>▪ DEBUG: Messages used for debugging.<br>**Notes:**<br>▪ The INFO and DEBUG messages are required only for advanced debugging and by default, they are not sent by the device.<br>▪ When viewing Syslog messages in the Web interface, these message types are color coded. |
| **Message Sequence Number [S=<number>]** | By default, Syslog messages are sequentially numbered in the format [S=<number>], for example, "[S=643]". A skip in the number sequence of messages indicates a loss of message packets. For example, in the below Syslog, messages 238 through 300 were not received. In other words, 63 Syslog messages were lost (the sequential numbers are indicated below in bold font): |

<table>
<tr><th>Message Item</th><th>Description</th></tr>
<tr><td></td><td>
<code>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]</code><br>
<code>18:38:14. 83 : 10.33.45.72 : NOTICE: [S=236][SID:2ed1c8:96:5] (lgr_flow)(620) #0:DIGIT_EV [File: Line:-1]</code><br>
<code>18:38:14. 83 : 10.33.45.72 : NOTICE: [S=237][SID:2ed1c8:96:5] (lgr_flow)(621) | #0:DIGIT_EV [File: Line:-1]</code><br>
<code>18:38:14.958 : 10.33.45.72 : NOTICE: [S=301][SID:2ed1c8:96:5] (lgr_flow)(625) | #0:DIGIT_EV [File: Line:-1]</code><br>
You can disable the inclusion of the message sequence number in Syslog messages, by setting the 'CDR Syslog Sequence Number' parameter to Disable (see "Configuring Syslog" on page 612).
</td></tr>
<tr><td>Log Number (lgr)(number)</td><td>Ignore this number; it has been replaced by the Message Sequence Number (described previously).</td></tr>
<tr><td>Session ID (SID)</td><td>
Unique SIP call session and device identifier. The device identifier facilitates debugging by clearly identifying the specific device that sent the log message, especially useful in deployments consisting of multiple devices. In addition, the benefit of unique numbering is that it enables you to filter the information (such as SIP, Syslog, and media) according to device or session ID.<br>
The syntax of the session and device identifiers are as follows:<br>
[SID=<last 6 characters of device's MAC address>:<number of times device has reset>:<unique SID counter indicating the call session; increments consecutively for each new session; resets to 1 after a device reset>]<br>
For example:<br>
<code>14:32:52.028: 10.33.8.70: NOTICE: [S=9369] [SID=2ed1c8:96:5] (lgr_psbrdex)(274) recv <-- OFF_HOOK Ch:4</code><br>
Where:
<ul>
<li>2ed1c8 is the device's MAC address.</li>
<li>96 is the number of times the device has reset.</li>
<li>5 is a unique SID session number (in other words, this is the fifth call session since the last device reset).
<ul>
<li>✓ 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 session number.</li>
<li>✓ Forked legs and alternative legs share the same session number.</li>
</ul></li>
</ul>
Note: You can configure the device to maintain the same SID value for calls traversing multiple AudioCodes' devices. For more information, see "Maintaining Same Syslog SID/BID over Multiple Devices" on page 607.
</td></tr>
</table>

| Message Item | Description |
|---|---|
| **Board ID (BID)** | Unique non-SIP session related (e.g., device reset or a Trunk alarm) and device identifier. The device identifier facilitates debugging by clearly identifying the specific device that sent the log message, especially useful in deployments consisting of multiple devices. In addition, the benefit of unique numbering is that it enables you to filter the information according to device.<br>The syntax of the BID is as follows:<br>[BID=<last 6 characters in MAC>:<number of times device has reset>]<br>For example:<br>`14:32:52.062: 10.33.8.70: WARNING: [S=9399] [BID=2ed1c8:96] invalid Physical index`<br>Where:<br>▪ *2ed1c8* is the device's MAC address.<br>▪ *96* is the number of times the device has reset.<br>▪ **Note:** You can configure the device to maintain the same BID value for calls traversing multiple AudioCodes' devices. For more information, see "Maintaining Same Syslog SID over Multiple Devices" on page 607. |
| **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. |

#### 39.2.1.1 Event Representation in Syslog Messages

The Syslog message events that the device sends are denoted by unique abbreviations. The following example shows an abbreviated event in a Syslog message indicating packet loss (PL):

```
Apr  4 12:00:12 172.30.1.14 PL:5  [Code:3a002] [CID:3294] [Time:
20:17:00]
```

The table below lists these unique event abbreviations:

**Table 39-3: Syslog Error Name Descriptions**

| Error Abbreviation | Error Name Description |
|---|---|
| **AA** | Invalid Accumulated Packets Counter |
| **AC** | Invalid Channel ID |
| **AL** | Invalid Header Length |
| **AO** | Invalid Codec Type |
| **AP** | Unknown Aggregation Payload Type |
| **AR** | Invalid Routing Flag Received |
| **AT** | Simple Aggregation Packets Lost |
| **CC** | Command Checksum Error |

| Error Abbreviation | Error Name Description |
|---|---|
| **CE** | Invalid Cell Coder Code |
| **CS** | Command Sequence Error |
| **ES** | 8 sec Timeout Before Disconnect |
| **HO** | Host Received Overrun |
| **IA** | Invalid AMR Payload |
| **IC** | Invalid CID Error |
| **IG** | Invalid G723 Code |
| **IP** | Invalid payload length |
| **IR** | Invalid RTCP Packet |
| **IS** | Invalid SID Length |
| **LC** | Transmitter Received Illegal Command |
| **LF** | Lost Fax Frames In High Speed Mode |
| **LM** | Lost Modem Frames In High Speed Mode |
| **MI** | Misalignment Error |
| **MR** | Modem Relay Is Not Supported |
| **OR** | DSP JB Overrun |
| **PH** | Packet Header Error |
| **PL** | RTP Packet Loss |
| **RB** | Counts the number of BFI Frames Received From The Host |
| **RD** | No Available Release Descriptor |
| **RO** | RTP Reorder |
| **RP** | Unknown RTP Payload Type |
| **RS** | RTP SSRC Error |
| **UF** | Unrecognized Fax Relay Command |
| **AA** | Invalid Accumulated Packets Counter |
| **AC** | Invalid Channel ID |
| **AL** | Invalid Header Length |
| **AO** | Invalid Codec Type |
| **AP** | Unknown Aggregation Payload Type |
| **AR** | Invalid Routing Flag Received |

#### 39.2.1.2 Identifying AudioCodes Syslog Messages using Facility Levels

The device's Syslog messages can easily be identified and distinguished from Syslog messages from other equipment, by setting its Facility level. The Facility levels of the device's Syslog messages are numerically coded with decimal values. Facility level may use any of the "local use" facilities (0 through 7), according to RFC 3164. Implementing Facility levels is useful, for example, if you collect the device's as well as other equipments' Syslog messages on the same server. Therefore, in addition to filtering Syslog messages according to IP address, the messages can be filtered according to Facility level.

The Facility level is configured using the SyslogFacility ini file parameter, which provides the following options:

**Table 39-4: Syslog Facility Levels**

| Numerical Value | Facility Level |
|---|---|
| **16 (default)** | local use 0 (local0) |
| **17** | local use 1 (local1) |
| **18** | local use 2 (local2) |
| **19** | local use 3 (local3) |
| **20** | local use 4 (local4) |
| **21** | local use 5 (local5) |
| **22** | local use 6 (local6) |
| **23** | local use 7 (local7) |

Syslog messages begin with a less-than ("**<**") character, followed by a number, which is followed by a greater-than ("**>**") character. This is optionally followed by a single ASCII space. The number is known as the *Priority* and represents both the Facility level and the Severity level. A Syslog message with Facility level 16 is shown below:

```
Facility: LOCAL0 - reserved for local use (16)
```

### 39.2.1.3 Syslog Fields for Answering Machine Detection (AMD)

The Syslog message can include information relating to the Answering Machine Detection (AMD) feature. AMD is used to detect whether a human (including a fax machine), an answering machine, silence, or answering machine beeps have answered the call on the remote side.

- AMDSignal – this field can acquire one of the following values:
  - voice (V)
  - answer machine (A)
  - silence (S)
  - unknown (U)
- AMDDecisionProbability – probability (in %) success that correctly detects answering type

Below is an example of such a Syslog message with AMD information:

```
CallMachine:EVENT_DETECTED_EV - AMDSignal = <type – V/A/S/U>,
AMDDecisionProbability = <percentage> %
```

If there is no AMD detection, the AMDSignal field is shown empty (i.e. AMDSignal = ).

For more information on the AMD feature, see "Answering Machine Detection (AMD)" on page 203.

### 39.2.1.4 SNMP Alarms in Syslog Messages

SNMP alerts are sent to the Syslog server using the following formats:

- **Raised Alarms:** RAISE-ALARM: <Alarm Name>; Textual Description: <Textual Description>; Severity <Alarm Severity>; Source <Alarm Source>; Unique ID: <Alarm Unique ID >.

  If additional information exists in the alarm, then these are also added: Additional Info1:/ Additional Info2:/ Additional Info3

  The Messages' Severity is as follows:

**Table 39-5: Syslog Message Severity**

| ITU Perceived Severity (SNMP Alarm's Severity) | AudioCodes' Syslog Severity |
|---|---|
| **Critical** | RecoverableMsg |
| **Major** | RecoverableMsg |
| **Minor** | RecoverableMsg |
| **Warning** | Notice |
| **Indeterminate** | Notice |
| **Cleared** | Notice |

- **Cleared Alarms:** CLEAR-ALARM: <Alarm Name>; Textual Description: <Textual Description>; Severity <Alarm Severity>; Source <Alarm Source>; Unique ID: <Alarm Unique ID >; If exists Additional Info1:/ Additional Info2:/ Additional Info3:

### 39.2.2 Configuring Web User Activities to Report to Syslog

The device can report operations (activities) performed in the Web interface by management users, by including them in Syslog messages. The Syslog message indicates these logs with the string, "Activity Log". Each logged user activity includes the following information:

- Username (e.g., "Admin") of the user that performed the action
- IP address of the client PC from where the Web user accessed the management interface
- Protocol used for the session (e.g., SSH or HTTP)

The following example shows a Web-user activity log (indicating a login action) with the above-mentioned information:

```
14:07:46.300 : 10.15.7.95 : Local 0   :NOTICE  : [S=3149]
[BID=3aad56:32]  Activity Log: WEB: Successful login at
10.15.7.95:80. User: Admin. Session: HTTP (10.13.22.54)
```

The device can report the following Web user activities:

- Modifications of individual parameters, for example:

  ```
  14:33:00.162 : 10.15.7.95 : Local 0   :NOTICE  : [S=3403]
  [BID=3aad56:32]  Activity Log: Max Login Attempts was changed
  from '3' to '2'. User: Admin. Session: HTTP (10.13.22.54)
  ```

- Modifications of table fields, and addition and deletion of table rows, for example:

  ```
  14:42:48.334 : 10.15.7.95 : NOTICE  : [S=3546] [BID=3aad56:32]
  Activity Log: Classification - remove line 2. User: Admin.
  Session: HTTP (10.13.22.54)
  ```

- Entered CLI commands (modifications of security-sensitive commands are logged without the entered value).
- Configuration file load (reported without per-parameter notifications).
- Auxiliary file load and software update.
- Device reset and burn to flash memory.
- Access to unauthorized Web pages according to the Web user's access level.
- Modifications of "sensitive" parameters.
- Login and logout.
- Actions that are not related to parameter changes (for example, file uploads, file delete, lock-unlock maintenance actions, LDAP clear cache, register-unregister, and start-stop trunk. In the Web, these actions are typically done by clicking a button (e.g., the LOCK button).

For more information on each of the above listed options, see "Syslog, CDR and Debug Parameters" on page 670.

You can also configure the device to send an SNMP trap each time a user performs an activity. To enable trap notification, use the parameter, EnableActivityTrap (see "Configuring SNMP Community Strings" on page 88).

**Notes:**

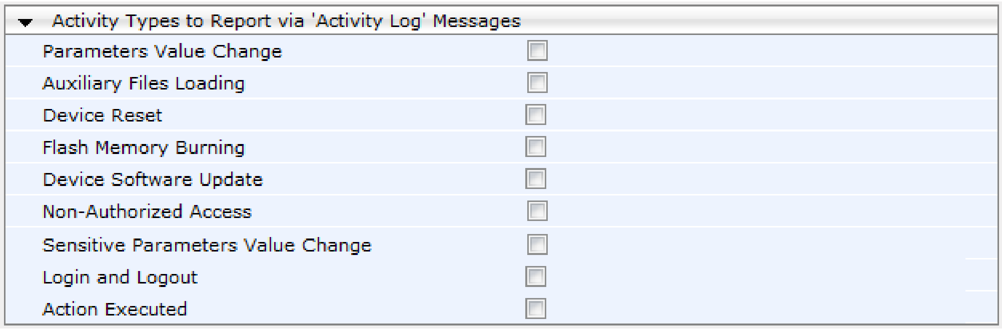
- You can also view logged user activities in the Web interface (see "Viewing Web User Activity Logs" on page 614).
- Logging of CLI commands can only be configured through CLI or ini file.

The following procedure describes how to configure Web user activity logging through the Web interface. You can also configure it through ini file (ActivityListToLog) or CLI (config-system > logging > activity-log).

➢ **To configure Web user activities to report to Syslog server:**

1. Open the Syslog Settings page (**Configuration** tab > **System** menu > **Syslog Settings**).
2. Under the Activity Types to Report via Activity Log Messages group, select the Web actions to report to the Syslog server.

**Figure 39-2: Web Activities to Report to Syslog**




3. Click **Submit**.

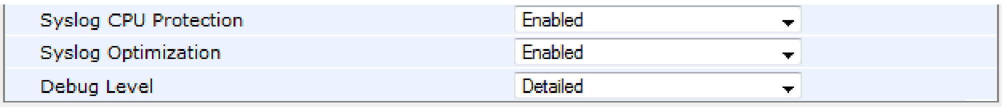
## 39.2.3 Configuring Syslog Debug Level

You can configure the amount of information (debug level) to include in Syslog messages. In addition, you can enable the device to send multiple Syslog messages bundled into a single packet as well as enable a protection mechanism that automatically lowers the debug level when the device's CPU resources become low, ensuring sufficient CPU resources are available for processing voice traffic.

➢ **To configure the Syslog debug level:**

1. Open the Syslog Settings page (**Configuration** tab > **System** menu > **Syslog Settings**).

**Figure 39-3: Configuring Syslog Debug Level**




2. From the 'Debug Level' (GwDebugLevel) drop-down list, select the desired debug level of the Syslog messages:
   - **No Debug:** Disables Syslog and no Syslog messages are sent.
   - **Basic:** Sends debug logs of incoming and outgoing SIP messages.
   - **Detailed:** Sends debug logs of incoming and outgoing SIP message as well as many other logged processes.
3. From the 'Syslog Optimization' (SyslogOptimization) drop-down list, select whether you want the device to accumulate and bundle multiple debug messages into a single UDP packet before sending it to a Syslog server. The benefit of this feature is that it reduces the number of UDP Syslog packets, thereby improving (optimizing) CPU utilization. The size of the bundled message is configured by the MaxBundleSyslogLength parameter.

4. From the 'Syslog CPU Protection' (SyslogCpuProtection) drop-down list, select whether you want to enable the protection feature for the device's CPU resources during debug reporting, ensuring voice traffic is unaffected. If CPU resources drop (i.e., high CPU usage) to a critical level (user-defined threshold), the device automatically lowers the debug level to free up CPU resources that were required for the previous debug-level functionality. When CPU resources become available again, the device increases the debug level to its' previous setting. For example, if you set the 'Debug Level' to **Detailed** and CPU resources decrease to the defined threshold, the device automatically changes the level to **Basic**, and if that is not enough, it changes the level to **No Debug**. Once CPU resources are returned to normal, the device automatically changes the debug level back to its' original setting (i.e., **Detailed**). The threshold is configured by the DebugLevelHighThreshold parameter.
5. Click **Submit**.

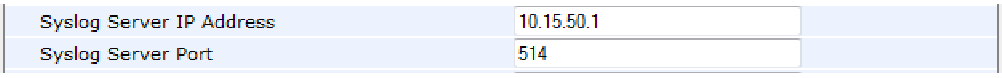
## 39.2.4 Configuring Address of Syslog Server

The following procedure describes how to configure the Syslog server's address to where the device sends the Syslog messages.

➢ **To configure the address of the Syslog server:**

1. Open the Syslog Settings page (**Configuration** tab > **System** menu > **Syslog Settings**).

**Figure 39-4: Configuring the Syslog Address**




2. In the 'Syslog Server IP Address' field, define the IP address of the Syslog server.
3. In the 'Syslog Server Port' field, define the port of the Syslog server.
4. Click **Submit**.

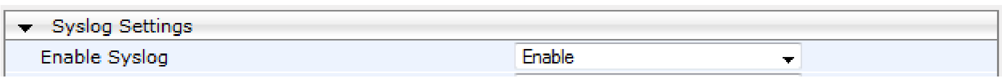
## 39.2.5 Enabling Syslog

The following procedure describes how to enable Syslog.

➢ **To enable Syslog:**

1. Open the Syslog Settings page (**Configuration** tab > **System** menu > **Syslog Settings**).

**Figure 39-5: Syslog Settings Page**




2. From the 'Enable Syslog' drop-down list, select **Enable**.
3. Click **Submit**.

## 39.2.6 Viewing Syslog Messages

You can receive and view Syslog messages generated by the device using any of the following Syslog server types:

- **Wireshark -** third-party network protocol analyzer (http://www.wireshark.org).

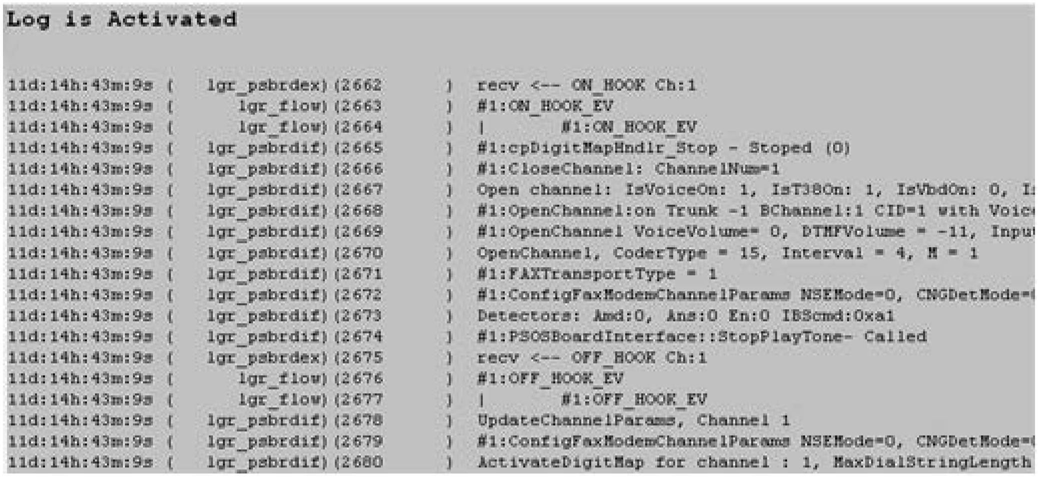
> **Note:** When debug recording is enabled and Syslog messages are also included in the debug recording, to view Syslog messages using Wireshark, you must install AudioCodes' Wireshark plug-in (acsyslog.dll). Once the plug-in is installed, the Syslog messages are decoded as "AC SYSLOG" and displayed using the "acsyslog" filter (instead of the regular "syslog" filter). For more information on debug recording, see "Debug Recording" on page 615.

- **Third-party, Syslog Server:** Any third-party Syslog server program that enables filtering of messages according to parameters such as priority, IP sender address, time, and date.
- Device's CLI Console: The device sends the error messages (e.g. Syslog messages) to the CLI console as well as to the configured destination. Use the following commands:

| | |
|---|---|
| debug log | ; Starts the debug |
| no debug log | ; Stops the debug |
| no debug log all | ; Stops all debug process |

- **Device's Web Interface:** The device provides an embedded Syslog server, which is accessed through the Web interface (**Status & Diagnostics** tab > **System Status** menu > **Message Log**). This provides limited Syslog server functionality.

**Figure 39-6: Message Log Page**



```
Log is Activated

11d:14h:43m:9s (   lgr_psbrdex)(2662      )  recv <-- ON_HOOK Ch:1
11d:14h:43m:9s (      lgr_flow)(2663      )  #1:ON_HOOK_EV
11d:14h:43m:9s (      lgr_flow)(2664      )  |        #1:ON_HOOK_EV
11d:14h:43m:9s (   lgr_psbrdif)(2665      )  #1:cpDigitMapHndlr_Stop - Stoped (0)
11d:14h:43m:9s (   lgr_psbrdif)(2666      )  #1:CloseChannel: ChannelNum=1
11d:14h:43m:9s (   lgr_psbrdif)(2667      )  Open channel: IsVoiceOn: 1, IsT38On: 1, IsVbdOn: 0, I
11d:14h:43m:9s (   lgr_psbrdif)(2668      )  #1:OpenChannel:on Trunk -1 BChannel:1 CID=1 with Voic
11d:14h:43m:9s (   lgr_psbrdif)(2669      )  #1:OpenChannel VoiceVolume= 0, DTMFVolume = -11, Inpu
11d:14h:43m:9s (   lgr_psbrdif)(2670      )  OpenChannel, CoderType = 15, Interval = 4, M = 1
11d:14h:43m:9s (   lgr_psbrdif)(2671      )  #1:FAXTransportType = 1
11d:14h:43m:9s (   lgr_psbrdif)(2672      )  #1:ConfigFaxModemChannelParams NSEMode=0, CNGDetMode=
11d:14h:43m:9s (   lgr_psbrdif)(2673      )  Detectors: Amd:0, Ans:0 En:0 IBScmd:0xa1
11d:14h:43m:9s (   lgr_psbrdif)(2674      )  #1:PSOSBoardInterface::StopPlayTone- Called
11d:14h:43m:9s (   lgr_psbrdex)(2675      )  recv <-- OFF_HOOK Ch:1
11d:14h:43m:9s (      lgr_flow)(2676      )  #1:OFF_HOOK_EV
11d:14h:43m:9s (      lgr_flow)(2677      )  |        #1:OFF_HOOK_EV
11d:14h:43m:9s (   lgr_psbrdif)(2678      )  UpdateChannelParams, Channel 1
11d:14h:43m:9s (   lgr_psbrdif)(2679      )  #1:ConfigFaxModemChannelParams NSEMode=0, CNGDetMode=
11d:14h:43m:9s (   lgr_psbrdif)(2680      )  ActivateDigitMap for channel : 1, MaxDialStringLength
```

The displayed logged messages are color-coded as follows:

- Yellow - fatal error message
- Blue - recoverable error message (i.e., non-fatal error)
- Black - notice message

To stop and clear the Message Log, close the Message Log page by accessing any another page in the Web interface.

> **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.

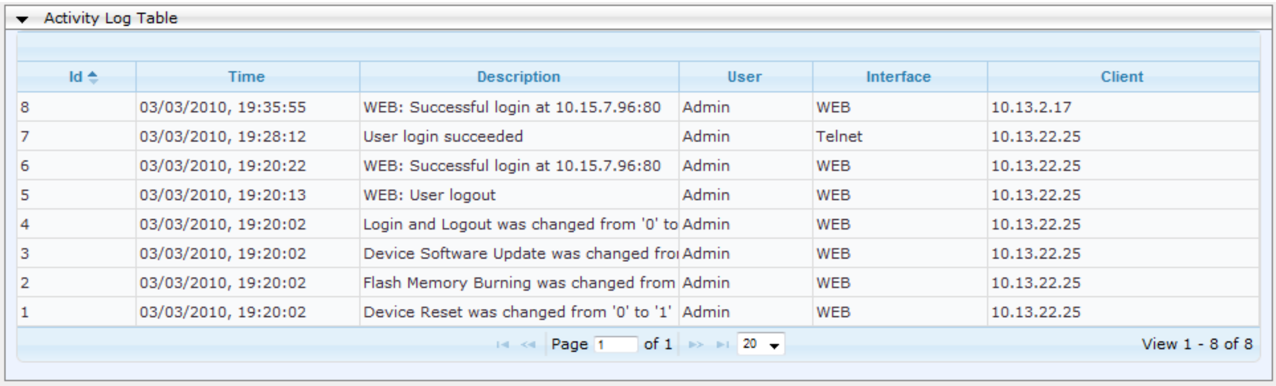
### 39.2.7 Viewing Web User Activity Logs

If you have enabled the reporting of Web user activities, you can view logged activities in the Web interface's Activity Log table (read-only). For enabling the logging of Web user activities, see "Configuring Web User Activities to Report to Syslog" on page 610.

➢ **To view Web user activity logs:**

- Open the Activity Log table (**Status & Diagnostics** tab > **System Status** menu > **Activity Log**).

**Figure 39-7: Activity Log Table**


Activity Log Table

| Id | Time | Description | User | Interface | Client |
|---|---|---|---|---|---|
| 8 | 03/03/2010, 19:35:55 | WEB: Successful login at 10.15.7.96:80 | Admin | WEB | 10.13.2.17 |
| 7 | 03/03/2010, 19:28:12 | User login succeeded | Admin | Telnet | 10.13.22.25 |
| 6 | 03/03/2010, 19:20:22 | WEB: Successful login at 10.15.7.96:80 | Admin | WEB | 10.13.22.25 |
| 5 | 03/03/2010, 19:20:13 | WEB: User logout | Admin | WEB | 10.13.22.25 |
| 4 | 03/03/2010, 19:20:02 | Login and Logout was changed from '0' to | Admin | WEB | 10.13.22.25 |
| 3 | 03/03/2010, 19:20:02 | Device Software Update was changed fro | Admin | WEB | 10.13.22.25 |
| 2 | 03/03/2010, 19:20:02 | Flash Memory Burning was changed from | Admin | WEB | 10.13.22.25 |
| 1 | 03/03/2010, 19:20:02 | Device Reset was changed from '0' to '1' | Admin | WEB | 10.13.22.25 |

Page 1 of 1 20 View 1 - 8 of 8

The table includes the following information:

**Table 39-6: Activity Log Table Description**

| Parameter | Description |
|---|---|
| Time | Date and time that the user activity was performed. |
| Description | Description of the user activity. |
| User | Username of the user that performed the activity. |
| Interface | Protocol used for the connection to the management interface (e.g., Telnet, SSH, Web, or HTTP). |
| Client | IP address of the client PC from where the user accessed the Web interface. |

# 39.3 Configuring Debug Recording

This section describes how to configure and activate debug recording, and how to collect debug recording packets. For filtering debug recording packets for specific calls, see "Configuring Log Filter Rules" on page 602.

**Notes:**

- Debug recording is collected only on the device's OAMP interface.
- For a detailed description of the debug recording parameters, see "Syslog, CDR and Debug Parameters" on page 670.

## 39.3.1 Configuring Address of Debug Recording Server

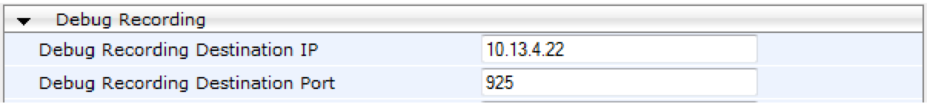
The procedure below describes how to configure the address of the debug recording (capturing) server to where the device sends the captured traffic. Once you configure an address, the device generates DR packets for all calls. However, you can configure the device to generate DR packets for specific calls, using Logging Filter rules in the Logging Filters table (see "Configuring Log Filter Rules" on page 602).

**Note:** You can also save debug recordings to an external USB hard drive that is connected to the device's USB port. For more information, see USB Storage Capabilities on page 541.

➢ **To configure the debug recording server's address:**

1. Open the Logging Settings page (**Configuration** tab > **System** menu > **Logging** > **Logging Settings**).

**Figure 39-8: Logging Settings Page**




2. In the 'Debug Recording Destination IP' field, configure the IP address of the debug capturing server.
3. In the 'Debug Recording Destination Port' field, configure the port of the debug capturing server.
4. Click **Submit**.

## 39.3.2 Collecting Debug Recording Messages

To collect debug recording packets, use the open source packet capturing program, Wireshark. AudioCodes proprietary plug-in files for Wireshark are required.

**Notes:**

- The default debug recording port is 925. You can change the port in Wireshark (**Edit** menu > **Preferences** > **Protocols** > **AC DR**).
- The plug-in files are per major software release of Wireshark. For more information, contact your AudioCodes sales representative.
- The plug-in files are applicable only to Wireshark 32-bit for Windows.

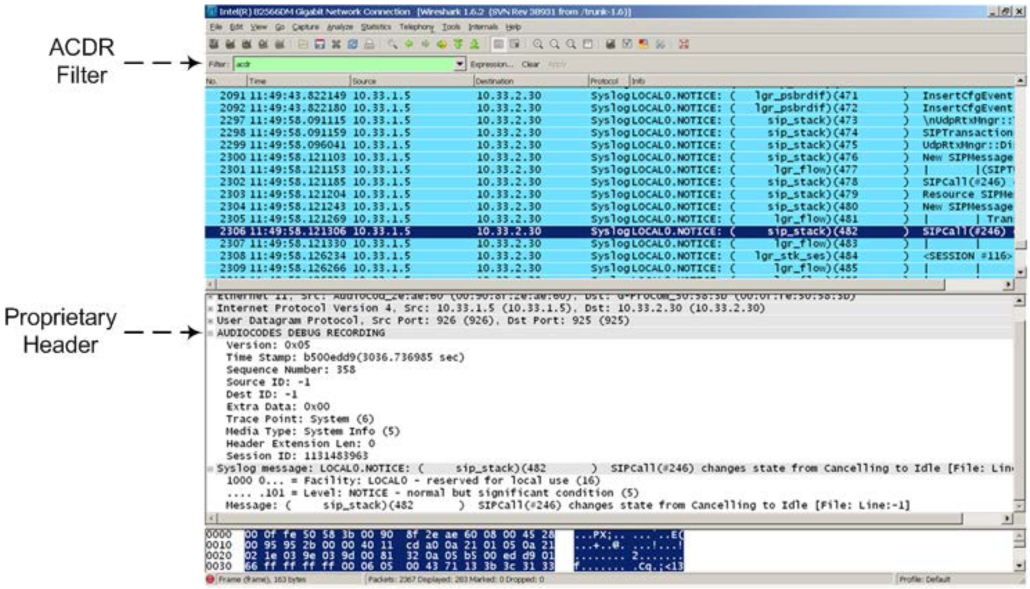
➢ **To install Wireshark and the plug-ins for debug recording:**

1. Install Wireshark on your computer. The Wireshark program can be downloaded from http://www.wireshark.org.
2. Download the proprietary plug-in files from www.audiocodes.com/downloads.
3. Copy the plug-in files to the directory in which you installed Wireshark, as follows:

| Copy this file | To this folder on your PC |
|---|---|
| ...\dtds\cdr.dtd | Wireshark\dtds\ |
| ...\plugins\<Wireshark ver.>*.dll | Wireshark\plugins\<Wireshark ver.> |
| ...\tpncp\tpncp.dat | Wireshark\tpncp |

4. Start Wireshark.
5. In the Filter field, type "acdr" (see the figure below) to view the debug recording messages. Note that the source IP address of the messages is always the OAMP IP address of the device.

The device adds the header "AUDIOCODES DEBUG RECORDING" to each debug recording message, as shown below:



### 39.3.3 Debug Capturing on Physical VoIP Interfaces

You can capture traffic on the device's physical (Ethernet LAN) VoIP interfaces (Layer-2 VLAN tagged packets). The captured traffic can be saved in a PCAP-format file (suitable for Wireshark) to a TFTP (default) or an FTP server. The generated PCAP file is in the Extensible Record Format (ERF). The capture can also be saved to a USB device. The maximum file size of debug captures that can be saved to the device is 20 MB.

To capture traffic on physical VoIP interfaces, use the following CLI commands:

- Starts physical VoIP debug capture:

```
# debug capture voip physical eth-lan
# debug capture voip physical start
```

- Captures packets continuously in a cyclical buffer (packets always captured until stop command):

```
# debug capture VoIP physical cyclic buffer
```

- Retrieves latest capture (PCAP file) saved on a specified server:

```
# debug capture VoIP physical get_last_capture <TFTP/FTP
server IP address>
```

  The file is saved to the device's memory (not flash) and erased after a device reset.

- Marks the captured file (useful for troubleshooting process):

```
# debug capture VoIP physical insert-pad
```

  Before running this command, the debug capture must be started.

- Displays debug status and configured rules:

```
# debug capture VoIP physical show
```

- Specifies the destination (FTP, TFTP, or USB) where you want the PCAP file sent:

```
# debug capture VoIP physical target <ftp|tftp|usb>
```

- Stops the debug capture, creates a file named debug-capture-voip-<timestamp>.pcap, and sends it to the TFTP or FTP server:

```
# debug capture voip physical stop <TFTP/FTP server IP
address>
```

  If no IP address is defined, the capture is saved on the device for later retrieval.

**This page is intentionally left blank.**

# 40 Self-Testing

The device features the following self-testing modes to identify faulty hardware components:

- **Detailed Test (Configurable):** This test verifies the correct functioning of the different hardware components on the device. This test is done when the device is taken out of service (i.e., not in regular service for processing calls). The test is performed on startup when initialization of the device completes.

  To enable this test, set the ini file parameter, EnableDiagnostics to 1 or 2, and then reset the device.  Upon completion of the test and if the test fails, the device sends information on the test results of each hardware component to the Syslog server.

  The following hardware components are tested:

  - Analog interfaces - when EnableDiagnostics = 1 or 2

**Notes:**

- To return the device to regular operation and service, disable the test by setting the ini file parameter, EnableDiagnostics to 0, and then reset the device.
- While the test is enabled, ignore errors sent to the Syslog server.

- **Startup Test (automatic):** This hardware test has minor impact in real-time. While this test is executed, the regular operation of the device is disabled. If an error is detected, an error message is sent to the Syslog.

**This page is intentionally left blank.**

# 41 Creating Core Dump and Debug Files upon Device Crash

For debugging purposes, you can create a core dump file and/or debug file. The files may assist you in identifying the cause of the crash. The core dump can either be included in or excluded from the debug file, or alternatively, sent separately to a TFTP server. You can then provide the files to AudioCodes support team for troubleshooting.

- **Core Dump File:** You can enable the device to send a core dump file to a remote destination upon a device crash. The core dump is a copy of the memory image at the time of the crash. It provides a powerful tool for determining the root cause of the crash. When enabled, the core dump file is sent to a user-defined TFTP server (IP address). If no address is configured, the core dump file is saved to the device's flash memory (if it has sufficient memory). The core dump file is saved as a binary file in the following name format: "**core**_<device name>_ver_<firmware version>_mac_<MAC address>_<date>_<time>", for example, *core_acMediant_ver_700-8-4_mac_00908F099096_1-02-2015_3-29-29*.
- **Debug File:** You can manually retrieve the debug file from the device and save it to a folder on your local PC. The debug file contains the following information:
    - Exception information, indicating the specific point in the code where the crash occurred and a list of up to 50 of the most recent SNMP alarms that were raised by the device before it crashed.
    - Latest log messages that were recorded prior to the crash.
    - Core dump (**only** if enabled, no IP address has been defined, and the device has sufficient memory on its flash).
    - May include additional application-proprietary debug information.

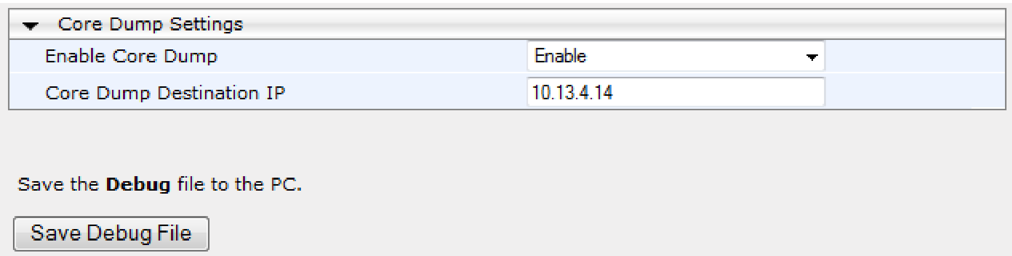
  The debug file is saved as a zipped file in the following name format: "**debug**_<device name>_ver_<firmware version>_mac_<MAC address>_<date>_<time>", for example, *debug_acMediant_ver_700-8-4_mac_00908F099096_1-03-2015_3-29-29*.

The following procedure describes how to configure core dump file creation through the Web interface.

### ➢ To enable core dump file generation:

1. Set up a TFTP server to where you want to send the core dump file.
2. Open the Debug Utilities page (**Maintenance** tab > **Maintenance** menu > **Debug Utilities**).

**Figure 41-1: Debug Utilities Page**




3. From the 'Enable Core Dump' drop-down list, select **Enable**.
4. In the 'Core Dump Destination IP' field, enter an IP address of the remote server to where you want the file to be sent (optional).
5. Click **Submit**, and then reset the device with a save-to-flash for your settings to take effect.

The following procedure describes how to retrieve the debug file from the device through the Web interface.

➢ **To save the debug file from the device:**

- In the Debug Utilities page, click the **Save Debug File** button.

# 42 Pinging a Remote Host or IP Address

You can verify the network connectivity with a remote host or IP address by pinging the network entity.

- IPv4: The ping to an IPv4 address can be done from any of the device's VoIP interfaces that is configured with an IPv4 address. The ping is done using the following CLI command:

```
# ping <IPv4 ip address or host name> source [voip|data]
interface
```

For a complete description of the ping command, refer to the *CLI Reference Guide*.

**This page is intentionally left blank.**

# Part IX

## Appendix

# 43 Testing SIP Signaling Calls

A simulated endpoint can be configured on the device to test SIP signaling of calls between it and a remote destination. This feature is useful in that it can remotely verify SIP message flow without involving the remote end side in the debug process. The SIP test call simulates the SIP signaling process - call setup, SIP 1xx responses, through to completing the SIP transaction with a 200 OK.

The test call sends Syslog messages to a Syslog server, showing the SIP message flow, tone signals (e.g., DTMF), termination reasons, as well as voice quality statistics and thresholds (e.g., MOS).

## 43.1 Configuring Test Call Endpoints

The Test Call table lets you test the SIP signaling (setup and registration) and media (DTMF signals) of calls between a simulated phone on the device and a remote endpoint. These tests involve both incoming and outgoing calls, where the test endpoint can be configured as the caller or called party. Test calls can be dialed automatically at a user-defined interval and/or manually when required. The simulated phone and remote endpoints are defined as SIP URIs (user@host) and the remote destination can be defined as an IP Group, IP address, or according to a Tel-to-IP routing rule. You can also enable automatic registration of the endpoint.

When a SIP test call is initiated, the device generates a SIP INVITE towards the remote endpoint (e.g., a SIP proxy server or softswitch). It simulates the SIP call setup process, managing SIP 1xx responses and completing the SIP transaction with a 200 OK.

**Note:** By default, you can configure up to five test calls. However, this number can be increased by installing the relevant Software License Key. For more information, contact your AudioCodes sales representative.

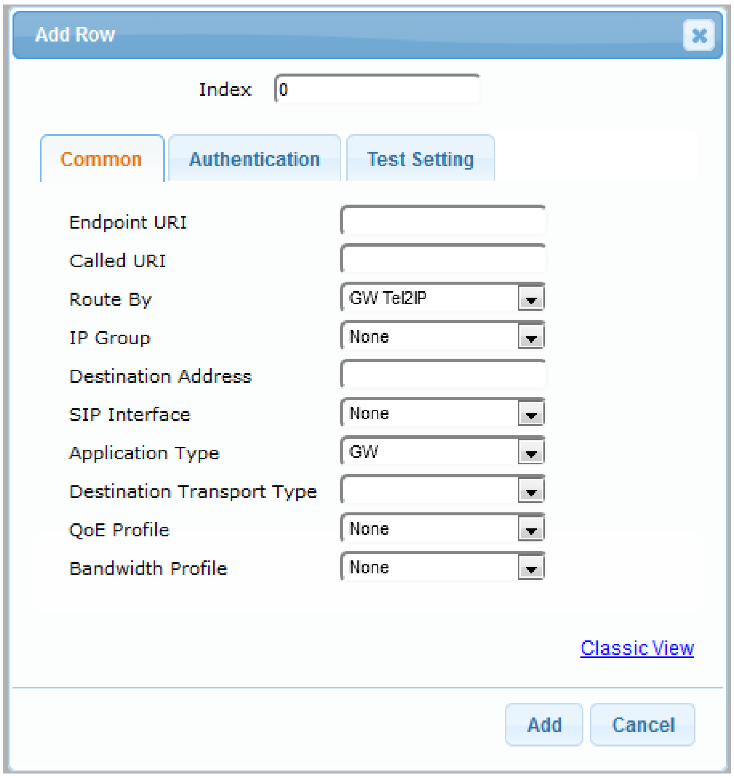
The following procedure describes how to configure test calls through the Web interface. You can also configure it through ini file (Test_Call) or CLI (configure system > test-call > test-call-table).

➢ **To configure a test call:**

1. Open the Test Call table (**Configuration** tab > **System** menu > **Test Call** > **Test Call Table**).

2. Click **Add**; the following dialog box appears:

**Figure 43-1: Test Call Table - Add Row Dialog Box**




3. Configure a test call according to the parameters described in the table below.
4. Click **Add**, and then save ("burn") your settings to flash memory.

**Table 43-1: Test Call Table Parameter Descriptions**

| Parameter | Description |
|---|---|
| **Common Tab** | |
| Endpoint URI<br>`endpoint-uri`<br>[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.<br>The valid value is a string of up to 150 characters. By default, the parameter is not configured.<br>**Note:** The parameter is mandatory. |
| Called URI<br>`called-uri`<br>[Test_Call_CalledURI] | Defines the destination (called) URI (user@host).<br>The valid value is a string of up to 150 characters. By default, the parameter is not configured. |
| Route By<br>`route-by`<br>[Test_Call_RouteBy] | Defines the type of routing method. This applies to incoming and outgoing calls.<br>▪ **[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).<br>▪ **[1]** IP Group = Calls are matched by (or routed to) an IP Group. To specify the IP Group, see the 'IP Group' parameter in the table. |

| Parameter | Description |
|---|---|
| | ▪ **[2]** Dest Address = Calls are matched by (or routed to) an SRD and application type. To specify the address, see the 'Destination Address' parameter in the table.<br>**Notes:**<br>▪ If configured to **GW Tel2IP** or **Dest Address**, you must assign a SIP Interface (see the 'SIP Interface' parameter in the table).<br>▪ For REGISTER messages:<br>✓ The **GW Tel2IP** option cannot be used as the routing method.<br>✓ If configured to **IP Group,** only Server-type IP Groups can be used. |
| IP Group<br>`ip-group-id`<br>[Test_Call_IPGroupName] | Assigns an IP Group to the rule, which is the IP Group that the test call is sent to or received from.<br>By default, no value is defined (**None**).<br>**Notes:**<br>▪ The parameter is applicable only if the 'Route By' parameter is configured to **IP Group** [1].<br>▪ The IP Group is used for incoming and outgoing calls. |
| Destination Address<br>`dst-address`<br>[Test_Call_DestAddress] | Defines the destination host. This can be defined as an IP address[:port] or DNS name[:port].<br>**Note:** The parameter is applicable only if the 'Route By' parameter is configured to **Dest Address** [2]. |
| SIP Interface<br>`sip-interface-name`<br>[Test_Call_SIPInterfaceName] | Assigns a SIP Interface to the rule, which is the SIP Interface to which the test call is sent and received from.<br>By default, no value is defined (**None**).<br>**Note:** The parameter is applicable only if the 'Route By' parameter is configured to GW Tel2IP or Dest Address. |
| Application Type<br>`application-type`<br>[Test_Call_ApplicationType] | Defines the application type for the endpoint. This associates the IP Group and SRD to a specific SIP interface.<br>▪ [0] GW (default) = Gateway application |
| Destination Transport Type<br>`dst-transport`<br>[Test_Call_DestTransportType] | Defines the transport type for outgoing calls.<br>▪ **[-1]** = Not configured (default)<br>▪ **[0]** UDP<br>▪ **[1]** TCP<br>▪ **[2]** TLS<br>**Note:** The parameter is applicable only if the 'Route By' parameter is set to [2] (Dest Address). |
| QoE Profile<br>`qoe-profile`<br>[Test_Call_QOEProfile] | Assigns a QoE Profile to the test call.<br>By default, no value is defined (**None**).<br>To configure QoE Profiles, see "Configuring Quality of Experience Profiles" on page 293. |
| Bandwidth Profile<br>`bandwidth-profile`<br>[Test_Call_BWProfile] | Assigns a Bandwidth Profile to the test call.<br>By default, no value is defined (**None**).<br>To configure Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 297. |

| Parameter | Description |
|---|---|
| **Authentication Tab**<br>**Note:** These parameters are applicable only if the Call Party parameter is set to **Caller.** | |
| Auto Register<br>`auto-register`<br>[Test_Call_AutoRegister] | Enables automatic registration of the endpoint. The endpoint can register to the device itself or to the 'Destination Address' or 'IP Group' parameter settings (see above).<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable |
| Username<br>`user-name`<br>[Test_Call_UserName] | Defines the authentication username.<br>By default, no username is defined. |
| Password<br>`password`<br>[Test_Call_Password] | Defines the authentication password.<br>By default, no password is defined. |
| **Test Setting Tab** | |
| Call Party<br>`call-party`<br>[Test_Call_CallParty] | Defines whether the test endpoint is the initiator or receiving side of the test call.<br>▪ **[0]** Caller (default)<br>▪ **[1]** Called |
| Maximum Channels for Session<br>`max-channels`<br>[Test_Call_MaxChannels] | Defines the maximum number of concurrent channels for the test session. For example, if you have configured an endpoint "101" and you set the parameter to "3", the device automatically creates three simulated endpoints - "101", "102" and "103" (i.e., consecutive endpoint URIs are assigned).<br>The default is 1. |
| Call Duration<br>`call-duration`<br>[Test_Call_CallDuration] | Defines the call duration (in seconds).<br>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.<br>**Note:** The parameter is applicable only if 'Call Party' is set to **Caller**. |
| Calls per Second<br>`calls-per-second`<br>[Test_Call_CallsPerSecond] | Defines the number of calls per second.<br>**Note:** The parameter is applicable only if 'Call Party' is set to **Caller**. |
| Test Mode<br>`test-mode`<br>[Test_Call_TestMode] | Defines the test session mode.<br>▪ **[0]** Once = (Default) The test runs until the lowest value between the following is reached:<br>✓ Maximum channels is reached for the test session, configured by 'Maximum Channels for Session'.<br>✓ Call duration ('Call Duration') multiplied by calls per second ('Calls per Second').<br>✓ Test duration expires, configured by 'Test Duration'.<br>▪ **[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, |

| Parameter | Description |
|---|---|
| | the test session stays within the configured maximum channels.<br>**Note:** The parameter is applicable only if 'Call Party' is set to **Caller**. |
| Test Duration<br>`test-duration`<br>[Test_Call_TestDuration] | Defines the test duration (in minutes).<br>The valid value is 0 to 100000. The default is 0 (i.e., unlimited).<br>**Note:** The parameter is applicable only if 'Call Party' is set to **Caller**. |
| Play<br>`play`<br>[Test_Call_Play] | Enables and defines the playing of a tone to the answered side of the call.<br>▪ **[0]** Disable<br>▪ **[1]** DTMF = (Default) Plays a user-defined DTMF string, configured in "Configuring DTMF Tones for Test Calls" on page 633.<br>▪ **[2]** PRT = Plays a non-DTMF tone from the PRT file (Test Call Tone). For this option, a PRT file must be loaded to the device (see "Prerecorded Tones File" on page 500).<br>**Notes:**<br>▪ Currently, play of DTMF is not supported.<br>▪ The parameter is applicable only if 'Call Party' is set to **Caller**. |
| Schedule Interval<br>`schedule-interval`<br>[Test_Call_ScheduleInterval] | Defines the interval (in minutes) between automatic outgoing test calls.<br>The valid value range is 0 to 100000. The default is 0 (i.e., scheduling is disabled).<br>**Note:** The parameter is applicable only if 'Call Party' is set to **Caller**. |

## 43.2 Starting and Stopping Test Calls

The following procedure describes how to start, stop, and restart test calls.

➢ **To start, stop, and restart a test call:**

1. In the Test Call table, select the required test call entry; the **Actions** button appears above the table.
2. From the **Actions** drop-down list, choose the required command:
   - **Dial:** starts the test call (this action is applicable only if the test call party is the caller).
   - **Drop Call:** stops the test call.
   - **Restart:** ends all established calls and then starts the test call session again.

The status of the test call is displayed in the 'Test Status' field of the Test Call table:

- "Idle": test call is not active.
- "Scheduled": test call is planned to run (according to 'Schedule Interval' parameter settings)
- "Running": test call has been started (i.e., the **Dial** command was clicked)

- "Receiving": test call has been automatically activated by calls received for the test call endpoint from the remote endpoint (when all these calls end, the status returns to "Idle")
- "Terminating": test call is in the process of terminating the currently established calls (this occurs if the **Drop Call** command is clicked to stop the test)
- "Done": test call has been successfully completed (or was prematurely stopped by clicking the **Drop Call** command)

A more detailed description of this field is displayed below the table when you click the **Show/Hide** button (see "Viewing Test Call Statistics" on page 632).

## 43.3 Viewing Test Call Statistics

In addition to viewing a brief status description of the test call in the 'Test Status' field (as described in "Starting, Stopping and Restarting Test Calls" on page 631), you can also view a more detailed status description which includes test call statistics.

➢ **To view statistics of a test call:**

1. Open the Test Call table (**Configuration** tab > **System** menu > **Test Call** > **Test Call Table**).
2. Select the test call table entry whose call statistics you want to view.
3. Click the **Show/Hide** button; the call statistics are displayed in the **Test Statistics** pane located below the table, as shown below:

**Figure 43-2: Viewing Test Call Statistics**

| Test Statistics | |
|---|---|
| Elapsed Time [HH:MM:SS]: | 00:01:44 |
| Active Calls: | 0 |
| Call Attempts: | 5 |
| Total Established Calls: | 5 |
| Total Failed Attempts: | 0 |
| Remote Disconnections Count: | 0 |
| Test Status: | Done |
| Average CPS: | 1.00 |
| Detailed Status: | Done - Established Calls: 5, ASR: 100% |
| MOS Status: | Local:12 (Red), Remote:25 (Red) |
| Delay Status: | Local:993 msec (Red), Remote:1006 msec (Red) |
| Jitter Status: | Local:1 msec (Green), Remote:0 msec (Green) |
| Packet Loss Status: | Local:51% (Red), Remote:49% (Red) |
| Bandwidth Status: | Rx:37 KBytes/s (Green), Tx:41 KBytes/s (Red) |

The 'Test Statistics' pane displays the following test session information:

- **Elapsed Time:** Duration of the test call since it was started (or restarted).
- **Active Calls:** Number of currently established test calls.
- **Call Attempts:** Number of calls that were attempted.
- **Total Established Calls:** Total number of calls that were successfully established.
- **Total Failed Attempts:** Total number of call attempts that failed.
- **Remote Disconnections Count:** Number of calls that were disconnected by the remote side.
- **Average CPS:** Average calls per second.
- **Test Status:** Displays the status (brief description) as displayed in the 'Test Status' field (see "Starting, Stopping and Restarting Test Calls" on page 631).
- **Average CPS:** Average calls per second.
- **Detailed Status:** Displays a detailed description of the test call status:
    - "Idle": test call is currently not active.

- "Scheduled - Established Calls: <number of established calls>, ASR: <%>": test call is planned to run (according to 'Schedule Interval' parameter settings) and also shows the following summary of completed test calls:
    - Total number of test calls that were established.
    - Number of successfully answered calls out of the total number of calls attempted (ASR).
- "Running (Calls: <number of active calls>, ASR: <%>)": test call has been started (i.e., the **Dial** command was clicked) and shows the following:
    - Number of currently active test calls.
    - Number of successfully answered calls out of the total number of calls attempted (Answer Seizure Ratio or ASR).
- "Receiving (<number of active calls>)": test call has been automatically activated by calls received for this configured test call endpoint from the configured remote endpoint. When all these calls terminate, the status returns to "Idle".
- "Terminating (<number of active calls>)": the **Drop Call** command has been clicked to stop the test call and the test call is in the process of terminating the currently active test calls.
- "Done - Established Calls: <number of established calls>, ASR: <%>": test call has been successfully completed (or was prematurely stopped by clicking the **Drop Call** command) and shows the following:
    - Total number of test calls that were established.
    - Number of successfully answered calls out of the total number of calls attempted (ASR).

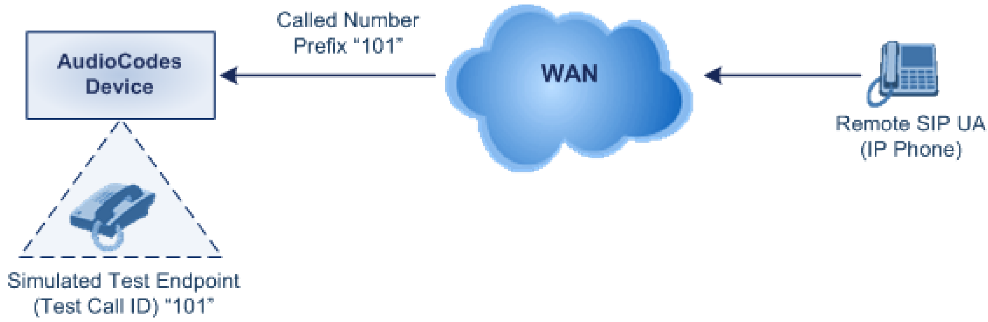
- **MOS Status:** MOS count and color threshold status of local and remote sides according to the assigned QoE Profile.
- **Delay Status:** Packet delay count and color-threshold status of local and remote sides according to the assigned QoE Profile.
- **Jitter Status:** Jitter count and color-threshold status of local and remote sides according to the assigned QoE Profile.
- **Packet Loss Status:** Packet loss count and color-threshold status of local and remote sides according to the assigned QoE Profile.
- **Bandwidth Status:** Tx/Rx bandwidth and color-threshold status according to the assigned Bandwidth Profile.

**Note:** On the receiving side, when the first call is accepted in "Idle" state, statistics are reset.

## 43.4 Configuring Basic Test Call

The Basic Test Call feature tests incoming calls from remote SIP (IP) endpoints to a simulated test endpoint on the device. The only required configuration is to assign a prefix number (test call ID) to the simulated endpoint. Incoming calls with this called (destination) prefix number are identified by the device as test calls and sent to the simulated endpoint. The figure below displays a basic test call example.

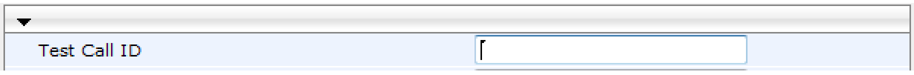
**Figure 43-3: Incoming Test Call Example**




### ➢ To configure basic call testing:

1. Open the Test Call Settings page (**Configuration** tab > **System** menu > **Test Call** > **Test Call Settings**).

**Figure 43-4: Test Call Settings Page**




2. In the 'Test Call ID' field, enter a prefix for the simulated endpoint.
3. Click **Submit**.

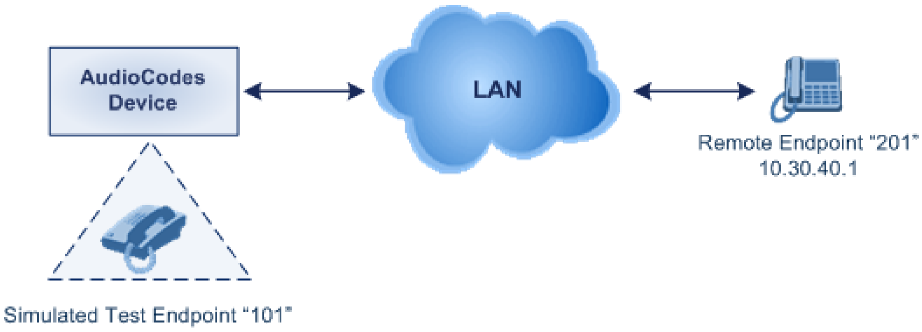
> **Note:** Test calls are done on all SIP Interfaces.

## 43.5 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 43-5: Single Test Call Example**




- Test Call table configuration:
  - Endpoint URI: "101"
  - Called URI: "201"
  - Route By: **Dest Address**
  - Destination Address: "10.30.40.01"
  - SIP Interface: SIPInterface_0
  - Call Party: **Caller**
  - Test Mode: **Once**

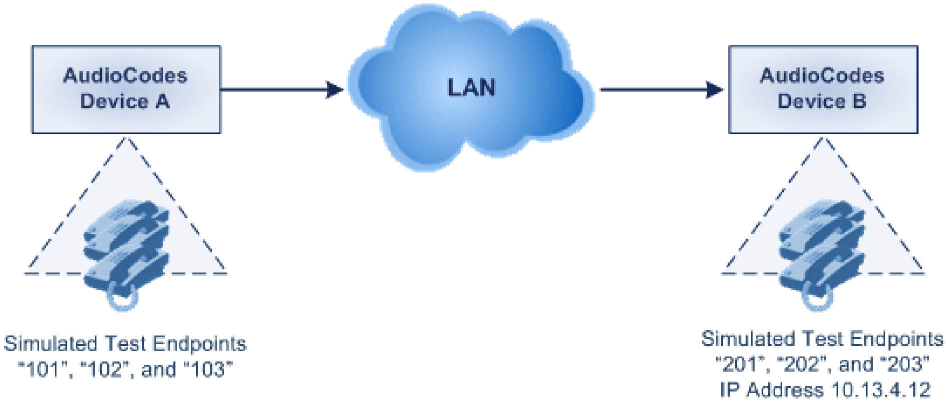
Alternatively, if you want to route the test call using the Tel-to-IP Routing table for the Gateway application, configure the following:

- Test Call table configuration:
  - Endpoint URI: 101@10.0.0.1
  - Route By: GW Tel2IP
  - SIP Interface: SIPInterface_0
  - Called URI: 201@10.30.40.1
  - Call Party: Caller
- Tel-to-IP Routing table configuration:
  - Destination Phone Prefix: 201 (i.e., the Called URI user-part)
  - Source Phone Prefix: 101 (i.e., the Endpoint URI user-part)
  - Destination 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.

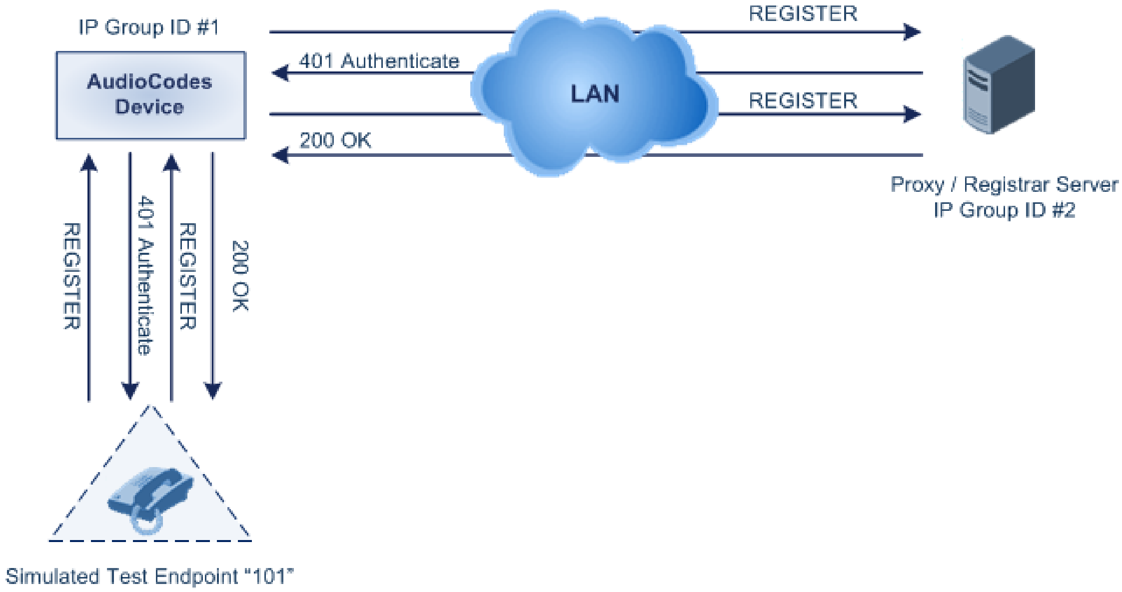
**Figure 43-6: Batch Test Call Example**




- Test Call table configuration at Device A:
    - Endpoint URI: "101"
    - Called URI: "201"
    - Route By: **Dest Address**
    - Destination Address: "10.13.4.12"
    - SIP Interface: SIPInterface_0
    - Call Party: **Caller**
    - Maximum Channels for Session: "3" (configures three endpoints - "101", "102" and "103)
    - Call Duration: "5" (seconds)
    - Calls per Sec: "1"
    - Test Mode: **Continuous**
    - Test Duration: "3" (minutes)
    - Schedule Interval: "180" (minutes)
- Test Call table configuration at Device B:
    - Endpoint URI: "201"
    - Maximum Channels for Session: "3" (configures three endpoints - "201", "202" and "203)

- **Registration Test Call Scenario:** This example describes the configuration for testing the registration and authentication (i.e., username and pas,sword) process of a simulated test endpoint on the device with an external proxy/registrar server. This is useful, for example, for verifying that endpoints located in the LAN can register with an external proxy and subsequently, communicate with one another.

**Figure 43-7: 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)
  - SIP Interface: SIPInterface_0
  - Auto Register: **Enable**
  - User Name: "testuser"
  - Password: "12345"
  - Call Party: **Caller**

**This page is intentionally left blank.**

# 44 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 44-1: Dialing Plan Notations for Prefixes and Suffixes**

<table>
<tr><th>Notation</th><th>Description</th></tr>
<tr><td>x (letter "x")</td><td>Wildcard that denotes any single digit or character.</td></tr>
<tr><td># (pound symbol)</td><td>▪ 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.<br>▪ 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.<br>▪ 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#.</td></tr>
<tr><td>* (asterisk symbol)</td><td>▪ When used on its own, it denotes any number or string.<br>▪ When used as part of a number, it denotes the asterisk key. For example, *345 represents a number that starts with *345.</td></tr>
<tr><td>$ (dollar sign)</td><td>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:<br>▪ Source and Destination Phone Prefix<br>▪ Source and Destination Username<br>▪ Source and Destination Calling Name Prefix</td></tr>
<tr><td colspan="2">Range of Digits<br>Notes:<br>▪ Dial plans denoting a prefix that is a range must be enclosed in square brackets, e.g., [4-8] or 23xx[456].<br>▪ Dial plans denoting a prefix that is not a range is not enclosed, e.g., 12345#.<br>▪ Dial plans denoting a suffix must be enclosed in parenthesis, e.g., (4) and (4-8).<br>▪ Dial plans denoting a suffix that include multiple ranges, the range must be enclosed in square brackets, e.g., (23xx[4,5,6]).<br>▪ 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]).</td></tr>
<tr><td>[n-m] or (n-m)</td><td>Represents a range of numbers.<br>Examples:<br>▪ To depict prefix numbers from 5551200 to 5551300:<br>✓ [5551200-5551300]#<br>▪ To depict prefix numbers from 123100 to 123200:<br>✓ 123[100-200]#<br>▪ To depict prefix and suffix numbers together:<br>✓ 03(100): for any number that starts with 03 and ends with 100.<br>✓ [100-199](100,101,105): for a number that starts with 100 to 199 and ends with 100, 101 or 105.</td></tr>
</table>

| Notation | Description |
|---|---|
| | ✓ 03(abc): for any number that starts with 03 and ends with abc.<br>✓ 03(5xx): for any number that starts with 03 and ends with 5xx.<br>✓ 03(400,401,405): for any number that starts with 03 and ends with 400 or 401 or 405.<br>**Notes:**<br>▪ The value *n* must be less than the value *m*.<br>▪ Only numerical ranges are supported (not alphabetical letters).<br>▪ For suffix ranges, the starting (*n*) and ending (*m*) numbers in the range must include the same number of digits. For example, (23-34) is correct, but (3-12) is not. |
| **[n,m,...]** or **(n,m,...)** | Represents multiple numbers. The value can include digits or characters.<br>Examples:<br>▪ To depict a one-digit number starting with 2, 3, 4, 5, or 6: **[2,3,4,5,6]**<br>▪ To depict a one-digit number ending with 7, 8, or 9: **(7,8,9)**<br>▪ Prefix with Suffix: **[2,3,4,5,6](7,8,9)** - prefix is denoted in square brackets; suffix in parenthesis<br>For **prefix only**, the notations *d[n,m]e* and *d[n-m]e* can also be used:<br>▪ To depict a five-digit number that starts with 11, 22, or 33: **[11,22,33]xxx#**<br>▪ 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:<br>▪ Prefix: **[123-130,455,766,780-790]**<br>▪ Suffix: **(123-130,455,766,780-790)**<br>**Note:** The ranges and the single numbers used in the dial plan must have the same number of digits. For example, each number range and single number in the dialing plan example above consists of three digits. |
| **Special ASCII Characters** | The device does not support the use of ASCII characters in manipulation rules and therefore, for LDAP-based queries, the device can use the hexadecimal (HEX) format of the ASCII characters for phone numbers instead. The HEX value must be preceded by a backslash “\”. For example, you can configure a manipulation rule that changes the received number **+49 (7303) 165-xxxxx** to **+49 \287303\29 165-xxxxx**, where *\28* is the ASCII HEX value for “(“ and *\29* is the ASCII HEX value for “)”. The manipulation rule in this example would denote the parenthesis in the destination number prefix using "x" wildcards (e.g., xx165xxxxx#); the prefix to add to the number would include the HEX values (e.g., +49 \287303\29 165-).<br>Below is a list of common ASCII characters and their corresponding HEX values:<br>**ASCII Character** **HEX Value**<br>* \2a<br>( \28<br>) \29<br>\ \5c<br>/ \2f |

**Note:** When configuring phone numbers or prefixes in the Web interface, enter them only as digits without any other characters. For example, if you wish to enter the phone number 555-1212, it must be entered as 5551212 without the hyphen (-). If the hyphen is entered, the entry is invalid.

**This page is intentionally left blank.**

# 45 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.

## 45.1 Management Parameters

This section describes the device's management-related parameters.

### 45.1.1 General Parameters

The general management parameters are described in the table below.

**Table 45-1: General Management Parameters**

| Parameter | Description |
|---|---|
| [WebLoginBlockAutoComplete] | Disables autocompletion when entering the management login username in the 'Username' field of the device's Web interface. Disabling autocompletion may be useful for security purposes by hiding previously entered usernames and thereby, preventing unauthorized access to the device's management interface.<br>▪ [0] Disable = (Default) Autocompletion is enabled and the 'Username' field automatically offers previously logged in usernames.<br>▪ [1] Enable = Autocompletion is disabled. |
| [EnforcePasswordComplexity] | Enables the enforcement of management login-password complexity requirements to ensure strong passwords.<br>▪ [0] Disable (default)<br>▪ [1] Enable<br>For more information on password complexity requirements, see the 'Password' parameter in Configuring Management User Accounts on page 65. |
| [CustomerSN] | Defines a serial number (S/N) for the device.<br>**Note:** The device's original S/N is automatically added at the end of the configured S/N. For example, if the original S/N is 8906721 and the configured S/N is "abc123", the resultant S/N is "abc1238906721". |

| Parameter | Description |
|---|---|
| Web and Telnet Access List Table<br>[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).<br>The default is 0.0.0.0 (i.e., the device can be accessed from any IP address).<br>For example:<br>WebAccessList_0 = 10.13.2.66<br>WebAccessList_1 = 10.13.77.7<br>For a description of the parameter, see "Configuring Web and Telnet Access List" on page 75. |

## 45.1.2 Web Parameters

The Web parameters are described in the table below.

**Table 45-2: Web Parameters**

| Parameter | Description |
|---|---|
| Enable web access from all interfaces<br>`web-access-from-all-interfaces`<br>[EnableWebAccessFromAllInterfaces] | Enables Web access from any of the device's IP network interfaces. This feature applies to HTTP and HTTPS protocols.<br>▪ [0] = (Default) Disable – Web access is only through the OAMP interface.<br>▪ [1] = Enable - Web access is through any network interface.<br>**Note:** For the parameter to take effect, a device reset is required. |
| Password Change Interval<br>[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.<br>The valid value is 0 to 100000, where 0 means that the password is always valid. The default is 1140.<br>**Note:** The parameter is applicable only when using the Web Users table, where the default value of the 'Password Age' parameter in the Web Users table inherits the parameter's value. |
| User Inactivity Timer<br>[UserInactivityTimer] | Defines the duration (in days) for which a user has not logged in to the Web interface, after which the status of the user becomes inactive and can no longer access the Web interface. These users can only log in to the Web interface if their status is changed (to New or Valid) by a System Administrator or Master user.<br>The valid value is 0 to 10000, where 0 means inactive. The default is 90.<br>**Note:** The parameter is applicable only when using the Web Users table. |
| Session Timeout<br>[WebSessionTimeout] | Defines the duration (in minutes) of inactivity of a logged-in user in the Web interface, after which the the user is |

| Parameter | Description |
|---|---|
| | automatically logged off the Web session. In other words, the session expires when the user has not performed any operations (activities) in the Web interface for the configured duration.<br>The valid value is 0-100000, where 0 means no timeout. The default is 15.<br>**Note:** You can also configure the functionality per user in the Web Users table (see Advanced User Accounts Configuration on page 67), which overrides this global setting. |
| Deny Access On Fail Count<br>[DenyAccessOnFailCount] | Defines the maximum number of failed login attempts, after which the requesting IP address is blocked.<br>The valid value range is 0 to 10. The values 0 and 1 mean immediate block. The default is 3. |
| Deny Authentication Timer<br>[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.<br>The valid value is 0 to 100000, where 0 means that login is not denied regardless of number of failed login attempts. The default is 60. |
| Display Login Information<br>[DisplayLoginInformation] | Enables display of user's login information on each successful login attempt.<br>▪ **[0]** = Disable (default)<br>▪ **[1]** = Enable |
| [EnableMgmtTwoFactorAuthentication] | Enables Web login authentication using a third-party, smart card.<br>▪ **[0]** = Disable (default)<br>▪ **[1]** = Enable<br>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.<br>Typically, a TLS connection is established between the smart card and the device's Web interface, and a RADIUS server is implemented to authenticate the password with the username. Thus, this feature implements a two-factor authentication - what the user has (the physical card) and what the user knows (i.e., the login password). |
| `http-port`<br>**[HTTPport]** | Defines the LAN HTTP port for Web management (default is 80). To enable Web management from the LAN, configure the desired port.<br>**Note:** For the parameter to take effect, a device reset is required. |
| [DisableWebConfig] | Determines whether the entire Web interface is read-only.<br>▪ **[0]** = (Default) Enables modifications of parameters. |

| Parameter | Description |
|---|---|
| | ▪ **[1]** = Web interface is read-only.<br>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').<br>**Note:** For the parameter to take effect, a device reset is required. |
| [ResetWebPassword] | Enables the device to restore the default management users:<br>▪ Security Administrator user (username "Admin"; password "Admin")<br>▪ Monitor user (username "User"; password "User")<br>In addition, all other users that may have been configured (in the Web Users table) are deleted.<br>▪ [0] = (Default) Disabled. Currently configured users (usernames and passwords) are retained.<br>▪ [1] = Enabled. Default users are restored (see description above) and all other configured users are deleted.<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ In addition to the ini file (see above), you can also restore the default user accounts through the following management platforms:<br>✓ SNMP (restores default users and retains other configured users:<br>1) Set acSysGenericINILine to WEBPasswordControlViaSNMP = 1, and reset the device with a flash burn (set acSysActionSetResetControl to 1 and acSysActionSetReset to 1).<br>2) Change the username and password in the acSysWEBAccessEntry table. Use the following format:<br>Username acSysWEBAccessUserName: old/pass/new<br>Password acSysWEBAccessUserCode: username/old/new |
| **Customizing Web GUI** | |

| Parameter | Description |
|---|---|
| [WelcomeMessage]<br>`configure system > welcome-msg` | Defines a welcome message displayed on the Web interface's Web Login page.<br>The format of the ini file table parameter is:<br>[WelcomeMessage ]<br>FORMAT WelcomeMessage_Index = WelcomeMessage_Text<br>[\WelcomeMessage]<br>For Example:<br>FORMAT WelcomeMessage_Index = WelcomeMessage_Text<br>WelcomeMessage 1 = "**********************************" ;<br>WelcomeMessage 2 = "********* This is a Welcome message ***" ;<br>WelcomeMessage 3 = "**********************************" ;<br>For more information, see Creating a Login Welcome Message on page 61.<br>**Note:**<br>▪ Each index row represents a line of text. Up to 20 lines (or rows) of text can be defined.<br>▪ The configured text message must be enclosed in double quotation marks (i.e., "...").<br>▪ If the parameter is not configured, no Welcome message is displayed. |
| [UseProductName] | Enables the option to customize the name of the device (product) that appears in the management interfaces.<br>▪ [0] = Disabled (default).<br>▪ [1] = Enables the display of a user-defined name, which is configured by the UserProductName parameter.<br>For more information, see Customizing the Product Name on page 60. |
| [UserProductName] | Defines a name for the device instead of the default name.<br>The value can be a string of up to 29 characters.<br>For more information, see Customizing the Product Name on page 60.<br>**Note:** To enable customization of the device name, see the UseProductName parameter. |
| [UseWebLogo] | Defines whether the Web interface displays a logo image or text.<br>▪ [0] = (Default) The Web interface displays a logo image, configured by the LogoFileName parameter.<br>▪ [1] = The Web interface displays text, configured by the WebLogoText parameter.<br>For more information, see Replacing the Corporate Logo on page 58. |

| Parameter | Description |
|---|---|
| [WebLogoText] | Defines the text that is displayed instead of the logo in the Web interface.<br>The valid value is a string of up to 15 characters.<br>For more information, see Replacing the Corporate Logo with Text on page 59.<br>**Note:** The parameter is applicable only when the UseWebLogo parameter is configured to 1. |
| [LogoWidth] | Defines the width (in pixels) of the logo image that you want displayed in the Web interface instead of the default logo.<br>The valid value is 0 to 199. The default is 145.<br>For more information, see Replacing the Corporate Logo with an Image on page 59.<br>**Notes:**<br>▪ The optimal setting depends on your screen resolution.<br>▪ If the width of the loaded image is greater than the maximum value, the device automatically resizes the image to the default width size.<br>▪ The height is limited to 24 pixels.<br>▪ The parameter is applicable only when the UseWebLogo parameter is configured to 0.<br>▪ To define the image file, see the LogoFileName parameter. |
| [LogoFileName] | Defines the name of the image file that you want loaded to the device. This image is displayed as the logo in the Web interface (instead of AudioCodes logo).<br>The file name can be up to 47 characters.<br>For more information, see Replacing the Corporate Logo with an Image on page 59.<br>**Notes:**<br>▪ The image file type can be one of the following: GIF, PNG, JPG, or JPEG.<br>▪ The size of the image file can be up to 64 Kbytes.<br>▪ The parameter is applicable only when the UseWebLogo parameter is configured to 0. |

### 45.1.3 Telnet Parameters

The Telnet parameters are described in the table below.

**Table 45-3: Telnet Parameters**

| Parameter | Description |
|---|---|
| Embedded Telnet Server<br>`telnet`<br>[TelnetServerEnable] | Enables the device's embedded Telnet server.<br>▪ **[0]** Disable<br>▪ **[1]** Enable Unsecured (default)<br>▪ **[2]** Enable Secured |

| Parameter | Description |
|---|---|
| | **Note:** Only management users with Security Administrator level, Administrator level, or Master level can access the device through Telnet (see "Configuring Web User Accounts" on page 65). |
| Telnet Server TCP Port<br>`telnet-port`<br>[TelnetServerPort] | Defines the port number for the embedded Telnet server.<br>The valid range is all valid port numbers. The default port is 23. |
| Telnet Server Idle Timeout<br>`idle-timeout`<br>[TelnetServerIdleDisconnect] | Defines the timeout (in minutes) for disconnection of an idle Telnet session. When set to zero, idle sessions are not disconnected.<br>The valid range is any value. The default is 0.<br>**Note:** For the parameter to take effect, a device reset is required. |
| Maximum Telnet Sessions<br>`telnet-max-sessions`<br>[TelnetMaxSessions] | Defines the maximum number of permitted, concurrent Telnet/SSH sessions.<br>The valid range is 1 to 5 sessions. The default is 2.<br>**Note:** Before changing the value, make sure that not more than this number of sessions are currently active; otherwise, the new setting will not take effect. |
| [CLIPrivPass] | Defines the password to access the Enable configuration mode in the CLI.<br>The valid value is a string of up to 50 characters. The default is "Admin".<br>**Note:** The password is case-sensitive. |

## 45.1.4 ini File Parameters

The parameters relating to ini-file management are described in the table below.

**Table 45-4: ini File Parameters**

| Parameter | Description |
|---|---|
| **[INIPasswordsDisplayType]** | Defines how passwords are displayed in the ini file.<br>▪ **[0]** Disable (default) = Passwords are obscured ("encoded"). The passwords are displayed in the following syntax: $1$<obscured password> (e.g., $1$S3p+fno=).<br>▪ **[1]** Enable = All passwords are hidden and replaced by an asterisk (*). |

## 45.1.5 SNMP Parameters

The SNMP parameters are described in the table below.

**Table 45-5: SNMP Parameters**

| Parameter | Description |
|---|---|
| Disable SNMP<br>`disable` | Enables and disables SNMP.<br>▪ **[0]** No = (Default) SNMP is enabled.<br>▪ **[1]** Yes = SNMP is disabled. |

| Parameter | Description |
|---|---|
| [DisableSNMP] | **Note:** For the parameter to take effect, a device reset is required. |
| `port`<br>[SNMPPort] | Defines the device's local (LAN) UDP port used for SNMP Get/Set commands.<br>The range is 100 to 3999. The default port is 161.<br>**Note:** For the parameter to take effect, a device reset is required. |
| [ChassisPhysicalAlias] | Defines the 'alias' name object for the physical entity as specified by a network manager, and provides a non-volatile 'handle' for the physical entity.<br>The valid range is a string of up to 255 characters. |
| [ChassisPhysicalAssetID] | Defines the user-assigned asset tracking identifier object for the device's chassis as specified by an EMS, and provides non-volatile storage of this information.<br>The valid range is a string of up to 255 characters. |
| [ifAlias] | Defines the textual name of the interface. The value is equal to the ifAlias SNMP MIB object.<br>The valid range is a string of up to 64 characters. |
| `auto-send-keep-alive`<br>[SendKeepAliveTrap] | Enables the device to send NAT keep-alive traps to the port of the SNMP network management station (e.g., AudioCodes EMS). This is used for NAT traversal, and allows SNMP communication with AudioCodes EMS management platform, located in the WAN, when the device is located behind NAT. It is needed to keep the NAT pinhole open for the SNMP messages sent from EMS to the device. The device sends the trap periodically - every 9/10 of the time configured by the NATBindingDefaultTimeout parameter. The trap that is sent is acKeepAlive. For more information on the SNMP trap, refer to the *SNMP Reference Guide*.<br>▪ **[0]** = (Default) Disable<br>▪ **[1]** = Enable<br>For configuring the port number, use the KeepAliveTrapPort parameter.<br>**Note:** For the parameter to take effect, a device reset is required. |
| [KeepAliveTrapPort] | Defines the port of the SNMP network management station to which the device sends keep-alive traps.<br>The valid range is 0 - 65534. The default is port 1161.<br>To enable NAT keep-alive traps, use the SendKeepAliveTrap parameter. |
| [PM_EnableThresholdAlarms] | Enables the sending of the SNMP trap event, acPerformanceMonitoringThresholdCrossing which is sent every time the threshold (high and low) of a Performance Monitored object (e.g., acPMMediaRealmAttributesMediaRealmBytesTxHighThreshold) is crossed.<br>▪ [0] = (Default) Disable<br>▪ [1] = Enable |
| `sys-oid`<br>[SNMPSysOid] | Defines the base product system OID.<br>The default is eSNMP_AC_PRODUCT_BASE_OID_D.<br>**Note:** For the parameter to take effect, a device reset is required. |
| [SNMPTrapEnterpriseOid] | Defines the Trap Enterprise OID. |

| Parameter | Description |
|---|---|
| | The default is eSNMP_AC_ENTERPRISE_OID.<br>The inner shift of the trap in the AcTrap subtree is added to the end of the OID in the parameter.<br>**Note:** For the parameter to take effect, a device reset is required. |
| [acUserInputAlarmDescription] | Defines the description of the input alarm. |
| [acUserInputAlarmSeverity] | Defines the severity of the input alarm. |
| [AlarmHistoryTableMaxSize] | Defines the maximum number of rows in the Alarm History table. The parameter can be controlled by the Config Global Entry Limit MIB (located in the Notification Log MIB).<br>The valid range is 50 to 1000. The default is 500.<br>**Note:** For the parameter to take effect, a device reset is required. |
| [ActiveAlarmTableMaxSize] | Defines the maximum number of currently active alarms that can be displayed in the Active Alarms table. When the table reaches this user-defined maximum capacity (i.e., full), the device sends the SNMP trap event, acActiveAlarmTableOverflow. If the table is full and a new alarm is raised by the <device>, the new alarm is not displayed in the table.<br>The valid range is 50 to 300. The default is 120.<br>For more information on the Active Alarms table, see Viewing Active Alarms on page 551.<br>**Note:**<br>▪ For the parameter to take effect, a <device> reset is required.<br>▪ 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`<br>[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 Trunk Group table.<br>▪ [0] Disable = (Default) The device sends the SNMP trap for non-configured ports.<br>▪ [1] Enable = The device does not send the SNMP trap for non-configured ports.<br>**Note:**<br>▪ The parameter is applicable to all telephony (analog) port types.<br>▪ The parameter is applicable only to the Gateway application. |
| `engine-id`<br>[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.<br>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<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required. |

| Parameter | Description |
|---|---|
| | ▪ Before setting the parameter, all SNMPv3 users must be deleted; otherwise, the parameter setting is ignored.<br>▪ If the supplied key does not pass validation of the 12 Hex values input or it is set with the default value, the engine ID is generated according to RFC 3411. |
| **SNMP Trap Destination Parameters** (configure system/snmp trap destination)<br>**Note:** Up to five SNMP trap managers can be defined. | |
| SNMP Manager<br>[SNMPManagerIsUsed_x] | Determines the validity of the parameters (IP address and port number) of the corresponding SNMP Manager used to receive SNMP traps.<br>▪ **[0]** (Check box cleared) = Disabled (default)<br>▪ **[1]** (Check box selected) = Enabled |
| IP Address<br>`ip-address`<br>[SNMPManagerTableIP_x] | Defines the IP address of the remote host used as an SNMP Manager. The device sends SNMP traps to this IP address.<br>Enter the IP address in dotted-decimal notation, e.g., 108.10.1.255. |
| Trap Port<br>`port`<br>[SNMPManagerTrapPort_x] | Defines the port number of the remote SNMP Manager. The device sends SNMP traps to this port.<br>The valid SNMP trap port range is 100 to 4000. The default port is 162. |
| Trap Enable<br>`send-trap`<br>[SNMPManagerTrapSendingEnable_x] | Enables the sending of traps to the corresponding SNMP manager.<br>▪ **[0]** Disable = Sending is disabled.<br>▪ **[1]** Enable = (Default) Sending is enabled. |
| Trap User<br>`trap-user`<br>[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).<br>The valid value is a string. |
| Trap Manager Host Name<br>`manager-host-name`<br>[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'.<br>The valid range is a string of up to 99 characters. |
| **SNMP Community String Parameters** | |
| Community String - Read Only<br>`configure system > snmp > ro-community-string`<br>[SNMPReadOnlyCommunityString_x] | Defines a read-only SNMP community string. Up to five read-only community strings can be configured.<br>The valid value is a string of up to 19 characters that can include only the following:<br>▪ Upper- and lower-case letters (a to z, and A to Z)<br>▪ Numbers (0 to 9)<br>▪ Hyphen (-)<br>▪ Underline (_)<br>For example, "Public-comm_string1".<br>The default is "public". |

| Parameter | Description |
|---|---|
| Community String - Read / Write<br>`configure system > snmp > rw-community-string`<br>[SNMPReadWriteCommunityString_x] | Defines a read-write SNMP community string. Up to five read-write community strings can be configured.<br>The valid value is a string of up to 19 characters that can include only the following:<br>▪ Upper- and lower-case letters (a to z, and A to Z)<br>▪ Numbers (0 to 9)<br>▪ Hyphen (-)<br>▪ Underline (_)<br>For example, "Private-comm_string1".<br>The default is "private". |
| Trap Community String<br>`configure system > snmp trap > community-string`<br>[SNMPTrapCommunityString] | Defines the community string for SNMP traps.<br>The valid value is a string of up to 19 characters that can include only the following:<br>▪ Upper- and lower-case letters (a to z, and A to Z)<br>▪ Numbers (0 to 9)<br>▪ Hyphen (-)<br>▪ Underline (_)<br>For example, "Trap-comm_string1".<br>The default is "trapuser". |
| **SNMP Trusted Managers Table** | |
| SNMP Trusted Managers<br>`configure system > snmp > trusted-managers`<br>[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.<br>**Notes:**<br>▪ 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.<br>▪ If no values are assigned to these parameters any manager can access the device.<br>▪ Trusted managers can work with all community strings. |
| **SNMP V3 Users Table** | |
| SNMP V3 Users<br>`configure system > snmp v3-users`<br>[SNMPUsers] | The p*arame*ter table defines SNMP v3 users.<br>The format of the ini file table parameter is:<br>[SNMPUsers]<br>FORMAT SNMPUsers_Index = SNMPUsers_Username, SNMPUsers_AuthProtocol, SNMPUsers_PrivProtocol, SNMPUsers_AuthKey, SNMPUsers_PrivKey, SNMPUsers_Group;<br>[\SNMPUsers]<br>For example:<br>SNMPUsers 1 = v3admin1, 1, 0, myauthkey, -, 1;<br>The example above configures user 'v3admin1' with security level authNoPriv(2), authentication protocol MD5, authentication text password 'myauthkey', and ReadWriteGroup2.<br>For a description of the table, see "Configuring SNMP V3 Users" on page 92. |

### 45.1.6 Serial Parameters

The serial interface parameters are described in the table below.

**Table 45-6: Serial Parameters**

| Parameter | Description |
|---|---|
| **[DisableRS232]** | Enables the device's RS-232 (serial) port.<br>▪ [0] = Enabled<br>▪ [1] = (Default) Disabled<br>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.<br>**Note:** For the parameter to take effect, a device reset is required. |
| **[SerialBaudRate]** | Defines the serial communication baud rate.<br>The valid values include the following: 1200, 2400, 9600, 14400, 19200, 38400, 57600, or 115200 (default).<br>**Note:** For the parameter to take effect, a device reset is required. |
| **[SerialData]** | Defines the serial communication data bit.<br>▪ **[7]** = 7-bit<br>▪ **[8]** = (Default) 8-bit<br>**Note:** For the parameter to take effect, a device reset is required. |
| **[SerialParity]** | Defines the serial communication polarity.<br>▪ **[0]** = (Default) None<br>▪ **[1]** = Odd<br>▪ **[2]** = Even<br>**Note:** For the parameter to take effect, a device reset is required. |
| **[SerialStop]** | Defines the serial communication stop bit.<br>▪ **[1]** = (Default) 1-bit (default)<br>▪ **[2]** = 2-bit<br>**Note:** For the parameter to take effect, a device reset is required. |
| **[SerialFlowControl]** | Defines the serial communication flow control.<br>▪ **[0]** = (Default) None<br>▪ **[1]** = Hardware<br>**Note:** For the parameter to take effect, a device reset is required. |

### 45.1.7 Auxiliary and Configuration File Name Parameters

The configuration files (i.e., Auxiliary files) can be loaded to the device using the Web interface. For loading these files using the *ini* file, you need to configure these files in the *ini* file and configured whether they must be stored in the non-volatile memory. The table below lists the *ini* file parameters associated with these Auxiliary files. For more information on Auxiliary files, see "Loading Auxiliary Files" on page 493.

**Table 45-7: Auxiliary and Configuration File Parameters**

| Parameter | Description |
|---|---|
| **General Parameters** | |
| [SetDefaultOnIniFileProcess] | Determines if all the device's parameters are set to their defaults before processing the updated *ini* file.<br>▪ **[0]** = Disable - parameters not included in the downloaded *ini* file are not returned to default settings (i.e., retain their current settings).<br>▪ **[1]** = Enable (default).<br>**Note:** The parameter is applicable only for automatic HTTP update or Web *ini* file upload (not applicable if the *ini* file is loaded using BootP). |
| [SaveConfiguration] | Determines if the device's configuration (parameters and files) is saved to flash (non-volatile memory).<br>▪ **[0]** = Configuration isn't saved to flash memory.<br>▪ **[1]** = (Default) Configuration is saved to flash memory. |
| **Auxiliary and Configuration File Name Parameters** | |
| Call Progress Tones File<br>[CallProgressTonesFilename] | Defines the name of the file containing the Call Progress Tones definitions.<br>For the ini file, the name must be enclosed by single apostrophes, for example, 'cpt_us.dat'.<br>For more information on how to create and load this file, refer to *DConvert Utility User's Guide*.<br>**Note:** For the parameter to take effect, a device reset is required. |
| Prerecorded Tones File<br>[PrerecordedTonesFileName] | Defines the name of the file containing the Prerecorded Tones.<br>**Note:** For the parameter to take effect, a device reset is required. |
| Dial Plan<br>[CasTrunkDialPlanName_x] | Defines the Dial Plan name (up to 11-character strings) per trunk.<br>For the ini file, the name must be enclosed by single apostrophes, for example, 'dial_plan_2.dat'.<br>**Note:** The x in the ini file parameter name denotes the trunk number, where 0 is Trunk 1. |
| Dial Plan File<br>[DialPlanFileName] | Defines the name of the Dial Plan file. This file should be created using AudioCodes DConvert utility (refer to *DConvert Utility User's Guide*).<br>For the ini file, the name must be enclosed by single apostrophes, for example, 'dial_plan.dat'. |
| [UserInfoFileName] | Defines the name of the file containing the User Information data.<br>For the ini file, the name must be enclosed by single apostrophes, for example, 'userinfo_us.dat'. |

### 45.1.8 Automatic Update Parameters

The automatic update of software and configuration files parameters are described in the table below.

**Table 45-8: Automatic Update of Software and Configuration Files Parameters**

| Parameter | Description |
|---|---|
| **General Automatic Update Parameters** | |
| `configure system/automatic-update/update-firmware`<br>[AutoUpdateCmpFile] | Enables the Automatic Update mechanism for the cmp file.<br>▪ **[0]** = (Default) The Automatic Update mechanism doesn't apply to the cmp file.<br>▪ **[1]** = The Automatic Update mechanism includes the cmp file.<br>**Note:** For the parameter to take effect, a device reset is required. |
| `configure system > automatic-update > update-frequency`<br>[AutoUpdateFrequency] | Defines the interval (in minutes) that the device waits between consecutive automatic updates.<br>The default is 0 (i.e., the update at fixed intervals mechanism is disabled).<br>**Note:** For the parameter to take effect, a device reset is required. |
| `configure system > automatic-update > predefined-time`<br>[AutoUpdatePredefinedTime] | Defines schedules (time of day) for performing automatic updates.<br>The format syntax of the parameter is 'hh:mm', where *hh* denotes the hour and *mm* the minutes. The value must be enclosed in single apostrophes. For example, '20:18'.<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ The actual update time is randomized by five minutes to reduce the load on the Web servers. |
| `automatic-update > http-user-agent`<br>[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.<br>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):<br>▪ <NAME>: product name, according to the installed Software License Key<br>▪ <MAC>: device's MAC address<br>▪ <VER>: software version currently installed on the device, e.g., "7.00.200.001"<br>▪ <CONF>: configuration version, as configured by the ini file parameter, INIFileVersion or CLI command, configuration-version<br>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:<br>User-Agent: Mozilla/4.0 (compatible; AudioCodes; <NAME>;<VER>;<MAC>;<CONF>)<br>For example, if you set AupdHttpUserAgent = MyWorld-<NAME>;<VER>(<MAC>), the device sends the following User-Agent header:<br>User-Agent: MyWorld-Mediant;7.00.200.001(00908F1DD0D3)<br>**Notes:**<br>▪ The variable tags are case-sensitive.<br>▪ 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. |

| Parameter | Description |
|---|---|
| | ▪ The tags can be defined in any order.<br>▪ The tags must be defined adjacent to one another (i.e., no spaces). |
| `automatic-update > auto-firmware`<br>[AutoCmpFileUrl] | Defines the filename and path (URL) to the provisioning server from where the software file (.cmp) can be downloaded, based on timestamp for the Automatic Updated mechanism.<br>The valid value is an IP address in dotted-decimal notation or an FQDN. |
| `system > tls > aupd-verify-cert`<br>[AUPDVerifyCertificates] | Determines whether the Automatic Update mechanism verifies server certificates when using HTTPS.<br>▪ **[0]** = Disable (default)<br>▪ **[1]** = Enable |
| [AUPDDigestUsername] | Defines the username for digest (MD5 cryptographic hashing) access authentication with the HTTP server used for the Automatic Update feature.<br>The valid value is a string of up to 50 characters. By default, no value is defined. |
| [AUPDDigestPassword] | Defines the password for digest (MD5 cryptographic hashing) access authentication with the HTTP server used for the Automatic Update feature.<br>The valid value is a string of up to 50 characters. By default, no value is defined. |
| `configure system > automatic-update > crc-check regular`<br>[AUPDCheckIfIniChanged] | Enables the device to perform cyclic redundancy checks (CRC) on downloaded configuration files (ini) during the Automatic Update process. The CRC checks whether the content (raw data) of the downloaded file is different to the content of the previously downloaded file from the previous Automatic Update process. The device compares the CRC check value (code) result with the check value of the previously downloaded file. If the check values are identical, it indicates that the file has no new configuration settings, and the device discards the file. If the check values are different, the device installs the downloaded file and applies the new configuration settings.<br>▪ **[0]** = (Default) Disable - the device does not perform CRC and installs the downloaded file regardless.<br>▪ **[1]** = Enable CRC for the entire file, including line order (i.e., same text must be on the same lines). If there are differences between the files, the device installs the downloaded file. If there are no differences, the device discards the newly downloaded file.<br>▪ **[2]** = Enable CRC for individual lines only. Same as option [1], except that the CRC ignores the order of lines (i.e., same text can be on different lines). |
| `config-system > automatic-update tftp-block-size`<br>[AUPDTftpBlockSize] | Defines the size of the TFTP data blocks (packets) when downloading a file from a TFTP server for the Automatic Update mechanism. This is in accordance to RFC 2348. TFTP block size is the physical packet size (in bytes) that a network can transmit. When configured to a value higher than the default (512 bytes), but lower than the client network's Maximum Transmission Unit (MTU), the file download speed can be significantly increased. |

| Parameter | Description |
|---|---|
| | The valid value is 512 to 8192. The default is 512.<br>**Notes:**<br>▪ A higher value does not necessarily mean better performance.<br>▪ The block size should be small enough to avoid IP fragmentation in the client network (i.e., below MTU).<br>▪ This feature is applicable only to TFTP servers that support this option. |
| [ResetNow] | Invokes an immediate device reset. This option can be used to activate offline (i.e., not on-the-fly) parameters that are loaded using the parameter IniFileUrl.<br>▪ **[0]** = (Default) The immediate restart mechanism is disabled.<br>▪ **[1]** = The device immediately resets after an *ini* file with the parameter set to 1 is loaded.<br>**Note:** If you use the parameter in an ini file for periodic automatic provisioning with non-HTTP (e.g., TFTP) and without CRC, the device resets upon every file download. |
| **Software/Configuration File URL Path for Automatic Update Parameters** | |
| `automatic-update > firmware`<br>[CmpFileURL] | Defines the name of the *cmp* file and the path to the server (IP address or FQDN) from where the device can load the *cmp* file and update itself. The *cmp* file can be loaded using HTTP/HTTPS. For example, http://192.168.0.1/filename.<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ When the parameter is configured, the device always loads the *cmp* file after it is reset.<br>▪ 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.<br>▪ The maximum length of the URL address is 255 characters. |
| `voice-configuration`<br>[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.<br>For example:<br>http://192.168.0.1/filename<br>http://192.8.77.13/config_<MAC>.ini<br>https://<username>:<password>@<IP address>/<file name><br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ When using HTTP or HTTPS, the date and time of the *ini* file are validated. Only more recently dated *ini* files are loaded.<br>▪ The case-sensitive string, "<MAC>" can be used in the file name for instructing the device to replace it with the device's MAC address. For more information, see "MAC Address Placeholder in Configuration File Name" on page 529. This option allows the loading of specific configurations for specific devices.<br>▪ The maximum length of the URL address is 99 characters. |
| cli-script <URL><br>`[AUPDCliScriptURL]` | Defines the URL of the server where the CLI Script file containing the device's configuration is located. This file is used for automatic provisioning. |

| Parameter | Description |
|---|---|
| | **Note:** The case-sensitive string, "<MAC>" can be used in the file name for instructing the device to replace it with the device's MAC address. For more information, see MAC Address Placeholder in Configuration File Name on page 529. |
| `prerecorded-tones`<br>[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.<br>For example: http://server_name/file, https://server_name/file.<br>**Note:** The maximum length of the URL address is 99 characters. |
| `call-progress-tones`<br>[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.<br>**Note:** The maximum length of the URL address is 99 characters. |
| `tls-root-cert`<br>[TLSRootFileUrl] | Defines the name of the TLS trusted root certificate file and the URL from where it can be downloaded.<br>**Note:** For the parameter to take effect, a device reset is required. |
| `tls-cert`<br>[TLSCertFileUrl] | Defines the name of the TLS certificate file and the URL from where it can be downloaded.<br>**Note:** For the parameter to take effect, a device reset is required. |
| `tls-private-key`<br>[TLSPkeyFileUrl] | Defines the URL for downloading a TLS private key file using the Automatic Update facility. |
| `user-info`<br>[UserInfoFileURL] | Defines the name of the User Information file and the path to the server (IP address or FQDN) on which it is located.<br>For example: http://server_name/file, https://server_name/file<br>**Note:** The maximum length of the URL address is 99 characters. |
| `configure system > automatic-update > feature-key`<br>[FeatureKeyURL] | Defines the name of the License Key file and the URL address of the server on which the file is located. |
| `configure system > automatic-update > template-url`<br>`[TemplateUrl]` | Defines the URL address in the File Template for automatic updates, of the provisioning server on which the files to download are located.<br>For more information, see File Template for Automatic Provisioning on page 530. |
| `configure system > automatic-update > template-files-list`<br>`[AupdFilesList]` | Defines the list of file types in the File Template for automatic updates, to download from the provisioning server.<br>For more information, see File Template for Automatic Provisioning on page 530. |
| `web-favicon`<br>[WebFaviconFileUrl] | Defines the name of the favicon image file and the URL address of the server on which the file is located. This is used for the Automatic Update feature.<br>For more information, see Customizing the Favicon on page 60. |

## 45.2 Networking Parameters

This subsection describes the device's networking parameters.

### 45.2.1 Ethernet Parameters

The Ethernet parameters are described in the table below.

**Table 45-9: Ethernet Parameters**

| Parameter | Description |
|---|---|
| Physical Ports Settings Table | |
| Physical Ports Settings<br>`configure voip/physical-port`<br>[PhysicalPortsTable] | The table configures the physical Ethernet ports.<br>The format of the ini file table parameter is as follows:<br>[ PhysicalPortsTable ]<br>FORMAT PhysicalPortsTable_Index = PhysicalPortsTable_Port, PhysicalPortsTable_Mode, PhysicalPortsTable_SpeedDuplex, PhysicalPortsTable_PortDescription, PhysicalPortsTable_GroupMember, PhysicalPortsTable_GroupStatus;<br>[ \PhysicalPortsTable ]<br>For a detailed description of the table, see Configuring Physical Ethernet Ports on page 127. |
| Ethernet Group Settings Table | |
| Ethernet Group Settings<br>`configure voip/ether-group`<br>[EtherGroupTable] | Defines the transmit (Tx) and receive (Rx) settings for the Ethernet port groups. The format of the ini file table parameter is:<br>[EtherGroupTable]<br>FORMAT EtherGroupTable_Index = EtherGroupTable_Group, EtherGroupTable_Mode, EtherGroupTable_Member1, EtherGroupTable_Member2;<br>[\EtherGroupTable]<br>For a detailed description of the table, see Configuring Ethernet Port Groups on page 129.<br>**Note:** For the parameter to take effect, a device reset is required. |
| Ethernet Device Table | |
| Ethernet Device Table<br>[DeviceTable] | Defines Ethernet Devices (VLANs).<br>The format of the ini file table parameter is as follows:<br>[ DeviceTable ]<br>FORMAT DeviceTable_Index = DeviceTable_VlanID, DeviceTable_UnderlyingInterface, DeviceTable_DeviceName, DeviceTable_Tagging;<br>[ \DeviceTable ]<br>For a detailed description of the table, see Configuring Underlying Ethernet Devices on page 132. |

### 45.2.2 Multiple VoIP Network Interfaces and VLAN Parameters

The IP network interfaces and VLAN parameters are described in the table below.

**Table 45-10: IP Network Interfaces and VLAN Parameters**

| Parameter | Description |
|---|---|
| **Interface Table** | |

| Parameter | Description |
|---|---|
| Interface Table<br>`configure voip > interface network-if display`<br>[InterfaceTable] | The table configures the Interface table.<br>The format of the ini file table parameter is as follows:<br>[InterfaceTable]<br>FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes, InterfaceTable_InterfaceMode, InterfaceTable_IPAddress, InterfaceTable_PrefixLength, InterfaceTable_Gateway, InterfaceTable_VlanID, InterfaceTable_InterfaceName, InterfaceTable_PrimaryDNSServerIPAddress, InterfaceTable_SecondaryDNSServerIPAddress, InterfaceTable_UnderlyingDevice;<br>[\InterfaceTable]<br>For a detailed description of the table, see "Configuring IP Network Interfaces" on page 135. |
| [EnableNTPasOAM] | Defines the application type for Network Time Protocol (NTP) services.<br>▪ **[1]** = OAMP (default)<br>▪ **[0]** = Control<br>**Note:** For the parameter to take effect, a device reset is required. |

## 45.2.3 Routing Parameters

The IP network routing parameters are described in the table below.

**Table 45-11: IP Network Routing Parameters**

| Parameter | Description |
|---|---|
| Send ICMP Unreachable Messages<br>[DisableICMPUnreachable] | Enables sending of ICMP Unreachable messages.<br>▪ [0] Enable = (Default) Device sends these messages.<br>▪ [1] Disable = Device does not send these messages. |
| Send and Receive ICMP Redirect Messages<br>[DisableICMPRedirects] | Enables sending and receiving of ICMP Redirect messages.<br>▪ [0] Enable = (Default) Device sends and accepts these messages.<br>▪ [1] Disable = Device rejects these messages and also does not send them. |
| **Static Route Table** | |
| Static Route Table<br>`configure voip > static`<br>[StaticRouteTable] | Defines up to 30 static IP routes for the device.<br>The format of the ini file table parameter is as follows:<br>[ StaticRouteTable ]<br>FORMAT StaticRouteTable_Index = StaticRouteTable_DeviceName, StaticRouteTable_Destination, StaticRouteTable_PrefixLength, StaticRouteTable_Gateway, StaticRouteTable_Description;<br>[ \StaticRouteTable ]<br>For a description of the parameter, see "Configuring Static IP Routes" on page 143. |

### 45.2.4 Quality of Service Parameters

The Quality of Service (QoS) parameters are described in the table below.

**Table 45-12: QoS Parameters**

| Parameter | Description |
|---|---|
| **Layer-2 Class Of Service (CoS) Parameters (VLAN Tag Priority Field)** | |
| DiffServ Table<br>`configure voip > vlan-mapping`<br>[DiffServToVlanPriority] | The table configures DiffServ-to-VLAN Priority mapping. For each packet sent to the LAN, the VLAN Priority of the packet is set according to the DiffServ value in the IP header of the packet.<br>The format of this ini file is as follows:<br>[ DiffServToVlanPriority ]<br>FORMAT DiffServToVlanPriority_Index = DiffServToVlanPriority_DiffServ, DiffServToVlanPriority_VlanPriority;<br>[ \DiffServToVlanPriority ]<br>For example:<br>DiffServToVlanPriority 0 = 46, 6;<br>DiffServToVlanPriority 1 = 40, 6;<br>DiffServToVlanPriority 2 = 26, 4;<br>DiffServToVlanPriority 3 = 10, 2;<br>For a description of the table, see Configuring Quality of Service on page 146.<br>**Note:** For the parameter to take effect, a device reset is required. |
| **Layer-3 Class of Service (TOS/DiffServ) Parameters** | |
| Media Premium QoS<br>`media-qos`<br>[PremiumServiceClassMediaDiffServ] | Global parameter that defines the DiffServ value for Premium Media CoS content. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_IPDiffServ). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| Control Premium QoS<br>`control-qos`<br>[PremiumServiceClassControlDiffServ] | Global parameter that defines the DiffServ value for Premium Control CoS content (Call Control applications). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SigIPDiffServ). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| Gold QoS<br>`gold-qos`<br>[GoldServiceClassDiffServ] | Defines the DiffServ value for the Gold CoS content (Streaming applications).<br>The valid range is 0 to 63. The default is 26. |
| Bronze QoS<br>`bronze-qos` | Defines the DiffServ value for the Bronze CoS content (OAMP applications). |

| Parameter | Description |
|---|---|
| [BronzeServiceClassDiffServ] | The valid range is 0 to 63. The default is 10. |

### 45.2.5 NAT and STUN Parameters

The Network Address Translation (NAT) parameters are described in the table below.

**Table 45-13: NAT Parameters**

| Parameter | Description |
|---|---|
| **NAT Parameters** | |
| NAT Mode<br>`disable-NAT-traversal`<br>[NATMode] | Enables the NAT feature for media when the device communicates with UAs located behind NAT.<br>▪ [0] Enable NAT Option = NAT traversal is performed only if the UA is located behind NAT:<br>✓ UA behind NAT: The device sends the media packets to the IP address:port obtained from the source address of the first media packet received from the UA.<br>✓ UA not behind NAT: The device sends the packets to the IP address:port specified in the SDP 'c=' line (Connection) of the first received SIP message.<br>Note: If the SIP session is established (ACK) and the device (not the UA) sends the first packet, it sends it to the address obtained from the SIP message and only after the device receives the first packet from the UA does it determine whether the UA is behind NAT.<br>▪ [1] Disable NAT = (Default) The device considers the UA as not located behind NAT and sends media packets to the UA using the IP address:port specified in the SDP 'c=' line (Connection) of the first received SIP message.<br>▪ [2] Force NAT = The device always considers the UA as behind NAT and sends the media packets to the IP address:port obtained from the source address of the first media packet received from the UA. The device only sends packets to the UA after it receives the first packet from the UA (to obtain the IP address).<br>For more information on handling calls from UAs behind NAT, see "First Incoming Packet Mechanism" on page 157. |
| NAT IP Address<br>`nat-ip-addr`<br>[StaticNatIP] | Defines the global (public) IP address of the device to enable static NAT between the device and the Internet.<br>**Note:** For the parameter to take effect, a device reset is required. |
| [NATBindingDefaultTimeout] | The device sends SNMP keep-alive traps periodically - every 9/10 of the time configured by the parameter (in seconds). Therefore, the parameter is applicable only if the SendKeepAliveTrap parameter is set to 1.<br>The parameter is used to allow SNMP communication with AudioCodes EMS management platform, located in the WAN, when the device is located behind NAT. It is needed to keep the NAT pinhole open for the SNMP messages sent from EMS to the device.<br><br>The valid range is 0 to 2,592,000. The default is 30. |

| Parameter | Description |
|---|---|
| | **Note:** For the parameter to take effect, a device reset is required. |
| SIP NAT Detection<br>`configure voip/sip-definition advanced-settings/sip-nat-detect`<br>[SIPNatDetection] | Enables the device to detect whether the incoming INVITE message is sent from an endpoint located behind NAT.<br>▪ **[0]** Disable = Disables the device's NAT Detection mechanism. Incoming SIP messages are processed as received from endpoints that are not located behind NAT and sent according to the SIP standard.<br>▪ **[1]** Enable (default) = Enables the device's NAT Detection mechanism. |

### 45.2.6 DNS Parameters

The Domain name System (DNS) parameters are described in the table below.

**Table 45-14: DNS Parameters**

| Parameter | Description |
|---|---|
| **Internal DNS Table** | |
| Internal DNS Table<br>`configure voip > voip-network dns Dns2Ip`<br>[DNS2IP] | The table defines the internal DNS table for resolving host names into IP addresses.<br>The format of the ini file table parameter is:<br>[Dns2Ip]<br>FORMAT Dns2Ip_Index = Dns2Ip_DomainName, Dns2Ip_FirstIpAddress, Dns2Ip_SecondIpAddress, Dns2Ip_ThirdIpAddress, Dns2Ip_FourthIpAddress;<br>[\Dns2Ip]<br>For example:<br>Dns2Ip 0 = DnsName, 1.1.1.1, 2.2.2.2, 3.3.3.3, ;<br>For a detailed description of the table, see "Configuring the Internal DNS Table" on page 149. |
| **Internal SRV Table** | |
| Internal SRV Table<br>`configure voip > voip-network dns Srv2Ip`<br>[SRV2IP] | The table defines the internal SRV table for resolving host names into DNS A-Records. Three different A-Records can be assigned to a host name. Each A-Record contains the host name, priority, weight, and port. The format of the ini file table parameter is:<br>[SRV2IP]<br>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;<br>[\SRV2IP]<br>For example:<br>SRV2IP 0 = SrvDomain,0,Dnsname1,1,1,500,Dnsname2,2,2,501,$$,0,0,0;<br>For a detailed description of the table, see "Configuring the Internal SRV Table" on page 150. |

## 45.2.7 DHCP Parameters

The Dynamic Host Control Protocol (DHCP) parameters are described in the table below.

**Table 45-15: DHCP Parameters**

| Parameter | Description |
|---|---|
| Enable DHCP<br>[DHCPEnable] | Enables DHCP client functionality.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ For a detailed description of DHCP, see "DHCP-based Provisioning" on page 519.<br>▪ The parameter is a "hidden" parameter. Once defined and saved to flash memory, its value doesn't revert to default even if the parameter doesn't appear in the *ini* file. |
| [DHCPSpeedFactor] | Defines the device's DHCP renewal speed for a leased IP address from a DHCP server.<br>▪ **[0]** = Disable<br>▪ **[1]** = (Default) Normal<br>▪ **[2]** to **[10]** = Fast<br>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.<br>**Note:** For the parameter to take effect, a device reset is required. |
| DHCP Servers Table | |
| DHCP Servers Table<br>`configure voip > dhcp server <index>`<br>[DhcpServer] | Defines the device's embedded DHCP server.<br>The format of the ini file table parameter is as follows:<br>[ DhcpServer ]<br>FORMAT DhcpServer_Index = DhcpServer_InterfaceName, DhcpServer_StartIPAddress, DhcpServer_EndIPAddress, DhcpServer_SubnetMask, DhcpServer_LeaseTime, DhcpServer_DNSServer1, DhcpServer_DNSServer2, DhcpServer_NetbiosNameServer, DhcpServer_NetbiosNodeType, DhcpServer_NTPServer1, DhcpServer_NTPServer2, DhcpServer_TimeOffset, DhcpServer_TftpServer, DhcpServer_BootFileName, DhcpServer_ExpandBootfileName, DhcpServer_OverrideRouter, DhcpServer_SipServer, DhcpServer_SipServerType;<br>[ \DhcpServer ]<br>For a detailed description of the table, see Configuring the Device's DHCP Server. |

| Parameter | Description |
|---|---|
| DHCP Vendor Class Table | |
| DHCP Vendor Class table<br>`configure voip > dhcp vendor-class`<br>[DhcpVendorClass] | Defines Vendor Class Identifier (VCI) names (DHCP Option 60) for the device's DHCP server. Only if the DHCPDiscover request message, received from the DHCP client, contains this value does the device provide DHCP services.<br>The format of the ini file table parameter is as follows:<br>[ DhcpVendorClass ]<br>FORMAT DhcpVendorClass_Index = DhcpVendorClass_DhcpServerIndex, DhcpVendorClass_VendorClassId;<br>[ \DhcpVendorClass ]<br>For a detailed description of the table, see Configuring the Vendor Class Identifier on page 218. |
| DHCP Option Table | |
| DHCP Option table<br>`configure voip > dhcp option`<br>[DhcpOption] | Defines additional DHCP Options that the device's DHCP server can use to service its DHCP clients.<br>The format of the ini file table parameter is as follows:<br>[ DhcpOption ]<br>FORMAT DhcpOption_Index = DhcpOption_DhcpServerIndex, DhcpOption_Option, DhcpOption_Type, DhcpOption_Value, DhcpOption_ExpandValue;<br>[ \DhcpOption ]<br>For a detailed description of the table, see Configuring Additional DHCP Options on page 219. |
| DHCP Static IP Table | |
| DHCP Static IP table<br>`configure voip > dhcp static-ip <index>`<br>[DhcpStaticIP] | Defines static "reserved" IP addresses that the device's DHCP server allocates to specific DHCP clients defined by MAC address.<br>The format of the ini file table parameter is as follows:<br>[ DhcpStaticIP ]<br>FORMAT DhcpStaticIP_Index = DhcpStaticIP_DhcpServerIndex, DhcpStaticIP_IPAddress, DhcpStaticIP_MACAddress;<br>[ \DhcpStaticIP ]<br>For a detailed description of the table, see Configuring Static IP Addresses for DHCP Clients on page 221. |

### 45.2.8 NTP and Daylight Saving Time Parameters

The Network Time Protocol (NTP) and daylight saving time parameters are described in the table below.

**Table 45-16: NTP and Daylight Saving Time Parameters**

| Parameter | Description |
|---|---|
| **NTP Parameters**<br>**Note:** For more information on Network Time Protocol (NTP), see "Simple Network Time Protocol Support" on page 119. | |

| Parameter | Description |
|---|---|
| NTP Server Address<br>`primary-server`<br>[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.<br>The default IP address is 0.0.0.0 (i.e., internal NTP client is disabled). |
| NTP Secondary Server Address<br>[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.<br>The default IP address is 0.0.0.0. |
| NTP Update Interval<br>`update-interval`<br>[NTPUpdateInterval] | Defines the time interval (in seconds) that the NTP client requests for a time update.<br>The default interval is 86400 (i.e., 24 hours). The range is 0 to 214783647.<br>**Note:** It is not recommend to set the parameter to beyond one month (i.e., 2592000 seconds). |
| NTP Authentication Key Identifier<br>`configure system > ntp > auth-key-id`<br>[NtpAuthKeyId] | Defines the NTP authentication key identifier for authenticating NTP messages. The identifier must match the value configured on the NTP server. The NTP server may have several keys configured for different clients; this number identifies which key is used.<br>The valid value is 1 to 65535. The default is 0 (i.e., no authentication is done). |
| NTP Authentication Secret Key<br>`configure system > ntp > auth-key-md5`<br>[ntpAuthMd5Key] | Defines the secret authentication key shared between the device (client) and the NTP server, for authenticating NTP messages.<br>The valid value is a string of up to 32 characters. By default, no key is defined. |
| **Regional Clock and Daylight Saving Time Parameters** | |
| UTC Offset<br>`utc-offset`<br>[NTPServerUTCOffset] | Defines the Universal Time Coordinate (UTC) offset (in seconds) from the local time.<br>The valid range is -43200 to 43200. The default is 0.<br>**Note:** The offset setting is applied only on the hour. For example, if you configure the parameter at 15:42, the device applies the setting only at 16:00. |
| Daylight Saving Time<br>`summer-time`<br>[DayLightSavingTimeEnable] | Enables daylight saving time (DST).<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable |
| Start Time / Day of Month Start<br>`start`<br>[DayLightSavingTimeStart] | Defines the date and time when DST begins. This value can be configured using any of the following formats:<br>▪ Day of year - *mm:dd:hh:mm*, where:<br>✓ *mm* denotes month<br>✓ *dd* denotes date of the month<br>✓ *hh* denotes hour<br>✓ *mm* denotes minutes<br>For example, "05:01:08:00" denotes daylight saving starting from May 1 at 8 A.M. |

| Parameter | Description |
|---|---|
| | ▪ Day of month - *mm:day/wk:hh:mm*, where:<br>✓ *mm* denotes month (e.g., 04)<br>✓ *day* denotes day of week (e.g., FRI)<br>✓ *wk* denotes week of the month (e.g., 03)<br>✓ *hh* denotes hour (e.g., 23)<br>✓ *mm* denotes minutes (e.g., 10)<br>For example, "04:FRI/03:23:00" denotes Friday, the third week of April, at 11 P.M. The week field can be 1-5, where 5 denotes the last occurrence of the specified day in the specified month. For example, "04:FRI/05:23:00" denotes the last Friday of April, at 11 P.M. |
| End Time / Day of Month End<br>`end`<br>[DayLightSavingTimeEnd] | Defines the date and time when DST ends. For a description of the format of this value, see the DayLightSavingTimeStart parameter. |
| Offset<br>`offset`<br>[DayLightSavingTimeOffset] | Defines the DST offset (in minutes).<br>The valid range is 0 to 120. The default is 60.<br>**Note:** The offset setting is applied only on the hour. For example, if you configure the parameter at 15:42, the device applies the setting only at 16:00. |

# 45.3 Debugging and Diagnostics Parameters

This subsection describes the device's debugging and diagnostic parameters.

## 45.3.1 General Parameters

The general debugging and diagnostic parameters are described in the table below.

**Table 45-17: General Debugging and Diagnostic Parameters**

| Parameter | Description |
|---|---|
| [EnableDiagnostics] | Determines the method for verifying correct functioning of the different hardware components on the device. On completion of the check and if the test fails, the device sends information on the test results of each hardware component to the Syslog server.<br>▪ **[0]** = (Default) Rapid and Enhanced self-test mode.<br>▪ **[1]** = Detailed self-test mode (full test of DSPs, PCM, Switch, LAN, PHY and Flash).<br>▪ **[2]** = A quicker version of the Detailed self-test mode (full test of DSPs, PCM, Switch, LAN, PHY, but partial test of Flash).<br>**Note:** For the parameter to take effect, a device reset is required. |
| Delay After Reset [sec]<br>`delay-after-reset`<br>**[GWAppDelayTime]** | Defines the time interval (in seconds) that the device's operation is delayed after a reset.<br>The valid range is 0 to 45. The default is 7 seconds. |

| Parameter | Description |
|---|---|
| | **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.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable |

### 45.3.2 SIP Test Call Parameters

The SIP Signaling Test Call parameters are described in the table below.

**Table 45-18: SIP Test Call Parameters**

| Parameter | Description |
|---|---|
| Test Call ID<br>`testcall-id`<br>[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.<br>This can be any string of up to 15 characters. By default, no number is defined.<br>**Note:** The parameter is only for testing incoming calls destined to this prefix number. |
| **Test Call Table** | |
| Test Call Table<br>`configure system > test-call > test-call-table`<br>[Test_Call] | Defines Test Call rules.<br>[ Test_Call ]<br>FORMAT Test_Call_Index = Test_Call_EndpointURI, Test_Call_CalledURI, Test_Call_RouteBy, Test_Call_IPGroupName, Test_Call_DestAddress, Test_Call_DestTransportType, Test_Call_SIPInterfaceName, Test_Call_ApplicationType, Test_Call_AutoRegister, Test_Call_UserName, Test_Call_Password, Test_Call_CallParty, Test_Call_MaxChannels, Test_Call_CallDuration, Test_Call_CallsPerSecond, Test_Call_TestMode, Test_Call_TestDuration, Test_Call_Play, Test_Call_ScheduleInterval, Test_Call_QOEProfile, Test_Call_BWProfile;<br>[ \Test_Call ]<br>For a description of the table, see "Configuring Test Call Endpoints" on page 627. |

### 45.3.3 Syslog, CDR and Debug Parameters

The Syslog, CDR and debug parameters are described in the table below.

**Table 45-19: Syslog, CDR and Debug Parameters**

| Parameter | Description |
|---|---|
| Enable Syslog<br>`syslog`<br>[EnableSyslog] | Determines whether the device sends logs and error messages (e.g., CDRs) generated by the device to a Syslog server.<br>▪ **[0]** Disable (default) |

| Parameter | Description |
|---|---|
| | ▪ **[1]** Enable<br>**Notes:**<br>▪ If you enable Syslog, you must enter an IP address of the Syslog server (using the SyslogServerIP parameter).<br>▪ Syslog messages may increase the network traffic.<br>▪ To configure Syslog SIP message logging levels, use the GwDebugLevel parameter.<br>▪ 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. |
| Syslog Server IP Address<br>`syslog-ip`<br>[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.<br>The default IP address is 0.0.0.0. |
| Syslog Server Port<br>`syslog-port`<br>[SyslogServerPort] | Defines the UDP port of the Syslog server.<br>The valid range is 0 to 65,535. The default port is 514. |
| CDR Server IP Address<br>`cdr-srvr-ip-adrr`<br>[CDRSyslogServerIP] | Defines the destination IP address to where CDR logs are sent.<br>The default value is a null string, which causes CDR messages to be sent with all Syslog messages to the Syslog server.<br>**Notes:**<br>▪ The CDR messages are sent to UDP port 514 (default Syslog port).<br>▪ This mechanism is active only when Syslog is enabled (i.e., the parameter EnableSyslog is set to 1). |
| CDR Report Level<br>`cdr-report-level`<br>[CDRReportLevel] | Enables media and signaling-related CDRs to be sent to a Syslog server and determines the call stage at which they are sent.<br>▪ **[0]** None = (Default) CDRs are not used.<br>▪ **[1]** End Call = CDR is sent to the Syslog server at the end of each call.<br>▪ **[2]** Start & End Call = CDR report is sent to Syslog at the start and end of each call.<br>▪ **[3]** Connect & End Call = CDR report is sent to Syslog at connection and at the end of each call.<br>▪ **[4]** Start & End & Connect Call = CDR report is sent to Syslog at the start, at connection, and at the end of each call.<br>**Notes:**<br>▪ The CDR Syslog message complies with RFC 3164 and is identified by: Facility = 17 (local1) and Severity = 6 (Informational).<br>▪ This mechanism is active only when Syslog is enabled (i.e., the parameter EnableSyslog is set to 1). |
| CDR Local Max File Size<br>`configure voip > services cdr > cdr-local-max-file-size` | Defines the size (in kilobytes) of each stored CDR file. Once the file size is reached, the device creates a new file for subsequent CDRs, and so on.<br>The valid value is 100 to 10000. The default is 1024. |

| Parameter | Description |
|---|---|
| [CDRLocalMaxFileSize] | |
| CDR Local Max Num Of Files<br>`configure voip > services cdr > cdr-local-max-files`<br>[CDRLocalMaxNomOfFiles] | Defines the maximum number of stored CDR files. If the maximum number is reached, the device replaces (overwrites) the oldest created file with a subsequent new file, and so on.<br>The valid value is 2 to 4096. The default is 5. |
| CDR Local Interval<br>`configure voip > services cdr > cdr-local-interval`<br>[CDRLocalInterval] | Defines how often (in minutes) the device creates a new CDR file. For example, if configured to 60, it creates a new file every hour. This occurs even if the maximum configured file size has not been reached (see the CDRLocalMaxFileSize parameter). However, if the maximum configured file size has been reached and the interval configured by the parameter has not been reached, a new CDR file is created.<br>The valid value is 2 to 1440. The default is 60. |
| Debug Level<br>`configure system/logging/debug-level`<br>[GwDebugLevel] | Enables Syslog debug reporting and logging level.<br>▪ **[0]** No Debug = (Default) Debug is disabled and Syslog messages are not sent.<br>▪ **[1]** Basic = Sends debug logs of incoming and outgoing SIP messages.<br>▪ **[5]** Detailed = Sends debug logs of incoming and outgoing SIP message as well as many other logged processes. |
| Syslog Optimization<br>`configure system/logging/syslog-optimization`<br>[SyslogOptimization] | Enables the device to accumulate and bundle multiple debug messages into a single UDP packet and then send it to a Syslog server. The benefit of this feature is that it reduces the number of UDP Syslog packets, thereby improving (optimizing) CPU utilization.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Note:** The size of the bundled message is configured by the MaxBundleSyslogLength parameter. |
| `mx-syslog-lgth`<br>[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.<br>The valid value range is 0 to 1220 (where 0 indicates that no bundling occurs). The default is 1220.<br>**Note:** The parameter is applicable only if the GWDebugLevel parameter is enabled. |
| Syslog CPU Protection<br>`configure system/logging/syslog-cpu-protection`<br>[SyslogCpuProtection] | Enables the protection of the device's CPU resources during debug reporting, ensuring voice traffic is unaffected. If CPU resources drop (i.e., high CPU usage) to a critical level (threshold), the device automatically lowers the debug level to free up CPU resources that were required for the previous debug-level functionality. When sufficient CPU resources become available again, the device increases the debug level. The threshold is configured by the 'Debug Level High Threshold' parameter (see below).<br>▪ **[0]** Disable<br>▪ **[1]** Enable (default) |
| Debug Level High Threshold | Defines the threshold (in percentage) for automatically switching to a different debug level, depending on CPU usage. The parameter |

| Parameter | Description |
|---|---|
| `debug-level-high-threshold`<br>[DebugLevelHighThreshold] | is applicable only if the 'Syslog CPU Protection' parameter is enabled.<br>The valid value is 0 to 100. The default is 90.<br>The debug level is changed upon the following scenarios:<br>▪ CPU usage equals threshold: Debug level is reduced one level.<br>▪ CPU usage is at least 5% greater than threshold: Debug level is reduced another level.<br>▪ CPU usage is 5 to 19% less than threshold: Debug level is increased by one level.<br>▪ CPU usage is at least 20% less than threshold: Debug level is increased by another level.<br>For example, assume that the threshold is set to 70% and the Debug Level to Detailed (5). When CPU usage reaches 70%, the debug level is reduced to Basic (1). When CPU usage increases by 5% or more than the threshold (i.e., greater than 75%), the debug level is disabled - No Debug (0). When the CPU usage decreases to 5% less than the threshold (e.g., 65%), the debug level is increased to Basic (1). When the CPU usage decreases to 20% less than the threshold (e.g., 50%), the debug level changes to Detailed (5).<br>**Note:** The device does not increase the debug level to a level that is higher than what you configured for the 'Debug Level' parameter. |
| Syslog Facility Number<br>[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.<br>▪ **[16]** = (Default) local use 0 (local0)<br>▪ **[17]** = local use 1 (local1)<br>▪ **[18]** = local use 2 (local2)<br>▪ **[19]** = local use 3 (local3)<br>▪ **[20]** = local use 4 (local4)<br>▪ **[21]** = local use 5 (local5)<br>▪ **[22]** = local use 6 (local6)<br>▪ **[23]** = local use 7 (local7) |
| `CDR Syslog Sequence Number`<br>`cdr-seq-num`<br>[CDRSyslogSeqNum] | Enables or disables the inclusion of the sequence number (S=) in CDR Syslog messages.<br>▪ **[0]** Disable<br>▪ **[1]** Enable (default) |
| `configure voip > sip-definition settings > time-zone-format`<br>`[TimeZoneFormat]` | Defines the time zone that is displayed with the timestamp in CDRs. The timestamp appears in the CDR fields "Setup Time", "Connect Time", and "Release Time".<br>The valid value is a string of up to six characters. The default is UTC. For example, if you configure the parameter |

| Parameter | Description |
|---|---|
| | TimeZoneFormat = GMT+11, the timestamp in CDRs are generated with the following time zone display:<br>`17:47:45.411 GMT+11 Sun Jan 03 2018`<br>**Note:** The time zone is only for display purposes; it does not configure the actual time zone. |
| Activity Types to Report via Activity Log Messages<br>`config-system > logging > activity-log`<br>[ActivityListToLog] | Defines the operations (activities) performed in the Web interface that are reported to a Syslog server.<br>▪ **[pvc]** Parameters Value Change = Changes made on-the-fly to parameters and tables, and Configuration file load. Note that the *ini* file parameter, EnableParametersMonitoring can also be used to set this option.<br>▪ **[afl]** Auxiliary Files Loading = Loading of Auxiliary files.<br>▪ **[dr]** Device Reset = Resetting of the device through the Maintenance Actions page.<br>**Note:** For this option to take effect, a device reset is required.<br>▪ **[fb]** Flash Memory Burning = Saving configuration with burn to flash (in the Maintenance Actions page).<br>▪ **[swu]** Device Software Update = Software updates (i.e., loading of cmp file) through the Software Upgrade Wizard.<br>▪ **[ard]** Access to Restricted Domains = Access to restricted Web pages:<br>✓ (1) ini parameters (AdminPage)<br>✓ (2) General Security Settings<br>✓ (3) Configuration File<br>✓<br>✓ (5) Software Upgrade Key Status<br>✓<br>✓ (7) Web & Telnet Access List<br>✓ (8) Web User Accounts<br>▪ **[naa]** Non-Authorized Access = Attempts to log in to the Web interface with a false or empty username or password.<br>▪ **[spc]** Sensitive Parameters Value Change = Changes made to "sensitive" parameters:<br>✓ (1) IP Address<br>✓ (2) Subnet Mask<br>✓ (3) Default Gateway IP Address<br>✓ (4) ActivityListToLog<br>▪ **[ll]** Login and Logout = Web login and logout attempts.<br>▪ **[cli]** = CLI commands entered by the user.<br>▪ **[ae]** Action Executed = Logs user actions that are not related to parameter changes. The actions can include, for example, file uploads, file delete, lock-unlock maintenance actions, LDAP clear cache, register-unregister, and start-stop trunk. In the Web, these actions are typically done by clicking a button (e.g., the **LOCK** button).<br>**Note:** For the *ini* file parameter, enclose values in single quotation marks, for example: ActivityListToLog = 'pvc', 'afl', 'dr', 'fb', 'swu', 'ard', 'naa', 'spc'. |
| Activity Trap<br>`activity-trap`<br>[EnableActivityTrap] | Enables the device to send an SNMP trap to notify of Web user activities in the Web interface. The activities to report are configured by the ActivityListToLog parameter.<br>▪ [0] Disable (default) |

| Parameter | Description |
|---|---|
| | ▪ [1] Enable |
| [EnableParametersMonitoring] | Enables the monitoring, through Syslog messages, of parameters that are modified on-the-fly.<br>▪ **[0]** = (Default) Disable<br>▪ **[1]** = Enable |
| Debug Recording Destination IP<br>`configure system > logging > dbg-rec-dest-ip`<br>[DebugRecordingDestIP] | Defines the IP address of the server for capturing debug recording. |
| Debug Recording Destination Port<br>`configure system > logging > dbg-rec-dest-port`<br>[DebugRecordingDestPort] | Defines the UDP port of the server for capturing debug recording. The default is 925. |
| Enable Core Dump<br>[EnableCoreDump] | Enables the automatic generation of a Core Dump file upon a device crash.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Note:** For the parameter to take effect, a device reset is required. |
| Core Dump Destination IP<br>[CoreDumpDestIP] | Defines the IP address of the remote server where you want the device to send the Core Dump file.<br>By default, no IP address is defined. |
| **Logging Filters Table** | |
| Logging Filters Table<br>`configure system > logging > logging-filters`<br>[LoggingFilters] | The table defines logging filtering rules for Syslog messages and debug recordings.<br>The format of the ini file table parameter is:<br>[ LoggingFilters ]<br>FORMAT LoggingFilters_Index = LoggingFilters_FilterType, LoggingFilters_Value, LoggingFilters_LogDestination, LoggingFilters_CaptureType, LoggingFilters_Mode;<br>[ \LoggingFilters ]<br>For a detailed description of the table, see "Configuring Log Filter Rules" on page 602. |
| **Gateway CDR Format Table** | |
| Gateway CDR Format Table<br>`configure voip > services cdr > cdr-format gw-cdr-format`<br>[GWCDRFormat] | The table defines CDR customization rules for Gateway calls.<br>The format of the ini file table parameter is:<br>[ GWCDRFormat ]<br>FORMAT GWCDRFormat_Index = GWCDRFormat_CDRType, GWCDRFormat_ColumnType, GWCDRFormat_Title, GWCDRFormat_RadiusType, GWCDRFormat_RadiusID;<br>[ \GWCDRFormat ] |

| Parameter | Description |
|---|---|
| | For a detailed description of the table, see Customizing CDRs for Gateway Calls on page 581. |

### 45.3.4 Resource Allocation Indication Parameters

The Resource Allocation Indication (RAI) parameters are described in the table below.

**Table 45-20: RAI Parameters**

| Parameter | Description |
|---|---|
| **[EnableRAI]** | Enables Resource Available Indication (RAI) alarm generation if the device's busy endpoints exceed a user-defined threshold, configured by the RAIHighThreshold parameter. When enabled and the threshold is crossed, the device sends the SNMP trap, acBoardCallResourcesAlarm.<br>▪ **[0]** = (Default) Disable<br>▪ **[1]** = Enable<br>**Note:** For the parameter to take effect, a device reset is required. |
| **[RAIHighThreshold]** | Defines the high threshold percentage of total calls that are active (busy endpoints). When the percentage of the device's busy endpoints exceeds this high threshold, the device sends the SNMP acBoardCallResourcesAlarm alarm trap with a 'major' alarm status.<br>The range is 0 to 100. The default is 90.<br>**Note:** The percentage of busy endpoints is calculated by dividing the number of busy endpoints by the total number of “enabled” endpoints (trunks are physically connected and synchronized with no alarms and endpoints are defined in the Hunt Group table). |
| **[RAILowThreshold]** | Defines the low threshold percentage of total calls that are active (busy endpoints).<br>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.<br>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.<br>The valid range is 1 to 200. The default is 10. |

## 45.4 Security Parameters

This subsection describes the device's security parameters.

### 45.4.1 General Security Parameters

The general security parameters are described in the table below.

**Table 45-21: General Security Parameters**

| Parameter | Description |
|---|---|
| **Firewall Table** | |
| Internal Firewall Parameters<br>`configure voip > access-list`<br>[AccessList] | The table defines the device's access list (firewall), which defines network traffic filtering rules.<br>The format of the ini file table parameter is:<br>[AccessList]<br>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;<br>[\AccessList]<br>For example:<br>AccessList 10 = mgmt.customer.com, , , 32, 0, 80, tcp, 1, OAMP, 0, 0, 0, allow;<br>AccessList 22 = 10.4.0.0, , , 16, 4000, 9000, any, 0, , 0, 0, 0, block;<br>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.<br>For a detailed description of the table, see "Configuring Firewall Settings" on page 161. |
| **Media Latching** | |
| Inbound Media Latch Mode<br>`inbound-media-latch-mode`<br>[InboundMediaLatchMode] | Enables the Media Latching feature.<br>▪ [0] Strict = Device latches onto the first original stream (IP address:port). It does not latch onto any other stream during the session.<br>▪ [1] Dynamic = (Default) Device latches onto the first stream. If it receives at least a minimum number of consecutive packets (configured by New<media type>StreamPackets) from a different source(s) and the device has not received packets from the current stream for a user-defined period (TimeoutToRelatch<media type>Msec), it latches onto the next packet received from any other stream. If other packets of a different media type are received from the new stream, based on IP address and SSRC for RTCP/RTP and based on IP address only for T.38, the packet is accepted immediately.<br>Note: If a packet from the original (first latched onto) IP address:port is received at any time, the device latches onto this stream.<br>▪ [2] Dynamic-Strict = Device latches onto the first stream. If it receives at least a minimum number of consecutive packets (configured by New<media type>StreamPackets) all from the same source which is different to the first stream and the device has not received packets from the current stream for a user-defined period (TimeoutToRelatch<media type>Msec), it latches onto the next packet received from any other stream. If other packets of different media type are received from the |

| Parameter | Description |
|---|---|
| | new stream based on IP address and SSRC for RTCP and based on IP address only for T.38, the packet is accepted immediately. Note: If a packet from the original (first latched onto) IP address:port is received at any time, the device latches onto this stream.<br>▪ [3] Strict-On-First = Typically used for NAT, where the correct IP address:port is initially unknown. The device latches onto the stream received in the first packet. The device does not change this stream unless a packet is later received from the original source. |
| New RTP Stream Packets<br>[NewRtpStreamPackets] | Defines the minimum number of continuous RTP packets received by the device's channel to allow latching onto the new incoming stream.<br>The valid range is 0 to 20. The default is 3. If set to 0, the device is left exposed to attacks against multiple packet streams. |
| New RTCP Stream Packets<br>[NewRtcpStreamPackets] | Defines the minimum number of continuous RTCP packets received by the device's channel to allow latching onto the new incoming stream.<br>The valid range is 0 to 20. The default is 3. If set to 0, the device is left exposed to attacks against multiple packet streams. |
| New SRTP Stream Packets<br>[NewSRTPStreamPackets] | Defines the minimum number of continuous SRTP packets received by the device's channel to allow latching onto the new incoming stream.<br>The valid range is 0 to 20. The default is 3. If set to 0, the device is left exposed to attacks against multiple packet streams. |
| New SRTCP Stream Packets<br>[NewSRTCPStreamPackets] | Defines the minimum number of continuous SRTCP packets received by the device's channel to allow latching onto the new incoming stream.<br>The valid range is 0 to 20. The default is 3. If set to 0, the device is left exposed to attacks against multiple packet streams. |
| Timeout To Relatch RTP<br>[TimeoutToRelatchRTPMsec] | Defines a period (msec) during which if no packets are received from the current RTP session, the channel can re-latch onto another stream.<br>The valid range is any value from 0. The default is 200. |
| Timeout To Relatch SRTP<br>[TimeoutToRelatchSRTPMsec] | Defines a period (msec) during which if no packets are received from the current SRTP session, the channel can re-latch onto another stream.<br>The valid range is any value from 0. The default is 200. |
| Timeout To Relatch Silence<br>[TimeoutToRelatchSilenceMsec] | Defines a period (msec) during which if no packets are received from the current RTP/SRTP session and the channel is in silence mode, the channel can re-latch onto another stream.<br>The valid range is any value from 0. The default is 200. |
| Timeout To Relatch RTCP<br>[TimeoutToRelatchRTCPMsec] | Defines a period (msec) during which if no packets are received from the current RTCP session, the channel can re-latch onto another RTCP stream.<br>The valid range is any value from 0. The default is 10,000. |
| Fax Relay Rx/Tx Timeout<br>[FaxRelayTimeoutSec] | Defines a period (sec) during which if no T.38 packets are received or sent from the current T.38 fax relay session, the channel can re-latch onto another stream. |

| Parameter | Description |
|---|---|
| | The valid range is 0 to 255. The default is 10. |

## 45.4.2 HTTPS Parameters

The Secure Hypertext Transport Protocol (HTTPS) parameters are described in the table below.

**Table 45-22: HTTPS Parameters**

| Parameter | Description |
|---|---|
| Secured Web Connection (HTTPS)<br>`secured-connection`<br>[HTTPSOnly] | Determines the protocol used to access the Web interface.<br>▪ **[0]** HTTP and HTTPS (default).<br>▪ **[1]** HTTPs Only = Unencrypted HTTP packets are blocked.<br>**Note:** For the parameter to take effect, a device reset is required. |
| `https-port`<br>[HTTPSPort] | Defines the local Secured HTTPS port of the device. The parameter allows secure remote device Web management from the LAN. To enable secure Web management from the LAN, configure the desired port.<br>The valid range is 1 to 65535 (other restrictions may apply within this range). The default port is 443.<br>**Note:** For the parameter to take effect, a device reset is required. |
| HTTPS Cipher String<br>`https-cipher-string`<br>[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.<br>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.<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ If the installed Software License Key includes the Strong Encryption feature, the default of the parameter is changed to ‘RC4:EXP’, enabling RC-128bit encryption.<br>▪ The value ‘ALL’ can be configured only if the installed Software License Key includes the Strong Encryption feature. |
| Requires Client Certificates for HTTPS connection<br>`req-client-cert`<br>[HTTPSRequireClientCertificate] | Enables the requirement of client certificates for HTTPS connection.<br>▪ **[0]** Disable = (Default) Client certificates are not required.<br>▪ **[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.<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ For a description on implementing client certificates, see "TLS for Remote Device Management" on page 115. |

## 45.4.3 SRTP Parameters

The Secure Real-Time Transport Protocol (SRTP) parameters are described in the table below.

**Table 45-23: SRTP Parameters**

| Parameter | Description |
|---|---|
| Media Security<br>`media-security-enable`<br>[EnableMediaSecurity] | Enables Secure Real-Time Transport Protocol (SRTP).<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Note:** For the parameter to take effect, a device reset is required. |
| Media Security Behavior<br>`media-sec-bhvior`<br>[MediaSecurityBehaviour] | Global parameter that defines the handling of SRTP (when the EnableMediaSecurity parameter is set to 1). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_MediaSecurityBehaviour). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.<br>**Note:** The parameter is applicable only to the Gateway application. |
| Master Key Identifier (MKI) Size<br>`SRTP-tx-packet-MKI-size`<br>[SRTPTxPacketMKISize] | Global parameter that defines the size (in bytes) of the Master Key Identifier (MKI) in SRTP Tx packets. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_MKISize). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| Symmetric MKI Negotiation<br>`symmetric-mki`<br>[EnableSymmetricMKI] | Global parameter that enables symmetric MKI negotiation. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableSymmetricMKI). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| Offered SRTP Cipher Suites<br>`offer-srtp-cipher` | Defines the offered crypto suites (cipher encryption algorithms) for SRTP.<br>▪ **[0]** All = (Default) All available crypto suites. |

| Parameter | Description |
| --- | --- |
| [SRTPofferedSuites] | ▪ **[1]** AES-CM-128-HMAC-SHA1-80 = device uses AES-CM encryption with a 128-bit key and HMAC-SHA1 message authentication with a 80-bit tag.<br>▪ **[2]** AES-CM-128-HMAC-SHA1-32 = device uses AES-CM encryption with a 128-bit key and HMAC-SHA1 message authentication with a 32-bit tag.<br>▪ [4] ARIA-CM-128-HMAC-SHA1-80 = device uses ARIA encryption algorithm with a 128-bit key and HMAC-SHA1 message authentication with a 32-bit tag.<br>▪ [8] ARIA-CM-192-HMAC-SHA1-80 = device uses ARIA encryption algorithm with a 192-bit key and HMAC-SHA1 message authentication with a 32-bit tag.<br>**Notes:**<br>▪ For enabling ARIA encryption, use the AriaProtocolSupport parameter.<br>▪ The parameter also affects the selection of the crypto in the device's answer. For example, if the device receives an offer with two crypto lines containing HMAC_SHA1_80 and HMAC_SHA_32, it uses the HMAC_SHA_32 key in its SIP 200 OK response if the parameter is set to 2. |
| Aria Protocol Support<br>`ARIA-protocol-support`<br>[AriaProtocolSupport] | Enables ARIA algorithm cipher encryption for SRTP. This is an alternative option to the existing support for the AES algorithm. ARIA is a symmetric key block cipher algorithm standard developed by the Korean National Security Research Institute.<br>▪ [0] Disable (default)<br>▪ [1] Enable<br>**Notes:**<br>▪ To configure the ARIA bit-key encryption size (128 or 192 bit) with HMAC SHA-1 cryptographic hash function, use the SRTPofferedSuites parameter.<br>▪ The ARIA feature is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see Software License Key on page 510. |
| Authentication On Transmitted RTP Packets<br>`RTP-authentication-disable-tx`<br>[RTPAuthenticationDisableTx] | Enables authentication on transmitted RTP packets in a secured RTP session.<br>▪ **[0]** Enable (default)<br>▪ **[1]** Disable |
| Encryption On Transmitted RTP Packets<br>`RTP-encryption-disable-tx` | Enables encryption on transmitted RTP packets in a secured RTP session.<br>▪ **[0]** Enable (default) |

| Parameter | Description |
|---|---|
| [RTPEncryptionDisableTx] | ▪ **[1]** Disable |
| Encryption On Transmitted RTCP Packets<br>`RTCP-encryption-disable-tx`<br>[RTCPEncryptionDisableTx] | Enables encryption on transmitted RTCP packets in a secured RTP session.<br>▪ **[0]** Enable (default)<br>▪ **[1]** Disable |
| SRTP Tunneling Authentication for RTP<br>`configure voip > media security > srtp-tnl-vld-rtp-auth`<br>[SRTPTunnelingValidateRTPRxAuthentication] | Enables validation of SRTP tunneling authentication for RTP.<br>▪ [0] Disable = (Default) The device does not perform any validation and forwards the packets as is.<br>▪ [1] Enable = The device validates the packets (e.g., sequence number) and if successful, forwards the packets. If validation fails, it drops the packets.<br>**Note:**<br>▪ The parameter is applicable only to SRTP-to-SRTP calls and when both endpoints use the same authentication keys.<br>The parameter is applicable only to the SBC application. |
| SRTP Tunneling Authentication for RTCP<br>`configure voip > media security > srtp-tnl-vld-rtcp-auth`<br>[SRTPTunnelingValidateRTCPRxAuthentication] | Enables validation of SRTP tunneling authentication for RTCP.<br>▪ [0] Disable = (Default) The device does not perform any validation and forwards the packets as is.<br>▪ [1] Enable = The device validates the packets (e.g., sequence number) and if successful, forwards the packets. If validation fails, it drops the packets.<br>**Note:**<br>▪ The parameter is applicable only to SRTP-to-SRTP calls and when both endpoints use the same authentication keys.<br>The parameter is applicable only to the SBC application. |
| `srtp-state-behavior-mode`<br>[ResetSRTPStateUponRekey] | Global parameter that enables synchronization of the SRTP state between the device and a server when a new SRTP key is generated upon a SIP session expire. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_ResetSRTPStateUponRekey). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |

### 45.4.4 TLS Parameters

The Transport Layer Security (TLS) parameters are described in the table below.

**Table 45-24: TLS Parameters**

| Parameter | Description |
|---|---|
| **TLS Contexts Table** | |
| TLS Contexts Table<br>`configure system > tls #`<br>[TLSContexts] | Defines SSL/TLS certificates.<br>The format of the ini file table parameter is as follows:<br>[ TLSContexts ]<br>FORMAT TLSContexts_Index = TLSContexts_Name, TLSContexts_TLSVersion, TLSContexts_ServerCipherString, TLSContexts_ClientCipherString, TLSContexts_OcspEnable, TLSContexts_OcspServerPrimary, TLSContexts_OcspServerSecondary, TLSContexts_OcspServerPort, TLSContexts_OcspDefaultResponse;<br>[ \TLSContexts ]<br>For a detailed description of the table, see "Configuring TLS Certificate Contexts" on page 103. |
| TLS Client Re-Handshake Interval<br>`tls-re-hndshk-int`<br>[TLSReHandshakeInterval] | Defines the time interval (in minutes) between TLS Re-Handshakes initiated by the device.<br>The interval range is 0 to 1,500 minutes. The default is 0 (i.e., no TLS Re-Handshake). |
| TLS Mutual Authentication<br>[SIPSRequireClientCertificate] | Defines the device's mode of operation regarding mutual authentication and certificate verification for TLS connections.<br>▪ **[0]** Disable = (Default)<br>✓ Device acts as a client: Verification of the server's certificate depends on the VerifyServerCertificate parameter.<br>✓ Device acts as a server: The device does not request the client certificate.<br>▪ **[1]** Enable =<br>✓ Device acts as a client: Verification of the server certificate is required to establish the TLS connection.<br>✓ Device acts as a server: The device requires the receipt and verification of the client certificate to establish the TLS connection.<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ This feature can be configured per SIP Interface (see "Configuring SIP Interfaces" on page 319).<br>▪ The SIPS certificate files can be changed using the parameters HTTPSCertFileName and HTTPSRootFileName. |
| Peer Host Name Verification Mode<br>[PeerHostNameVerificationMode] | Determines whether the device verifies the Subject Name of a remote certificate when establishing TLS connections.<br>▪ **[0]** Disable (default).<br>▪ **[1]** Server Only = Verify Subject Name only when acting as a client for the TLS connection. |

| Parameter | Description |
|---|---|
| | ▪ **[2]** Server & Client = Verify Subject Name when acting as a server or client for the TLS connection.<br>When the device receives a remote certificate and the parameter is not disabled, the IP address from which the certificate is received is compared with the addresses defined for the Proxy Sets. If no Proxy Set with the source address is found, the connection is refused. Otherwise, the value of SubjectAltName field in the certificate is compared with the addresses\ DNS Names of the classified Proxy Set. If a match is found for any of the configured Proxies, the TLS connection is established.<br>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.<br>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.<br>**Note:** If you set the parameter to [2] (Server & Client), for this functionality to operate you also need to set the SIPSRequireClientCertificate parameter to [1] (Enable). |
| TLS Client Verify Server Certificate<br>`tls-vrfy-srvr-cert`<br>[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.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Note:** If Subject Name verification is necessary, the parameter PeerHostNameVerificationMode must be used as well. |
| Strict Certificate Extension Validation<br>`require-strict-cert`<br>[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.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable |
| TLS Remote Subject Name<br>`tls-rmt-subs-name`<br>[TLSRemoteSubjectName] | Defines the Subject Name that is compared with the name defined in the remote side certificate when establishing TLS connections.<br>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.<br>The valid range is a string of up to 49 characters.<br>**Note:** The parameter is applicable only if the parameter PeerHostNameVerificationMode is set to 1 or 2. |

| Parameter | Description |
|---|---|
| TLS Expiry Check Start<br>`expiry-check-start`<br>[TLSExpiryCheckStart] | Defines the number of days before the installed TLS server certificate is to expire at which the device must send a trap (acCertificateExpiryNotification) to notify of this.<br>The valid value is 0 to 3650. The default is 60. |
| TLS Expiry Check Period<br>`expiry-check-period`<br>[TLSExpiryCheckPeriod] | Defines the periodical interval (in days) for checking the TLS server certificate expiry date.<br>The valid value is 1 to 3650. The default is 7. |
| TLS FIPS 140 Mode<br>[TLS_Fips140_Mode] | Enables FIPS 140-2 conformance mode for TLS.<br>▪ [0] Disable (default)<br>▪ [1] Enable |

## 45.4.5 SSH Parameters

Secure Shell (SSH) parameters are described in the table below.

**Table 45-25: SSH Parameters**

| Parameter | Description |
|---|---|
| Enable SSH Server<br>`ssh`<br>[SSHServerEnable] | Enables the device's embedded SSH server.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable |
| Server Port<br>`ssh-port`<br>[SSHServerPort] | Defines the port number for the embedded SSH server.<br>Range is any valid port number. The default port is 22. |
| SSH Admin Key<br>`ssh-admin-key`<br>[SSHAdminKey] | Defines the RSA public key for strong authentication for logging in to the SSH interface (if enabled).<br>The value should be a base64-encoded string. The value can be a maximum length of 511 characters. |
| Require Public Key<br>`ssh-require-public-key`<br>[SSHRequirePublicKey] | Enables RSA public keys for SSH.<br>▪ **[0]** = (Default) RSA public keys are optional if a value is configured for the parameter SSHAdminKey.<br>▪ **[1]** = RSA public keys are mandatory.<br>**Note:** To define the key size, use the TLSPkeySize parameter. |
| Max Payload Size<br>`ssh-max-payload-size`<br>[SSHMaxPayloadSize] | Defines the maximum uncompressed payload size (in bytes) for SSH packets.<br>The valid value is 550 to 32768. The default is 32768. |
| Max Binary Packet Size<br>`ssh-max-binary-packet-size`<br>[SSHMaxBinaryPacketSize] | Defines the maximum packet size (in bytes) for SSH packets.<br>The valid value is 582 to 35000. The default is 35000. |
| Maximum SSH Sessions<br>`ssh-max-sessions`<br>[SSHMaxSessions] | Defines the maximum number of simultaneous SSH sessions.<br>The valid range is 1 to 5. The default is 5 sessions. |

| Parameter | Description |
|---|---|
| Enable Last Login Message<br>`ssh-last-login-message`<br>[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.<br>▪ **[0]** Disable<br>▪ **[1]** Enable (default)<br>**Note:** The last SSH login information is cleared when the device is reset. |
| Max Login Attempts<br>`ssh-max-login-attempts`<br>**[SSHMaxLoginAttempts]** | Defines the maximum SSH login attempts allowed for entering an incorrect password by an administrator before the SSH session is rejected.<br>The valid range is 1 to 5. The default is 3.<br>**Note:** The new setting takes effect only for new subsequent SSH connections. |

## 45.4.6 IDS Parameters

The Intrusion Detection System (IDS) parameters are described in the table below.

**Table 45-26: IDS Parameters**

| Parameter | Description |
|---|---|
| Intrusion Detection System (IDS)<br>`enable-ids`<br>[EnableIDS] | Enables the IDS feature.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Note:** For the parameter to take effect, a device reset is required. |
| `ids-clear-period`<br>[IDSAlarmClearPeriod] | Defines the interval (in seconds) after which an IDS alarm is cleared from the Active Alarms table if no thresholds are crossed during this time. However, this "quiet" period must be at least twice the Threshold Window value. For example, if IDSAlarmClearPeriod is set to 20 sec and the Threshold Window is set to 15 sec, the IDSAlarmClearPeriod parameter is ignored and the alarm is cleared only after 30 seconds (2 x 15 sec).<br>The valid value is 0 to 86400. The default is 300. |
| **IDS Policy Table** | |
| IDS Policy Table<br>[IDSPolicy] | Defines IDS Policies.<br>The format of the ini file parameter is:<br>[ IDSPolicy ]<br>FORMAT IDSPolicy_Index = IDSPolicy_Name, IDSPolicy_Description;<br>[ \IDSPolicy ]<br>For a detailed description of the table, see "Configuring IDS Policies" on page 168. |
| **IDS Rule Table** | |

| Parameter | Description |
|---|---|
| IDS Rule Table<br>[IDSRule] | Defines rules for IDS Policies.<br>The format of the ini file parameter is:<br>[ IDSRule ]<br>FORMAT IDSRule_Index = IDSRule_Policy, IDSRule_RuleID, IDSRule_Reason, IDSRule_ThresholdScope, IDSRule_ThresholdWindow, IDSRule_MinorAlarmThreshold, IDSRule_MajorAlarmThreshold, IDSRule_CriticalAlarmThreshold, IDSRule_DenyThreshold, IDSRule_DenyPeriod;<br>[ \IDSRule ]<br>For a detailed description of the table, see "Configuring IDS Policies" on page 168. |
| **IDS Match Table** | |
| IDS Match Table<br>[IDSMatch] | Defines target rules per IDS Policy.<br>The format of the ini file parameter is:<br>[ IDSMatch ]<br>FORMAT IDSMatch_Index = IDSMatch_SIPInterface, IDSMatch_ProxySet, IDSMatch_Subnet, IDSMatch_Policy;<br>[ \IDSMatch ]<br>For a detailed description of the table, see "Assigning IDS Policies" on page 172. |

## 45.4.7 OCSP Parameters

The Online Certificate Status Protocol (OCSP) parameters are described in the table below.

**Table 45-27: OCSP Parameters**

| Parameter | Description |
|---|---|
| Enable OCSP Server<br>`enable`<br>[OCSPEnable] | Enables or disables certificate checking using OCSP.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>For a description of OCSP, see Configuring Certificate Revocation Checking (OCSP). |
| Primary Server IP<br>`server-ip`<br>[OCSPServerIP] | Defines the IP address of the OCSP server.<br>The default IP address is 0.0.0.0. |
| Secondary Server IP<br>`secondary-server-ip`<br>[OCSPSecondaryServerIP] | Defines the IP address (in dotted-decimal notation) of the secondary OCSP server (optional).<br>The default IP address is 0.0.0.0. |
| Server Port<br>`server-port`<br>[OCSPServerPort] | Defines the OCSP server's TCP port number.<br>The default port number is 2560. |
| Default Response When Server Unreachable<br>`default-response` | Determines whether the device allows or rejects peer certificates when the OCSP server cannot be contacted.<br>▪ **[0]** Reject (default) |

| Parameter | Description |
|---|---|
| [OCSPDefaultResponse] | ▪ **[1]** Allow |

# 45.5 Quality of Experience Parameters

The Quality of Experience (QoE) parameters are described in the table below.

**Table 45-28: Quality of Experience Parameters**

| Parameter | Description |
|---|---|
| **SEM Parameters** | |
| Server IP<br>`configure voip/qoe configuration/server-ip`<br>[QOEServerIP] | Defines the IP address of the primary Session Experience Manager (SEM) server to where the quality experience reports are sent.<br>**Note:** For the parameter to take effect, a device reset is required. |
| Redundant Server IP<br>`configure voip > qoe configuration > set secondary-server-ip`<br>[QOESecondaryServerIp] | Defines the IP address of the secondary SEM server to where the quality experience reports are sent. This is applicable when the SEM/EMS server is in Geographical Redundancy HA mode.<br>**Note:** For the parameter to take effect, a device reset is required. |
| Interface Name<br>`configure voip/qoe configuration/interface-name`<br>[QOEInterfaceName] | Defines the IP network interface on which the quality experience reports are sent.<br>The default is the OAMP interface.<br>**Note:** For the parameter to take effect, a device reset is required. |
| QoE Connection by TLS<br>`configure voip > qoe configuration > tls-enable`<br>[QOEEnableTLS] | Enables a TLS connection with the SEM server.<br>▪ [0] Disable (default)<br>▪ [1] Enable<br>**Note:** For the parameter to take effect, a device reset is required. |
| QOE TLS Context Name<br>`configure voip/qoe configuration/tls-context-name`<br>[QoETLSContextName] | Selects a TLS Context (configured in the TLS Contexts table) for the TLS connection with the SEM server.<br>The valid value is a string representing the name of the TLS Context as configured in the 'Name' field of the TLS Contexts table. The default is the default TLS Context (ID 0). |
| QoE Report Mode<br>`report-mode`<br>[QoeReportMode] | Defines at what stage of the call the device sends the QoE data of the call to the SEM server.<br>▪ [0] Report QoE During Call (default)<br>▪ [1] Report QoE At End Call<br>**Note:** If a QoE traffic overflow between SEM and the device occurs, the device sends the QoE data only at the end of the call, regardless of the settings of the parameter. |
| **Quality of Experience Profile Table** | |

| Parameter | Description |
|---|---|
| Quality of Experience Profile<br>`configure voip/qoe qoe-profile`<br>[QOEProfile] | The table defines Quality of Experience Profiles.<br>The format of the ini file table parameter is as follows:<br>[QOEProfile]<br>FORMAT QOEProfile_Index = QOEProfile_Name, QOEProfile_SensitivityLevel;<br>[\QOEProfile]<br>For a detailed description of the table, see "Configuring Quality of Experience Profiles" on page 293. |
| **Quality of Experience Color Rules Table** | |
| Quality of Experience Color Rules<br>`configure voip/qoe qoe-profile qoe-color-rules`<br>[QOEColorRules] | The table defines Quality of Experience Color Rules.<br>The format of the ini file table parameter is as follows:<br>[QOEColorRules]<br>FORMAT QOEColorRules_Index = QOEColorRules_QoeProfile, QOEColorRules_ColorRuleIndex, QOEColorRules_monitoredParam, QOEColorRules_direction, QOEColorRules_profile, QOEColorRules_GreenYellowThreshold, QOEColorRules_GreenYellowHysteresis, QOEColorRules_YellowRedThreshold, QOEColorRules_YellowRedHysteresis;<br>[\QOEColorRules]<br>For a detailed description of the table, see "Configuring Quality of Experience Profiles" on page 293. |
| **Bandwidth Profile Table** | |
| Bandwidth Profile<br>`configure voip/qoe bw-profile`<br>[BWProfile] | The table defines Bandwidth Profiles.<br>The format of the ini file table parameter is as follows:<br>[BWProfile]<br>FORMAT BWProfile_Index = BWProfile_Name, BWProfile_EgressAudioBandwidth, BWProfile_IngressAudioBandwidth, BWProfile_EgressVideoBandwidth, BWProfile_IngressVideoBandwidth, BWProfile_TotalEgressBandwidth, BWProfile_TotalIngressBandwidth, BWProfile_WarningThreshold, BWProfile_hysteresis, BWProfile_GenerateAlarms;<br>[\BWProfile]<br>For a detailed description of the table, see "Configuring Bandwidth Profiles" on page 297.<br>**Note:** For the parameter to take effect, a device reset is required. |
| **Media Enhancement Profile Table** | |
| Media Enhancement Profile<br>`configure voip/qoe media-enhancement`<br>[MediaEnhancementProfile] | The table defines Media Enhancement Profiles.<br>The format of the ini file table parameter is as follows:<br>[MediaEnhancementProfile]<br>FORMAT MediaEnhancementProfile_Index = MediaEnhancementProfile_ProfileName;<br>[\MediaEnhancementProfile] |

| Parameter | Description |
|---|---|
| | For a detailed description of the table, see "Configuring Media Enhancement Profiles" on page 300. |
| **Media Enhancement Rules Table** | |
| Media Enhancement Rules<br>`configure voip/qoe media-enhancement-rules`<br>[MediaEnhancementRules] | The table defines Media Enhancement Rules.<br>The format of the ini file table parameter is as follows:<br>[MediaEnhancementRules]<br>FORMAT MediaEnhancementRules_Index = MediaEnhancementRules_MediaEnhancementProfile, MediaEnhancementRules_RuleIndex, MediaEnhancementRules_Trigger, MediaEnhancementRules_Color, MediaEnhancementRules_ActionRule, MediaEnhancementRules_ActionValue;<br>[\MediaEnhancementRules]<br>For a detailed description of the table, see "Configuring Media Enhancement Profiles" on page 300. |

## 45.6 Control Network Parameters

### 45.6.1 IP Group, Proxy, Registration and Authentication Parameters

The proxy server, registration and authentication SIP parameters are described in the table below.

**Table 45-29: Proxy, Registration and Authentication SIP Parameters**

| Parameter | Description |
|---|---|
| **IP Group Table** | |
| IP Group Table<br>`configure voip > voip-network ip-group`<br>[IPGroup] | This table configures IP Groups.<br>The format of the ini file table parameter is:<br>[ IPGroup ]<br>FORMAT IPGroup_Index = IPGroup_Type, IPGroup_Name, IPGroup_ProxySetName, IPGroup_SIPGroupName, IPGroup_ContactUser, IPGroup_SipReRoutingMode, IPGroup_AlwaysUseRouteTable, IPGroup_SRDName, IPGroup_MediaRealm, IPGroup_ClassifyByProxySet, IPGroup_ProfileName, IPGroup_MaxNumOfRegUsers, IPGroup_InboundManSet, IPGroup_OutboundManSet, IPGroup_RegistrationMode, IPGroup_AuthenticationMode, IPGroup_MethodList, IPGroup_SourceUriInput, IPGroup_DestUriInput, IPGroup_ContactName, IPGroup_Username, IPGroup_Password, IPGroup_UUIFormat, IPGroup_QOEProfile, IPGroup_BWProfile, IPGroup_MediaEnhancementProfile, IPGroup_AlwaysUseSourceAddr, IPGroup_MsgManUserDef1, IPGroup_MsgManUserDef2, IPGroup_SIPConnect, IPGroup_DTLSContext, IPGroup_CreatedByRoutingServer, IPGroup_UsedByRoutingServer;<br>[/IPGroup] |

| Parameter | Description |
|---|---|
| | For a description of the table, see "Configuring IP Groups" on page 323.<br>**Note:** For the parameter to take effect, a device reset is required. |
| Authentication per Port Table | |
| Authentication Table<br>`configure voip/gw analoggw authentication`<br>[Authentication] | The table defines a user name and password for authenticating each device port. The format of the ini file table parameter is as follows:<br>[Authentication]<br>FORMAT Authentication_Index = Authentication_UserId, Authentication_UserPassword, Authentication_Module, Authentication_Port;<br>[\Authentication]<br>Where,<br>▪<br>▪ Module = Module number, where 1 denotes the module in Slot 1<br>▪ Port = Port number, where 1 denotes the Port 1 of the module<br>For example:<br>Authentication 1 = lee,1552,1,2; (user name "lee" with password 1552 for authenticating Port 2 of Module 1)<br>For a description o this table, see Configuring Authentication on page 472.<br>**Note:** The parameter is applicable only to FXS interfaces. |
| **Account Table** | |
| Account Table<br>`configure voip > sip-definition account`<br>[Account] | Defines user accounts for registering and/or authenticating (digest) Hunt Groups or IP Groups (e.g., an IP-PBX) with a Serving IP Group (e.g., a registrar server).<br>The format of the ini file table parameter is as follows:<br>[Account]<br>FORMAT Account_Index = Account_ServedTrunkGroup, Account_ServedIPGroupName, Account_ServingIPGroupName, Account_Username, Account_Password, Account_HostName, Account_Register, Account_ContactUser, Account_ApplicationType;<br>[\Account]<br>For a detailed description of the table, see "Configuring Registration Accounts" on page 341. |
| **Proxy Registration Parameters** | |
| Use Default Proxy<br>`enable-proxy`<br>[IsProxyUsed] | Enables the use of Proxy Set ID 0 (for backward compatibility).<br>▪ [0] No = (Default) Proxy Set 0 is not used.<br>▪ [1] Yes = Proxy Set ID 0 is used.<br>**Notes:**<br>▪ The parameter must be used only for backward compatibility. If not required for backward compatibility, make sure that the parameter is disabled, and use the Proxy |

| Parameter | Description |
|---|---|
| | Set table for configuring all your Proxy Sets (except for Proxy Set ID 0).<br>▪ If you are not using a proxy server, you must configure routing rules to route the call.<br>▪ The parameter is applicable only to the Gateway application. |
| Proxy Name<br>`proxy-name`<br>[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.<br>The valid value is a string of up to 49 characters.<br>**Note:** The parameter functions together with the UseProxyIPasHost parameter. |
| Use Proxy IP as Host<br>`use-proxy-ip-as-host`<br>[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.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>If the parameter is disabled and the device registers to an IP Group (i.e., proxy server), it uses the string configured by the ProxyName parameter as the host name in the REGISTER's Request-URI and uses the string configured by the IP Group table parameter, SIPGroupName as the host name in the To and From headers. If the IP Group is configured with a Proxy Set that has multiple IP addresses, all the REGISTER messages sent to these proxies are sent with the same host name.<br>**Note:** If the parameter is disabled and the ProxyName parameter is not configured, the proxy's IP address is used as the host name in the REGISTER Request-URI. |
| Redundancy Mode<br>`redundancy-mode`<br>[ProxyRedundancyMode] | Determines whether the device switches back to the primary Proxy after using a redundant Proxy.<br>▪ **[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.<br>▪ **[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).<br>**Note:** To use this Proxy Redundancy mechanism, you need to enable the keep-alive with Proxy option, by setting the parameter EnableProxyKeepAlive to 1 or 2. |
| Proxy IP List Refresh Time<br>`proxy-ip-lst-rfrsh-time`<br>[ProxyIPListRefreshTime] | Defines the time interval (in seconds) between each Proxy IP list refresh.<br>The range is 5 to 2,000,000. The default interval is 60. |
| Enable Fallback to Routing Table<br>`fallback-to-routing`<br>[IsFallbackUsed] | Determines whether the device falls back to the Tel-to-IP Routing table for call routing when Proxy servers are unavailable.<br>▪ [0] Disable = (Default) Fallback is not used.<br>▪ [1] Enable = The Tel-to-IP Routing table is used when Proxy servers are unavailable. |

| Parameter | Description |
|---|---|
| | When the device falls back to the Tel-to-IP Routing table, it continues scanning for a Proxy. When the device locates an active Proxy, it switches from internal routing back to Proxy routing.<br>**Note:** To enable the redundant Proxies mechanism, set the parameter EnableProxyKeepAlive to 1 or 2. |
| Prefer Routing Table<br>`prefer-routing-table`<br>[PreferRouteTable] | Determines whether the device's routing table takes precedence over a Proxy for routing calls.<br>▪ [0] No = (Default) Only a Proxy server is used to route calls.<br>▪ [1] Yes = The device checks the routing rules in the Tel-to-IP Routing table for a match with the Tel-to-IP call. Only if a match is not found is a Proxy used. |
| Always Use Proxy<br>`always-use-proxy`<br>[AlwaysSendToProxy] | Determines whether the device sends SIP messages and responses through a Proxy server.<br>▪ **[0]** Disable = (Default) Use standard SIP routing rules.<br>▪ **[1]** Enable = All SIP messages and responses are sent to the Proxy server.<br>**Note:** The parameter is applicable only if a Proxy server is used (i.e., the parameter IsProxyUsed is set to 1). |
| SIP ReRouting Mode<br>`sip-rerouting-mode`<br>[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).<br>▪ [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.<br>▪ [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.<br>▪ [2] Routing Table = Uses the Routing table to locate the destination and then sends a new INVITE to this destination.<br>**Notes:**<br>▪ The parameter is applicable only to the Gateway application.<br>▪ When the 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].<br>▪ When the 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.<br>▪ When the parameter is set to [2], the XferPrefix parameter can be used to define different routing rules for redirect calls.<br>▪ The parameter is disregarded if the parameter AlwaysSendToProxy is set to 1. |
| DNS Query Type<br>`dns-query`<br>[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. |

| Parameter | Description |
|---|---|
| | ▪ **[0]** A-Record = (Default) No NAPTR or SRV queries are performed.<br>▪ **[1]** SRV = If the Proxy/Registrar IP address parameter, Contact/Record-Route headers, or IP address configured in the routing tables contain a domain name, an SRV query is performed. The device uses the first host name received from the SRV query. The device then performs a DNS A-record query for the host name to locate an IP address.<br>▪ **[2]** NAPTR = An NAPTR query is performed. If it is successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is performed according to the configured transport type.<br>**Notes:**<br>▪ If the Proxy/Registrar IP address parameter, the domain name in the Contact/Record-Route headers, or the IP address configured in the routing tables contain a domain name with a port definition, the device performs a regular DNS A-record query.<br>▪ If a specific Transport Type is configured, a NAPTR query is not performed.<br>▪ To enable NAPTR/SRV queries for Proxy servers only, use the global parameter ProxyDNSQueryType, or use the proxy Set table. |
| Proxy DNS Query Type<br>`proxy-dns-query`<br>[ProxyDNSQueryType] | Global parameter that defines the DNS query record type for resolving the Proxy server's configured domain name (FQDN) into an IP address.<br>▪ **[0]** A-Record (default) = A-record DNS query.<br>▪ **[1]** SRV = If the Proxy IP address parameter contains a domain name without port definition (e.g., ProxyIP = domain.com), an SRV query is performed. The SRV query returns up to four Proxy host names and their weights. The device then performs DNS A-record queries for each Proxy host name (according to the received weights) to locate up to four Proxy IP addresses. Thus, if the first SRV query returns two domain names and the A-record queries return two IP addresses each, no additional searches are performed.<br>▪ **[2]** NAPTR = NAPTR query is done. If successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is done according to the configured transport type. If the Proxy IP address parameter contains a domain name with port definition (e.g., ProxyIP = domain.com:5080), the device performs a regular DNS A-record query. If a specific Transport Type is defined, a NAPTR query is not performed.<br>**Notes:**<br>▪ This functionality can be configured per Proxy Set in the Proxy Set table (see "Configuring Proxy Sets" on page 329).<br>▪ When enabled, NAPTR/SRV queries are used to discover Proxy servers even if the parameter DNSQueryType is disabled. |

| Parameter | Description |
|---|---|
| Use Gateway Name for OPTIONS<br>`use-gw-name-for-opt`<br>[UseGatewayNameForOptions] | Determines whether the device uses its IP address or string name ("gateway name") in keep-alive SIP OPTIONS messages (host part of the Request-URI). To configure the "gateway name", use the SIPGatewayName parameter. The device uses the OPTIONS request as a keep-alive message with its primary and redundant SIP proxy servers (i.e., the EnableProxyKeepAlive parameter is set to 1).<br>▪ **[0]** No = (Default) Device's IP address is used in keep-alive OPTIONS messages.<br>▪ **[1]** Yes = Device's "gateway name" is used in keep-alive OPTIONS messages.<br>▪ **[2]** Server = Device's IP address is used in the From and To headers in keep-alive OPTIONS messages. |
| User Name<br>`user-name-4-auth`<br>[UserName] | Defines the username for registration and Basic/Digest authentication with a Proxy/Registrar server.<br>By default, no value is defined.<br>**Notes:**<br>▪ The parameter is applicable only to the Gateway application.<br>▪ The parameter is applicable only if single device registration is used (i.e., the parameter AuthenticationMode is set to authentication per gateway).<br>▪ Instead of configuring the parameter, the Authentication table can be used (see Authentication on page 472). |
| Password<br>`password-4-auth`<br>[Password] | Defines the password for Basic/Digest authentication with a Proxy/Registrar server. A single password is used for all device ports.<br>The default is 'Default_Passwd'.<br>**Note:** Instead of configuring the parameter, the Authentication table can be used (see Authentication on page 472). |
| Cnonce<br>`cnonce-4-auth`<br>[Cnonce] | Defines the Cnonce string used by the SIP server and client to provide mutual authentication.<br>The value is free format, i.e., 'Cnonce = 0a4f113b'. The default is 'Default_Cnonce'. |
| Mutual Authentication Mode<br>`mutual-authentication`<br>[MutualAuthenticationMode] | Determines the device's mode of operation when Authentication and Key Agreement (AKA) Digest Authentication is used.<br>▪ **[0]** Optional = (Default) Incoming requests that don't include AKA authentication information are accepted.<br>▪ **[1]** Mandatory = Incoming requests that don't include AKA authentication information are rejected. |
| Challenge Caching Mode<br>`challenge-caching`<br>[SIPChallengeCachingMode] | Enables local caching of SIP message authorization challenges from Proxy servers.<br>The device sends the first request to the Proxy without authorization. The Proxy sends a 401/407 response with a challenge for credentials. The device saves (caches) the response for further uses. The device sends a new request with the appropriate credentials. Subsequent requests to the Proxy are automatically sent with credentials (calculated from the saved challenge). If the Proxy doesn't accept the new request and sends another challenge, the old challenge is replaced with |

| Parameter | Description |
|---|---|
| | the new one. One of the benefits of the feature is that it may reduce the number of SIP messages transmitted through the network.<br>▪ [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.<br>▪ [1] INVITE Only = Challenges issued for INVITE requests are cached. This prevents a mixture of REGISTER and INVITE authorizations.<br>▪ [2] Full = Caches all challenges from the proxies.<br>**Note:**<br>▪ Challenge caching is used with all proxies and not only with the active one.<br>▪ The challenge can be cached per endpoint or per Account. |
| **Proxy Address Table** | |
| Proxy IP Table<br>`configure voip > voip-network proxy-ip`<br>[ProxyIP] | The table defines proxy addresses per Proxy Set.<br>The format of the ini file table parameter is as follows:<br>[ProxyIP]<br>FORMAT ProxyIp_Index = ProxyIp_ProxySetId, ProxyIp_ProxyIpIndex, ProxyIp_IpAddress, ProxyIp_TransportType;<br>[\ProxyIP]<br>For a description of the table, see "Configuring Proxy Sets" on page 329. |
| **Proxy Sets Table** | |
| Proxy Set Table<br>`configure voip > voip-network proxy-set`<br>[ProxySet] | Defines the Proxy Sets.<br>The format of the ini file table parameter is as follows:<br>[ ProxySet ]<br>FORMAT ProxySet_Index = ProxySet_ProxyName, ProxySet_EnableProxyKeepAlive, ProxySet_ProxyKeepAliveTime, ProxySet_ProxyLoadBalancingMethod, ProxySet_IsProxyHotSwap, ProxySet_SRDName, ProxySet_ClassificationInput, ProxySet_TLSContextName, ProxySet_ProxyRedundancyMode, ProxySet_DNSResolveMethod, ProxySet_KeepAliveFailureResp, ProxySet_GWIPv4SIPInterfaceName, ProxySet_SBCIPv4SIPInterfaceName, ProxySet_GWIPv6SIPInterfaceName, ProxySet_SBCIPv6SIPInterfaceName, ProxySet_MinActiveServersLB, ProxySet_SuccessDetectionRetries, ProxySet_SuccessDetectionInterval, ProxySet_FailureDetectionRetransmissions;<br>[ \ProxySet ]<br>For a description of the table, see "Configuring Proxy Sets" on page 329. |
| **Registrar Parameters** | |

| Parameter | Description |
|---|---|
| Enable Registration<br>`enable-registration`<br>[IsRegisterNeeded] | Enables the device to register to a Proxy/Registrar server.<br>▪ [0] Disable = (Default) The device doesn't register to Proxy/Registrar server.<br>▪ [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).<br>**Notes:**<br>▪ The parameter is applicable only to the Gateway application.<br>▪ The device sends a REGISTER request for each channel or for the entire device (according to the AuthenticationMode parameter). |
| Registrar Name<br>`registrar-name`<br>[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.<br>The valid range is up to 100 characters.<br>**Note:** The parameter is applicable only to the Gateway application. |
| Registrar IP Address<br>`ip-addrr-rgstrr`<br>[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>.<br>**Notes:**<br>▪ The parameter is applicable only to the Gateway application.<br>▪ If not specified, the REGISTER request is sent to the primary Proxy server.<br>▪ When a port number is specified, DNS NAPTR/SRV queries aren't performed, even if the parameter DNSQueryType is set to 1 or 2.<br>▪ 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.<br>▪ 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. |
| Registrar Transport Type<br>`registrar-transport`<br>[RegistrarTransportType] | Determines the transport layer used for outgoing SIP dialogs initiated by the device to the Registrar.<br>▪ [-1] Not Configured (default)<br>▪ [0] UDP<br>▪ [1] TCP<br>▪ [2] TLS<br>**Notes:**<br>▪ The parameter is applicable only to the Gateway application. |

| Parameter | Description |
|---|---|
| | ▪ When set to 'Not Configured', the value of the parameter SIPTransportType is used. |
| Registration Time<br>`registration-time`<br>[RegistrationTime] | Defines the time interval (in seconds) for registering to a Proxy server. The value is used in the SIP Expires header. The parameter also defines the time interval between Keep-Alive messages when the parameter EnableProxyKeepAlive is set to 2 (REGISTER).<br>Typically, the device registers every 3,600 sec (i.e., one hour). The device resumes registration according to the parameter RegistrationTimeDivider.<br>The valid range is 10 to 2,000,000. The default is 180. |
| Re-registration Timing [%]<br>`re-registration-timing`<br>[RegistrationTimeDivider] | Defines the re-registration timing (in percentage). The timing is a percentage of the re-register timing set by the Registrar server.<br>The valid range is 50 to 100. The default is 50.<br>For example: If the parameter is set to 70% and the Registration Expires time is 3600, the device re-sends its registration request after 3600 x 70% (i.e., 2520 sec).<br>**Notes**:<br>▪ The parameter may be overridden if the parameter RegistrationTimeThreshold is greater than 0. |
| Registration Retry Time<br>`registration-retry-time`<br>[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.<br>The default is 30 seconds. The range is 10 to 3600. |
| Registration Time Threshold<br>`registration-time-thres`<br>[RegistrationTimeThreshold] | Defines a threshold (in seconds) for re-registration timing. If the parameter is greater than 0, but lower than the computed re-registration timing (according to the parameter RegistrationTimeDivider), the re-registration timing is set to the following: timing set by the Registration server in the SIP Expires header minus the value of the parameter RegistrationTimeThreshold.<br>The valid range is 0 to 2,000,000. The default is 0. |
| Re-register On INVITE Failure<br>`reg-on-invite-fail`<br>[RegisterOnInviteFailure] | Enables immediate re-registration if no response is received for an INVITE request sent by the device.<br>▪ [0] Disable (default)<br>▪ [1] Enable = The device immediately expires its re-registration timer and commences re-registration to the same Proxy upon any of the following scenarios:<br>✓ The response to an INVITE request is 407 (Proxy Authentication Required) without an authentication header included.<br>✓ The remote SIP UA abandons a call before the device has received any provisional response (indicative of an outbound proxy server failure).<br>✓ 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). |

| Parameter | Description |
|---|---|
| | ✓ 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).<br>✓ 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).<br>**Note:** The parameter is applicable only to the Gateway application. |
| ReRegister On Connection Failure<br>`reg-on-conn-failure`<br>[ReRegisterOnConnectionFailure] | Enables the device to perform SIP re-registration upon TCP/TLS connection failure.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable |
| Gateway Registration Name<br>`gw-registration-name`<br>[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 the parameter, the UserName parameter is used instead.<br>**Notes:**<br>▪ The parameter is applicable only to the Gateway application.<br>▪ The 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. |
| Registration Mode<br>`authentication-mode`<br>[AuthenticationMode] | Determines the device's registration and authentication method.<br>▪ [0] Per Endpoint = Registration and authentication is performed separately for each endpoint/B-channel. This is typically used for FXS interfaces, where each endpoint registers (and authenticates) separately with its user name and password.<br>▪ [1] Per Gateway = (Default) Single registration and authentication for the entire device.<br>▪ [3] Per FXS = Registration and authentication for FXS endpoints.<br>**Note:** The parameter is applicable only to the Gateway application. |
| Set Out-Of-Service On Registration Failure<br>`set-oos-on-reg-failure`<br>[OOSOnRegistrationFail] | Enables setting the endpoint or entire device (i.e., all endpoints) to out-of-service if registration fails.<br>▪ [0] Disable (default)<br>▪ [1] Enable<br>If the registration is per endpoint (i.e., AuthenticationMode is set to 0) or per Account (see Configuring Hunt Group Settings on page 386) and a specific endpoint/Account registration fails (SIP 4xx or no response), then that endpoint is set to out-of-service until a success response is received in a subsequent registration request. When the registration is per the entire device (i.e., AuthenticationMode is set to 1) and registration fails, all endpoints are set to out-of-service. If all the Accounts of |

| Parameter | Description |
|---|---|
| | a specific Hunt Group fail registration and if the Hunt Group comprises a complete trunk, then the entire trunk is set to out-of-service.<br>**Notes:**<br>▪ The parameter is applicable only to the Gateway application.<br>▪ The out-of-service method is configured using the FXSOOSBehavior parameter. |
| `expl-un-reg`<br>[UnregistrationMode] | Enables the device to perform explicit unregisters.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable = The device sends an asterisk ("*") value in the SIP Contact header, instructing the Registrar server to remove all previous registration bindings. The device removes SIP User Agent (UA) registration bindings in a Registrar, according to RFC 3261. Registrations are soft state and expire unless refreshed, but they can also be explicitly removed. A client can attempt to influence the expiration interval selected by the Registrar. A UA requests the immediate removal of a binding by specifying an expiration interval of "0" for that contact address in a REGISTER request. UA's should support this mechanism so that bindings can be removed before their expiration interval has passed. Use of the "*" Contact header field value allows a registering UA to remove all bindings associated with an address-of-record (AOR) without knowing their precise values.<br>**Note:** The REGISTER-specific Contact header field value of "*" applies to all registrations, but it can only be used if the Expires header field is present with a value of "0". |
| Add Empty Authorization Header<br>`add-empty-author-hdr`<br>[EmptyAuthorizationHeader] | Enables the inclusion of the SIP Authorization header in initial registration (REGISTER) requests sent by the device.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>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:<br>▪ username - set to the value of the private user identity<br>▪ realm - set to the domain name of the home network<br>▪ uri - set to the SIP URI of the domain name of the home network<br>▪ nonce - set to an empty value<br>▪ response - set to an empty value<br>For example:<br>`Authorization: Digest username=alice_private@home1.net, realm=”home1.net”, nonce=””, response=”e56131d19580cd833064787ecc”`<br>**Note:** This registration header is according to the IMS 3GPP TS24.229 and PKT-SP-24.220 specifications. |

| Parameter | Description |
|---|---|
| Add initial Route Header<br>`add-init-rte-hdr`<br>[InitialRouteHeader] | Enables the inclusion of the SIP Route header in initial registration or re-registration (REGISTER) requests sent by the device.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>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:<br>`Route: <sip:10.10.10.10;lr;transport=udp>`<br>or<br>`Route: <sip: pcscf-gm.ims.rr.com;lr;transport=udp>` |
| [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.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>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.<br>**Note:** The device sends a CRLF message to the Proxy Set only if the Proxy Keep-Alive feature (EnableProxyKeepAlive parameter) is enabled and its transport type is set to TCP or TLS. The device first sends a SIP OPTION message to establish the TCP/TLS connection and if it receives any SIP response, it continues sending the CRLF keep-alive sequences. |
| [PingPongKeepAliveTime] | Defines the periodic interval (in seconds) after which a “ping” (double-CRLF) keep-alive is sent to a proxy/registrar, using the CRLF Keep-Alive mechanism.<br>The default range is 5 to 2,000,000. The default is 120.<br>The device uses the range of 80-100% of this user-defined value as the actual interval. For example, if the parameter value is set to 200 sec, the interval used is any random time between 160 to 200 seconds. This prevents an “avalanche” of keep-alive by multiple SIP UAs to a specific server. |

| Parameter | Description |
|---|---|
| Max Generated Register Rate<br>configure voip > sip-definition settings > max-gen-reg-rate<br>[MaxGeneratedRegistersRate] | Defines the maximum number of user register requests (REGISTER messages) that the device sends (to a proxy or registrar server) at a user-defined rate configured by the GeneratedRegistersInterval parameter. The parameter is useful in that it may be used to prevent an overload on the device's CPU caused by sending many registration requests at a given time.<br>The valid value is 30 to 300 register requests per second. The default is 150.<br>For configuration examples, see the description of the GeneratedRegistersInterval parameter. |
| Generated Registers interval<br>gen-reg-int<br>[GeneratedRegistersInterval] | Defines the rate (in seconds) at which the device sends user register requests (REGISTER messages). The parameter is based on the maximum number of REGISTER messages that can be sent at this rate, configured by the MaxGeneratedRegistersRate parameter.<br>The valid value is 1 to 5. The default is 1.<br>Configuration examples:<br>▪ If you configure the MaxGeneratedRegistersRate parameter to 100 and the GeneratedRegistersInterval to 5, the device sends a maximum of 20 REGISTER messages per second (i.e., 100 messages divided by 5 sec; 100 per 5 seconds).<br>▪ If you configure the MaxGeneratedRegistersRate parameter to 100 and the GeneratedRegistersInterval to 1, the device sends a maximum of a 100 REGISTER messages per second. |

## 45.6.2 Network Application Parameters

The SIP network application parameters are described in the table below.

**Table 45-30: SIP Network Application Parameters**

| Parameter | Description |
|---|---|
| **SRD Table** | |
| SRD Table<br>`configure voip > voip-network srd`<br>[SRD] | Defines Signaling Routing Domains (SRD).<br>The format of the ini file table parameter is as follows:<br>[ SRD ]<br>FORMAT SRD_Index = SRD_Name, SRD_IntraSRDMediaAnchoring, SRD_BlockUnRegUsers, SRD_MaxNumOfRegUsers, SRD_EnableUnAuthenticatedRegistrations, SRD_SharingPolicy, SRD_UsedByRoutingServer;<br>[ \SRD ]<br>For a detailed description of the table, see "Configuring SRDs" on page 311. |
| **SIP Interface Table** | |
| SIP Interface Table | Defines SIP Interfaces.<br>The format of the ini file table parameter is as follows: |

| Parameter | Description |
|---|---|
| `configure voip > voip-network sip-interface`<br>[SIPInterface] | [ SIPInterface ]<br>FORMAT SIPInterface_Index = SIPInterface_InterfaceName, SIPInterface_NetworkInterface, SIPInterface_ApplicationType, SIPInterface_UDPPort, SIPInterface_TCPPort, SIPInterface_TLSPort, SIPInterface_SRDName, SIPInterface_MessagePolicyName, SIPInterface_TLSContext, SIPInterface_TLSMutualAuthentication, SIPInterface_TCPKeepaliveEnable, SIPInterface_ClassificationFailureResponseType, SIPInterface_PreClassificationManSet, SIPInterface_EncapsulatingProtocol, SIPInterface_MediaRealm, SIPInterface_BlockUnRegUsers, SIPInterface_MaxNumOfRegUsers, SIPInterface_EnableUnAuthenticatedRegistrations, SIPInterface_UsedByRoutingServer;<br>[ \SIPInterface ]<br>For a detailed description of the table, see "Configuring SIP Interfaces" on page 319. |
| [TCPKeepAliveTime] | Defines the interval (in sec) between the last data packet sent and the first keep-alive probe to send.<br>The valid value is 10 to 65,000. The default is 60.<br>**Notes:**<br>▪ Simple ACKs such as keepalives are not considered data packets.<br>▪ TCP keepalive is enabled per SIP Interface in the SIP Interface table. |
| [TCPKeepAliveInterval] | Defines the interval (in sec) between consecutive keep-alive probes, regardless of what the connection has exchanged in the meantime.<br>The valid value is 10 to 65,000. The default is 10.<br>**Note:** TCP keepalive is enabled per SIP Interface in the SIP Interface table. |
| [TCPKeepAliveRetry] | Defines the number of unacknowledged keep-alive probes to send before considering the connection down.<br>The valid value is 1 to 100. The default is 5.<br>**Note:** TCP keepalive is enabled per SIP Interface in the SIP Interface table. |
| **NAT Translation Table** | |
| NAT Translation Table<br>`configure voip > voip-network NATTranslation`<br>[NATTranslation] | Defines NAT rules for translating source IP addresses per VoIP interface (SIP control and RTP media traffic) into NAT IP addresses.<br>The format of the ini file table parameter is as follows:<br>[ NATTranslation ]<br>FORMAT NATTranslation_Index = NATTranslation_SrcIPInterfaceName, NATTranslation_TargetIPAddress, NATTranslation_SourceStartPort, NATTranslation_SourceEndPort, NATTranslation_TargetStartPort, NATTranslation_TargetEndPort;<br>[ \NATTranslation ]<br>For a detailed description of the table, see "Configuring NAT Translation per IP Interface" on page 154. |
| **Media Realm Table** | |

| Parameter | Description |
|---|---|
| Media Realm Table<br>`configure voip > voip-network realm`<br>[CpMediaRealm] | Defines Media Realms.<br>The format of the ini file table parameter is as follows:<br>[ CpMediaRealm ]<br>FORMAT CpMediaRealm_Index = CpMediaRealm_MediaRealmName, CpMediaRealm_IPv4IF, CpMediaRealm_IPv6IF, CpMediaRealm_PortRangeStart, CpMediaRealm_MediaSessionLeg, CpMediaRealm_PortRangeEnd, CpMediaRealm_IsDefault, CpMediaRealm_QoeProfile, CpMediaRealm_BWProfile;<br>[ \CpMediaRealm ]<br>For a detailed description of the table, see "Configuring Media Realms" on page 303. |
| **Remote Media Subnet Table** | |
| Remote Media Subnet<br>`configure voip > voip-network realm remote-media-subnet`<br>[SubRealm] | Defines Remote Media Subnets.<br>The format of the ini file table parameter is as follows:<br>[RemoteMediaSubnet]<br>FORMAT RemoteMediaSubnet_Index = RemoteMediaSubnet_Realm, RemoteMediaSubnet_RemoteMediaSubnetIndex, RemoteMediaSubnet_RemoteMediaSubnetName, RemoteMediaSubnet_PrefixLength, RemoteMediaSubnet_AddressFamily, RemoteMediaSubnet_DstIPAddress, RemoteMediaSubnet_QOEProfileName, RemoteMediaSubnet_BWProfileName;<br>[\RemoteMediaSubnet]<br>For a detailed description of the table, see "Configuring Remote Media Subnets" on page 307. |
| **Media Realm Extension Table** | |
| Media Realm Extension<br>[MediaRealmExtension] | Defines Media Realm Extensions.<br>The format of the ini file table parameter is as follows:<br>[ MediaRealmExtension ]<br>FORMAT MediaRealmExtension_Index = MediaRealmExtension_MediaRealmIndex, MediaRealmExtension_ExtensionIndex, MediaRealmExtension_IPv4IF, MediaRealmExtension_IPv6IF, MediaRealmExtension_PortRangeStart, MediaRealmExtension_PortRangeEnd, MediaRealmExtension_MediaSessionLeg;<br>[ \MediaRealmExtension ]<br>For a detailed description of the table, see "Configuring Media Realm Extensions" on page 309. |

## 45.7 General SIP Parameters

The general SIP parameters are described in the table below.

**Table 45-31: General SIP Parameters**

| Parameter | Description |
|---|---|
| Max Call Duration (min)<br>`mx-call-duration`<br>[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.<br>The valid range is 0 to 35,791. The default is 0 (i.e., no limitation). |
| Send reject on overload<br>`configure voip/sip-definition advanced-settings/reject-on-ovrld`<br>[SendRejectOnOverload] | Disables the sending of SIP 503 (Service Unavailable) responses upon receipt of new SIP dialog-initiating requests when the device's CPU is overloaded and thus, unable to accept and process new SIP messages.<br>▪ **[0]** Disable = No SIP 503 response is sent when CPU overloaded.<br>▪ **[1]** Enable (default) = SIP 503 response is sent when CPU overloaded.<br>▪ **Note:** Even if the parameter is disabled (i.e., 503 is not sent), the device still discards the new SIP dialog-initiating requests when the CPU is overloaded. |
| SIP 408 Response upon non-INVITE<br>`enbl-non-inv-408`<br>[EnableNonInvite408Reply] | Enables the device to send SIP 408 responses (Request Timeout) upon receipt of non-INVITE transactions. Disabling this response complies with RFC 4320/4321. By default, and in certain circumstances such as a timeout expiry, the device sends a SIP 408 Request Timeout in response to non-INVITE requests (e.g., REGISTER).<br>▪ **[0]** Disable = SIP 408 response is not sent upon receipt of non-INVITE messages (to comply with RFC 4320).<br>▪ **[1]** Enable = (Default) SIP 408 response is sent upon receipt of non-INVITE messages, if necessary. |
| SIP Remote Reset<br>`sip-remote-reset`<br>[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.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>The action depends on the Event header value:<br>▪ 'check-sync;reboot=false': triggers the regular Automatic Update feature (if Automatic Update has been enabled on the device)<br>▪ 'check-sync;reboot=true': triggers a device reset<br>**Note:**<br>▪ The Event header value is proprietary to AudioCodes.<br>▪ The parameter is applicable only to the Gateway application. |
| Max SIP Message Length [KB]<br>[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.<br>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. |

| Parameter | Description |
|---|---|
| | ▪ **[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.<br>▪ **[1]** = Enabled. SIP responses are sent to the UDP port from where SIP requests are received even if the 'rport' parameter is not present in the Via header. |
| Reject Cancel after Connect<br>`reject-cancel-after-connect`<br>[RejectCancelAfterConnect] | Enables or disables the device to accept or reject SIP CANCEL requests received after the receipt of a 200 OK in response to an INVITE (i.e., call established). According to the SIP standard, a CANCEL can be sent only during the INVITE transaction (before 200 OK), and once a 200 OK response is received the call can be rejected only by a BYE request.<br>▪ **[0]** Disable = (Default) Accepts a CANCEL request received during the INVITE transaction by sending a 200 OK response and terminates the call session.<br>▪ **[1]** Enable = Rejects a CANCEL request received during the INVITE transaction by sending a SIP 481 (Call/Transaction Does Not Exist) response and maintains the call session. |
| Verify Received RequestURI<br>`verify-rcvd-requri`<br>[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.<br>▪ **[0]** Disable = (Default) Even if the user is different, the device accepts the SIP request.<br>▪ **[1]** Enable = If the user is different, the device rejects the SIP request (BYE is responded with 481; re-INVITE is responded with 404; ACK is ignored). |
| Max Number of Active Calls<br>`max-nb-of--act-calls`<br>[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.<br>The valid range is 1 to the maximum number of supported channels. The default value is the maximum available channels (i.e., no restriction on the maximum number of calls). |
| Number of Calls Limit<br>[IpProfile_CallLimit,] | Defines the maximum number of concurrent calls per IP Profile (see "Configuring IP Profiles" on page 366). |
| QoS statistics in SIP Release Call<br>[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.<br>▪ **[0]** = Disable (default)<br>▪ **[1]** = Enable<br>The X-RTP-Stat header provides the following statistics:<br>▪ Number of received and sent voice packets<br>▪ Number of received and sent voice octets<br>▪ Received packet loss, jitter (in ms), and latency (in ms)<br>The X-RTP-Stat header contains the following fields:<br>▪ PS=<voice packets sent><br>▪ OS=<voice octets sent><br>▪ PR=<voice packets received><br>▪ OR=<voice octets received><br>▪ PL=<receive packet loss> |

| Parameter | Description |
|---|---|
| | ▪ JI=<jitter in ms><br>▪ LA=<latency in ms><br>Below is an example of the X-RTP-Stat header in a SIP BYE message:<br>`BYE sip:302@10.33.4.125 SIP/2.0`<br>`Via: SIP/2.0/UDP 10.33.4.126;branch=z9hG4bKac2127550866`<br>`Max-Forwards: 70`<br>`From: <sip:401@10.33.4.126;user=phone>;tag=1c2113553324`<br>`To: <sip:302@company.com>;tag=1c991751121`<br>`Call-ID: 991750671245200001912@10.33.4.125`<br>`CSeq: 1 BYE`<br>**`X-RTP-Stat: PS=207;OS=49680;;PR=314;OR=50240;PL=0;JI=600;LA=40;`**<br>`Supported: em,timer,replaces,path,resource-priority`<br>`Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUBSCRIBE,UPDATE`<br>`User-Agent: Sip-Gateway-/v.6.80A.227.005`<br>`Reason: Q.850 ;cause=16 ;text="local"`<br>`Content-Length: 0` |
| PRACK Mode<br>`prack-mode`<br>[PrackMode] | Determines the PRACK (Provisional Acknowledgment) mechanism mode for SIP 1xx reliable responses.<br>▪ **[0]** Disable<br>▪ **[1]** Supported (default)<br>▪ **[2]** Required<br>**Notes:**<br>▪ The Supported and Required headers contain the '100rel' tag.<br>▪ The device sends PRACK messages if 180/183 responses are received with '100rel' in the Supported or Required headers. |
| Enable Early Media<br>`early-media`<br>[EnableEarlyMedia] | Global parameter that enables the Early Media feature for sending media (e.g., ringing) before the call is established. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableEarlyMedia) or Tel Profiles. For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366 or in the Tel Profile table, see Configuring Tel Profiles on page 362.<br>**Notes:**<br>▪ If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.<br>▪ The parameter is applicable only to the Gateway application. |
| Enable Early 183<br>`early-183`<br>[EnableEarly183] | Global parameter that enables the device to send SIP 183 responses with SDP to the IP upon receipt of INVITE messages. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableEarly183). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 366. |

| Parameter | Description |
|---|---|
| | **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| 183 Message Behavior<br>`183-msg-behavior`<br>[SIP183Behaviour] | Defines the response of the device upon receipt of a SIP 183 response.<br>▪ [0] Progress = (Default) A 183 response (without SDP) does not cause the device to play a ringback tone.<br>▪ [1] Alert = 183 response is handled by the device as if a 180 Ringing response is received, and the device plays a ringback tone.<br>**Note:** The parameter is applicable only to the Gateway application. |
| Session-Expires Time<br>`session-expires-time`<br>[SIPSessionExpires] | Defines the numerical value sent in the Session-Expires header in the first INVITE request or response (if the call is answered).<br>The valid range is 1 to 86,400 sec. The default is 0 (i.e., the Session-Expires header is disabled).<br>**Note:** The parameter is applicable only to the Gateway application. |
| Minimum Session-Expires<br>`min-session-expires`<br>[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.<br>The valid range is 10 to 100,000. The default is 90.<br>**Note:** The parameter is applicable only to the Gateway application. |
| Session Expires Disconnect Time<br>`session-exp-disconnect-time`<br>[SessionExpiresDisconnectTime] | Defines a session expiry timeout.<br>The new session expiry timeout is calculated by subtracting the configured value from the original timeout as specified in the Session-Expires header. However, the new timeout must be greater than or equal to one-third (1/3) of the Session-Expires value. If the refresher does not send a refresh request within the new timeout, the device disconnects the session (i.e., sends a SIP BYE).<br>For example, if you configure the parameter to 32 seconds and the Session-Expires value is 180 seconds, the session timeout occurs 148 seconds (i.e., 180 minus 32) after the last session refresh. If the Session-Expires header value is 90 seconds, the timeout occurs 60 seconds after the last refresh. This is because 90 minus 32 is 58 seconds, which is less than one third of the Session-Expires value (i.e., 60/3 is 30, and 90 minus 30 is 60).<br>The valid range is 0 to 32 (in seconds). The default is 32. |
| Session Expires Method<br>`session-exp-method`<br>[SessionExpiresMethod] | Determines the SIP method used for session-timer updates.<br>▪ **[0]** Re-INVITE = (Default) Uses re-INVITE messages for session-timer updates.<br>▪ **[1]** UPDATE = Uses UPDATE messages.<br>**Notes:**<br>▪ The parameter is applicable only to the Gateway application.<br>▪ The device can receive session-timer refreshes using both methods.<br>▪ The UPDATE message used for session-timer is excluded from the SDP body. |
| [RemoveToTagInFailureResponse] | Determines whether the device removes the ‘to’ header tag from final SIP failure responses to INVITE transactions.<br>▪ **[0]** = (Default) Do not remove tag.<br>▪ **[1]** = Remove tag. |
| [EnableRTCPAttribute] | Enables the use of the 'rtcp' attribute in the outgoing SDP. |

| Parameter | Description |
|---|---|
| | ▪ **[0]** = Disable (default)<br>▪ **[1]** = Enable<br>**Note:** The parameter is applicable only to the Gateway application. |
| [OPTIONSUserPart] | Defines the user part value of the Request-URI for outgoing SIP OPTIONS requests. If no value is configured, the endpoint number (analog) is used.<br>A special value is ‘empty’, indicating that no user part in the Request-URI (host part only) is used.<br>The valid range is a 30-character string. By default, this value is not defined. |
| Fax Signaling Method<br>`fax-sig-method`<br>[IsFaxUsed] | Global parameter that defines the SIP signaling method for establishing and transmitting a fax session when the device detects a fax. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_IsFaxUsed). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| [HandleG711asVBD] | Enables the handling of G.711 as a G.711 Voice Band Data (VBD) coder.<br>▪ **[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.<br>▪ **[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.<br>**Note:** The parameter is applicable only if G.711 VBD coder(s) with regular G.711 payload types 0 or 8 are configured for the device (using the CodersGroup parameter). |
| `fax-vbd-behvr`<br>[FaxVBDBehavior] | Determines the device's fax transport behavior when G.711 VBD coder is negotiated at call start.<br>▪ **[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).<br>▪ **[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.<br>**Notes:** |

| Parameter | Description |
|---|---|
| | ▪ 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.<br>▪ 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.<br>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:<br>`a=rtpmap:120 NoAudio/8000\r\n`<br>**Note:** For incoming SDP offers, NoAudio is always supported. |
| SIP Transport Type<br>`app-sip-transport-type`<br>[SIPTransportType] | Determines the default transport layer for outgoing SIP calls initiated by the device.<br>▪ **[0]** UDP (default)<br>▪ **[1]** TCP<br>▪ **[2]** TLS (SIPS)<br>**Notes:**<br>▪ It's recommended to use TLS for communication with a SIP Proxy and not for direct device-to-device communication.<br>▪ For received calls (i.e., incoming), the device accepts all these protocols. |
| Display Default SIP Port<br>`display-default-sip-port`<br>[DisplayDefaultSIPPort] | Enables the device to add the default SIP port 5060 (UDP/TCP) or 5061 (TLS) to outgoing messages that are received without a port. This condition also applies to manipulated messages where the resulting message has no port number. The device adds the default port number to the following SIP headers: Request-Uri, To, From, P-Asserted-Identity, P-Preferred-Identity, and P-Called-Party-ID. If the message is received with a port number other than the default, for example, 5070, the port number is not changed.<br>An example of a SIP From header with the default port is shown below:<br>`From: <sip:+4000@10.8.4.105:`**`5060`**`;user=phone>;tag=f25419a96a;epid=009FAB8F3E`<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable |
| Enable SIPS<br>`enable-sips`<br>[EnableSIPS] | Enables secured SIP (SIPS URI) connections over multiple hops.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>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).<br>**Note:** If the parameter is enabled and the parameter SIPTransportType is set to 0 (i.e., UDP), the connection fails. |
| Enable TCP Connection Reuse | Enables the reuse of the same TCP connection for all calls to the same destination. |

| Parameter | Description |
|---|---|
| `tcp-conn-reuse`<br>[EnableTCPConnectionReuse] | ▪ **[0]** Disable = Uses a separate TCP connection for each call.<br>▪ **[1]** Enable = (Default) Uses the same TCP connection for all calls. |
| Fake TCP alias<br>`fake-tcp-alias`<br>[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.<br>▪ **[0]** Disable = (Default) TCP/TLS connection reuse is done only if the "alias" parameter is present in the Via header of the first INVITE.<br>▪ **[1]** Enable<br>**Note:** To enable TCP/TLS connection re-use, set the EnableTCPConnectionReuse parameter to 1. |
| Reliable Connection Persistent Mode<br>`reliable-conn-persistent`<br>[ReliableConnectionPersistentMode] | Enables setting of all TCP/TLS connections as persistent and therefore, not released.<br>▪ **[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.<br>▪ **[1]** = Enable - TCP connections to all destinations are persistent and not released unless the device reaches 70% of its maximum TCP resources.<br>While trying to send a SIP message connection, reuse policy determines whether live connections to the specific destination are re-used.<br>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.<br>**Note:** If the destination is a Proxy server, the TCP/TLS connection is persistent regardless of the settings of the parameter. |
| TCP Timeout<br>`tcp-timeout`<br>[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.<br>The valid range is 0 to 40 sec. The default is 64 * SipT1Rtx parameter value. For example, if SipT1Rtx is set to 500 msec, then the default of SIPTCPTimeout is 32 sec. |
| SIP Destination Port<br>`sip-dst-port`<br>[SIPDestinationPort] | Defines the SIP destination port for sending initial SIP requests.<br>The valid range is 1 to 65534. The default port is 5060.<br>**Note:** SIP responses are sent to the port specified in the Via header. |
| Use user=phone in SIP URL<br>`user=phone-in-url`<br>[IsUserPhone] | Determines whether the 'user=phone' string is added to the SIP URI and SIP To header.<br>▪ **[0]** No = 'user=phone' string is not added.<br>▪ **[1]** Yes = (Default) 'user=phone' string is part of the SIP URI and SIP To header. |
| Use user=phone in From Header<br>`phone-in-from-hdr`<br>[IsUserPhoneInFrom] | Determines whether the 'user=phone' string is added to the From and Contact SIP headers.<br>▪ **[0]** No = (Default) Doesn't add 'user=phone' string. |

| Parameter | Description |
|---|---|
| | ▪ **[1]** Yes = 'user=phone' string is part of the From and Contact headers. |
| Use Tel URI for Asserted Identity<br>`uri-for-assert-id`<br>[UseTelURIForAssertedID] | Determines the format of the URI in the P-Asserted-Identity and P-Preferred-Identity headers.<br>▪ **[0]** Disable = (Default) 'sip:'<br>▪ **[1]** Enable = 'tel:' |
| Tel to IP No Answer Timeout<br>`tel2ip-no-ans-timeout`<br>[IPAlertTimeout] | Defines the time (in seconds) that the device waits for a 200 OK response from the called party (IP side) after sending an INVITE message, for Tel-to-IP calls. If the timer expires, the call is released.<br>The valid range is 0 to 3600. The default is 180. |
| Enable Remote Party ID<br>`remote-party-id`<br>[EnableRPIheader] | Enables Remote-Party-Identity headers for calling and called numbers for Tel-to-IP calls.<br>▪ [0] Disable (default).<br>▪ [1] Enable = Remote-Party-Identity headers are generated in SIP INVITE messages for both called and calling numbers. |
| Use Tgrp Information<br>`use-tgrp-inf`<br>[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:<br>INVITE sip::+16305550100;tgrp=1;trunk-context=example.com@10.1.0.3;user=phone SIP/2.0<br>▪ [0] Disable = (Default) The 'tgrp' parameter isn't used.<br>▪ [1] Send Only = The Hunt Group number or name (configured in the Hunt Group Settings table) 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.<br>▪ [2] Send and Receive = The functionality of outgoing SIP messages is identical to the functionality described for option [1]. In addition, for incoming SIP INVITEs, if the Request-URI includes a 'tgrp' parameter, the device routes the call according to that value (if possible). The Contact header in the outgoing SIP INVITE (Tel-to-IP call) contains "tgrp=<source trunk group ID>;trunk-context=<gateway IP address>". The <source Hunt 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 Hunt Group ID>;trunk-context=<gateway IP address>”. The <destination Hunt 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.<br>**Note:** IP-to-Tel configuration (using the PSTNPrefix parameter) overrides the 'tgrp' parameter in incoming INVITE messages. |
| TGRP Routing Precedence<br>`tgrp-routing-prec`<br>[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.<br>▪ [0] = (Default) IP-to-Tel routing is determined by the IP to Hunt Group Routing table (PSTNPrefix parameter). If a matching rule is |

| Parameter | Description |
|---|---|
| | not found in this table, the device uses the Hunt Group parameters for routing the call.<br>▪ [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, the IP to Hunt Group Routing table is used for routing the call.<br>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:<br>`INVITE sip:200;tgrp=7;trunk-context=example.com@10.33.2.68;user=phone SIP/2.0`<br>**Note:** For IP-to-Tel routing based on the 'dtg' parameter (instead of the 'tgrp' parameter), use the parameter UseBroadsoftDTG. |
| `use-dtg`<br>[UseBroadsoftDTG] | Determines whether the device uses the 'dtg' parameter for routing IP-to-Tel calls to a specific Hunt Group.<br>▪ [0] Disable (default)<br>▪ [1] Enable<br>When the 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. The parameter is used instead of the 'tgrp/trunk-context' parameters. The 'dtg' parameter appears in the INVITE Request-URI (and in the To header).<br>For example, the received SIP message below routes the call to Hunt Group ID 56:<br>`INVITE sip:123456@192.168.1.2;dtg=56;user=phone SIP/2.0`<br>**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. |
| Enable GRUU<br>`enable-gruu`<br>[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.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>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:<br>`REFER sip:C@domain.com SIP/2.0`<br>`From: sip:A@domain.com;tag=99asd`<br>`To: sip:C@domain.com`<br>`Refer-To: (URI that identifies B's UA)`<br>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. |

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>
<ul>
<li>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:
<ul>
<li>✓ If the REGISTER is per the device’s client (endpoint), it is the MAC address concatenated with the phone number of the client.</li>
<li>✓ If the REGISTER is per device, it is the MAC address only.</li>
<li>✓ 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.</li>
</ul></li>
</ul>
If the remote server doesn’t support GRUU, it ignores the parameters of the GRUU. Otherwise, if the remote side also supports GRUU, the REGISTER responses contain the “gruu” parameter in each Contact header. The parameter contains a SIP or SIPS URI that represents a GRUU corresponding to the UA instance that registered the contact. The server provides the same GRUU for the same AOR and instance-id when sending REGISTER again after registration expiration. RFC 5627 specifies that the remote target is a GRUU target if its’ Contact URL has the "gr" parameter with or without a value.
<ul>
<li>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.</li>
</ul>
</td></tr>
<tr><td>[IsCiscoSCEMode]</td><td>
Determines whether a Cisco gateway exists at the remote side.
<ul>
<li>[0] = (Default) No Cisco gateway exists at the remote side.</li>
<li>[1] = A Cisco gateway exists at the remote side.</li>
</ul>
When a Cisco gateway exists at the remote side, the device must set the value of the 'annexb' parameter of the fmtp attribute in the SDP to 'no'. This logic is used if Silence Suppression for the used coder is configured to 2 (enable without adaptation). In this case, Silence Suppression is used on the channel but not declared in the SDP.
<ul>
<li>Note:</li>
<li>The parameter is applicable only to the Gateway application.</li>
<li>The IsCiscoSCEMode parameter is applicable only when the selected coder is G.729.</li>
</ul>
</td></tr>
<tr><td>User-Agent Information<br><code>user-agent-info</code><br>[UserAgentDisplayInfo]</td><td>
Defines the string that is used in the SIP User-Agent and Server response headers. When configured, the string <UserAgentDisplayInfo value>/software version' is used, for example:
<pre>User-Agent: myproduct/v.6.80A.227.005</pre>
If not configured, the default string, <AudioCodes product-name>/software version' is used, for example:
<pre>User-Agent: Audiocodes-Sip-Gateway-Mediant 800B
Gateway and E-SBC/v.6.80A.227.005</pre>
The maximum string length is 50 characters.<br>
Note: The software version number and preceding forward slash (/) cannot be modified. Therefore, it is recommended not to include a
</td></tr>
</table>

| Parameter | Description |
|---|---|
| | forward slash in the parameter's value (to avoid two forward slashes in the SIP header, which may cause problems). |
| SDP Session Owner<br>`sdp-session-owner`<br>[SIPSDPSessionOwner] | Defines the value of the Owner line ('o' field) in outgoing SDP messages.<br>The valid range is a string of up to 39 characters. The default is "AudiocodesGW".<br>For example:<br>`o=AudiocodesGW 1145023829 1145023705 IN IP4 10.33.4.126` |
| `sdp-ver-nego`<br>[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.<br>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.<br>▪ **[0]** Disable = (Default) The device negotiates any new SDP re-offer, regardless of the origin field.<br>▪ **[1]** Enable = The device negotiates only an SDP re-offer with an incremented origin field. |
| Subject<br>`usr-def-subject`<br>[SIPSubject] | Defines the Subject header value in outgoing INVITE messages. If not specified, the Subject header isn't included (default).<br>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.<br>▪ [0] = (Default) Coder negotiation is given higher priority to the remote UA's list of supported coders.<br>▪ [1] = Coder negotiation is given higher priority to the device's (local) supported coders list.<br>**Note:** The parameter is applicable only to the Gateway application. |
| Send All Coders on Retrieve<br>`send-all-cdrs-on-rtrv`<br>[SendAllCodersOnRetrieve] | Enables coder re-negotiation in the sent re-INVITE for retrieving an on-hold call.<br>▪ [0] Disable = (Default) Sends only the initially chosen coder when the call was first established and then put on-hold.<br>▪ [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.<br>The parameter is useful in the following call scenario example:<br>1 Party A calls party B and coder G.711 is chosen.<br>2 Party B is put on-hold while Party A blind transfers Party B to Party C.<br>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. |

| Parameter | Description |
|---|---|
| | **Note:** The parameter is applicable only to the Gateway application. |
| Multiple Packetization Time Format<br>`mult-ptime-format`<br>[MultiPtimeFormat] | Determines whether the 'mptime' attribute is included in the outgoing SDP.<br>▪ **[0]** None = (Default) Disabled.<br>▪ **[1]** PacketCable = Includes the 'mptime' attribute in the outgoing SDP - PacketCable-defined format.<br>The mptime' attribute enables the device to define a separate packetization period for each negotiated coder in the SDP. The 'mptime' attribute is only included if the parameter is enabled even if the remote side includes it in the SDP offer. Upon receipt, each coder receives its 'ptime' value in the following precedence: from 'mptime' attribute, from 'ptime' attribute, and then from default value. |
| [EnablePtime] | Determines whether the 'ptime' attribute is included in the SDP.<br>▪ **[0]** = Remove the 'ptime' attribute from SDP.<br>▪ **[1]** = (Default) Include the 'ptime' attribute in SDP. |
| 3xx Behavior<br>`3xx-behavior`<br>[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.<br>▪ **[0]** Forward = (Default) Use different call identifiers for a redirected INVITE message.<br>▪ **[1]** Redirect = Use the same call identifiers. |
| Enable P-Charging Vector<br>`p-charging-vector`<br>[EnablePChargingVector] | Enables the inclusion of the P-Charging-Vector header to all outgoing INVITE messages.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Note:** The parameter is applicable only to the Gateway application. |
| Retry-After Time<br>`retry-aftr-time`<br>[RetryAfterTime] | Defines the time (in seconds) used in the Retry-After header when a 503 (Service Unavailable) response is generated by the device.<br>The time range is 0 to 3,600. The default is 0. |
| Fake Retry After<br>`fake-retry-after`<br>[FakeRetryAfter] | Determines whether the device, upon receipt of a SIP 503 response without a Retry-After header, behaves as if the 503 response included a Retry-After header and with the period (in seconds) specified by the parameter.<br>▪ **[0]** Disable (default)<br>▪ Any positive value (in seconds) for defining the period<br>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.<br>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.<br>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. |

| Parameter | Description |
|---|---|
| Enable P-Associated-URI Header<br>`p-associated-uri-hdr`<br>[EnablePAssociatedURIHeader] | Determines the device usage of the P-Associated-URI header. This header can be received in 200 OK responses to REGISTER requests. When enabled, the first URI in the P-Associated-URI header is used in subsequent requests as the From/P-Asserted-Identity headers value.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Note:** P-Associated-URIs in registration responses is handled only if the device is registered per endpoint (using the User Information file). |
| Source Number Preference<br>`src-nb-preference`<br>[SourceNumberPreference] | Determines from which SIP header the source (calling) number is obtained in incoming INVITE messages.<br>▪ If not configured or if any string other than "From" or "Pai2" is configured, the calling number is obtained from a specific header using the following logic:<br>a. P-Preferred-Identity header.<br>b. If the above header is not present, then the first P-Asserted-Identity header is used.<br>c. If the above header is not present, then the Remote-Party-ID header is used.<br>d. If the above header is not present, then the From header is used.<br>▪ "**From**" = The calling number is obtained from the From header.<br>▪ "**Pai2**" = The calling number is obtained using the following logic:<br>a. If a P-Preferred-Identity header is present, the number is obtained from it.<br>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.<br>c. If only one P-Asserted-Identity header is present, the calling number is obtained from it.<br>**Notes:**<br>▪ The "From" and "Pai2" values are not case-sensitive.<br>▪ 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. |
| `src-hdr-4-called-nb`<br>[SelectSourceHeaderForCalledNumber] | Determines the SIP header used for obtaining the called number (destination) for IP-to-Tel calls.<br>▪ [0] Request-URI header = (Default) Obtains the destination number from the user part of the Request-URI.<br>▪ [1] To header = Obtains the destination number from the user part of the To header.<br>▪ [2] P-Called-Party-ID header = Obtains the destination number from the P-Called-Party-ID header. |
| Enable Reason Header<br>`reason-header`<br>[EnableReasonHeader] | Enables the usage of the SIP Reason header.<br>▪ **[0]** Disable<br>▪ **[1]** Enable (default) |
| Gateway Name<br>`gw-name`<br>[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).<br>**Notes:** |

| Parameter | Description |
|---|---|
| | ▪ Ensure that the parameter value is the one with which the Proxy has been configured with to identify the device.<br>▪ The parameter can also be configured for an IP Group (in the IP Group table). |
| [ZeroSDPHandling] | Determines the device's response to an incoming SDP that includes an IP address of 0.0.0.0 in the SDP's Connection Information field (i.e., "c=IN IP4 0.0.0.0").<br>▪ **[0]** = (Default) Sets the IP address of the outgoing SDP's c= field to 0.0.0.0.<br>▪ **[1]** = Sets the IP address of the outgoing SDP c= field to the IP address of the device. If the incoming SDP doesn't contain the "a=inactive" line, the returned SDP contains the "a=recvonly" line. |
| Enable Delayed Offer<br>`delayed-offer`<br>[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.)<br>▪ **[0]** Disable = (Default) The device sends the initial INVITE message with an SDP.<br>▪ **[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".<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable |
| Enable Contact Restriction<br>`contact-restriction`<br>[EnableContactRestriction] | Determines whether the device sets the Contact header of outgoing INVITE requests to 'anonymous' for restricted calls.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable |
| `anonymous-mode`<br>[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.<br>▪ [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><br>▪ [1] = The device's IP address is used as the URI host part instead of "anonymous.invalid".<br>The 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 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 |

| Parameter | Description |
|---|---|
| | 3325. However, when the parameter is set to 1, the device replaces the "anonymous.invalid" with its IP address. |
| `p-assrtd-usr-name`<br>[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.<br>The default is null. |
| [UseAORInReferToHeader] | Defines the source for the SIP URI set in the Refer-To header of outgoing REFER messages.<br>▪ **[0]** = (Default) Use SIP URI from Contact header of the initial call.<br>▪ **[1]** = Use SIP URI from To/From header of the initial call. |
| Enable User-Information Usage<br>`user-inf-usage`<br>[EnableUserInfoUsage] | Enables the usage of the User Information, which is loaded to the idevice> in the User Information Auxiliary file. For more nformation on User Information, see "User Information File" on page 504.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Note:** For the parameter to take effect, a device reset is required. |
| [HandleReasonHeader] | Determines whether the device uses the value of the incoming SIP Reason header for Release Reason mapping.<br>▪ **[0]** = Disregard Reason header in incoming SIP messages.<br>▪ **[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.<br>▪ **[0]** = (Default) Disregard the 'silecesupp' attribute.<br>▪ **[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.<br>**Note:** The parameter is applicable only if the G.711 coder is used. |
| [EnableRport] | Enables the usage of the 'rport' parameter in the Via header.<br>▪ **[0]** = Disabled (default)<br>▪ **[1]** = Enabled<br>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.<br>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.<br>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.<br>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. |
| Enable X-Channel Header | Determines whether the SIP X-Channel header is added to SIP messages for providing information on the physical Trunk/B-channel on which the call is received or placed. |

| Parameter | Description |
|---|---|
| `x-channel-header`<br>[XChannelHeader] | ▪ **[0]** Disable = (Default) X-Channel header is not used.<br>▪ **[1]** Enable = X-Channel header is generated by the device and sent in INVITE messages and 180, 183, and 200 OK SIP responses. The header includes the Trunk number, B-channel, and the device's IP address.<br>For example, 'x-channel: DS/DS1-1/8;IP=192.168.13.1', where:<br>✓ 'DS/DS-1' is a constant string<br>✓ '8' is the channel<br>✓ 'IP=192.168.13.1' is the device's IP address |
| Progress Indicator to IP<br>`prog-ind-2ip`<br>[ProgressIndicator2IP] | Global parameter that defines the progress indicator (PI) sent to the IP. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_ProgressIndicator2IP) or Tel Profiles. For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 366 or in the Tel Profile table, see Configuring Tel Profiles on page 362.<br>**Note:** If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile. |
| [EnableRekeyAfter181] | Enables the device to send a re-INVITE with a new (different) SRTP key (in the SDP) if a SIP 181 response is received ("call is being forwarded"). The re-INVITE is sent immediately upon receipt of the 200 OK (when the call is answered).<br>▪ **[0]** = Disable (default)<br>▪ **[1]** = Enable<br>**Note:** The parameter is applicable only if SRTP is used. |
| [NumberOfActiveDialogs] | Defines the maximum number of concurrent, outgoing SIP REGISTER dialogs. The parameter is used to control the registration rate.<br>The valid range is 1 to 20. The default is 20.<br>**Notes:**<br>▪ Once a 200 OK is received in response to a REGISTER message, the REGISTER message is not considered in this maximum count limit.<br>▪ The parameter applies only to outgoing REGISTER messages (i.e., incoming is unlimited). |
| Network Node ID<br>net-node-id<br>[NetworkNodeId] | Defines the Network Node Identifier of the device for Avaya UCID.<br>The valid value range is1 to 0x7FFF. The default is 0.<br>**Notes:**<br>▪ To use this feature, you must set the parameter to any value other than 0.<br>▪ To enable the generation by the device of the Avaya UCID value and adding it to the outgoing INVITE sent to the IP Group (Avaya entity), use the IP Group table's parameter 'UUI Format'. |
| Default Release Cause<br>`dflt-release-cse`<br>[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.<br>The default release cause is NO_ROUTE_TO_DESTINATION (3). Other common values include NO_CIRCUIT_AVAILABLE (34), DESTINATION_OUT_OF_ORDER (27), etc.<br>**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).<br>▪ Analog: For more information on mapping PSTN release causes to SIP responses, see Mapping PSTN Release Cause to SIP Response on page 408.<br>▪ When the Trunk is disconnected or is not synchronized, the internal cause is 27. This cause is mapped, by default, to SIP 502.<br>▪ For mapping SIP-to-Q.931 and Q.931-to-SIP release causes, see Configuring Release Cause Mapping on page 408.<br>▪ 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 435. |
| Enable Microsoft Extension<br>`microsoft-ext`<br>[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.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>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. |
| [UseSIPURIForDiversionHeader] | Defines the URI format in the SIP Diversion header.<br>▪ **[0]** = 'tel:' (default)<br>▪ **[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.<br>The valid range is 0 to 180,000 (i.e., 3 minutes). The default is 32000 (i.e., 32 sec). |
| [EnableImmediateTrying] | Determines if and when the device sends a 100 Trying in response to an incoming INVITE request.<br>▪ [0] = 100 Trying response is sent upon receipt of a Proceeding message from the PSTN.<br>▪ [1] = (Default) 100 Trying response is sent immediately upon receipt of INVITE request. |
| [TransparentCoderPresentation] | Determines the format of the Transparent coder representation in the SDP.<br>▪ [0] = clearmode (default)<br>▪ [1] = X-CCD |

| Parameter | Description |
|---|---|
| [IgnoreRemoteSDPMKI] | Determines whether the device ignores the Master Key Identifier (MKI) if present in the SDP received from the remote side.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable |
| Comfort Noise Generation Negotiation<br>`com-noise-gen-nego`<br>[ComfortNoiseNegotiation] | Enables negotiation and usage of Comfort Noise (CN) for Gateway calls.<br>▪ [0] Disable<br>▪ [1] Enable (default)<br>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.<br>To determine CNG support, the device uses the ComfortNoiseNegotiation parameter and the codec's SCE (silence suppression setting) using the CodersGroup parameter.<br>If the ComfortNoiseNegotiation parameter is enabled, then the following occurs:<br>▪ 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.<br>▪ 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.<br>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.<br>**Note:** The parameter is applicable only to the Gateway application. |
| `sdp-ecan-frmt`<br>[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.<br>▪ **[0]** = (Default) The 'ecan' attribute appears on the 'a=gpmd' line.<br>▪ **[1]** = The 'ecan' attribute appears as a separate attribute.<br>▪ **[2]** = The 'ecan' attribute is not included in the SDP.<br>▪ **[3]** = The 'ecan' attribute and the 'vbd' parameter are not included in the SDP.<br>**Note:** The parameter is applicable only when the IsFaxUsed parameter is set to 2, and for re-INVITE messages generated by the device as result of modem or fax tone detection. |
| First Call Ringback Tone ID<br>`1st-call-rbt-id`<br>[FirstCallRBTId] | Defines the index of the first ringback tone in the CPT file. This option enables an Application server to request the device to play a distinctive ringback tone to the calling party according to the destination of the call. The tone is played according to the Alert-Info header received in the 180 Ringing SIP response (the value of the Alert-Info header is added to the value of the parameter).<br>The valid range is -1 to 1,000. The default is -1 (i.e., play standard ringback tone).<br>**Notes:** |

| Parameter | Description |
|---|---|
| | ▪ It is assumed that all ringback tones are defined in sequence in the CPT file.<br>▪ In case of an MLPP call, the device uses the value of the parameter plus 1 as the index of the ringback tone in the CPT file (e.g., if this value is set to 1, then the index is 2, i.e., 1 + 1). |
| Reanswer Time<br>`reanswer-time`<br>[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.<br>The valid range is 0 to 255 (in seconds). The default is 0. |
| Enable Reanswering Info<br>`reans-info-enbl`<br>[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.<br>▪ [0] Disable (default)<br>▪ [1] Enable<br>The 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).<br>The INFO message format is as follows:<br>INFO sip:12345@10.50.228.164:5082 SIP/2.0<br>Via: SIP/2.0/UDP 127.0.0.1;branch=z9hG4bK_05_905924040-90579<br>From: <sip:+551137077803@ims.acme.com.br:5080;user=phone>;tag=008277765<br>To: <sip:notavailable@unknown.invalid>;tag=svw-0-1229428367<br>Call-ID: ConorCCR-0-LU-1229417827103300@dtas-stdn.fs5000group0-000.l<br>CSeq: 1 INFO<br>Contact: sip:10.20.7.70:5060<br>Content-Type: application/On-Hook (application/Off-Hook)<br>Content-Length: 0<br>**Notes:**<br>▪ The parameter is applicable only if the parameter RegretTime is configured.<br>▪ The parameter is applicable only to FXS interfaces. |
| PSTN Alert Timeout<br>`pstn-alert-timeout`<br>[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), 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 the parameter, the device cancels the call by sending a SIP 408 response.<br>The valid value range is 1 to 600 (in seconds). The default is 180. |

| Parameter | Description |
|---|---|
| RTP Only Mode<br>`rtp-only-mode`<br>[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 Tel-to-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).<br>▪ **[0]** Disable (default)<br>▪ **[1]** Transmit & Receive = Send and receive RTP packets.<br>▪ **[2]** Transmit Only= Send RTP packets only.<br>▪ **[3]** Receive Only= Receive RTP packets only.<br>**Notes:**<br>▪ To configure the RTP Only mode per trunk, use the RTPOnlyModeForTrunk_x parameter.<br>▪ If per trunk configuration (using the RTPOnlyModeForTrunk_ID parameter) is set to a value other than the default, the RTPOnlyMode parameter value is ignored. |
| Media IP Version Preference<br>`media-ip-ver-pref`<br>[MediaIPVersionPreference] | Global parameter that defines the preferred RTP media IP addressing version (IPv4 or IPv6) for outgoing SIP calls. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_MediaIPVersionPreference). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 366. |
| SIT Q850 Cause<br>[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.<br>The valid range is 0 to 127. The default is 34.<br>**Note:** For mapping specific SIT tones, you can use the SITQ850CauseForNC, SITQ850CauseForIC, SITQ850CauseForVC, and SITQ850CauseForRO parameters. |
| SIT Q850 Cause For IC<br>[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.<br>The valid range is 0 to 127. The default is -1 (not configured).<br>**Notes:**<br>▪ When not configured (i.e., default), the SITQ850Cause parameter is used.<br>▪ |
| SIT Q850 Cause For VC<br>[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.<br>The valid range is 0 to 127. The default is -1 (not configured).<br>**Note:** When not configured (i.e., default), the SITQ850Cause parameter is used. |
| SIT Q850 Cause For RO<br>[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.<br>The valid range is 0 to 127. The default is -1 (not configured). |

| Parameter | Description |
|---|---|
| | **Note:** When not configured (i.e., default), the SITQ850Cause parameter is used. |
| [GWInboundManipulationSet] | Selects the Manipulation Set ID for manipulating all inbound INVITE messages. The Manipulation Set is defined using the MessageManipulations parameter. By default, no manipulation is done (i.e. Manipulation Set ID is set to -1).<br>**Note:** The parameter is applicable only to the Gateway application. |
| [GWOutboundManipulationSet] | Selects the Manipulation Set ID for manipulating all outbound INVITE messages. The Manipulation Set is defined using the MessageManipulations parameter. By default, no manipulation is done (i.e. Manipulation Set ID is set to -1).<br>**Notes:**<br>▪ The parameter is used only if the Outbound Message Manipulation Set parameter of the destination IP Group is not set.<br>▪ The parameter is applicable only to the Gateway application. |
| Out-of-Service (Busy Out) Parameters | |
| Enable Busy Out<br>`busy-out`<br>[EnableBusyOut] | Enables the Busy Out feature.<br>▪ [0] Disable (Default)<br>▪ [1] Enable<br>When Busy Out is enabled and certain scenarios exist, the device does the following:<br>▪ Analog: The FXS port behaves according to the settings of the FXSOOSBehavior parameter such as plays a reorder tone when the phone is off-hooked, or changes the line polarity.<br>The above behavior is done upon one of the following scenarios:<br>▪ The device is physically disconnected from the network (i.e., Ethernet cable is disconnected).<br>▪ The device can't communicate with the Proxy Sets (according to the Proxy Keep-Alive mechanism) associated with the destination IP Groups for matching routing rules in the Tel-to-IP Routing table, and no other alternative route exists to send the call.<br>▪ The IP Connectivity mechanism is enabled (see the AltRoutingTel2IPEnable parameter) and there is no connectivity to any destination IP address configured for matching routing rules in the Tel-to-IP Routing table.<br>**Note:**<br>▪ If the AltRoutingTel2IPEnable parameter is enabled, the Busy Out feature does not function with the Proxy Set keep-alive mechanism. To use the Busy Out feature with the Proxy Set keep-alive mechanism (for IP Groups), disable the AltRoutingTel2IPEnable parameter.<br>▪ Analog:<br>✓ The FXSOOSBehavior parameter determines the behavior of the FXS endpoints when a Busy Out or Graceful Lock occurs. |
| Out-Of-Service Behavior<br>`oos-behavior`<br>[FXSOOSBehavior] | Determines the behavior of FXS endpoints when a Busy Out condition exists.<br>▪ [0] None = Silence is heard when the FXS endpoint goes off-hook. |

| Parameter | Description |
|---|---|
| | ▪ [1] Reorder Tone = (Default) The device plays a reorder tone to the connected phone / PBX.<br>▪ [2] Polarity Reversal = The device reverses the polarity of the endpoint making it unusable (relevant, for example, for PBX DID lines).<br>▪ [3] Reorder Tone + Polarity Reversal = Same as options [1] and [2].<br>▪ [4] Current Disconnect = The device disconnects the current to the FXS endpoint.<br>**Notes:**<br>▪ A device reset is required for the parameter to take effect when it is set to [2], [3], or [4].<br>▪ The parameter is applicable only to FXS interfaces. |
| **Retransmission Parameters** | |
| SIP T1 Retransmission Timer<br>`t1-re-tx-time`<br>[SipT1Rtx] | Defines the time interval (in msec) between the first transmission of a SIP message and the first retransmission of the same message.<br>The default is 500.<br>**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:<br>▪ The first retransmission is sent after 500 msec.<br>▪ The second retransmission is sent after 1000 (2*500) msec.<br>▪ The third retransmission is sent after 2000 (2*1000) msec.<br>▪ The fourth retransmission and subsequent retransmissions until SIPMaxRtx are sent after 4000 (2*2000) msec. |
| SIP T2 Retransmission Timer<br>`t2-re-tx-time`<br>[SipT2Rtx] | Defines the maximum interval (in msec) between retransmissions of SIP messages (except for INVITE requests).<br>The default is 4000.<br>**Note:** The time interval between subsequent retransmissions of the same SIP message starts with SipT1Rtx and is multiplied by two until SipT2Rtx. |
| SIP Maximum RTX<br>`sip-max-rtx`<br>[SIPMaxRtx] | Defines the maximum number of UDP transmissions of SIP messages (first transmission plus retransmissions).<br>The range is 1 to 30. The default is 7. |
| Number of RTX Before Hot-Swap<br>`nb-of-rtx-b4-hot-swap`<br>[HotSwapRtx] | Defines the number of retransmitted INVITE/REGISTER messages before the call is routed (hot swap) to another Proxy/Registrar.<br>The valid range is 1 to 30. The default is 3.<br>For example, if configured to 3 and no response is received from an IP destination, the device attempts another three times to send the call to the IP destination. If still unsuccessful, it attempts to redirect the call to another IP destination.<br>**Note:** The parameter is also used for alternative routing (see "Alternative Routing Based on IP Connectivity" on page 429. |
| SIP Message Manipulations Table | |
| Message Manipulations<br>`configure voip > sbc manipulations` | Defines manipulation rules for SIP header messages.<br>The format of the ini file table parameter is as follows: |

| Parameter | Description |
|---|---|
| `message-manipulations`<br>[MessageManipulations] | [ MessageManipulations]<br>FORMAT MessageManipulations_Index = MessageManipulations_ManSetID, MessageManipulations_MessageType, MessageManipulations_Condition, MessageManipulations_ActionSubject, MessageManipulations_ActionType, MessageManipulations_ActionValue, MessageManipulations_RowRole;<br>[\MessageManipulations]<br>For example, the below configuration changes the user part of the SIP From header to 200:<br>MessageManipulations 1 = 0, Invite.Request, , Header.From.Url.User, 2, 200, 0;<br>For a detailed description of the table, see Configuring SIP Message Manipulation on page 349. |
| Message Policy Table | |
| Message Policy Table<br>`configure voip > sbc message-policy`<br>[MessagePolicy] | Defines SIP message policy rules for blocking (blacklist) unwanted incoming SIP messages or allowing (whitelist) receipt of desired messages.<br>The format of the ini file table parameter is as follows:<br>[MessagePolicy]<br>FORMAT MessagePolicy_Index = MessagePolicy_Name, MessagePolicy_MaxMessageLength, MessagePolicy_MaxHeaderLength, MessagePolicy_MaxBodyLength, MessagePolicy_MaxNumHeaders, MessagePolicy_MaxNumBodies, MessagePolicy_SendRejection, MessagePolicy_MethodList, MessagePolicy_MethodListType, MessagePolicy_BodyList, MessagePolicy_BodyListType;<br>[/MessagePolicy]<br>For a detailed description of the table, see Configuring SIP Message Policy Rules. |

## 45.8 Coders and Profile Parameters

The profile parameters are described in the table below.

**Table 45-32: Profile Parameters**

| Parameter | Description |
|---|---|
| Coders Table / Coder Groups Table | |
| Coders Table/Coder Group Settings<br>`configure voip > coders-and-profiles coders-group`<br>[CodersGroup0]<br>[CodersGroup1]<br>[CodersGroup2]<br>[CodersGroup3] | 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.<br>The format of the ini file table parameter is as follows:<br>[ CodersGroup<0-9> ]<br>FORMAT CodersGroup0_Index = CodersGroup0_Name, CodersGroup0_pTime, CodersGroup0_rate, CodersGroup0_PayloadType, CodersGroup0_Sce, |

| Parameter | Description |
|---|---|
| [CodersGroup4]<br>[CodersGroup5]<br>[CodersGroup6]<br>[CodersGroup7]<br>[CodersGroup8]<br>[CodersGroup9] | CodersGroup0_CoderSpecific;<br>[ \CodersGroup<0-9 ]<br>**Notes:**<br>▪ For a list of supported coders and a description of the table, see Configuring Default Coders on page 357.<br>▪ For configuring Coder Groups, see "Configuring Coder Groups" on page 360.<br>▪ The coder name is case-sensitive. |
| **IP Profile Settings Table** | |
| IP Profile Settings<br>`configure voip > coders-and-profiles ip-profile`<br>[IPProfile] | Defines the IP Profile table. The format of the ini file table parameter is as follows:<br>[IPProfile]<br>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_MediaIPVersionPreference, IpProfile_AMDSensitivityParameterSuit, IpProfile_AMDSensitivityLevel, IpProfile_AMDMaxGreetingTime, IpProfile_AMDMaxPostSilenceGreetingTime, IpProfile_EnableQSIGTunneling, IpProfile_EnableSymmetricMKI, IpProfile_MKISize, IpProfile_EnableEarly183, IpProfile_EarlyAnswerTimeout, IpProfile_ResetSRTPStateUponRekey, IpProfile_AmdMode, IpProfile_GenerateSRTPKeys, IpProfile_JitterBufMaxDelay;<br>[\IPProfile]<br>For a description of the table, see "Configuring IP Profiles" on page 366. |
| Tel Profile Table | |
| Tel Profile Settings<br>`configure voip > coders-and-profiles tel-profile`<br>[TelProfile] | Defines 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 Hunt Group table (TrunkGroup parameter). Therefore, Tel Profiles allow you to apply the same settings of a group of parameters to multiple channels, or apply specific settings to different channels.<br>The format of the ini file table parameter is as follows:<br>[TelProfile]<br>FORMAT TelProfile_Index = TelProfile_ProfileName, TelProfile_TelPreference, TelProfile_CodersGroupID, TelProfile_IsFaxUsed, TelProfile_JitterBufMinDelay, TelProfile_JitterBufOptFactor, TelProfile_IPDiffServ, |

| Parameter | Description |
|---|---|
| | 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_CallPriorityMode;<br>[\TelProfile]<br>For a description of the parameter, see Configuring Tel Profiles on page 362. |

## 45.9 Channel Parameters

This subsection describes the device's channel parameters.

### 45.9.1 Voice Parameters

The voice parameters are described in the table below.

**Table 45-33: Voice Parameters**

| Parameter | Description |
|---|---|
| Input Gain<br>`input-gain`<br>[InputGain] | Global parameter that defines the pulse-code modulation (PCM) input (received) gain control level (in decibels). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_InputGain) or Tel Profiles. For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366 or in the Tel Profile table, see Configuring Tel Profiles on page 362.<br>**Note:** If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile. |
| Voice Volume<br>`voice-volume`<br>[VoiceVolume] | Global parameter that defines the voice gain control (in decibels). This defines the level of the transmitted (IP-to-Tel) signal. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_VoiceVolume) or Tel Profiles. For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366 or in the Tel Profile table, see Configuring Tel Profiles on page 362.<br>**Note:** If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile. |
| `G726-voice-payload-format`<br>[VoicePayloadFormat] | Determines the bit ordering of the G.726 voice payload format.<br>▪ **[0]** = (Default) Little Endian<br>▪ **[1]** = Big Endian<br>**Note:** To ensure high voice quality when using G.726, both communicating ends should use the same endianness format. Therefore, when the device communicates with a third-party entity that uses the G.726 voice coder and voice quality is poor, change the settings of the parameter (between Big Endian and Little Endian). |
| MF Transport Type<br>`MF-transport-type`<br>[MFTransportType] | Currently, not supported. |
| Echo Canceler<br>`echo-canceller-enable`<br>[EnableEchoCanceller] | Global parameter that enables echo cancellation (i.e., echo from voice calls is removed). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableEchoCanceller) or Tel Profiles. For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP |

| Parameter | Description |
|---|---|
| | Profiles" on page 366 or Tel Profile table, see Configuring Tel Profiles on page 362.<br>**Note:** If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile. |
| Echo Canceller Type<br>`echo-canceller-type`<br>[EchoCancellerType] | Defines the echo canceller type.<br>▪ [0] Line echo canceller = (Default) Echo canceller for Tel side.<br>▪ [1] Acoustic Echo suppressor - netw = Echo canceller for IP side. |
| `echo-canceller-hybrid-loss`<br>[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.<br>▪ **[0]** = (Default) 6 dB<br>▪ **[1]** = N/A<br>▪ **[2]** = 0 dB<br>▪ **[3]** = 3 dB |
| `echo-canceller-NLP-mode`<br>[ECNLPMode] | Enables Non-Linear Processing (NLP) mode for echo cancellation.<br>▪ **[0]** = (Default) NLP adapts according to echo changes<br>▪ **[1]** = Disables NLP<br>**Note:** The parameter can also be configured in a Tel Profile. |
| `echo-canceller-aggressive-NLP`<br>[EchoCancellerAggressiveNLP] | Enables the Aggressive NLP at the first 0.5 second of the call.<br>▪ **[0]** = Disable<br>▪ **[1]** = (Default) Enable. The echo is removed only in the first half of a second of the incoming IP signal.<br>**Note:** For the parameter to take effect, a device reset is required. |
| `number-of-SID-coefficients`<br>[RTPSIDCoeffNum] | Defines the number of spectral coefficients added to an SID packet being sent according to RFC 3389.<br>The valid values are **[0]** (default), **[4]**, **[6]**, **[8]** and **[10]**. |
| **Answer Detector (AD) Parameters** | |
| Enable Answer Detector<br>[EnableAnswerDetector] | Currently, not supported. |
| Answer Detector Activity Delay<br>`answer-detector-activativity-delay`<br>[AnswerDetectorActivityDelay] | Defines the time (in 100-msec resolution) between activating the Answer Detector and the time that the detector actually starts to operate.<br>The valid range is 0 to 1023. The default is 0. |
| Answer Detector Silence Time<br>[AnswerDetectorSilenceTime] | Currently, not supported. |
| Answer Detector Redirection<br>[AnswerDetectorRedirection] | Currently, not supported. |

| Parameter | Description |
|---|---|
| Answer Detector Sensitivity<br>`answer-detector-sensitivity`<br>[AnswerDetectorSensitivity] | Defines the Answer Detector sensitivity.<br>The range is 0 (most sensitive) to 2 (least sensitive). The default is 0. |

## 45.9.2 Coder Parameters

The coder parameters are described in the table below.

**Table 45-34: Coder Parameters**

| Parameter | Description |
|---|---|
| Silk Tx Inband FEC<br>`silk-tx-inband-fec`<br>[SilkTxInbandFEC] | Enables forward error correction (FEC) for the SILK coder.<br>▪ [0] Disable (default)<br>▪ [1] Enable |
| Silk Max Average Bit Rate<br>`silk-max-average-bitrate`<br>[SilkMaxAverageBitRate] | Defines the maximum average bit rate for the SILK coder.<br>The valid value range is 5000 to 30000. The default is 16000.<br>The SILK coder is Skype's default audio codec used for Skype-to-Skype calls. |
| `vbr-coder-header-format`<br>[VBRCoderHeaderFormat] | Determines the format of the RTP header for VBR coders.<br>▪ [0] = (Default) Payload only (no header, TOC, or m-factor) - similar to RFC 3558 Header Free format.<br>▪ [1] = Supports RFC 2658 - 1 byte for interleaving header (always 0), TOC, no m-factor.<br>▪ [2] = Payload including TOC only, allow m-factor.<br>▪ [3] = RFC 3558 Interleave/Bundled format. |
| `vbr-coder-hangover`<br>[VBRCoderHangover] | Defines the required number of silence frames at the beginning of each silence period when using the VBR coder silence suppression.<br>The range is 0 to 255. The default is 1. |
| AMR Payload Format<br>[AmrOctetAlignedEnable] | Defines the AMR payload format type.<br>▪ [0] Bandwidth Efficient<br>▪ [1] Octet Aligned (default)<br>**Note:** The AMR payload type can also be configured per Coder Group (see Configuring Coder Groups on page 360). The Coder Group configuration overrides the parameter. |
| [AMRCoderHeaderFormat] | Determines the payload format of the AMR header.<br>▪ [0] = Non-standard multiple frames packing in a single RTP frame. Each frame has a CMR and TOC header.<br>▪ [1] = AMR frame according to RFC 3267 bundling.<br>▪ [2] = AMR frame according to RFC 3267 interleaving.<br>▪ [3] = AMR is passed using the AMR IF2 format.<br>**Note:** Bandwidth Efficient mode is not supported; the mode is always Octet-aligned. |

### 45.9.3 DTMF Parameters

The dual-tone multi-frequency (DTMF) parameters are described in the table below.

**Table 45-35: DTMF Parameters**

| Parameter | Description |
|---|---|
| DTMF Transport Type<br>`DTMF-transport-type`<br>[DTMFTransportType] | Determines the DTMF transport type.<br>▪ **[0]** Mute DTMF = DTMF digits are removed from the voice stream and are not relayed to remote side.<br>▪ **[2]** Transparent DTMF = DTMF digits remain in the voice stream.<br>▪ **[3]** RFC 2833 Relay DTMF = (Default) DTMF digits are removed from the voice stream and are relayed to remote side according to RFC 2833.<br>▪ **[7]** RFC 2833 Relay Decoder Mute = DTMF digits are sent according to RFC 2833 and muted when received.<br>**Note:** The parameter is automatically updated if the parameters FirstTxDTMFOption or RxDTMFOption are configured. |
| DTMF Volume (-31 to 0 dB)<br>`DTMF-volume`<br>[DTMFVolume] | Defines the DTMF gain control value (in decibels) to the Tel side. The valid range is -31 to 0 dB. The default is -11 dB.<br>**Note:** The parameter can also be configured in a Tel Profile. |
| DTMF Generation Twist<br>`DTMF-generation-twist`<br>[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.<br>The valid range is -10 to 10 dB. The default is 0 dB.<br>**Note:** For the parameter to take effect, a device reset is required. |
| `inter-digit-interval`<br>[DTMFInterDigitInterval] | Defines the time (in msec) between generated DTMF digits to the Tel side (if FirstTxDTMFOption = 1, 2 or 3).<br>The valid range is 0 to 32767. The default is 100. |
| [DTMFDigitLength] | Defines the time (in msec) for generating DTMF tones to the Tel side (if FirstTxDTMFOption = 1, 2 or 3). It also configures the duration that is sent in INFO (Cisco) messages.<br>The valid range is 0 to 32767. The default is 100. |
| `default-dtmf-signal-duration`<br>[RxDTMFHangOverTime] | Defines the Voice Silence time (in msec) after playing DTMF or MF digits to the Tel side that arrive as Relay from the IP side.<br>Valid range is 0 to 2,000 msec. The default is 1,000 msec. |
| `digit-hangover-time-tx`<br>[TxDTMFHangOverTime] | Defines the Voice Silence time (in msec) after detecting the end of DTMF or MF digits at the Tel side when the DTMF Transport Type is either Relay or Mute.<br>Valid range is 0 to 2,000 msec. The default is 1,000 msec. |
| NTE Max Duration<br>`telephony-events-max-duration`<br>[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. |

| Parameter | Description |
|---|---|
| | The range is -1 to 200,000,000 msec. The default is -1 (i.e., NTE stops only upon detection of an End event). |

## 45.9.4 RTP, RTCP and T.38 Parameters

The RTP, RTCP and T.38 parameters are described in the table below.

**Table 45-36: RTP/RTCP and T.38 Parameters**

| Parameter | Description |
|---|---|
| Dynamic Jitter Buffer Minimum Delay<br>`jitter-buffer-minimum-delay`<br>[DJBufMinDelay] | Global parameter that defines the minimum delay (in msec) of the device's dynamic Jitter Buffer.<br>You can also configure this functionality per specific calls, using IP Profiles (IpProfile_JitterBufMinDelay) or Tel Profiles. For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 366, or in the Tel Profile table, see Configuring Tel Profiles on page 362.<br>**Note:** If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile. |
| Dynamic Jitter Buffer Optimization Factor<br>`jitter-buffer-optimization-factor`<br>[DJBufOptFactor] | Global parameter that defines the Dynamic Jitter Buffer frame error/delay optimization factor.<br>You can also configure this functionality per specific calls, using IP Profiles or Tel Profiles. For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 366, or in the Tel Profile table, see Configuring Tel Profiles on page 362.<br>**Note:** If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile. |
| Analog Signal Transport Type<br>[AnalogSignalTransportType] | Determines the analog signal transport type.<br>▪ [0] Ignore Analog Signals = (Default) Ignore.<br>▪ [1] RFC 2833 Analog Signal Relay = Transfer hookflash using RFC 2833.<br>**Note:** The parameter is applicable only to FXS interfaces. |
| RTP Redundancy Depth<br>`RTP-redundancy-depth`<br>[RTPRedundancyDepth] | Global parameter that enables the device to generate RFC 2198 redundant packets. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_RTPRedundancyDepth). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| Enable RTP Redundancy Negotiation<br>`rtp-rdcy-nego-enbl` | Enables the device to include the RTP redundancy dynamic payload type in the SDP (according to RFC 2198).<br>▪ [0] Disable (default) |

| Parameter | Description |
|---|---|
| [EnableRTPRedundancyNegotiation] | ▪ [1] Enable = The device includes in the SDP message the RTP payload type "RED" and the payload type configured by the parameter RFC2198PayloadType.<br>a=rtpmap:<PT> RED/8000<br>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.<br>**Notes:**<br>▪ The parameter is applicable only to the Gateway application.<br>▪ For this feature to be functional, you must also set the parameter RTPRedundancyDepth to 1 (i.e., enabled).<br>▪ Currently, the negotiation of “RED” payload type is not supported and therefore, it should be configured to the same PT value for both parties. |
| RFC 2198 Payload Type<br>`RTP-redundancy-payload-type`<br>[RFC2198PayloadType] | Defines the RTP redundancy packet payload type (according to RFC 2198).<br>The valid value is 96 to 127. The default is 104.<br>**Note:** The parameter is applicable only if the RTPRedundancyDepth parameter is set to 1. |
| Packing Factor<br>[RTPPackingFactor] | N/A. Controlled internally by the device according to the selected coder. |
| RFC 2833 TX Payload Type<br>`telephony-events-payload-type-tx`<br>[RFC2833TxPayloadType] | Defines the Tx RFC 2833 DTMF relay dynamic payload type for outbound calls.<br>The valid range is 96 to 127. The default is 96.<br>**Note:** When RFC 2833 payload type negotiation is used (i.e., the parameter FirstTxDTMFOption is set to 4), this payload type is used for the received DTMF packets. If negotiation isn't used, this payload type is used for receive and for transmit. |

| Parameter | Description |
|---|---|
| RFC 2833 RX Payload Type<br>`telephony-events-payload-type-rx`<br>[RFC2833RxPayloadType] | Defines the Rx RFC 2833 DTMF relay dynamic payload type for inbound calls.<br>The valid range is 96 to 127. The default is 96.<br>**Note:** When RFC 2833 payload type negotiation is used (i.e., the parameter FirstTxDTMFOption is set to 4), this payload type is used for the received DTMF packets. If negotiation isn't used, this payload type is used for receive and for transmit. |
| [EnableDetectRemoteMACChange] | Determines whether the device changes the RTP packets according to the MAC address of received RTP packets and according to Gratuitous Address Resolution Protocol (GARP) messages.<br>▪ [0] = Nothing is changed.<br>▪ [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.<br>▪ [2] = (Default) The device uses the received GARP packets to change the MAC address of the transmitted RTP packets.<br>▪ [3] = Options 1 and 2 are used.<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ If the device is located in a network subnet which is connected to other gateways using a router that uses Virtual Router Redundancy Protocol (VRRP) for redundancy, then set the parameter to 0 or 2. |
| RTP Base UDP Port<br>[BaseUDPport] | Global parameter that defines the lower boundary of the UDP port used for RTP, RTCP (RTP port + 1) and T.38 (RTP port + 2). For more information on configuring the UDP port range, see "Configuring RTP Base UDP Port" on page 197.<br>The range of possible UDP ports is 6,000 to 65,535. The default base UDP port is 6000.<br>**Note:** For the parameter to take effect, a device reset is required. |
| `no-operation-enable`<br>[NoOpEnable] | Enables the transmission of RTP or T.38 No-Op packets.<br>▪ **[0]** = Disable (default)<br>▪ **[1]** = Enable<br>This mechanism ensures that the NAT binding remains open during RTP or T.38 silence periods. |
| [NoOpInterval] | Defines the time interval in which RTP or T.38 No-Op packets are sent in the case of silence (no RTP/T.38 traffic) when No-Op packet transmission is enabled.<br>The valid range is 20 to 65,000 msec. The default is 10,000.<br>**Note:** To enable No-Op packet transmission, use the NoOpEnable parameter. |

| Parameter | Description |
|---|---|
| `no-operation-interval`<br>[RTPNoOpPayloadType] | Defines the payload type of No-Op packets.<br>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.<br>**Note:** When defining the parameter, ensure that it doesn't cause collision with other payload types. |
| **RTP Control Protocol Extended Reports (RTCP XR) Parameters** | |
| Enable RTCP XR<br>`voice-quality-monitoring-enable`<br>[VQMonEnable] | Enables voice quality monitoring and RTCP XR, according to RFC 3611.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable Fully = Calculates voice quality metrics, uses them for QoE calculations, reports them to SEM (if configured), and sends them to remote side using RTCP XR.<br>▪ **[2]** Enable Calculation Only = Calculates voice quality metrics, uses them for QoE calculations, reports them to SEM (if configured), but does not send them to remote side using RTCP XR.<br>**Note:** For the parameter to take effect, a device reset is required. |
| Minimum Gap Size<br>[VQMonGMin] | Defines the voice quality monitoring - minimum gap size (number of frames).<br>The default is 16. |
| Burst Threshold<br>[VQMonBurstHR] | Defines the voice quality monitoring - excessive burst alert threshold.<br>The default is -1 (i.e., no alerts are issued). |
| Delay Threshold<br>[VQMonDelayTHR] | Defines the voice quality monitoring - excessive delay alert threshold.<br>The default is -1 (i.e., no alerts are issued). |
| R-Value Delay Threshold<br>[VQMonEOCRValTHR] | Defines the voice quality monitoring - end of call low quality alert threshold.<br>The default is -1 (i.e., no alerts are issued). |
| RTCP XR Packet Interval<br>`rtcp-interval`<br>[RTCPInterval] | Defines the time interval (in msec) between adjacent RTCP XR reports. This interval starts from call establishment. Thus, the device can send RTCP XR reports during the call, in addition to at the end of the call. If the duration of the call is shorter than this interval, RTCP XR is sent only at the end of the call.<br>The valid value range is 0 to 65,535. The default is 5,000. |
| Disable RTCP XR Interval Randomization<br>`disable-RTCP-randomization`<br>[DisableRTCPRandomize] | Determines whether RTCP report intervals are randomized or whether each report interval accords exactly to the parameter RTCPInterval.<br>▪ **[0]** Disable = (Default) Randomize<br>▪ **[1]** Enable = No Randomize |

| Parameter | Description |
|---|---|
| Gateway RTCP XR Report Mode<br>`rtcp-xr-rep-mode`<br>[RTCPXRReportMode] | Determines whether RTCP XR reports are sent and defines the interval at which they are sent.<br>▪ [0] Disable = (Default) RTCP XR reports are not sent.<br>▪ [1] End Call = RTCP XR reports are sent at the end of each call.<br>▪ [2] End Call & Periodic = RTCP XR reports are sent at the end of each call and periodically according to the RTCPInterval parameter.<br>**Note:** The parameter is applicable only to the Gateway application. |
| publication-ip-group-id<br>[PublicationIPGroupID] | Defines the IP Group to where the RTCP XR is sent. |

# 45.10 Gateway Application Parameters

## 45.10.1 Fax and Modem Parameters

The fax and modem parameters are described in the table below.

**Table 45-37: Fax and Modem Parameters**

| Parameter | Description |
|---|---|
| Fax Transport Mode<br>`fax-transport-mode`<br>[FaxTransportMode] | Determines the fax transport mode used by the device.<br>▪ **[0]** Disable = transparent mode<br>▪ **[1]** T.38 Relay (default)<br>▪ **[2]** Bypass<br>▪ **[3]** Events Only<br>**Note:** The 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). |
| `V34-fax-transport-type`<br>[V34FaxTransportType] | Determines the V.34 fax transport method (whether V34 fax falls back to T.30 or pass over Bypass).<br>▪ [0] = Transparent<br>▪ [1] = (Default) Relay<br>▪ [2] = Bypass<br>▪ [3] = Transparent with Events<br>**Note:** To configure V34FaxTransportType to 1 (i.e., fax relay), you also need to configure FaxTransportMode to 1 (fax relay). |
| V.21 Modem Transport Type<br>`V21-modem-transport-type`<br>[V21ModemTransportType] | Determines the V.21 modem transport type.<br>▪ **[0]** Disable = (Default) Transparent.<br>▪ **[2]** Enable Bypass<br>▪ **[3]** Events Only = Transparent with Events.<br>**Note:** You can also configure this functionality per specific calls, using IP Profiles (IpProfile_VxxTransportType). For more information, see "Configuring IP Profiles" on page 366. |
| V.22 Modem Transport Type<br>`V22-modem-transport-type`<br>[V22ModemTransportType] | Determines the V.22 modem transport type.<br>▪ **[0]** Disable = Transparent.<br>▪ **[2]** Enable Bypass (default)<br>▪ **[3]** Events Only = Transparent with Events.<br>**Note:** You can also configure this functionality per specific calls, using IP Profiles (IpProfile_VxxTransportType). For more information, see "Configuring IP Profiles" on page 366. |
| V.23 Modem Transport Type | Determines the V.23 modem transport type. |

| Parameter | Description |
|---|---|
| `V23-modem-transport-type`<br>[V23ModemTransportType] | ▪ **[0]** Disable = Transparent.<br>▪ **[2]** Enable Bypass (default)<br>▪ **[3]** Events Only = Transparent with Events.<br>**Note:** You can also configure this functionality per specific calls, using IP Profiles (IpProfile_VxxTransportType). For more information, see "Configuring IP Profiles" on page 366. |
| V.32 Modem Transport Type<br>`V32-modem-transport-type`<br>[V32ModemTransportType] | Determines the V.32 modem transport type.<br>▪ **[0]** Disable = Transparent.<br>▪ **[2]** Enable Bypass (default)<br>▪ **[3]** Events Only = Transparent with Events.<br>**Notes:**<br>▪ The parameter applies only to V.32 and V.32bis modems.<br>▪ You can also configure this functionality per specific calls, using IP Profiles (IpProfile_VxxTransportType). For more information, see "Configuring IP Profiles" on page 366. |
| V.34 Modem Transport Type<br>`V34-modem-transport-type`<br>[V34ModemTransportType] | Determines the V.90/V.34 modem transport type.<br>▪ **[0]** Disable = Transparent.<br>▪ **[2]** Enable Bypass (default)<br>▪ **[3]** Events Only = Transparent with Events.<br>**Note:** You can also configure this functionality per specific calls, using IP Profiles (IpProfile_VxxTransportType). For more information, see "Configuring IP Profiles" on page 366. |
| `bell-modem-transport-type`<br>[BellModemTransportType] | Determines the Bell modem transport method.<br>▪ **[0]** = Transparent (default)<br>▪ **[2]** = Bypass<br>▪ **[3]** = Transparent with events |
| Fax CNG Mode<br>`fax_cng_mode`<br>[FaxCNGMode] | Determines the device's handling of fax relay upon detection of a fax CNG tone or a V.34/Super G3 V8-CM (Call Menu) signal from originating faxes.<br>▪ **[0]** Doesn't send T.38 Re-INVITE = (Default) SIP re-INVITE is not sent.<br>▪ **[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.<br>▪ [2] Sends on CNG or v8-cn = Sends a SIP re-INVITE with T.38 parameters in SDP to the terminating fax upon detection of a fax CNG tone (if the CNGDetectorMode |

| Parameter | Description |
|---|---|
| | parameter is set to 1) or upon detection of a V8-CM signal.<br>**Notes:**<br>▪<br>▪ If the parameter is set to [2] and the CNGDetectorMode parameter is set to [0], the device sends a re-INVITE only if it detects a V8-CM signal from the originating fax.<br>▪ This feature is applicable only if the IsFaxUsed parameter is set to [1] or [3].<br>▪ The device also sends T.38 re-INVITE if the CNGDetectorMode parameter is set to [2], regardless of the FaxCNGMode parameter settings. |
| CNG Detector Mode<br>`coder`<br>[CNGDetectorMode] | Global parameter that enables the detection of the fax calling tone (CNG) and defines the detection method. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_CNGmode). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| Fax Detect Timeout Since Connect<br>`configure voip > sip-definition general-settings > fax-detect-timeout-since-connect`<br>[FaxDetectTimeoutSinceConnect] | Defines a timeout (in msec) for detecting fax from the Tel side during an established voice call. The interval starts from when the voice call is established. If the device detects a fax tone within the interval, it ends the voice session and sends a T.38 or VBD re-INVITE message to the IP side and processes the fax. If the interval expires without any received fax event, the device ignores all subsequent fax events during the voice session.<br>The valid value is 0 to 120000. The default is 0. If set to 0, the device can detect fax during the entire voice call. |
| SIP T.38 Version<br>`sip-t38-ver`<br>[SIPT38Version] | Determines the T.38 fax relay version.<br>▪ [-1] Not Configured = (Default) No T.38<br>▪ [0] Version 0<br>▪ [3] Version 3 = T.38 Version 3 (V.34 over T.38)<br>**Note:** For a description on V.34 over T.38 fax relay, see V.34 Fax Support on page 187. |
| Fax Relay Enhanced Redundancy Depth<br>`enhanced-redundancy-depth`<br>[FaxRelayEnhancedRedundancyDepth] | Defines the number of times that control packets are retransmitted when using the T.38 standard. |

| Parameter | Description |
|---|---|
| | The valid range is 0 to 4. The default is 2. |
| Fax Relay Redundancy Depth<br>`redundancy-depth`<br>[FaxRelayRedundancyDepth] | Defines the number of times that each fax relay payload is retransmitted to the network.<br>▪ **[0]** = (Default) No redundancy<br>▪ **[1]** = One packet redundancy<br>▪ **[2]** = Two packet redundancy<br>**Note:** The parameter is applicable only to non-V.21 packets. |
| Fax Relay Max Rate (bps)<br>`max-rate`<br>[FaxRelayMaxRate] | Defines the maximum rate (in bps) at which fax relay messages are transmitted (outgoing calls).<br>▪ **[0]** 2400 = 2.4 kbps<br>▪ **[1]** 4800 = 4.8 kbps<br>▪ **[2]** 7200 = 7.2 kbps<br>▪ **[3]** 9600 = 9.6 kbps<br>▪ **[4]** 12000 = 12.0 kbps<br>▪ **[5]** 14400 = 14.4 kbps (default)<br>▪ [6] 16800bps = 16.8 kbps<br>▪ [7] 19200bps = 19.2 kbps<br>▪ [8] 21600bps = 21.6 kbps<br>▪ [9] 24000bps = 24 kbps<br>▪ [10] 26400bps = 26.4 kbps<br>▪ [11] 28800bps = 28.8 kbps<br>▪ [12] 31200bps = 31.2 kbps<br>▪ [13] 33600bps = 33.6 kbps<br>**Notes:**<br>▪ 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.<br>▪ Fax relay rates greater than 14.4 kbps are applicable only to V.34 / T.38 fax relay. For non-T.38 V.34 supporting devices, configuration greater than 14.4 kbps is truncated to 14.4 kbps. |
| Fax Relay ECM Enable<br>`ecm-mode`<br>[FaxRelayECMEnable] | Enables Error Correction Mode (ECM) mode during fax relay.<br>▪ **[0]** Disable<br>▪ **[1]** Enable (default) |
| Fax/Modem Bypass Coder Type<br>[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.<br>▪ **[0]** G.711Alaw= (Default) G.711 A-law 64<br>▪ **[1]** G.711Mulaw = G.711 μ-law |

| Parameter | Description |
|---|---|
| Fax/Modem Bypass Packing Factor<br>`packing-factor`<br>[FaxModemBypassM] | Defines the number (20 msec) of coder payloads used to generate a fax/modem bypass packet.<br>The valid range is 1, 2, or 3 coder payloads. The default is 1 coder payload. |
| `fax-modem-telephony-events-mode`<br>[FaxModemNTEMode] | Determines whether the device sends RFC 2833 ANS/ANSam events upon detection of fax and/or modem Answer tones (i.e., CED tone).<br>▪ **[0]** = Disabled (default)<br>▪ **[1]** = Enabled<br>**Note:** The parameter is applicable only when the fax or modem transport type is set to bypass or Transparent-with-Events. |
| Fax Bypass Payload Type<br>`fax-bypass-payload-type`<br>[FaxBypassPayloadType] | Defines the fax bypass RTP dynamic payload type.<br>The valid range is 0 to 127. The default is 102. |
| `modem-bypass-payload-type`<br>[ModemBypassPayloadType] | Defines the modem bypass dynamic payload type.<br>The range is 0 to 127. The default is 103. |
| `volume`<br>[FaxModemRelayVolume] | Defines the fax gain control.<br>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. |
| Fax Bypass Output Gain<br>`fax-bypass-output-gain`<br>[FaxBypassOutputGain] | Defines the fax bypass output gain control.<br>The range is -31 to +31 dB, in 1-dB steps. The default is 0 (i.e., no gain). |
| Modem Bypass Output Gain<br>[ModemBypassOutputGain] | Defines the modem bypass output gain control.<br>The range is -31 dB to +31 dB, in 1-dB steps. The default is 0 (i.e., no gain). |
| `modem-bypass-output-gain`<br>[FaxModemBypassBasicRTPPacketInterval] | Defines the basic frame size used during fax/modem bypass sessions.<br>▪ **[0]** = (Default) Determined internally<br>▪ **[1]** = 5 msec (not recommended)<br>▪ **[2]** = 10 msec<br>▪ **[3]** = 20 msec<br>**Note:** When set to 5 msec (1), the maximum number of simultaneous channels supported is 120. |
| `jitter-buffer-minimum-delay`<br>[FaxModemBypasDJBufMinDelay] | Defines the Jitter Buffer delay (in milliseconds) during fax and modem bypass session.<br>The range is 0 to 150 msec. The default is 40. |
| `enable-fax-modem-inband-network-detection`<br>[EnableFaxModemInbandNetworkDetection] | Enables in-band network detection related to fax/modem.<br>▪ **[0]** = (Default) Disable. |

| Parameter | Description |
|---|---|
| | ▪ **[1]** = Enable. When the 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. |
| `NSE-mode`<br>[NSEMode] | Global parameter that enables Cisco's compatible fax and modem bypass mode, Named Signaling Event (NSE) packets. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_NSEMode). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see ''Configuring IP Profiles'' on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| `NSE-payload-type`<br>[NSEPayloadType] | Defines the NSE payload type for Cisco Bypass compatible mode.<br>The valid range is 96-127. The default is 105.<br>**Notes:**<br>▪ The parameter is applicable only to the Gateway application.<br>▪ Cisco gateways usually use NSE payload type of 100. |
| [T38UseRTPPort] | Defines the port (with relation to RTP port) for sending and receiving T.38 packets.<br>▪ **[0]** = (Default) Use the RTP port +2 to send/receive T.38 packets.<br>▪ **[1]** = Use the same port as the RTP port to send/receive T.38 packets.<br>**Notes:**<br>▪ For the parameter to take effect, you must reset the device.<br>▪ 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. |
| T.38 Max Datagram Size<br>`t38-mx-datagram-sz`<br>[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. |

| Parameter | Description |
|---|---|
| | The valid range is 120 to 600. The default is 560. |
| T38 Fax Max Buffer<br>`t38-fax-mx-buff`<br>[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.<br>The valid range is 500 to 3000. The default is 3000. |
| Detect Fax on Answer Tone<br>`det-fax-on-ans-tone`<br>[DetFaxOnAnswerTone] | Determines when the device initiates a T.38 session for fax transmission.<br>▪ **[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.<br>▪ **[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.<br>**Note:** The parameters is applicable only if the IsFaxUsed parameter is set to 1 (T.38 Relay) or 3 (Fax Fallback). |
| CED Transfer Mode<br>[CEDTransferMode] | Defines the method for sending fax/modem CED (answering) tones.<br>▪ [0] Fax Relay or VBD = (Default) The device transfers the CED tone in Relay mode and starts the fax session immediately.<br>▪ [1] Voice Mode or VBD = The device transfers the CED tone in either Voice or Bypass mode and starts the fax session on V21 preamble.<br>▪ [2] RFC 4733 Blocking RTP VBD = The device transfers the CED tone in RFC 2833. This is applicable only to V.150.1 modem relay and fax bypass.<br>▪ [3] RFC 4733 Along with RTP VBD = The device transfers the CED tone in RFC 2833 and bypass, in parallel. For combined V.150.1 modem relay and fax relay, use this option.<br>**Note:** The parameter is applicable only to the Gateway application. |
| T.38 Fax Session<br>`t38-sess-imm-strt`<br>[T38FaxSessionImmediateStart] | Enables fax transmission of T.38 "no-signal" packets to the terminating fax machine.<br>▪ **[0]** Disable (default) |

| Parameter | Description |
|---|---|
| | ▪ **[1]** Immediate Start on Fax = Device activates T.38 fax relay upon receipt of a re-INVITE with T.38 only in the SDP.<br>▪ **[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.<br>The 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.<br>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.<br>**Note:** To enable No-Op packet transmission, use the NoOpEnable and NoOpInterval parameters. |
| V.150.1 Modem over IP<br>**Note:** These parameters are applicable only to the Gateway application. | |
| Profile Number<br>[V1501AllocationProfile] | Defines the V.150.1 profile, which determines how many DSP channels support V.150.1.<br>The value range is 0 to 20. The default is 0.<br>**Note:** For the parameter to take effect, a device reset is required. |
| SSE Payload Type Rx<br>`V1501-SSE-payload-type-rx`<br>[V1501SSEPayloadTypeRx] | Defines the V.150.1 (modem relay protocol) State Signaling Event (SSE) payload type Rx.<br>The value range is 96 to 127. The default is 105. |
| SSE Redundancy Depth<br>`SSE-redundancy-depth`<br>[V1501SSERedundancyDepth] | Defines the SSE redundancy depth.<br>The value range is 1-6. The default is 3. |
| SPRT Transport Ch.0 Max Payload Size<br>`SPRT-transport-channel0-max-payload-size`<br>[V1501SPRTTransportChannel0MaxPayloadSize] | Defines the maximum payload size for V.150.1 SPRT Transport Channel 0.<br>The range is 140 to 256. The default is 140. |
| SPRT Transport Ch.2 Max Payload Size<br>`SPRT-transport-channel2-max-payload-size`<br>[V1501SPRTTransportChannel2MaxPayloadSize] | Defines the maximum payload size for V.150.1 SPRT Transport Channel 2.<br>The range is 132 to 256. The default is 132. |

| Parameter | Description |
|---|---|
| SPRT Transport Ch.2 Max Window Size<br>`SPRT-transport-channel2-max-window-size`<br>[V1501SPRTTransportChannel2MaxWindowSize] | Defines the maximum window size of SPRT transport channel 2.<br>The value range is 8 to 32. The default is 8. |
| SPRT Transport Ch.3 Max Payload Size<br>`SPRT-transport-channel3-max-payload-size`<br>[V1501SPRTTransportChannel3MaxPayloadSize] | Defines the maximum payload size for V.150.1 SPRT Transport Channel 3.<br>The range is 140 to 256. The default is 140. |

## 45.10.2 DTMF and Hook-Flash Parameters

The DTMF and hook-flash parameters are described in the table below.

**Table 45-38: DTMF and Hook-Flash Parameters**

| Parameter | Description |
|---|---|
| **Hook-Flash Parameters** | |
| Hook-Flash Code<br>`hook-flash-code`<br>[HookFlashCode] | Defines the digit pattern that when received from the Tel side, indicates a Hook Flash event.<br>The valid range is a 25-character string. The default is a null string.<br>**Note:** The parameter can also be configured in a Tel Profile. |
| Hook-Flash Option<br>`hook-flash-option`<br>[HookFlashOption] | Defines the hook-flash transport type (i.e., method by which hook-flash is sent and received).<br>▪ **[0]** Not Supported = (Default) Hook-Flash indication is not sent.<br>▪ **[1]** INFO = Sends proprietary INFO message (Broadsoft) with Hook-Flash indication. The device sends the INFO message as follows:<br>Content-Type: application/broadsoft; version=1.0<br>Content-Length: 17<br>event flashhook<br>▪ **[4]** RFC 2833 = This option is currently not supported.<br>▪ **[5]** INFO (Lucent) = Sends proprietary SIP INFO message with Hook-Flash indication. The device sends the INFO message as follows:<br>Content-Type: application/hook-flash<br>Content-Length: 11<br>signal=hf<br>▪ **[6]** INFO (NetCentrex) = Sends proprietary SIP INFO message with Hook-Flash indication. The device sends the INFO message as follows:<br>`Content-Type: application/dtmf-relay`<br>`Signal=16`<br>Where 16 is the DTMF code for hook flash.<br>▪ **[7]** INFO (HUAWEI) = Sends a SIP INFO message with Hook-Flash indication. The device sends the INFO message as follows:<br>`Content-Length: 17`<br>`Content-Type: application/sscc`<br>`event=flashhook`<br>**Notes:**<br>▪ FXS interfaces send Hook-Flash signals only if the EnableHold parameter is set to 0. |
| Min. Flash-Hook Detection Period<br>`min-flash-hook-time`<br>[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.<br>The valid range is 25 to 300. The default is 300. |

| Parameter | Description |
|---|---|
| | **Notes:**<br>▪ The parameter is applicable only to FXS interfaces.<br>▪ 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). |
| Max. Flash-Hook Detection Period<br>`flash-hook-period`<br>[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.<br>For the analog side, it defines the following:<br>▪ FXS interfaces:<br>✓ Maximum hook-flash detection period. A longer signal is considered an off-hook or on-hook event.<br>✓ Hook-flash generation period upon detection of a SIP INFO message containing a hook-flash signal.<br>The valid range is 25 to 3,000. The default is 700.<br>**Note:** The parameter can also be configured in a Tel Profile. |
| **DTMF Parameters** | |
| `notify-on-sig-end`<br>[MGCPDTMFDetectionPoint] | Determines when the detection of DTMF events is notified.<br>▪ **[0]** = DTMF event is reported at the end of a detected DTMF digit.<br>▪ **[1]** = (Default) DTMF event is reported at the start of a detected DTMF digit. |
| Declare RFC 2833 in SDP<br>`rfc-2833-in-sdp`<br>[RxDTMFOption] | Global parameter that enables the device to declare the RFC 2833 'telephony-event' parameter in the SDP. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_RxDTMFOption). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| First Tx DTMF Option<br>`configure voip > gw dtmf-and-suppl dtmf-and-dialing > first-dtmf-option-type`<br>[FirstTxDTMFOption] | Defines the first preferred transmit (Tx) DTMF negotiation method.<br>▪ **[0]** Not Supported = (Default) No negotiation - DTMF digits are sent according to the parameters DTMFTransportType and RFC2833PayloadType. The RFC 2833 payload type is according to the RFC2833PayloadType parameter for transmit and receive.<br>▪ **[1]** Info NORTEL = Sends DTMF digits according to IETF Internet-Draft draft-choudhuri-sip-info-digit-00.<br>▪ **[2]** NOTIFY = Sends DTMF digits according to IETF Internet-Draft draft-mahy-sipping-signaled-digits-01.<br>▪ **[3]** Info Cisco = Sends DTMF digits according to Cisco format.<br>▪ **[4]** RFC 2833 = The device handles DTMF as follows: |

| Parameter | Description |
|---|---|
| | ✓ Negotiates RFC 2833 payload type using local and remote SDPs.<br>✓ Sends DTMF packets using RFC 2833 payload type according to the payload type in the received SDP.<br>✓ Expects to receive RFC 2833 packets with the same payload type according to the RFC2833PayloadType parameter.<br>✓ Removes DTMF digits in transparent mode (as part of the voice stream).<br>▪ **[5]** Info KOREA = Sends DTMF digits according to Korea Telecom format.<br>**Notes:**<br>▪ When out-of-band DTMF transfer is used ([1], [2], [3], or [5]), the DTMFTransportType parameter is automatically set to [0] (DTMF digits are erased from the RTP stream).<br>▪ For more information on DTMF transport, see "Configuring DTMF Transport Types" on page 194.<br>▪ You can also configure the parameter per specific calls, using IP Profiles (IpProfile_FirstTxDtmfOption). For configuring IP Profiles, see "Configuring IP Profiles" on page 366. |
| Second Tx DTMF Option<br>`configure voip > gw dtmf-and-suppl dtmf-and-dialing > second-dtmf-option-type`<br>[SecondTxDTMFOption] | Defines the second preferred transmit (Tx) DTMF negotiation method. The first preferred method is configured by the FirstTxDTMFOption parameter. For a description of the optional values for the parameter, see the FirstTxDTMFOption parameter above.<br>**Note:** You can also configure the parameter per specific calls, using IP Profiles (IpProfile_SecondTxDtmfOption). For configuring IP Profiles, see "Configuring IP Profiles" on page 366. |
| [DisableAutoDTMFMute] | Enables the automatic muting of DTMF digits when out-of-band DTMF transmission is used.<br>▪ **[0]** = (Default) Automatic mute is used.<br>▪ **[1]** = No automatic mute of in-band DTMF.<br>When the 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 (FirstTxDTMFOption 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.<br>**Note:** Usually this mode is not recommended. |
| Enable Digit Delivery to IP<br>`digit-delivery-2ip`<br>[EnableDigitDelivery2IP] | Enables the Digit Delivery feature whereby DTMF digits are sent to the destination IP address after the Tel-to-IP call is answered.<br>▪ **[0]** Disable (default).<br>▪ **[1]** Enable = Enable digit delivery to IP.<br>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 |

| Parameter | Description |
|---|---|
| | (number of 'p' multiplied by 1.5 seconds), and then sends the rest of the DTMF digits using the method chosen (in-band or out-of-band).<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ The called number can include several 'p' characters (1.5 seconds pause), for example, 1001pp699, 8888p9p300. |
| Enable Digit Delivery to Tel<br>`digit-delivery-2tel`<br>[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 for IP-to-Tel calls.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>Additional examples: 1664wpp102, 66644ppp503, and 7774w100pp200.<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ For analog interfaces: The called number can include characters 'p' (1.5 seconds pause) and 'd' (detection of dial tone). If character 'd' is used, it must be the first 'digit' in the called number. The character 'p' can be used several times.<br>For example (for FXS interfaces), the called number can be as follows: d1005, dpp699, p9p300. To add the 'd' and 'p' digits, use the usual number manipulation rules.<br>▪ If the parameter is enabled, it is possible to configure the FXS 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.<br>▪ For analog interfaces: The FXS interface send SIP 200 OK responses only after the DTMF dialing is complete.<br>▪ The parameter can also be configured in a Tel Profile. |
| `replace-nb-sign-w-esc`<br>[ReplaceNumberSignWithEscapeChar] | Determines whether to replace the number sign (#) with the escape character (%23) in outgoing SIP messages for Tel-to-IP calls.<br>▪ [0] Disable (default).<br>▪ [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.<br>**Notes:**<br>▪ The parameter is applicable only if the parameter IsSpecialDigits is set 1.<br>▪ The parameter is applicable only to analog interfaces. |

| Parameter | Description |
|---|---|
| Special Digit Representation<br>`special-digit-rep`<br>[UseDigitForSpecialDTMF] | Defines the representation for 'special' digits ('*' and '#') that are used for out-of-band DTMF signaling (using SIP INFO/NOTIFY).<br>▪ **[0]** Special = (Default) Uses the strings '*' and '#'.<br>▪ **[1]** Numeric = Uses the numerical values 10 and 11. |

## 45.10.3 Digit Collection and Dial Plan Parameters

The digit collection and dial plan parameters are described in the table below.

**Table 45-39: Digit Collection and Dial Plan Parameters**

| Parameter | Description |
|---|---|
| Dial Plan Index<br>`dial-plan-index`<br>[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.<br>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.<br>**Notes:**<br>▪ If the parameter is configured to select a Dial Plan index, the settings of the parameter DigitMapping are ignored.<br>▪ If the 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.<br>▪ The parameter can also be configured in a Tel Profile.<br>▪ For more information on the Dial Plan file, see "Dialing Plans for Digit Collection" on page 501. |
| `tel2ip-src-nb-map-dial-index`<br>[Tel2IPSourceNumberMapping DialPlanIndex] | Defines the Dial Plan index in the external Dial Plan file for the Tel-to-IP Source Number Mapping feature.<br>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).<br>For more information on this feature, see "Modifying ISDN-to-IP Calling Party Number using Dial Plan File" on page 504. |
| Digit Mapping Rules<br>`default-dm`<br>[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.<br>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:<br>▪ **[n-m]:** Range of numbers (not letters).<br>▪ **.** (single dot): Repeat digits until next notation (e.g., T).<br>▪ **x:** Any single digit. |

| Parameter | Description |
|---|---|
| | ▪ **T:** Dial timeout (configured by the TimeBetweenDigits parameter).<br>▪ **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.<br>An example of a digit map is shown below:<br>11xS\|00T\|[1-7]xxx\|8xxxxxxx\|#xxxxxxx\|*xx\|91xxxxxxxxxx\|9011x.T<br>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.').<br>**Notes:**<br>▪ 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.<br>▪ For more information on digit mapping, see "Digit Mapping" on page 440. |
| Max Digits in Phone Num<br>`mxdig-b4-dialing`<br>[MaxDigits] | Defines the maximum number of collected destination number digits that can be received (i.e., dialed) from the Tel side (analog). When the number of collected digits reaches this maximum, the device uses these digits for the called destination number.<br>The valid range is 1 to 49. The default is 5 for analog.<br>**Notes:**<br>▪ Instead of using the parameter, Digit Mapping rules can be configured.<br>▪ For FXS interfaces: Dialing ends when any of the following scenarios occur:<br>✓ Maximum number of digits is dialed<br>✓ Interdigit Timeout (TimeBetweenDigits) expires<br>✓ Pound (#) key is pressed<br>✓ Digit map pattern is matched |
| Inter Digit Timeout for Overlap Dialing<br>`time-btwn-dial-digs`<br>[TimeBetweenDigits] | Analog: Defines the time (in seconds) that the device waits between digits that are dialed by the user.<br>When this inter-digit timeout expires, the device uses the collected digits to dial the called destination number.<br>The valid range is 1 to 10. The default is 4. |
| Enable Special Digits<br>`special-digits`<br>[IsSpecialDigits] | Determines whether the asterisk (*) and pound (#) digits can be used in DTMF.<br>▪ [0] Disable = Use '*' or '#' to terminate number collection (refer to the parameter UseDigitForSpecialDTMF). (Default.)<br>▪ [1] Enable = Allows '*' and '#' for telephone numbers dialed by a user or for the endpoint telephone number.<br>**Notes:** |

| Parameter | Description |
|---|---|
| | ▪ The symbols can always be used as the first digit of a dialed number even if you disable the parameter.<br>▪ The parameter is applicable only to FXS interfaces. |

### 45.10.4 Voice Mail Parameters

The voice mail parameters are described in the table below.

**Table 45-40: Voice Mail Parameters**

| Parameter | Description |
|---|---|
| Enable VoiceMail URI<br>`voicemail-uri`<br>[EnableVMURI] | Enables the interworking of target and cause for redirection from Tel to IP and vice versa, according to RFC 4468.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable |
| SMDI Parameters | |
| Enable SMDI<br>[SMDI] | Enables Simplified Message Desk Interface (SMDI) interface on the device.<br>▪ [0] Disable = (Default) Normal serial<br>▪ [1] Enable (Bellcore)<br>▪ [2] Ericsson MD-110<br>▪ [3] NEC (ICS)<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ 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). |
| SMDI Timeout<br>[SMDITimeOut] | Defines the time (in msec) that the device waits for an SMDI Call Status message before or after a Setup message is received. The parameter synchronizes the SMDI and analog CAS interfaces.<br>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.<br>The valid range is 0 to 10000 (i.e., 10 seconds). The default is 2000. |
| **Message Waiting Indication (MWI) Parameters** | |
| MWI Off Digit Pattern<br>`mwi-off-dig-ptrn`<br>[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.<br>The valid range is a 25-character string. |
| MWI On Digit Pattern<br>`mwi-on-dig-ptrn`<br>[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.<br>The valid range is a 25-character string. |

| Parameter | Description |
|---|---|
| MWI Suffix Pattern<br>`mwi-suffix-pattern`<br>[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.<br>The valid range is a 25-character string. |
| MWI Source Number<br>`mwi-source-number`<br>[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. |
| `mwi-subs-ipgrpid`<br>[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.<br>For example, if the 'SIP Group Name' field of the IP Group is set to "company.com", the device sends the following SUBSCRIBE message:<br>`SUBSCRIBE sip:company.com...`<br>Instead of:<br>`SUBSCRIBE sip:10.33.10.10...`<br>**Note:** If the parameter is not configured, the MWI SUBSCRIBE message is sent to the MWI server as defined by the MWIServerIP parameter. |
| [NotificationIPGroupID] | Defines the IP Group ID to which the device sends SIP NOTIFY MWI messages.<br>**Notes:**<br>▪ This is used for MWI Interrogation. For more information on the interworking of QSIG MWI to IP, see Message Waiting Indication on page 453.<br>▪ To determine the handling method of MWI Interrogation messages, use the TrunkGroupSettings_MWIInterrogationType, parameter (in the Hunt Group Settings table). |
| [MWIQsigMsgCentreldIDPartyNumber] | Defines the Message Centred ID party number used for QSIG MWI messages. If not configured (default), the parameter is not included in MWI (activate and deactivate) QSIG messages.<br>The valid value is a string. |
| **Digit Patterns** The following digit pattern parameters apply only to voice mail applications that use the DTMF communication method. For the available pattern syntaxes, refer to the *CPE Configuration Guide for Voice Mail*. | |
| Forward on Busy Digit Pattern (Internal)<br>`fwd-bsy-dig-ptrn-int`<br>[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.<br>The valid range is a 120-character string. |

| Parameter | Description |
|---|---|
| Forward on No Answer Digit Pattern (Internal)<br>`fwd-no-ans-dig-pat-int`<br>[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.<br>The valid range is a 120-character string. |
| Forward on Do Not Disturb Digit Pattern (Internal)<br>`fwd-dnd-dig-ptrn-int`<br>[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.<br>The valid range is a 120-character string. |
| Forward on No Reason Digit Pattern (Internal)<br>`fwd-no-rsn-dig-ptrn-int`<br>[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.<br>The valid range is a 120-character string. |
| Forward on Busy Digit Pattern (External)<br>`fwd-bsy-dig-ptrn-ext`<br>[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).<br>The valid range is a 120-character string. |
| Forward on No Answer Digit Pattern (External)<br>`fwd-no-ans-dig-pat-ext`<br>[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).<br>The valid range is a 120-character string. |
| Forward on Do Not Disturb Digit Pattern (External)<br>`fwd-dnd-dig-ptrn-ext`<br>[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).<br>The valid range is a 120-character string. |
| Forward on No Reason Digit Pattern (External)<br>`fwd-no-rsn-dig-ptrn-ext`<br>[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).<br>The valid range is a 120-character string. |
| Internal Call Digit Pattern<br>`int-call-dig-ptrn`<br>[DigitPatternInternalCall] | Defines the digit pattern used by the PBX to indicate an internal call.<br>The valid range is a 120-character string. |
| External Call Digit Pattern<br>`ext-call-dig-ptrn`<br>[DigitPatternExternalCall] | Defines the digit pattern used by the PBX to indicate an external call.<br>The valid range is a 120-character string. |
| Disconnect Call Digit Pattern<br>`disc-call-dig-ptrn`<br>[TelDisconnectCode] | Defines a digit pattern that when received from the Tel side, indicates the device to disconnect the call.<br>The valid range is a 25-character string. |
| Digit To Ignore Digit Pattern<br>`dig-to-ignore-dig-pattern`<br>[DigitPatternDigitToIgnore] | Defines a digit pattern that if received as Src (S) or Redirect (R) numbers is ignored and not added to that number.<br>The valid range is a 25-character string. |

## 45.10.5 Supplementary Services Parameters

This subsection describes the device's supplementary telephony services parameters.

### 45.10.5.1 Caller ID Parameters

The caller ID parameters are described in the table below.

**Table 45-41: Caller ID Parameters**

| Parameter | Description |
|---|---|
| Caller ID Permissions Table | |
| Caller ID Permissions Table<br>`configure voip > gw analoggw enable-caller-id`<br>[EnableCallerID] | This table parameter enables (per port) Caller ID generation (for FXS interfaces).<br>The format of the ini file table parameter is as follows:<br>[EnableCallerID]<br>FORMAT EnableCallerID_Index = EnableCallerID_IsEnabled, EnableCallerID_Module, EnableCallerID_Port;<br>[\EnableCallerID]<br>Where,<br>▪ Module = Module number, where 1 denotes the module in Slot 1.<br>▪ Port = Port number, where 1 denotes Port 1 of a module.<br>For example:<br>EnableCallerID 0 = 1,3,1; (caller ID enabled on Port 1 of Module 3)<br>EnableCallerID 1 = 0,3,2; (caller ID disabled on Port 2 of Module 3)<br>For a detailed description of the table, see Configuring Caller ID Permissions on page 479.<br>**Notes:**<br>▪ The indexing of the parameter starts at 0.<br>▪ The parameter is applicable only to FXS interfaces. |
| Caller Display Information Table | |
| Caller Display Information Table<br>`configure voip > gw analoggw caller-display-info`<br>[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.<br>The format of the ini file table parameter is as follows:<br>[CallerDisplayInfo]<br>FORMAT CallerDisplayInfo_Index = CallerDisplayInfo_DisplayString, CallerDisplayInfo_IsCidRestricted, CallerDisplayInfo_Module, CallerDisplayInfo_Port;<br>[\CallerDisplayInfo]<br>Where,<br>▪ Module = Module number, where 1 denotes the module in Slot 1.<br>▪ Port = Port number, where 1 denotes Port 1 of a module.<br>For example:<br>CallerDisplayInfo 0 = Susan C.,0,1,1; ("Susan C." is sent as the Caller ID for Port 1 of Module 1) |

| Parameter | Description |
|---|---|
| | CallerDisplayInfo 1 = Mark M.,0,1,2; ("Mark M." is sent as Caller ID for Port 2 of Module 1)<br>For a detailed description of the table, see Configuring Caller Display Information on page 476.<br>**Notes:**<br>▪ The indexing of this table ini file parameter starts at 0.<br>▪ The parameter is applicable only to FXS interfaces. |
| Enable Caller ID<br>`enable-caller-id`<br>[EnableCallerID] | Global parameter that enables Caller ID.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable =<br>✓ FXS: The calling number and display text (from IP) are sent to the device's port.<br>To configure the Caller ID string per port, see Configuring Caller Display Information on page 476. To enable or disable caller ID generation / detection per port, see Configuring Caller ID Permissions on page 479. |
| Caller ID Type<br>`caller-ID-type`<br>[CallerIDType] | Determines the standard used for generation (FXS) of Caller ID, and generation (FXS) of MWI (when specified) signals:<br>▪ [0] Standard Bellcore = (Default) Caller ID and MWI<br>▪ [1] Standard ETSI = Caller ID and MWI<br>▪ [2] Standard NTT<br>▪ [4] Standard BT = Britain<br>▪ [16] Standard DTMF Based ETSI<br>▪ [17] Standard Denmark = Caller ID and MWI<br>▪ [18] Standard India<br>▪ [19] Standard Brazil<br>**Notes:**<br>▪ The parameter is applicable only to FXS interfaces.<br>▪ 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.<br>▪ Only FXS ports can generate the Britain [4] Caller ID.<br>▪ To select the Bellcore Caller ID sub standard, use the BellcoreCallerIDTypeOneSubStandard parameter. To select the ETSI Caller ID substandard, use the ETSICallerIDTypeOneSubStandard parameter.<br>▪ To select the Bellcore MWI sub standard, use the BellcoreVMWITypeOneStandard parameter. To select the ETSI MWI sub standard, use the ETSIVMWITypeOneStandard parameter.<br>▪ 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. |
| Enable FXS Caller ID Category Digit For Brazil Telecom<br>`fxs-callid-cat-brazil`<br>[AddCPCPrefix2BrazilCallerID] | Enables the interworking of Calling Party Category (cpc) code from SIP INVITE messages to FXS Caller ID first digit.<br>▪ [0] Disable (default)<br>▪ [1] Enable |

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>When the 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:<br>
<code>From:<sip:+551137077801;cpc=payphone@10.20.7.35>;tag=53700</code><br>
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.<br>
If the incoming INVITE message doesn't contain the 'cpc' parameter, nothing is added to the Caller ID number.
<table>
<tr><th>CPC Value in Received INVITE</th><th>CPC Code Prefixed to Caller ID (Sent to FXS Endpoint)</th><th>Description</th></tr>
<tr><td>cpc=unknown</td><td>1</td><td>Unknown user</td></tr>
<tr><td>cpc=subscribe</td><td>1</td><td>-</td></tr>
<tr><td>cpc=ordinary</td><td>1</td><td>Ordinary user</td></tr>
<tr><td>cpc=priority</td><td>2</td><td>Pre-paid user</td></tr>
<tr><td>cpc=test</td><td>3</td><td>Test user</td></tr>
<tr><td>cpc=operator</td><td>5</td><td>Operator</td></tr>
<tr><td>cpc=data</td><td>6</td><td>Data call</td></tr>
<tr><td>cpc=payphone</td><td>7</td><td>Payphone user</td></tr>
</table>
Notes:
<ul>
<li>The parameter is applicable only to FXS interfaces.</li>
<li>For the parameter to be enabled, you must also set the parameter EnableCallingPartyCategory to 1.</li>
</ul></td></tr>
<tr><td>[EnableCallerIDTypeTwo]</td><td>Disables the generation of Caller ID type 2 when the phone is off-hooked. Caller ID type 2 (also known as off-hook Caller ID) is sent to a currently busy telephone to display the caller ID of the waiting call.
<ul>
<li>[0] = Caller ID type 2 isn't played.</li>
<li>[1] = (Default) Caller ID type 2 is played.</li>
<li>Note: The parameter is applicable only to FXS interfaces.</li>
</ul></td></tr>
<tr><td><code>caller-id-timing-mode</code><br>[AnalogCallerIDTimingMode]</td><td>Determines when Caller ID is generated.
<ul>
<li>[0] = (Default) Caller ID is generated between the first two rings.</li>
<li>[1] = The device attempts to find an optimized timing to generate the Caller ID according to the selected Caller ID type.</li>
</ul>
Notes:
<ul>
<li>The parameter is applicable only to FXS interfaces.</li>
</ul></td></tr>
</table>

| Parameter | Description |
|---|---|
| | ▪ If the parameter is set to 1 and used with distinctive ringing, the Caller ID signal doesn't change the distinctive ringing timing.<br>▪ For the parameter to take effect, a device reset is required. |
| `bellcore-callerid-type-one-sub-standard`<br>[BellcoreCallerIDTypeOneSubStandard] | Determines the Bellcore Caller ID sub-standard.<br>▪ [0] = (Default) Between rings.<br>▪ [1] = Not ring related.<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ The parameter is applicable only to FXS interfaces. |
| `etsi-callerid-type-one-sub-standard`<br>[ETSICallerIDTypeOneSubStandard] | Determines the ETSI FSK Caller ID Type 1 sub-standard (FXS only).<br>▪ [0] = (Default) ETSI between rings.<br>▪ [1] = ETSI before ring DT_AS.<br>▪ [2] = ETSI before ring RP_AS.<br>▪ [3] = ETSI before ring LR_DT_AS.<br>▪ [4] = ETSI not ring related DT_AS.<br>▪ [5] = ETSI not ring related RP_AS.<br>▪ [6] = ETSI not ring related LR_DT_AS.<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ The parameter is applicable only to FXS interfaces. |
| Asserted Identity Mode<br>`asserted-identity-m`<br>[AssertedIdMode] | Determines whether the SIP header P-Asserted-Identity or P-Preferred-Identity is added to the sent INVITE, 200 OK, or UPDATE request for Caller ID (or privacy). These headers are used to present the calling party's Caller ID, which is composed of a Calling Number and a Calling Name (optional).<br>▪ **[0]** Disabled = (Default) P-Asserted-Identity and P-Preferred-Identity headers are not added.<br>▪ **[1]** Add P-Asserted-Identity<br>▪ **[2]** Add P-Preferred-Identity<br>The used header also depends on the calling Privacy (allowed or restricted). These headers are used together with the Privacy header. If Caller ID is restricted (i.e., P-Asserted-Identity is not sent), the Privacy header includes the value 'id' ('Privacy: id'). Otherwise, for allowed Caller ID, 'Privacy: none' is used. If Caller ID is restricted (received from Tel or configured in the device), the From header is set to <anonymous@anonymous.invalid>. |
| Reject Anonymous Calls Per Port Table | |
| `configure voip > gw analoggw reject-anonymous-calls`<br>[RejectAnonymousCallPerPort] | This table parameter determines whether the device rejects incoming anonymous calls per FXS port. If enabled, when a device's FXS interface receives an anonymous call, it rejects the call and responds with a SIP 433 (Anonymity Disallowed) response.<br>The format of the ini file table parameter is as follows:<br>[RejectAnonymousCallPerPort]<br>FORMAT RejectAnonymousCallPerPort_Index = |

| Parameter | Description |
|---|---|
| | RejectAnonymousCallPerPort_Enable, RejectAnonymousCallPerPort_Port, RejectAnonymousCallPerPort_Module; [\RejectAnonymousCallPerPort]<br>Where,<br>▪ Enable = accept [0] (default) or reject [1] incoming anonymous calls.<br>▪ Port = Port number.<br>▪ Module = Module number.<br>For example:<br>RejectAnonymousCallPerPort 0 = 0,1,1;<br>RejectAnonymousCallPerPort 1 = 1,2,1;<br>**Note:** The parameter is applicable only to FXS interfaces. |

### 45.10.5.2 Call Waiting Parameters

The call waiting parameters are described in the table below.

**Table 45-42: Call Waiting Parameters**

| Parameter | Description |
|---|---|
| Enable Call Waiting<br>`call-waiting`<br>[EnableCallWaiting] | Enables the Call Waiting feature.<br>▪ **[0]** Disable<br>▪ **[1]** Enable (Default)<br>For FXS interface: If enabled, when an FXS interface receives a call on a busy endpoint, it responds with a 182 response (and not with a 486 busy). The device plays a call waiting indication signal. When hook-flash is detected, the device switches to the waiting call. The device that initiated the waiting call plays a call waiting ringback tone to the calling party after a 182 response is received.<br>**Notes:**<br>▪ 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).<br>▪ FXS interfaces: The EnableHold parameter must be enabled on both the calling and the called side.<br>▪ Analog interfaces: You can use the table parameter CallWaitingPerPort to enable Call Waiting per port.<br>▪ Analog interfaces: For information on the Call Waiting feature, see Call Waiting on page 452. |
| [Send180ForCallWaiting] | Determines the SIP response code for indicating Call Waiting.<br>▪ **[0]** = (Default) Use 182 Queued response to indicate call waiting.<br>▪ **[1]** = Use 180 Ringing response to indicate call waiting. |
| Call Waiting Table | |
| Call Waiting Table<br>`configure voip > gw analoggw call-waiting` | Defines call waiting per FXS port. The format of this The format of the ini file table parameter format of the ini file table parameter is as follows: |

| Parameter | Description |
|---|---|
| [CallWaitingPerPort] | [CallWaitingPerPort]<br>FORMAT CallWaitingPerPort_Index = CallWaitingPerPort_IsEnabled, CallWaitingPerPort_Module, CallWaitingPerPort_Port;<br>[\CallWaitingPerPort]<br>For example:<br>CallWaitingPerPort 0 = 0,1,1; (call waiting disabled for Port 1 of Module 1)<br>CallWaitingPerPort 1 = 1,1,2; (call waiting enabled for Port 2 of Module 1)<br>**Notes:**<br>▪ The parameter is applicable only to FXS ports.<br>▪ For a detailed description of the table, see Configuring Call Waiting on page 480. |
| Number of Call Waiting Indications<br>`nb-of-cw-ind`<br>[NumberOfWaitingIndications] | Defines the number of call waiting indications that are played to the called telephone that is connected to the device for Call Waiting.<br>The valid range is 1 to 100 indications. The default is 2.<br>**Note:** The parameter is applicable only to FXS ports. |
| Time Between Call Waiting Indications<br>`time-between-cw`<br>[TimeBetweenWaitingIndications] | Defines the time (in seconds) between consecutive call waiting indications for call waiting.<br>The valid range is 1 to 100. The default is 10.<br>**Note:** The parameter is applicable only to FXS ports. |
| Time Before Waiting Indications<br>`time-b4-cw-ind`<br>[TimeBeforeWaitingIndications] | Defines the interval (in seconds) before a call waiting indication is played to the port that is currently in a call.<br>The valid range is 0 to 100. The default time is 0 seconds.<br>**Note:** The parameter is applicable only to FXS ports. |
| Waiting Beep Duration<br>`waiting-beep-dur`<br>[WaitingBeepDuration] | Defines the duration (in msec) of call waiting indications that are played to the port that is receiving the call.<br>The valid range is 100 to 65535. The default is 300.<br>**Note:** The parameter is applicable only to FXS ports. |
| [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).<br>There are three ways to use the distinctive call waiting tones:<br>▪ 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.<br>▪ Playing the call waiting tone according to PriorityIndex in the ToneIndex table parameter.<br>▪ Playing the call waiting tone according to the parameter “CallWaitingTone#' of a SIP INFO message.<br>The device plays the tone received in the 'play tone CallWaitingTone#' parameter of an INFO message plus the value of the parameter minus 1.<br>The valid range is -1 to 1,000. The default is -1 (i.e., not used). |

| Parameter | Description |
|---|---|
| | **Notes:**<br>▪ The parameter is applicable only to analog interfaces.<br>▪ It is assumed that all Call Waiting Tones are defined in sequence in the CPT file.<br>▪ SIP Alert-Info header examples:<br>✓ Alert-Info:<Bellcore-dr2><br>✓ Alert-Info:<http://…/Bellcore-dr2> (where "dr2" defines call waiting tone #2)<br>▪ The SIP INFO message is according to Broadsoft's application server definition. Below is an example of such an INFO message:<br>INFO sip:06@192.168.13.2:5060 SIP/2.0<br>Via:SIP/2.0/UDP 192.168.13.40:5060;branch=z9hG4bK040066422630<br>From: <sip:4505656002@192.168.13.40:5060>;tag=1455352915<br>To: <sip:06@192.168.13.2:5060><br>Call-ID:0010-0008@192.168.13.2<br>CSeq:342168303 INFO<br>Content-Length:28<br>Content-Type:application/broadsoft<br>play tone CallWaitingTone1 |

### 45.10.5.3 Call Forwarding Parameters

The call forwarding parameters are described in the table below.

**Table 45-43: Call Forwarding Parameters**

| Parameter | Description |
|---|---|
| Enable Call Forward<br>`call-forward`<br>[EnableForward] | Enables the Call Forwarding feature.<br>▪ **[0]** Disable<br>▪ **[1]** Enable (Default)<br>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.<br>**Notes:**<br>▪ To use this service, the devices at both ends must support this option.<br>▪ For the device to respond to SIP 3xx responses with a new SIP request (forwarding the original request), set the parameter to **Enable**. |
| Call Forwarding Table | |
| Call Forwarding Table<br>`configure voip > gw analoggw call-forward`<br>[FwdInfo] | Defines 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.<br>The format of the ini file table parameter is as follows:<br>[FwdInfo]<br>FORMAT FwdInfo_Index = FwdInfo_Type, FwdInfo_Destination, |

| Parameter | Description |
|---|---|
| | FwdInfo_NoReplyTime, FwdInfo_Module, FwdInfo_Port;<br>[\FwdInfo]<br>Where,<br>▪ Module = Module number, where 1 denotes the module in Slot 1.<br>▪ Port = Port number, where 1 denotes Port 1 of a module.<br>For example:<br>▪ Below configuration forwards calls originally destined to Port 1 of Module 1 to "1001" upon On Busy:<br>FwdInfo 0 = 1,1001,30,1,1;<br>▪ Below configuration forwards calls originally destined to Port 2 of Module 1 to an IP address upon On Busy:<br>FwdInfo 1 = 1,2003@10.5.1.1,30,1,2;<br>For a detailed description of the table, see Configuring Call Forward on page 477.<br>**Note:** The parameter is applicable only to FXS interfaces. |
| Call Forward Reminder Ring Parameters<br>**Notes:**<br>▪ These parameters are applicable only to FXS interfaces.<br>▪ For a description of this feature, see Call Forward Reminder Ring on page 450. | |
| Enable NRT Subscription<br>`nrt-subscription`<br>[EnableNRTSubscription] | Enables endpoint subscription for Ring reminder event notification feature.<br>▪ [0] Disable (default)<br>▪ [1] Enable |
| AS Subscribe IPGroupID<br>`as-subs-ipgroupid`<br>[ASSubscribeIPGroupID] | Defines the IP Group ID that contains the Application server for Subscription.<br>The valid value range is 1 to 8. The default is -1 (i.e., not configured). |
| NRT Retry Subscription Time<br>[NRTSubscribeRetryTime] | Defines the Retry period (in seconds) for Dialog subscription if a previous request failed.<br>The valid value range is 10 to 7200. The default is 120. |
| Call Forward Ring Tone ID<br>`cfe-ring-tone-id`<br>[CallForwardRingToneID] | Defines the ringing tone type played when call forward notification is accepted.<br>The valid value range is 1 to 5. The default is 1. |

### 45.10.5.4 Message Waiting Indication Parameters

The message waiting indication (MWI) parameters are described in the table below.

**Table 45-44: MWI Parameters**

| Parameter | Description |
|---|---|
| Enable MWI<br>`enable-mwi`<br>[EnableMWI] | Enables Message Waiting Indication (MWI).<br>▪ **[0]** Disable (default).<br>▪ **[1]** Enable<br>**Notes:** |

| Parameter | Description |
|---|---|
| | ▪ The parameter is applicable only to FXS interfaces.<br>▪ The device supports only the receipt of SIP MWI NOTIFY messages (the device doesn't generate these messages).<br>▪ For more information on MWI, see "Message Waiting Indication" on page 453. |
| MWI Analog Lamp<br>`mwi-analog-lamp`<br>[MWIAnalogLamp] | Enables the visual display of MWI.<br>▪ **[0]** Disable (default).<br>▪ **[1]** Enable = Enables visual MWI by supplying line voltage of approximately 100 VDC to activate the phone's lamp.<br>**Notes:**<br>▪ The parameter is applicable only for FXS interfaces.<br>▪ The parameter can also be configured in a Tel Profile. |
| MWI Display<br>`enable-mwi`<br>[MWIDisplay] | Enables sending MWI information to the phone display.<br>▪ **[0]** Disable = (Default) MWI information isn't sent to display.<br>▪ **[1]** Enable = The device generates an MWI message (determined by the parameter CallerIDType), which is displayed on the MWI display.<br>**Note:**<br>▪ The parameter is applicable only to FXS interfaces.<br>▪ The parameter can also be configured in a Tel Profile. |
| Subscribe to MWI<br>`subscribe-to-mwi`<br>[EnableMWISubscription] | Enables subscription to an MWI server.<br>▪ **[0]** No (default)<br>▪ **[1]** Yes<br>**Notes:**<br>▪ To configure the MWI server address, use the MWIServerIP parameter.<br>▪ To configure whether the device subscribes per endpoint or per the entire device, use the parameter SubscriptionMode. |
| MWI Server IP Address<br>`mwi-srvr-ip-addr`<br>[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. |
| MWI Server Transport Type<br>`mwi-srvr-transp-type`<br>[MWIServerTransportType] | Determines the transport layer used for outgoing SIP dialogs initiated by the device to the MWI server.<br>▪ **[-1]** Not Configured (default)<br>▪ **[0]** UDP<br>▪ **[1]** TCP<br>▪ **[2]** TLS<br>**Note:** When set to 'Not Configured', the value of the parameter SIPTransportType is used. |
| MWI Subscribe Expiration Time<br>`mwi-subs-expr-time`<br>[MWIExpirationTime] | Defines the MWI subscription expiration time in seconds.<br>The default is 7200 seconds. The range is 10 to 2,000,000. |
| MWI Subscribe Retry Time<br>`mwi-subs-rtry-time`<br>[SubscribeRetryTime] | Defines the subscription retry time (in seconds) after last subscription failure.<br>The default is 120 seconds. The range is 10 to 2,000,000. |

| Parameter | Description |
|---|---|
| Subscription Mode<br>`subscription-mode`<br>[SubscriptionMode] | Determines the method the device uses to subscribe to an MWI server.<br>▪ **[0]** Per Endpoint = (Default) Each endpoint subscribes separately - typically used for FXS interfaces.<br>▪ **[1]** Per Gateway = Single subscription for the entire device. |
| `etsi-vmwi-type-one-standard`<br>[ETSIVMWITypeOneStandard] | Determines the ETSI Visual Message Waiting Indication (VMWI) Type 1 sub-standard.<br>▪ **[0]** = (Default) ETSI VMWI between rings<br>▪ **[1]** = ETSI VMWI before ring DT_AS<br>▪ **[2]** = ETSI VMWI before ring RP_AS<br>▪ **[3]** = ETSI VMWI before ring LR_DT_AS<br>▪ **[4]** = ETSI VMWI not ring related DT_AS<br>▪ **[5]** = ETSI VMWI not ring related RP_AS<br>▪ **[6]** = ETSI VMWI not ring related LR_DT_AS<br>**Note:** For the parameter to take effect, a device reset is required. |
| `bellcore-vmwi-type-one-standard`<br>[BellcoreVMWITypeOneStandard] | Determines the Bellcore VMWI sub-standard.<br>▪ **[0]** = (Default) Between rings.<br>▪ **[1]** = Not ring related.<br>**Note:** For the parameter to take effect, a device reset is required. |

### 45.10.5.5 Call Hold Parameters

The call hold parameters are described in the table below.

**Table 45-45: Call Hold Parameters**

| Parameter | Description |
|---|---|
| Enable Hold<br>`hold`<br>[EnableHold] | Global parameter that enables the Call Hold feature. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableHold). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| Hold Format<br>`hold-format`<br>[HoldFormat] | Defines the format of the SDP in the sent re-INVITE hold request.<br>▪ **[0]** 0.0.0.0 = (Default) The SDP "c=" field contains the IP address "0.0.0.0" and the "a=inactive" attribute.<br>▪ **[1]** Send Only = The SDP "c=" field contains the device's IP address and the "a=sendonly" attribute.<br>▪ **[2]** x.y.z.t = The SDP "c=" field contains the device's IP address and the "a=inactive" attribute.<br>**Note:** The device does not send any RTP packets when it is in hold state. |

| Parameter | Description |
|---|---|
| Held Timeout<br>`held-timeout`<br>[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).<br>▪ **[-1]** = (Default) The call is placed on hold indefinitely until the initiator of the on hold retrieves the call again.<br>▪ **[0 - 2400]** = Time to wait (in seconds) after which the call is released. |
| Call Hold Reminder Ring Timeout<br>`call-hold-remnd-rng`<br>[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 the parameter. If the user off-hooks the phone, the call becomes active.<br>The valid range is 0 to 600. The default is 30.<br>**Notes:**<br>▪ The parameter is applicable only to FXS interfaces.<br>▪ This Reminder Ring feature can be disabled using the DisableReminderRing parameter. |
| `dis-reminder-ring`<br>[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.<br>▪ [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.<br>▪ [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).<br>**Notes:**<br>▪ The parameter is applicable only to FXS interfaces.<br>▪ The parameter is typically used for MLPP, allowing preemption to clear held calls. |
| `dtmf-during-hold`<br>[PlayDTMFduringHold] | Determines whether the device sends DTMF signals (or DTMF SIP INFO message) when a call is on hold.<br>▪ **[0]** = (Default) Disable.<br>▪ **[1]** = Enable - If the call is on hold, the device stops playing the Held tone (if it is played) and sends DTMF:<br>✓ To Tel side: plays DTMF digits according to the received SIP INFO message(s). (The stopped held tone is not played again.)<br>✓ To IP side: sends DTMF SIP INFO messages to an IP destination if it detects DTMF digits from the Tel side. |

### 45.10.5.6 Call Transfer Parameters

The call transfer parameters are described in the table below.

**Table 45-46: Call Transfer Parameters**

| Parameter | Description |
|---|---|
| Enable Transfer<br>`enable-transfer`<br>[EnableTransfer] | Enables the Call Transfer feature.<br>▪ **[0]** Disable<br>▪ **[1]** Enable = (Default) The device responds to a REFER message with the Referred-To header to initiate a call transfer. For analog interfaces: If the transfer service is enabled, the user can activate Transfer using hook-flash signaling. If this service is enabled, the remote party performs the call transfer.<br>**Notes:**<br>▪ To use call transfer, the devices at both ends must support this option.<br>▪ To use call transfer, set the parameter EnableHold to 1. |
| Transfer Prefix<br>`transfer-prefix`<br>[xferPrefix] | Defines the string that is added as a prefix to the transferred/forwarded called number when the REFER/3xx message is received.<br>**Notes:**<br>▪ 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.<br>▪ The parameter can be used to apply different manipulation rules to differentiate transferred/forwarded number from the originally dialed number. |
| Enable Semi-Attended Transfer<br>`semi-att-transfer`<br>[EnableSemiAttendedTransfer] | Determines the device behavior when Transfer is initiated while in Alerting state.<br>▪ **[0]** Disable = (Default) Send REFER with the Replaces header.<br>▪ **[1]** Enable = Send CANCEL, and after a 487 response is received, send REFER without the Replaces header. |
| Blind<br>`blind-transfer`<br>[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.<br>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.<br>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.<br>**Note:** For FXS interfaces, it is possible to configure whether the KeyBlindTransfer code is added as a prefix to the dialed |

| Parameter | Description |
|---|---|
| | destination number, by using the parameter KeyBlindTransferAddPrefix. |
| `blind-xfer-add-prefix`<br>[KeyBlindTransferAddPrefix] | Determines whether the device adds the Blind Transfer code (defined by the KeyBlindTransfer parameter) to the dialed destination number.<br>▪ [0] Disable (default)<br>▪ [1] Enable<br>**Note:** The parameter is applicable only to FXS interfaces. |
| `blind-xfer-disc-tmo`<br>[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 the parameter is set to 0, the REFER message is immediately sent.<br>The valid value range is 0 to 1,000,000. The default is 0. |
| QSIG Path Replacement Mode<br>`qsig-path-replacement-md`<br>[QSIGPathReplacementMode] | Enables QSIG transfer for IP-to-Tel and Tel-to-IP calls.<br>▪ [0] IP2QSIGTransfer = (Default) Enables IP-to-QSIG transfer.<br>▪ [1] QSIG2IPTransfer = Enables QSIG-to-IP transfer. |
| `replace-tel2ip-calnum-to`<br>[ReplaceTel2IPCallingNumTimeout] | Defines the maximum duration (timeout) to wait between call Setup and Facility with Redirecting Number for replacing the calling number (for Tel-to-IP calls).<br>The valid value range is 0 to 10,000 msec. The default is 0.<br>The interworking of the received Setup message to a SIP INVITE is suspended when the parameter is set to any value greater than 0. This means that the redirecting number in the Setup message is not checked. When a subsequent Facility with Call Transfer Complete/Update is received with a non-empty Redirection Number, the Calling Number is replaced with the received redirect number in the sent INVITE message.<br>If the timeout expires, the device sends the INVITE without changing the calling number.<br>**Notes:**<br>▪ The suspension of the INVITE message occurs for all calls.<br>▪ The parameter is applicable to QSIG. |
| Call Transfer using re-INVITEs<br>`enable-call-transfer-using-reinvites`<br>[EnableCallTransferUsingReinvites] | Enables call transfer using re-INVITEs.<br>▪ [0] Disable = (Default) Call transfer is done using REFER messages.<br>▪ [1] Enable = Call transfer is done by sending re-INVITE messages (instead of REFER).<br>**Notes:**<br>▪ The device uses two DSP channels per transferred call. Thus, to use this feature, you also need to configure the maximum number of available DSP channels, using the MediaChannels parameter.<br>▪ The parameter is applicable only to FXS interfaces. |

### 45.10.5.7 Multi-Line Extensions and Supplementary Services Parameters

The multi-line extensions and supplementary services parameters are described in the table below.

**Table 45-47: Multi-line Extensions and Supplementary Services Parameters**

| Parameter | Description |
|---|---|
| **Supplementary Services Table** | |
| Supplementary Services Table<br>`configure voip/gw digitalgw isdn-supp-serv`<br>[ISDNSuppServ] | Defines phone extension numbers per FXS port and configures various supplementary services per endpoint.<br>The format of the ini file table parameter is as follows:<br>[ ISDNSuppServ ]<br>FORMAT ISDNSuppServ_Index = ISDNSuppServ_PhoneNumber, ISDNSuppServ_LocalPhoneNumber, ISDNSuppServ_Module, ISDNSuppServ_Port, ISDNSuppServ_UserId, ISDNSuppServ_UserPassword, ISDNSuppServ_CallerID, ISDNSuppServ_IsPresentationRestricted, ISDNSuppServ_IsCallerIDEnabled;<br>[ \ISDNSuppServ ]<br>For a detailed description of the table, see "Configuring Multi-Line Extensions and Supplementary Services" on page 460. |

### 45.10.5.8 Three-Way Conferencing Parameters

The three-way conferencing parameters are described in the table below.

**Table 45-48: Three-Way Conferencing Parameters**

| Parameter | Description |
|---|---|
| Enable 3-Way Conference<br>`enable-3w-conf`<br>[Enable3WayConference] | Enables the 3-Way Conference feature.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Note:** For the parameter to take effect, a device reset is required. |
| 3-Way Conference Mode<br>`3w-conf-mode`<br>[3WayConferenceMode] | Defines the mode of operation for three-way conferencing.<br>▪ **[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.<br>▪ **[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. |

| Parameter | Description |
|---|---|
| | ▪ **[2]** On Board = On-board, three-way conference. The conference is established on the device without the need of an external Conference server. You can limit the number of simultaneous, on-board 3-way conference calls, by using the MaxInBoardConferenceCalls parameter.<br>▪ **[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.<br>**Notes:**<br>▪ The parameter is applicable only to FXS interfaces.<br>▪ Three-way conferencing using an external conference server is supported only by FXS interfaces.<br>▪ When using an external Conferencing server, a conference call with up to six participants can be established. |
| Max. 3-Way Conference<br>[MaxInBoardConferenceCalls] | Defines the maximum number of simultaneous, on-board three-way conference calls.<br>The valid range is 0 to 5. The default is 2.<br>**Notes:**<br>▪ For enabling on-board, three-way conferencing, use the 3WayConferenceMode parameter.<br>▪ The parameter is applicable only to FXS interfaces. |
| Establish Conference Code<br>`estb-conf-code`<br>[ConferenceCode] | Defines the DTMF digit pattern, which upon detection generates the conference call when three-way conferencing is enabled (Enable3WayConference is set to 1).<br>The valid range is a 25-character string. The default is “!” (Hook-Flash).<br>**Note:** If the FlashKeysSequenceStyle parameter is set to 1 or 2, the setting of the ConferenceCode parameter is overridden. |
| Conference ID<br>`conf-id`<br>[ConferenceID] | Defines the Conference Identification string.<br>The valid value is a string of up to 16 characters. The default is "conf".<br>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. |
| Use Different RTP port After Hold<br>`configure voip > sip-definition advanced-settings > dfrnt-port-after-hold`<br>[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.<br>▪ **[0]** Disable = (Default) The FXS endpoint makes the first and second calls on the same RTP port in the initial outgoing INVITE request. If a three-way conference is then made, the device sends a re-INVITE to the held call to retrieve it and to change the RTP port to a different port number. |

| Parameter | Description |
|---|---|
| | For example: A first calls B on port 6000 and places B on hold. A then calls C, also on port 6000. The device sends a re-INVITE to the held call to retrieve it and changes the port to 6010.<br>▪ **[1]** Enable = The FXS endpoint makes the first and second calls on different RTP ports in the initial outgoing INVITE request. If a three-way conference is then made, the device sends a re-INVITE to the held call to retrieve it, without changing the port of the held call.<br>For example: A first calls B on port 6000 and places B on hold. A then calls C on port 6010. The device sends a re-INVITE to the held call to retrieve it (without changing the port, i.e., remains 6010).<br>**Notes:**<br>▪ 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.<br>▪ 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.<br>▪ The parameter is applicable only to FXS interfaces. |

### 45.10.5.9 MLPP and Emergency Call Parameters

The Multilevel Precedence and Preemption (MLPP) and emergency E911 call parameters are described in the table below.

**Table 45-49: MLPP and Emergency E911 Call Parameters**

| Parameter | Description |
|---|---|
| Call Priority Mode<br>`call-prio-mode`<br>[CallPriorityMode] | Defines priority call handling, for all calls.<br>▪ **[0]** Disable (default).<br>▪ **[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.<br>▪ **[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 Phone Number (0). The preemption is done only if the incoming IP-to-Tel call is identified as an emergency call. The device identifies emergency calls by one of the following: |

| Parameter | Description |
|---|---|
| | ✓ The destination number of the IP call matches one of the numbers defined by the EmergencyNumbers parameter. (For E911, you must define the parameter with the value "911".)<br>✓ The incoming SIP INVITE message contains the “emergency” value in the Priority header.<br>**Notes:**<br>▪ The parameter is applicable to FXS.<br>▪ MLPP and Emergency services can also be configured in a Tel Profile.<br>▪ For more information, see "Pre-empting Existing Call for E911 IP-to-Tel Call" on page 460. |
| **Emergency E911 Parameters** | |
| E911 Gateway<br>[E911Gateway] | Enables Enhanced 9-1-1(E9-1-1) support for ELIN handling in a Microsoft Lync Server environment and routing to a PSTN-based emergency service provider.<br>▪ [0] None = (Default) Disable<br>▪ [1] NG911 Gateway = Enables the ELIN Gateway.<br>For more information on E9-1-1 in a Lync environment, see Enhanced 9-1-1 Support for Lync Server on page 289.<br>**Note:** The parameter is applicable only to Gateway calls. |
| [E911CallbackTimeout] | Defines the maximum interval within which the PSAP can use the ELIN to call back the E9-1-1 caller. This interval starts from when the initial call established with the PSAP is terminated.<br>The valid range is 1 to 60 (minutes). The default is 30. |
| Emergency Special Release Cause<br>`emrg-spcl-rel-cse`<br>[EmergencySpecialReleaseCause] | Enables the device to send a SIP 503 "Service Unavailable" response if an emergency call cannot be established (i.e., rejected). This can occur, for example, due to the PSTN (for example, the destination is busy or not found) or ELIN Gateway (for example, lack of resources or an internal error).<br>▪ [0] = Disable (default)<br>▪ [1] = Enable |
| Emergency Numbers<br>`emerg-nbs`<br>[EmergencyNumbers] | Defines a list of “emergency” numbers.<br>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.<br>The list can include up to four different numbers, where each number can be up to four digits long.<br>Example: EmergencyNumbers = ‘100’,’911’,’112’ |
| Emergency Calls Regret Timeout<br>`emerg-calls-regrt-t-out`<br>[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). |

| Parameter | Description |
|---|---|
| | The valid range is 1 to 30. The default is 10.<br>**Note:** The parameter is applicable only to FXS interfaces. |
| **Multilevel Precedence and Preemption (MLPP) Parameters** | |
| `MLPP DiffServ`<br>`mlpp-diffserv`<br>[MLPPDiffserv] | Defines the DiffServ value (differentiated services code point/DSCP) used in IP packets containing SIP messages that are related to MLPP calls. The parameter defines DiffServ for incoming and outgoing MLPP calls with the Resource-Priority header.<br>The valid range is 0 to 63. The default is 50. |
| Precedence Ringing Type<br>`precedence-ringing`<br>[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.<br>The valid range is -1 to 16. The default is -1 (i.e., plays standard ringing tone).<br>**Note:** The parameter is applicable only to analog interfaces. |
| `e911-mlpp-bhvr`<br>[E911MLPPBehavior] | Defines the E911 (or Emergency Telecommunication Services/ETS) MLPP Preemption mode:<br>▪ [0] = (Default) Standard Mode - ETS calls have the highest priority and preempt any MLPP call.<br>▪ [1] = Treat as routine mode - ETS calls are handled as routine calls.<br>**Note:** The parameter is applicable only to analog interfaces. |
| `resource-prio-req`<br>[RPRequired] | Determines whether the SIP resource-priority tag is added in the SIP Require header of the INVITE message for Tel-to-IP calls.<br>▪ **[0]** Disable = Excludes the SIP resource-priority tag from the SIP Require header.<br>▪ **[1]** Enable = (Default) Adds the SIP resource-priority tag in the SIP Require header.<br>**Note:** The parameter is applicable only to MLPP priority call handling (i.e., only when the CallPriorityMode parameter is set to 1). |

| Parameter | Description |
|---|---|
| **Multiple Differentiated Services Code Points (DSCP) per MLPP Call Priority Level (Precedence) Parameters**<br>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:<br>**MLPP Precedence Level** — **Precedence Level in Resource-Priority SIP Header**<br>0 (lowest) — routine<br>2 — priority<br>4 — immediate<br>6 — flash<br>8 — flash-override<br>9 (highest) — flash-override-override | |
| RTP DSCP for MLPP Routine<br>`dscp-4-mlpp-rtn`<br>[MLPPRoutineRTPDSCP] | Defines the RTP DSCP for MLPP Routine precedence call level.<br>The valid range is -1 to 63. The default is -1.<br>**Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call. |
| RTP DSCP for MLPP Priority<br>`dscp-4-mlpp-prio`<br>[MLPPPriorityRTPDSCP] | Defines the RTP DSCP for MLPP Priority precedence call level.<br>The valid range is -1 to 63. The default is -1.<br>**Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call. |
| RTP DSCP for MLPP Immediate<br>`dscp-4-mlpp-immed`<br>[MLPPImmediateRTPDSCP] | Defines the RTP DSCP for MLPP Immediate precedence call level.<br>The valid range is -1 to 63. The default is -1.<br>**Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call. |
| RTP DSCP for MLPP Flash<br>`dscp-4-mlpp-flsh`<br>[MLPPFlashRTPDSCP] | Defines the RTP DSCP for MLPP Flash precedence call level.<br>The valid range is -1 to 63. The default is -1.<br>**Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call. |
| RTP DSCP for MLPP Flash Override<br>`dscp-4-mlpp-flsh-ov`<br>[MLPPFlashOverRTPDSCP] | Defines the RTP DSCP for MLPP Flash-Override precedence call level.<br>The valid range is -1 to 63. The default is -1.<br>**Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call. |
| RTP DSCP for MLPP Flash-Override-Override | Defines the RTP DSCP for MLPP Flash-Override-Override precedence call level. |

| Parameter | Description |
|---|---|
| `dscp-4-mlpp-flsh-ov-ov`<br>[MLPPFlashOverOverRTPDSCP] | The valid range is -1 to 63. The default is -1.<br>**Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call. |

### 45.10.5.10 Call Cut-Through Parameters

The call cut-through parameters are described in the table below.

**Table 45-50: Call Cut-Through Parameters**

| Parameter | Description |
|---|---|
| Enable Calls Cut Through<br>`calls-cut-through`<br>[CutThrough] | Enables FXS endpoints to receive incoming IP calls while the port is in off-hook state.<br>▪ [0] Disable (default)<br>▪ [1] Enable<br>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.<br>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.<br>The waiting call is automatically answered by the device when the current call is terminated (configured by setting the parameter EnableCallWaiting to 1).<br>**Note:** This feature is applicable only to FXS interfaces. |

### 45.10.5.11 Automatic Dialing Parameters

The automatic dialing upon off-hook parameters are described in the table below.

**Table 45-51: Automatic Dialing Parameters**

| Parameter | Description |
|---|---|
| **Automatic Dialing Table** | |
| Automatic Dialing Table<br>`configure voip/gw analoggw automatic-dialing`<br>[TargetOfChannel] | Defines telephone numbers that are automatically dialed when a specific FXS port is off-hooked. The format of the ini file table parameter is as follows:<br>[TargetOfChannel]<br>FORMAT TargetOfChannel_Index = TargetOfChannel_Destination, TargetOfChannel_Type, TargetOfChannel_Module, TargetOfChannel_Port, TargetOfChannel_HotLineToneDuration;<br>[\TargetOfChannel]<br>For example, the below configuration defines automatic dialing of phone number 911 when the phone connected to Port 1 of Module 1 is off-hooked for over 10 seconds:<br>TargetOfChannel 0 = 911, 1, 1, 1 ,10;<br>**Notes:**<br>▪ The first index of this table ini file parameter is 0.<br>▪ TargetOfChannel_Module is the module number, where 1 denotes the module in Slot 1.<br>▪ TargetOfChannel_Port is the port number, where 1 denotes Port 1 on the module.<br>▪ This is parameter is applicable only to FXS interfaces. |

| Parameter | Description |
|---|---|
| | ▪ For a detailed description of the table, see "Configuring Automatic Dialing" on page 473. |

### 45.10.5.12 Direct Inward Dialing Parameters

The Direct Inward Dialing (DID) parameters are described in the table below.

**Table 45-52: DID Parameters**

| Parameter | Description |
|---|---|
| DID Wink<br>`did-wink-enbl`<br>[EnableDIDWink] | Enables Direct Inward Dialing (DID) using Wink-Start signaling, typically used for signaling between an E-911 switch and the PSAP.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Single = The device can be used for connection to EIA/TIA-464B DID Loop Start lines. FXS (generation) are supported:<br>✓ The FXS interface generates a Wink signal upon detection of an off-hook state, instead of playing a dial tone.<br>For example: (Wink) KP I(I) xxx-xxxx ST (Off Hook)<br>Where:<br>✓ I = one or two information digits<br>✓ x = ANI<br>▪ **[2]** Double Wink = Double-wink signaling. This is applicable to FXS interfaces only. The FXS interface generates the first Wink upon detection of an off-hook state in the line. The second Wink is generated after a user-defined interval (configured by the TimeBetweenDIDWinks parameter) after which the DTMF/MF digits are collected by the device. Digits that arrive between the first and second Wink are ignored as they contain the same number. For example:<br>(Wink) KP 911 ST (Wink) KP I(I) xxx-xxxx ST (Off Hook)<br>▪ **[3]** Wink & Polarity=<br>✓ **FXS:** 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 by the device only after this polarity change. Digits that arrive between the first Wink and the polarity change are ignored as they always contain the same number. In this mode, the FXS interface does not generate a polarity change to normal if the Tel-to-IP call is answered by an IP party. Polarity reverts to normal when the call is released. For example:<br>(Wink) KP 911 ST (Polarity) KP I(I) xxx-xxxx ST (Off Hook)<br>**Notes:**<br>▪ The EnableReversalPolarity and PolarityReversalType parameters must be set to 1 for FXS interfaces.<br>▪ The parameter can also be configured in a Tel Profile. |

| Parameter | Description |
|---|---|
| [TimeBetweenDIDWinks] | Defines the interval (in msec) for wink signaling:<br>▪ Double-wink signaling [2]: interval between the first and second wink<br>▪ Wink and Polarity signaling [3]: interval between wink and polarity change<br>The valid range is 100 to 2000. The default is 1000.<br>**Note:** See the EnableDIDWink parameter for configuring the wink signaling type. |
| Delay Before DID Wink<br>`delay-b4-did-wink`<br>[DelayBeforeDIDWink] | Defines the time interval (in msec) between the detection of the off-hook and the generation of the DID Wink.<br>The valid range is 0 to 1,000. The default is 0.<br>**Note:** The parameter is applicable only to FXS interfaces. |
| `NTT-DID-signaling-form`<br>[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.<br>▪ **[0]** = (Default) FSK-based signaling<br>▪ **[1]** = DTMF-based signaling<br>**Note:** The parameter is applicable only to FXS interfaces. |
| [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.<br>The format of the ini file table parameter is as follows:<br>[EnableDID]<br>FORMAT EnableDID_**Index** = EnableDID_**IsEnable**, EnableDID_Port, EnableDID_Module;<br>[\EnableDID]<br>Where,<br>▪ IsEnable = Enables [1] or disables [0] (default) Japan NTT Modem DID support.<br>▪ Port = Port number.<br>▪ Module = Module number.<br>For example:<br>EnableDID 0 = 1,1,2; (DID is enabled on Port 1 of Module 2)<br>**Note:** The parameter is applicable only to FXS interfaces. |
| `wink-time`<br>[WinkTime] | Defines the time (in msec) elapsed between two consecutive polarity reversals. The parameter can be used for DID signaling, for example, E911 lines to the Public Safety Answering Point (PSAP), according to the Bellcore GR-350-CORE standard (refer to the ini file parameter Enable911PSAP).<br>The valid range is 0 to 4,294,967,295. The default is 200.<br>**Notes:**<br>▪ The parameter is applicable to FXS interfaces.<br>▪ For the parameter to take effect, a device reset is required. |

### 45.10.5.13 Character Conversion Parameters

The Character Conversion table  parameter is described in the table below.

**Table 45-53: Char Conversion Table Parameters**

| **Char Conversion Table** | |
|---|---|
| Char Conversion<br>`configure voip > gw dtmf-and-suppl dtmf-and-dialing char-conversion display`<br>[CharConversion] | Defines Unicode-to-ASCII character conversion rules. The format of the ini file table parameter is as follows:<br>[CharConversion]<br>FORMAT CharConversion_Index = CharConversion_CharName, CharConversion_FirstByte, CharConversion_SecondByte, CharConversion_ConvertedOutput;<br>[\CharConversion]<br>For a detailed description of the table, see "Converting Accented Characters from IP to Tel" on page 466. |

## 45.10.6 Answer and Disconnect Supervision Parameters

The answer and disconnect supervision parameters are described in the table below.

**Table 45-54: Answer and Disconnect Parameters**

| Parameter | Description |
|---|---|
| Answer Supervision<br>`answer-supervision`<br>[EnableVoiceDetection] | Enables the sending of SIP 200 OK upon detection of speech, fax, or modem.<br>▪ **[1]** Yes = The device sends a SIP 200 OK (in response to an INVITE message) when speech, fax, or modem is detected (from the Tel side, for analog interfaces).<br>▪ **[0]** No = (Default) The device sends a SIP 200 OK only after it completes dialing (to the Tel side, for analog interfaces).<br>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. |
| Disconnect on Broken Connection<br>`disc-broken-conn`<br>[DisconnectOnBrokenConnection] | Global parameter that defines the device's handling of calls if RTP packets are not received within a user-defined timeout (configured by the BrokenConnectionEventTimeout parameter). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_DisconnectOnBrokenConnection). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| Broken Connection Timeout<br>`broken-connection-event-timeout`<br>[BrokenConnectionEventTimeout] | Defines the time period (in 100-msec units) after which a call is disconnected if an RTP packet is not received.<br>The valid range is from 3 (i.e., 300 msec) to approx. 2684354 (i.e., 74.5 hours). The default is 100 (i.e., 10000 msec or 10 seconds).<br>**Notes:** |

| Parameter | Description |
|---|---|
| | ▪ The parameter is applicable only if the parameter DisconnectOnBrokenConnection is set to 1.<br>▪ Currently, this feature functions only if Silence Suppression is disabled. |
| Polarity (Current) Reversal for Call Release (Analog Interfaces) Parameters | |
| Enable Polarity Reversal<br>`polarity-rvrsl`<br>[EnableReversalPolarity] | Enables the polarity reversal feature for call release.<br>▪ [0] Disable = (Default) Disable the polarity reversal service.<br>▪ [1] Enable = Enable the polarity reversal service.<br>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.<br>**Note:** The parameter can also be configured in a Tel Profile. |
| Enable Current Disconnect<br>`current-disc`<br>[EnableCurrentDisconnect] | Enables call release upon detection of a Current Disconnect signal.<br>▪ [0] Disable = (Default) Disable the current disconnect service.<br>▪ [1] Enable = Enable the current disconnect service.<br>If the current disconnect service is enabled:<br>▪ The FXS interface generates a 'Current Disconnect Pulse' after a call is released from IP.<br>The current disconnect duration is configured by the CurrentDisconnectDuration parameter. The frequency at which the analog line voltage is sampled is configured by the TimeToSampleAnalogLineVoltage parameter.<br>**Note:** The parameter can also be configured in a Tel Profile. |
| `polarity-reversal-type`<br>[PolarityReversalType] | Defines the voltage change slope during polarity reversal or wink.<br>▪ [0] = (Default) Soft reverse polarity.<br>▪ [1] = Hard reverse polarity.<br>**Notes:**<br>▪ The parameter is applicable only to FXS interfaces.<br>▪ 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).<br>▪ For the parameter to take effect, a device reset is required. |
| `current-disconnect-duration`<br>[CurrentDisconnectDuration] | Defines the duration (in msec) of the current disconnect pulse.<br>The range is 200 to 1500. The default is 900.<br>**Notes:**<br>▪ The parameter is applicable for FXS interfaces.<br>▪ For the parameter to take effect, a device reset is required. |

## 45.10.7 Tone Parameters

This subsection describes the device's tone parameters.

### 45.10.7.1 Telephony Tone Parameters

The telephony tone parameters are described in the table below.

**Table 45-55: Tone Parameters**

| Parameter | Description |
|---|---|
| SIP Hold Behavior<br>`sip-hold-behavior`<br>[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.<br>▪ [0] Disable (default)<br>▪ [1] Enable |
| Dial Tone Duration<br>`dt-duration`<br>[TimeForDialTone] | Defines the duration (in seconds) that the dial tone is played.<br>For analog interfaces: FXS interfaces play the dial tone after the phone is picked up (off-hook).<br>The valid range is 0 to 60. The default time is 16.<br>Notes for analog interfaces:<br>▪ During play of dial tone, the device waits for DTMF digits.<br>▪ The parameter is not applicable when Automatic Dialing is enabled. |
| Stutter Tone Duration<br>`sttr-tone-duration`<br>[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.<br>The range is 1,000 to 60,000. The default is 2,000 (i.e., 2 seconds).<br>**Notes:**<br>▪ The parameter is applicable only to FXS interfaces.<br>▪ 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.<br>▪ The MWI tone overrides the call forwarding reminder tone. For more information on MWI, see Message Waiting Indication on page 453. |
| Hotline Dial Tone Duration<br>`hotline-dt-dur`<br>[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 473).<br>The valid range is 0 to 60. The default is 16.<br>**Notes:**<br>▪ The parameter is applicable to FXS interfaces. |

| Parameter | Description |
|---|---|
| | ▪ You can define the Hotline duration per FXS port using the Automatic Dialing table. |
| Reorder Tone Duration<br>`reorder-tone-duration`<br>[TimeForReorderTone] | For analog interfaces: Defines the duration (in seconds) that the device plays a busy or reorder tone before releasing the line. Typically, after playing the busy or reorder tone for this duration, the device starts playing an offhook warning tone.<br>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".<br>**Notes:**<br>▪ The selected busy or reorder tone is according to the SIP release cause code received from IP.<br>▪ The parameter can also be configured for a Tel Profile (in the Tel Profile table). |
| Time Before Reorder Tone<br>`time-b4-reordr-tn`<br>[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.<br>The valid range is 0 to 60. The default is 0.<br>**Note:** The parameter is applicable only to FXS interfaces. |
| Cut Through Reorder Tone Duration<br>`cut-thru-reord-dur`<br>[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.<br>The valid values are 0 to 30. The default is 0 (i.e., no reorder tone is played).<br>**Note:** To enable the Cut-Through feature, use the CutThrough (for FXS channels) parameter. |
| Enable Comfort Tone<br>`comfort-tone`<br>[EnableComfortTone] | Determines whether the device plays a comfort tone (Tone Type #18) to the FXS endpoint after a SIP INVITE is sent and before a SIP 18x response is received.<br>▪ [0] Disable (default)<br>▪ [1] Enable<br>**Note:** The parameter is applicable to FXS interfaces. |
| [WarningToneDuration] | Defines the duration (in seconds) for which the offhook warning tone is played to the user.<br>The valid range is -1 to 2,147,483,647. The default is 600.<br>**Notes:**<br>▪ A negative value indicates that the tone is played infinitely.<br>▪ The parameter is applicable only to analog interfaces. |
| Play Ringback Tone to Tel<br>`play-rbt2tel`<br>[PlayRBTone2Tel] | Determines the playing method of the ringback tone to the Tel (for analog interfaces).<br>▪ **[0]** Don't Play =<br>✓ Analog Interfaces: Ringback tone is not played.<br>▪ **[1]** Play on Local =<br>✓ Analog Interfaces: Plays a ringback tone to the Tel side of the call when a SIP 180/183 response is received. |

| Parameter | Description |
|---|---|
| | ▪ **[2]** Prefer IP = (Default):<br>✓ Analog Interfaces: Plays a ringback tone to the Tel side only if a 180/183 response without SDP is received. If 180/183 with SDP message is received, the device cuts through the voice channel and doesn't play the ringback tone.<br>▪ **[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.<br>**Note:** The parameter is applicable only to the Gateway application. |
| Play Ringback Tone to IP<br>`play-rbt-2ip`<br>[PlayRBTone2IP] | Global parameter that enables the device to play a ringback tone to the IP side for IP-to-Tel calls. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_PlayRBTone2IP). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| Tone Index Table | |
| Tone Index Table<br>`configure voip > gw analoggw tone-index`<br>[ToneIndex] | Defines 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).<br>The format of the ini file table parameter is as follows:<br>[ToneIndex]<br>FORMAT ToneIndex_Index = ToneIndex_FXSPort_First, ToneIndex_FXSPort_Last, ToneIndex_SourcePrefix, ToneIndex_DestinationPrefix, ToneIndex_PriorityIndex;<br>[\ToneIndex]<br>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.<br>ToneIndex 1 = 1, 2, 20*, , 3;<br>For a detailed description of the table, see Configuring FXS Distinctive Ringing and Call Waiting Tones per Source/Destination Number.<br>**Note:** The parameter is applicable only to FXS interfaces. |

### 45.10.7.2 Tone Detection Parameters

The signal tone detection parameters are described in the table below.

**Table 45-56: Tone Detection Parameters**

| Parameter | Description |
|---|---|
| `DTMF-detector-enable`<br>[DTMFDetectorEnable] | Enables the detection of DTMF signaling.<br>▪ **[0]** = Disable<br>▪ **[1]** = Enable (default) |
| `MFR1-detector-enable`<br>[MFR1DetectorEnable] | Enables the detection of MF-R1 signaling.<br>▪ **[0]** = Disable (default)<br>▪ **[1]** = Enable |
| [R1DetectionStandard] | Determines the MF-R1 protocol used for detection.<br>▪ [0] = ITU (default)<br>▪ [1] = R1.5<br>**Note:** For the parameter to take effect, a device reset is required. |
| `user-defined-tones-detector-enable`<br>[UserDefinedToneDetectorEnable] | Enables the detection of User Defined Tones signaling, applicable for Special Information Tone (SIT) detection.<br>▪ **[0]** = Disable (default)<br>▪ **[1]** = Enable |
| `sit-detector-enable`<br>[SITDetectorEnable] | Enables SIT detection according to the ITU-T recommendation E.180/Q.35.<br>▪ **[0]** = Disable (default)<br>▪ **[1]** = Enable<br>**Note:** For the parameter to take effect, a device reset is required. |
| `udt-detector-frequency-deviation`<br>[UDTDetectorFrequencyDeviation] | Defines the deviation (in Hz) allowed for the detection of each signal frequency.<br>The valid range is 1 to 50. The default is 50.<br>**Note:** For the parameter to take effect, a device reset is required. |
| `cpt-detector-frequency-deviation`<br>[CPTDetectorFrequencyDeviation] | Defines the deviation (in Hz) allowed for the detection of each CPT signal frequency.<br>The valid range is 1 to 30. The default is 10.<br>**Note:** For the parameter to take effect, a device reset is required. |

### 45.10.7.3 Metering Tone Parameters

The metering tone parameters are described in the table below.

**Table 45-57: Metering Tone Parameters**

| Parameter | Description |
|---|---|
| Generate Metering Tones<br>`gen-mtr-tones`<br>[PayPhoneMeteringMode] | Defines the method for configuring metering tones that are generated to the Tel side.<br>▪ **[0]** Disable = (Default) Metering tones are not generated.<br>▪ **[1]** Internal Table = Metering tones are generated by the device according to the Charge Code table (see Configuring Charge Codes on page 464) and sent to the Tel side.<br>**Note:** The parameter is applicable only to FXS interfaces. |
| Analog Metering Type<br>`metering-type`<br>[MeteringType] | Determines the metering method for generating pulses (sinusoidal metering burst frequency) by the FXS port.<br>▪ [0] 12 KHz = (Default) 12 kHz sinusoidal bursts.<br>▪ [1] 16 KHz = 16 kHz sinusoidal bursts.<br>▪ [2] = Polarity Reversal pulses.<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ The parameter is applicable only to FXS interfaces. |
| Analog TTX Voltage Level<br>[AnalogTTXVoltageLevel]<br> | Determines the metering signal/pulse voltage level (TTX).<br>▪ [0] 0V = 0 Vrms sinusoidal bursts.<br>▪ [1] 0.5V = (Default) 0.5 Vrms sinusoidal bursts.<br>▪ [2] 1V = 1 Vrms  sinusoidal bursts<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ The parameter is applicable only to FXS interfaces. |
| **Charge Codes Table** | |
| Charge Codes Table<br>`configure voip > gw analoggw ChargeCode`<br>[ChargeCode] | Defines metering tones and their time intervals that the FXS interface generates to the Tel side.<br>The format of the ini file table parameter is as follows:<br>[ChargeCode]<br>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;<br>[\ChargeCode]<br>**Notes:**<br>▪ To associate a configured Charge Code to an outgoing Tel-to-IP call, use the Tel-to-IP Routing table.<br>▪ To configure the Charge Codes table using the Web interface, see Configuring Charge Codes Table on page 464. |

## 45.10.8 Telephone Keypad Sequence Parameters

The telephony keypad sequence parameters are described in the table below.

**Table 45-58: Telephone Keypad Sequence Parameters**

| Parameter | Description |
|---|---|
| Call Pickup Key<br>`sip-definition advanced-settings > call-pickup-key`<br>[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.<br>The valid value is a string of up to 15 characters (0-9, #, and *). By default, no value is defined.<br>**Notes:**<br>▪ Call pick-up is configured only for FXS endpoints pertaining to the same Hunt Group.<br>▪ The 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.<br>The valid range is a one-character string. By default, no value is defined.<br>**Notes:**<br>▪ You can enable the device to add this string as the prefix to the collected (and sent) digits, using the parameter AddPrefix2ExtLine.<br>▪ The parameter is applicable only to FXS interfaces. |
| `prefix-2-ext-line`<br>[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).<br>▪ **[0]** = Disable (default)<br>▪ **[1]** = Enable<br>For example, if the 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.<br>**Note:** The parameter is applicable only to FXS interfaces. |
| **Hook Flash Parameters** | |

| Parameter | Description |
|---|---|
| Flash Keys Sequence Style<br>`flash-key-seq-style`<br>[FlashKeysSequenceStyle] | Determines the hook-flash key sequence for FXS interfaces.<br>▪ **[0]** Flash hook = (Default) Only the phone's flash button is used, according to the following scenarios:<br>✓ 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.<br>✓ 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.<br>▪ **[1]** Sequence 1 = Sequence of flash button with digit:<br>✓ Flash + 1: holds a call or toggles between two existing calls<br>✓ Flash + 2: makes a call transfer.<br>✓ 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).<br>▪ **[2]** Sequence 2 = Sequence of flash button with digit:<br>✓ Flash only: Places a call on hold.<br>✓ Flash + 1:<br>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.<br>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.<br>✓ Flash + 2: Places a call on hold and answers a call-waiting call, or toggles between active and on-hold calls.<br>✓ 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.<br>✓ Flash + 4: Makes a call transfer.<br>**Note:** The parameter is applicable only to FXS interfaces. |
| Flash Keys Sequence Timeout<br>`flash-key-seq-tmout`<br>[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).<br>The valid range is 100 to 5,000. The default is 2,000. |
| **Keypad Feature - Call Forward Parameters** | |
| Forward Unconditional<br>`fwd-unconditional`<br>[KeyCFUnCond] | Defines the keypad sequence to activate the immediate call forward option. |

| Parameter | Description |
|---|---|
| Forward No Answer<br>`fwd-no-answer`<br>[KeyCFNoAnswer] | Defines the keypad sequence to activate the forward on no answer option. |
| Forward On Busy<br>`fwd-on-busy`<br>[KeyCFBusy] | Defines the keypad sequence to activate the forward on busy option. |
| Forward On Busy or No Answer<br>`fwd-busy-or-no-ans`<br>[KeyCFBusyOrNoAnswer] | Defines the keypad sequence to activate the forward on 'busy or no answer' option. |
| Do Not Disturb<br>`fwd-dnd`<br>[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:<br>1 Dial the user-defined sequence number on the keypad; a dial tone is heard.<br>2 Dial the telephone number to which the call is forwarded (terminate the number with #); a confirmation tone is heard. | |
| Forward Deactivate<br>`fwd-deactivate`<br>[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** | |
| Restricted Caller ID Activate<br>`id-restriction-act`<br>[KeyCLIR] | Defines the keypad sequence to activate the restricted Caller ID option. After the sequence is pressed, a confirmation tone is heard. |
| Restricted Caller ID Deactivate<br>`id-restriction-deact`<br>[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** | |
| Hot-line Activate<br>`hotline-act`<br>[KeyHotLine] | Defines the keypad sequence to activate the delayed hotline option.<br>To activate the delayed hotline option from the telephone, perform the following:<br>1 Dial the user-defined sequence number on the keypad; a dial tone is heard.<br>2 Dial the telephone number to which the phone automatically dials after a configurable delay (terminate the number with #); a confirmation tone is heard. |
| Hot-line Deactivate<br>`hotline-deact`<br>[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**<br>**Note:** See the description of the KeyBlindTransfer parameter for this feature. | |
| **Keypad Feature - Call Waiting Parameters** | |

| Parameter | Description |
|---|---|
| Call Waiting Activate<br>`cw-act`<br>[KeyCallWaiting] | Defines the keypad sequence to activate the Call Waiting option. After the sequence is pressed, a confirmation tone is heard. |
| Call Waiting Deactivate<br>`cw-deact`<br>[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** | |
| Reject Anonymous Call Activate<br>[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. |
| Reject Anonymous Call Deactivate<br>[KeyRejectAnonymousCallDe**act]** | Defines the keypad sequence that de-activates the reject anonymous call option. After the sequence is pressed, a confirmation tone is heard. |

## 45.10.9 FXS Parameters

The general nd FXS parameters are described in the table below.

**Table 45-59: General FXS Parameters**

| Parameter | Description |
|---|---|
| Update Port Info<br>[AnalogPortInfo_x **]** | Defines an arbitrary name for an analog (FXS) port. This can be used to easily identify the port.<br>The valid value is a string of up to 40 characters. By default, the value is undefined.<br>**Notes:**<br>▪ For the ini file parameter, the *x* denotes the port number.<br>▪ For configuring a port name through the Web interface, see "Assigning a Port Name" on page 64. |
| **FXS Parameters** | |
| FXS Coefficient Type<br>`fxs-country-coefficients`<br>[FXSCountryCoefficients] | Determines the FXS line characteristics (AC and DC) according to USA or Europe (TBR21) standards.<br>▪ **[66]** Europe = TBR21<br>▪ **[70]** USA = (Default) United States<br>**Note:** For the parameter to take effect, a device reset is required. |

## 45.10.10 Hunt Groups and Routing Parameters

The routing parameters are described in the table below.

**Table 45-60: Routing Parameters**

| Parameter | Description |
|---|---|
| Hunt Group Table | |

| Parameter | Description |
|---|---|
| Hunt Group Table<br>`configure voip > gw hunt-or-trunk-group TrunkGroup`<br>[TrunkGroup] | Defines and activates Hunt Groups.<br>The format of the ini file table parameter is as follows:<br>[TrunkGroup]<br>FORMAT TrunkGroup_Index = TrunkGroup_TrunkGroupNum, TrunkGroup_FirstTrunkId, TrunkGroup_FirstBChannel, TrunkGroup_LastBChannel, TrunkGroup_FirstPhoneNumber, TrunkGroup_ProfileName, TrunkGroup_LastTrunkId, TrunkGroup_Module;<br>[\TrunkGroup]<br>For a description of the table, see Configuring Hunt Groups on page 385.<br>**Note:** Hunt Group ID 1 is denoted as 0 in the table. |
| **Hunt Group Settings Table** | |
| Hunt Group Settings<br>`configure voip > gw hunt-or-trunk-group trunk-group-setting`<br>[TrunkGroupSettings] | Defines the rules for channel allocation per Hunt Group.<br>The format of the ini file table parameter is as follows:<br>[TrunkGroupSettings]<br>FORMAT TrunkGroupSettings_Index = TrunkGroupSettings_TrunkGroupId, TrunkGroupSettings_ChannelSelectMode, TrunkGroupSettings_RegistrationMode, TrunkGroupSettings_GatewayName, TrunkGroupSettings_ContactUser, TrunkGroupSettings_ServingIPGroupName, TrunkGroupSettings_MWIInterrogationType, TrunkGroupSettings_TrunkGroupName, TrunkGroupSettings_UsedByRoutingServer;<br>[\TrunkGroupSettings]<br>For a description of the table, see "Configuring Hunt Group Settings" on page 386. |
| Channel Select Mode<br>`ch-select-mode`<br>[ChannelSelectMode] | Defines the method for allocating incoming IP-to-Tel calls to a channel. The parameter applies to the following:<br>▪ All Hunt Groups configured without a channel select mode in the Hunt Group Settings table (see "Configuring Hunt Group Settings" on page 386).<br>▪ All channels configured without a Hunt Group ID.<br>for all Hunt Groups channels that are configured without a Hunt Group ID,.<br>▪ **[0]** By Dest Phone Number<br>▪ **[1]** Cyclic Ascending (default)<br>▪ **[2]** Ascending<br>▪ **[3]** Cyclic Descending<br>▪ **[4]** Descending<br>▪ **[5]** Dest Number + Cyclic Ascending.<br>▪ **[6]** By Source Phone Number<br>▪ [7] Trunk Cyclic Ascending<br>▪ [8] Trunk & Channel Cyclic Ascending<br>▪ [9] Ring to Hunt Group<br>▪ [10] Select Trunk By Supplementary Service Table |

| Parameter | Description |
|---|---|
| | ▪ **[11]** Dest Number + Ascending<br>For a detailed description of the parameter's options, see "Configuring Hunt Group Settings" on page 386. |
| Default Destination Number<br>`dflt-dest-nb`<br>[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 Hunt Group table (see Configuring the Hunt Groups on page 385). The parameter is used as a starting number for the list of channels comprising all the device's Hunt Groups.<br>The default is 1000. |
| Source IP Address Input<br>`src-ip-addr-input`<br>[SourceIPAddressInput] | Determines which IP address the device uses to determine the source of incoming INVITE messages for IP-to-Tel routing.<br>▪ **[-1]** = (Default) Auto Decision - the parameter is automatically set to SIP Contact Header (1).<br>▪ **[0]** SIP Contact Header = The IP address in the Contact header of the incoming INVITE message is used.<br>▪ **[1]** Layer 3 Source IP = The actual IP address (Layer 3) from where the SIP packet was received is used. |
| Use Source Number As Display Name<br>`src-nb-as-disp-name`<br>[UseSourceNumberAsDisplayName] | Determines the use of Tel Source Number and Display Name for Tel-to-IP calls.<br>▪ **[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.<br>▪ **[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.<br>▪ **[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).<br>▪ **[3]** Original = Similar to option [2], except that the operation is done before regular calling number manipulation. |
| Use Display Name as Source Number<br>`disp-name-as-src-nb`<br>[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.<br>▪ [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.<br>▪ [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:<br>✓ If 'From: 100 <sip:200@201.202.203.204>' is received from the IP side, the outgoing source |

| Parameter | Description |
|---|---|
| | number (and display name) are set to "100" and Presentation is set to Allowed (0).<br>✓ 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).<br>▪ [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. |
| ENUM Resolution<br>`enum-service-domain`<br>[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.<br>The valid value is a string of up to 50 characters. The default is "e164.arpa".<br>**Note:** ENUM-based routing is configured in the Tel-to-IP Routing table using the "ENUM" string value as the destination address to denote the parameter's value. |
| Use Routing Table for Host Names and Profiles<br>`rte-tbl-4-host-names`<br>[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.<br>▪ **[0]** Disable = (Default) Don't use the Tel-to-IP Routing table.<br>▪ **[1]** Enable = Use the Tel-to-IP Routing table.<br>**Notes:**<br>▪ The parameter appears only if the 'Use Default Proxy' parameter is enabled.<br>▪ The domain name is used instead of a Proxy name or IP address in the INVITE SIP URI. |
| Tel to IP Routing Mode<br>onfigure voip > gw routing general-setting > tel2ip-rte-mode<br>[RouteModeTel2IP] | Determines whether to route Tel calls to an IP destination before or after manipulation of the destination number. This applies to Tel-to-IP routing rules configured in the Tel-to-IP Routing table.<br>▪ **[0]** Route calls before manipulation = Calls are routed before the number manipulation rules are applied (default).<br>▪ **[1]** Route calls after manipulation = Calls are routed after the number manipulation rules are applied.<br>**Notes:**<br>▪ The parameter is not applicable if outbound proxy routing is used.<br>▪ For number manipulation, see "Configuring Source/Destination Number Manipulation" on page 393.<br>▪ For configuring Tel-to-IP routing rules, see "Configuring Tel-to-IP Routing Rules" on page 409. |
| **Tel-to-IP Routing table** | |
| Tel-to-IP Routing table<br>`configure voip > gw routing tel2ip-routing`<br>[Prefix] | Defines Tel-to-IP routing rules for routing Tel-to-IP calls.<br>The format of the ini file table parameter is:<br>[PREFIX]<br>FORMAT PREFIX_Index = PREFIX_RouteName, |

| Parameter | Description |
|---|---|
| | PREFIX_DestinationPrefix, PREFIX_DestAddress, PREFIX_SourcePrefix, PREFIX_ProfileName, PREFIX_MeteringCode, PREFIX_DestPort, PREFIX_DestIPGroupName, PREFIX_TransportType, PREFIX_SrcTrunkGroupID, PREFIX_DestSIPInterfaceName, PREFIX_CostGroup, PREFIX_ForkingGroup, PREFIX_CallSetupRulesSetId, PREFIX_ConnectivityStatus;<br>[\PREFIX]<br>For a detailed description of the table, see "Configuring Tel-to-IP Routing Rules" on page 409. |
| **IP to Hunt Group Routing Table** | |
| IP to Hunt Group Routing<br>`configure voip > gw routing ip2tel-routing`<br>[PSTNPrefix] | Defines the routing of IP-to-Hunt Groups.<br>The format of the ini file table parameter is as follows:<br>[PSTNPrefix]<br>FORMAT PstnPrefix_Index = PstnPrefix_RouteName, PstnPrefix_DestPrefix, PstnPrefix_TrunkGroupId, PstnPrefix_SourcePrefix, PstnPrefix_SourceAddress, PstnPrefix_ProfileName, PstnPrefix_SrcIPGroupName, PstnPrefix_DestHostPrefix, PstnPrefix_SrcHostPrefix, PstnPrefix_SrcSIPInterfaceName, PstnPrefix_TrunkId, PstnPrefix_CallSetupRulesSetId, PstnPrefix_DestType;<br>[\PSTNPrefix]<br>For a detailed description of the table, see "Configuring IP-to-Hunt Group Routing Rules" on page 421. |
| IP to Tel Routing Mode<br>`ip2tel-rte-mode`<br>[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 393).<br>▪ **[0]** Route calls before manipulation = (Default) Calls are routed before the number manipulation rules are applied.<br>▪ **[1]** Route calls after manipulation = Calls are routed after the number manipulation rules are applied. |
| IP Security<br>`ip-security`<br>[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.<br>▪ **[0]** Disable = (Default) The device accepts all SIP calls.<br>▪ **[1]** Secure Incoming calls = The device accepts SIP calls (i.e., calls from the IP side) only from IP addresses that are configured in the Tel-to-IP Routing table or Proxy Set table, or IP addresses resolved from DNS servers from FQDN values configured in the Proxy Set table. All other incoming calls are rejected.<br>▪ **[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 |

| Parameter | Description |
|---|---|
| | and option [1] is that this option is concerned only about numerical IP addresses that are defined in the tables.<br>**Note:** If the 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. |
| Filter Calls to IP<br>`filter-calls-to-ip`<br>[FilterCalls2IP] | Enables filtering of Tel-to-IP calls when a Proxy Set is used.<br>▪ **[0]** Don't Filter = (Default) The device doesn't filter calls when using a proxy.<br>▪ **[1]** Filter = Filtering is enabled.<br>When the parameter is enabled and a proxy is used, the device first checks the Tel-to-IP Routing table before making a call through the proxy. If the number is not allowed (i.e., number isn't listed in the table or a call restriction routing rule of IP address 0.0.0.0 is applied), the call is released.<br>**Note:** When no proxy is used, the parameter must be disabled and filtering is according to the Tel-to-IP Routing table. |
| `etsi-diversion`<br>[EnableETSIDiversion] | Determines the method in which the Redirect Number is sent to the Tel side.<br>▪ [0] = (Default) Q.931 Redirecting Number Information Element (IE).<br>▪ [1] = ETSI DivertingLegInformation2 in a Facility IE. |
| Add CIC<br>`add-cic`<br>[AddCicAsPrefix] | Determines whether to add the Carrier Identification Code (CIC) as a prefix to the destination phone number for IP-to-Tel calls. When the 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 the parameter's value.<br>▪ **[0]** No (default)<br>▪ **[1]** Yes<br>For example, as a result of receiving the below INVITE, the destination number after number manipulation is cic+167895550001:<br>INVITE sip:5550001;cic=+16789@172.18.202.60:5060;user=phone SIP/2.0<br>**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. |
| [FaxReroutingMode] | Enables the re-routing of incoming Tel-to-IP calls that are identified as fax calls. If a CNG tone is detected on the Tel side of a Tel-to-IP call, the device adds the string, "FAX" as a prefix to the destination number before routing and manipulation. A routing rule in the Tel-to-IP Routing table having the value "FAX" (case-sensitive) as the destination number is then used to re-route the call to a fax destination |

| Parameter | Description |
|---|---|
| | and the destination number manipulation mechanism is used to remove the "FAX" prefix before sending the fax, if required. If the initial INVITE used to establish the voice call (not fax) was already sent, a CANCEL (if not connected yet) or a BYE (if already connected) is sent to release the voice call.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Rerouting without Delay = Upon detection of a CNG tone, the device immediately releases the call of the initial INVITE and then sends a new INVITE to a specific IP Group or fax server according to the Tel-to-IP Routing table. To enable this feature, set the CNGDetectorMode parameter to 2 and the IsFaxUsed parameter to 1, 2, or 3.<br>**Note:** The parameter has replaced the EnableFaxRerouting parameter. For backward compatibility, the EnableFaxRerouting parameter set to 1 is equivalent to the FaxReroutingMode parameter set to 1. |
| [FaxReroutingDelay] | Defines the maximum time interval (in seconds) that the device waits for CNG detection before re-routing calls identified as fax calls to fax destinations (terminating fax machine).<br>The valid value range is 1-10. The default is 5. |
| **Call Forking Parameters** | |
| Forking Handling Mode<br>`forking-handling`<br>[ForkingHandlingMode] | Determines how the device handles the receipt of multiple SIP 18x forking responses for Tel-to-IP calls. The forking 18x response is the response with a different SIP to-tag than the previous 18x response. These responses are typically generated (initiated) by Proxy / Application servers that perform call forking, sending the device's originating INVITE (received from SIP clients) to several destinations, using the same Call ID.<br>▪ **[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.<br>▪ **[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.<br>**Note:** Regardless of the parameter setting, once a SIP 200 OK response is received, the device uses the RTP information and re-opens the voice stream, if necessary. |
| Forking Timeout<br>`forking-timeout`<br>[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 |

| Parameter | Description |
|---|---|
| | 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.<br>The number of supported forking calls per channel is 20. In other words, for an INVITE message, the device can receive up to 20 forking responses from the Proxy server.<br>The valid range is 0 to 30. The default is 30. |
| Tel2IP Call Forking Mode<br>`tel2ip-call-forking-mode`<br>[Tel2IPCallForkingMode] | Enables Tel-to-IP call forking, whereby a Tel call can be routed to multiple IP destinations.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Note:** Once enabled, routing rules must be assigned Forking Groups in the Tel-to-IP Routing table. |
| `configure voip/sip-definition advanced-settings/forking-delay-time-invite`<br>[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.<br>The valid value range is 0 to 40. The default is 0 (i.e., sends immediately). |
| **Gateway Routing Policy Table** | |
| Gateway Routing Policy<br>`configure voip > gw routing gw-routing-policy`<br>[GWRoutingPolicy] | Edits the Gateway Routing Policy.<br>The format of the ini file table parameter is as follows:<br>[ GwRoutingPolicy ]<br>FORMAT GwRoutingPolicy_Index = GwRoutingPolicy_Name, GwRoutingPolicy_LCREnable, GwRoutingPolicy_LCRAverageCallLength, GwRoutingPolicy_LCRDefaultCost, GwRoutingPolicy_LdapServersGroupName;<br>[ \GwRoutingPolicy ]<br>For a description of the table, see "Configuring a Gateway Routing Policy Rule" on page 426. |

### 45.10.11 IP Connectivity Parameters

The IP connectivity parameters are described in the table below.

**Table 45-61: IP Connectivity Parameters**

| Parameter | Description |
|---|---|
| Enable Alt Routing Tel to IP<br>`alt-routing-tel2ip`<br>[AltRoutingTel2IPEnable] | Enables the Alternative Routing feature for Tel-to-IP calls.<br>▪ **[0]** Disable = (Default) Disables the Alternative Routing feature.<br>▪ **[1]** Enable = Enables the Alternative Routing feature.<br>▪ **[2]** Status Only = The Alternative Routing feature is disabled, but read-only information on the QoS of the destination IP addresses is provided.<br>**Note:** If the parameter is enabled, the Busy Out feature (see EnableBusyOut parameter) does not function with the Proxy Set keep-alive mechanism. To use the Busy Out feature with the Proxy Set keep-alive mechanism (for IP Groups), disable the parameter. |
| Alt Routing Tel to IP Mode<br>`alt-rte-tel2ip-mode`<br>[AltRoutingTel2IPMode] | Determines the IP Connectivity event(s) reason for triggering Alternative Routing.<br>▪ **[0]** None = Alternative routing is not used.<br>▪ **[1]** Connectivity = Alternative routing is performed if SIP OPTIONS message to the initial destination fails (determined according to the AltRoutingTel2IPConnMethod parameter).<br>▪ **[2]** QoS = Alternative routing is performed if poor QoS is detected.<br>▪ **[3]** Both = (Default) Alternative routing is performed if either SIP OPTIONS to initial destination fails, poor QoS is detected, or the DNS host name is not resolved.<br>**Notes:**<br>▪ 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.<br>▪ To receive quality information (displayed in the 'Quality Status' and 'Quality Info.' fields in "Viewing IP Connectivity" on page 568) per destination, the parameter must be set to 2 or 3. |
| Alt Routing Tel to IP Connectivity Method<br>`alt-rte-tel2ip-method`<br>[AltRoutingTel2IPConnMethod] | Determines the method used by the device for periodically querying the connectivity status of a destination IP address.<br>▪ **[0]** ICMP Ping = (Default) Internet Control Message Protocol (ICMP) ping messages.<br>▪ **[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.<br>**Note:** ICMP Ping is currently not supported for the IP Connectivity feature. |
| Alt Routing Tel to IP Keep Alive Time<br>`alt-rte-tel2ip-keep-alive`<br>[AltRoutingTel2IPKeepAliveTime] | Defines the time interval (in seconds) between SIP OPTIONS Keep-Alive messages used for the IP Connectivity application.<br>The valid range is 5 to 2,000,000. The default is 60. |

| Parameter | Description |
|---|---|
| Max Allowed Packet Loss for Alt Routing [%]<br>`mx-pkt-loss-4-alt-rte`<br>[IPConnQoSMaxAllowedPL] | Defines the packet loss (in percentage) at which the IP connection is considered a failure and Alternative Routing mechanism is activated.<br>The default is 20%. |
| Max Allowed Delay for Alt Routing<br>`mx-all-dly-4-alt-rte`<br>[IPConnQoSMaxAllowedDelay] | Defines the transmission delay (in msec) at which the IP connection is considered a failure and the Alternative Routing mechanism is activated.<br>The range is 100 to 10,000. The default is 250. |

## 45.10.12 Alternative Routing Parameters

The alternative routing parameters are described in the table below.

**Table 45-62: Alternative Routing Parameters**

| Parameter | Description |
|---|---|
| 3xx Use Alt Route Reasons<br>`configure voip/sip-definition advanced-settings/3xx-use-alt-route`<br>[UseAltRouteReasonsFor3xx] | Defines the handling of received SIP 3xx responses regarding call redirection to listed contacts in the Contact header.<br>▪ **[0]** No = (Default) Upon receipt of a 3xx response, the device tries each contact, one by one, listed in the Contact headers, until a successful destination is found. However, if a contact responds with a 486 or 600, the device does not try to redirect the call to next contact, and drops the call.<br>▪ **[1]** No if 6xx = Upon receipt of a 3xx response, the device tries each contact, one by one, listed in the Contact headers. However, if a 6xx Global Failure response is received during this process (e.g., 600 Busy Everywhere) the device does not try to redirect the call to the next contact, and drops the call.<br>▪ **[2]** Yes = Upon receipt of a 3xx response, the device redirects the call to the first contact listed in the Contact header. If the contact responds with a SIP response that is defined in the Reasons for Tel-to-IP Alternative Routing table, the device tries to redirect the call to the next contact, and so on. If a contact responds with a response that is not configured in the table, the device does not try to redirect the call to the next contact, and drops the call. |
| Redundant Routing Mode<br>`redundant-routing-m`<br>[RedundantRoutingMode] | Determines the type of redundant routing mechanism when a call can't be completed using the main route.<br>▪ **[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.<br>▪ **[1]** Routing Table = (Default) Internal routing table is used to locate a redundant route.<br>▪ **[2]** Proxy = Proxy list is used to locate a redundant route.<br>**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 |

| Parameter | Description |
|---|---|
| | 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).<br>▪ **[0]** = Disable (default)<br>▪ **[1]** = Enable |
| Alternative Routing Tone Duration<br>`alt-rte-tone-duration`<br>[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).<br>The valid range is 0 to 20,000. The default is 0 (i.e., no tone is played).<br>**Notes:**<br>▪ The parameter is applicable only to FXS interfaces.<br>▪ The parameter is applicable only to Tel-to-IP alternative routing based on SIP responses (see Alternative Routing Based on SIP Responses on page 431). |
| **Reasons for Alternative Tel-to-IP Routing Table** | |
| Reasons for Alternative Routing<br>`configure voip > gw manipulations general-setting alt-route-cause-tel2ip`<br>[AltRouteCauseTel2IP] | Defines 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 table (if a Proxy is not used) or used as a redundant Proxy (you need to set the parameter RedundantRoutingMode to 2). The release reason for Tel-to-IP calls is provided in SIP 4xx, 5xx, and 6xx response codes.<br>The format of the ini file table parameter is as follows:<br>[AltRouteCauseTel2IP]<br>FORMAT AltRouteCauseTel2IP_Index = AltRouteCauseTel2IP_ReleaseCause;<br>[\AltRouteCauseTel2IP]<br>For example:<br>AltRouteCauseTel2IP 0 = 486; (Busy Here)<br>AltRouteCauseTel2IP 1 = 480; (Temporarily Unavailable)<br>AltRouteCauseTel2IP 2 = 408; (No Response)<br>For a detailed description of the table, see "Alternative Routing Based on SIP Responses" on page 431. |
| **Reasons for Alternative IP-to-Tel Routing Table** | |
| Reasons for Alternative IP-to-Tel Routing<br>`configure voip > gw manipulations general-setting alt-route-cause-ip2tel`<br>[AltRouteCauseIP2Tel] | Defines call failure reason values received from the Tel side (in Q.931 presentation). If a call is released as a result of one of these reasons, the device attempts to locate an alternative Hunt Group for the call in the IP to Hunt Group Routing table.<br>The format of the ini file table parameter is as follows:<br>[AltRouteCauseIP2Tel]<br>FORMAT AltRouteCauseIP2Tel_Index = AltRouteCauseIP2Tel_ReleaseCause;<br>[\AltRouteCauseIP2Tel]<br>For example:<br>AltRouteCauseIP2Tel 0 = 3 (No Route to Destination)<br>AltRouteCauseIP2Tel 1 = 1 (Unallocated Number) |

| Parameter | Description |
|---|---|
| | AltRouteCauseIP2Tel 2 = 17 (Busy Here)<br>AltRouteCauseIP2Tel 2 = 27 (Destination Out of Order)<br>For a detailed description of the table, see "Alternative Routing to Hunt upon Q.931 Call Release Cause Code" on page 435. |
| **Forward On Busy Hunt Destination Table** | |
| Forward On Busy Hunt Destination<br>`configure voip > gw routing fwd-on-bsy-trk-dest`<br>[ForwardOnBusyTrunkDest] | Defines the Forward On Busy Hunt Destination table. This table allows you to define an alternative IP destination if a Hunt Group is busy for IP-to-Tel calls.<br>The format of the ini file table parameter is as follows:<br>[ForwardOnBusyTrunkDest]<br>FORMAT ForwardOnBusyTrunkDest_Index = ForwardOnBusyTrunkDest_TrunkGroupId, ForwardOnBusyTrunkDest_ForwardDestination;<br>[\ForwardOnBusyTrunkDest]<br>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 Hunt Group ID 2 is unavailable:<br>ForwardOnBusyTrunkDest 1 = 2, 112@10.13.4.12:5060;transport=tcp;<br>For a detailed description of the table, see "Alternative Routing to IP Destination upon Busy Hunt" on page 436. |

## 45.10.13 Number Manipulation Parameters

The number manipulation parameters are described in the table below.

**Table 45-63: Number Manipulation Parameters**

| Parameter | Description |
|---|---|
| Use EndPoint Number As Calling Number Tel2IP<br>`epn-as-cpn-tel2ip`<br>[UseEPNumAsCallingNumTel2IP] | Enables the use of the B-channel number as the calling number (sent in the From field of the INVITE) instead of the number received in the Q.931 Setup message, for Tel-to-IP calls.<br>▪ [0] Disable (default)<br>▪ [1] Enable<br>For example, if the incoming calling party number in the Q.931 Setup message is "12345" and the B-channel number is 17, then the outgoing INVITE From header is set to "17" instead of "12345".<br>**Note:** When enabled, this feature is applied before routing and manipulation on the source number. |
| Use EndPoint Number As Calling Number IP2Tel<br>`epn-as-cpn-ip2tel`<br>[UseEPNumAsCallingNumIP2Tel] | Enables the use of the B-channel number as the calling party number (sent in the Q.931 Setup message) instead of the number received in the From header of the INVITE, for IP-to-Tel calls.<br>▪ [0] Disable (default)<br>▪ [1] Enable<br>For example, if the incoming INVITE From header contains "12345" and the destined B-channel number is 17, then the outgoing calling party number in the Q.931 Setup message is set to "17" instead of "12345". |

| Parameter | Description |
|---|---|
| | **Note:** When enabled, this feature is applied after routing and manipulation on the source number (i.e., just before sending to the Tel side). |
| Copy Destination Number to Redirect Number<br>`cp-dst-nb-2-redir-nb`<br>[CopyDest2RedirectNumber] | Enables the device to copy 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.<br>▪ **[0]** Don't copy = (Default) Disable.<br>▪ **[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.<br>▪ **[2]** Copy before phone number manipulation = Copies the called number before manipulation. The device first copies the original called number to the SIP Diversion header, and then performs Tel-to-IP destination phone number manipulation. Therefore, this allows you to have different numbers for the called (i.e., SIP To header) and redirect (i.e., SIP Diversion header) numbers. |
| Add Trunk Group ID as Prefix<br>`trkgrpid-prefix`<br>[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.<br>▪ **[0]** No = (Default) Don't add Hunt Group ID as prefix.<br>▪ **[1]** Yes = Add Hunt Group ID as prefix to called number.<br>**Notes:**<br>▪ This option can be used to define various routing rules.<br>▪ To use this feature, you must configure the Hunt Group IDs (see Configuring Hunt Groups on page 385). |
| Add Trunk ID as Prefix<br>`trk-id-as-prefix`<br>[AddPortAsPrefix] | Determines whether or not the slot number/port number/Hunt ID is added as a prefix to the called (destination) number for Tel-to-IP calls.<br>▪ **[0]** No (Default)<br>▪ **[1]** Yes<br>If enabled, the device adds the following prefix to the called phone number: slot number (a single digit in the range of 1 to 6) and port number/Hunt ID (single digit in the range 1 to 8). For example, for the first channel located in the first slot, the number "11" is added as the prefix.<br>This option can be used to define various routing rules. |
| Add Trunk Group ID as Prefix to Source<br>`trkgrpid-pref2source`<br>[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).<br>▪ **[0]** No (default)<br>▪ **[1]** Yes |
| IP to Tel Remove Routing Table Prefix<br>`ip2tel-rmv-rte-tbl`<br>[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 Rules" on page 421) from the destination number for IP-to-Tel calls, before sending it to the Tel.<br>▪ **[0]** No (default) |

| Parameter | Description |
|---|---|
| | ▪ **[1]** Yes<br>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.<br>**Notes:**<br>▪ The 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).<br>▪ 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.<br>▪ **[0]** = (Default) Disabled<br>▪ **[1]** = Swap calling and called numbers<br>**Note:** The parameter can also be configured for a Tel Profile (in the Tel Profile table). |
| Add Number Plan and Type to RPI Header<br>`np-n-type-to-rpi-hdr`<br>[AddTON2RPI] | Determines whether the TON/PLAN parameters are included in the Remote-Party-ID (RPID) header.<br>▪ **[0]** No<br>▪ **[1]** Yes (default)<br>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. |
| Source Manipulation Mode<br>`src-manipulation`<br>[SourceManipulationMode] | Determines the SIP headers containing the source number after manipulation:<br>▪ **[0]** = (Default) The SIP From and P-Asserted-Identity headers contain the source number after manipulation.<br>▪ **[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** | |
| `configure voip > gw manipulations calling-name-map-ip2tel`<br>[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:<br>[ CallingNameMapIp2Tel ]<br>FORMAT CallingNameMapIp2Tel_Index = CallingNameMapIp2Tel_ManipulationName, CallingNameMapIp2Tel_DestinationPrefix, CallingNameMapIp2Tel_SourcePrefix, CallingNameMapIp2Tel_CallingNamePrefix, CallingNameMapIp2Tel_SourceAddress, CallingNameMapIp2Tel_RemoveFromLeft, CallingNameMapIp2Tel_RemoveFromRight, |

| Parameter | Description |
|---|---|
| | CallingNameMapIp2Tel_LeaveFromRight,<br>CallingNameMapIp2Tel_Prefix2Add,<br>CallingNameMapIp2Tel_Suffix2Add;<br>[ \CallingNameMapIp2Tel ]<br>For a detailed description of the table, see "Configuring SIP Calling Name Manipulation" on page 400. |
| **Calling Name Manipulations Tel-to-IP Table** | |
| `configure voip > gw manipulations calling-name-map-tel2ip`<br>[CallingNameMapTel2Ip] | Defines rules for manipulating the calling name (caller ID) for Tel-to-IP calls. This can include modifying or removing the calling name.<br>[ CallingNameMapTel2Ip ]<br>FORMAT CallingNameMapTel2Ip_Index =<br>CallingNameMapTel2Ip_ManipulationName,<br>CallingNameMapTel2Ip_DestinationPrefix,<br>CallingNameMapTel2Ip_SourcePrefix,<br>CallingNameMapTel2Ip_CallingNamePrefix,<br>CallingNameMapTel2Ip_SrcTrunkGroupID,<br>CallingNameMapTel2Ip_RemoveFromLeft,<br>CallingNameMapTel2Ip_RemoveFromRight,<br>CallingNameMapTel2Ip_LeaveFromRight,<br>CallingNameMapTel2Ip_Prefix2Add,<br>CallingNameMapTel2Ip_Suffix2Add;<br>[ \CallingNameMapTel2Ip ]<br>For a detailed description of the table, see "Configuring SIP Calling Name Manipulation" on page 400. |
| **Destination Phone Number Manipulation for IP-to-Tel Calls Table** | |
| Destination Phone Number Manipulation Table for IP-to-Tel Calls<br>`configure voip > gw manipulations NumberMapIp2Tel2`<br>[NumberMapIP2Tel] | This table parameter manipulates the destination number of IP-to-Tel calls. The format of the ini file table parameter is as follows:<br>[NumberMapIp2Tel]<br>FORMAT NumberMapIp2Tel_Index =<br>NumberMapIp2Tel_ManipulationName,<br>NumberMapIp2Tel_DestinationPrefix,<br>NumberMapIp2Tel_SourcePrefix,<br>NumberMapIp2Tel_SourceAddress,<br>NumberMapIp2Tel_NumberType, NumberMapIp2Tel_NumberPlan,<br>NumberMapIp2Tel_RemoveFromLeft,<br>NumberMapIp2Tel_RemoveFromRight,<br>NumberMapIp2Tel_LeaveFromRight,<br>NumberMapIp2Tel_Prefix2Add, NumberMapIp2Tel_Suffix2Add,<br>NumberMapIp2Tel_IsPresentationRestricted;<br>[\NumberMapIp2Tel]<br>For a detailed description of the table, see "Configuring Source/Destination Number Manipulation" on page 393. |
| `prfm-ip-to-tel-dst-map`<br>[PerformAdditionalIP2TELDestinationManipulation] | 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).<br>▪ **[0]** = Disable (default)<br>▪ **[1]** = Enable |

| Parameter | Description |
|---|---|
| **Destination Phone Number Manipulation for Tel-to-IP Calls Table** | |
| Destination Phone Number Manipulation Table for Tel-to-IP Calls<br>`configure voip > gw manipulations NumberMapTel2Ip`<br>[NumberMapTel2IP] | This table parameter manipulates the destination number of Tel-to-IP calls. The format of the ini file table parameter is as follows:<br>[NumberMapTel2Ip]<br>FORMAT NumberMapTel2Ip_Index = NumberMapTel2Ip_ManipulationName, NumberMapTel2Ip_DestinationPrefix, NumberMapTel2Ip_SourcePrefix, NumberMapTel2Ip_SourceAddress, NumberMapTel2Ip_NumberType, NumberMapTel2Ip_NumberPlan, NumberMapTel2Ip_RemoveFromLeft, NumberMapTel2Ip_RemoveFromRight, NumberMapTel2Ip_LeaveFromRight, NumberMapTel2Ip_Prefix2Add, NumberMapTel2Ip_Suffix2Add, NumberMapTel2Ip_IsPresentationRestricted, NumberMapTel2Ip_SrcTrunkGroupID, NumberMapTel2Ip_ SrcIPGroupID;<br>[\NumberMapTel2Ip]<br>For a detailed description of the table, see "Configuring Source/Destination Number Manipulation" on page 393. |
| **Source Phone Number Manipulation for IP-to-Tel Calls Table** | |
| Source Phone Number Manipulation Table for IP-to-Tel Calls<br>`configure voip > gw manipulations SourceNumberMapIp2Tel`<br>[SourceNumberMapIP2Tel] | The parameter table manipulates the source number for IP-to-Tel calls. The format of the ini file table parameter is as follows:<br>[SourceNumberMapIp2Tel]<br>FORMAT SourceNumberMapIp2Tel_Index = SourceNumberMapIp2Tel_ManipulationName, SourceNumberMapIp2Tel_DestinationPrefix, SourceNumberMapIp2Tel_SourcePrefix, SourceNumberMapIp2Tel_SourceAddress, SourceNumberMapIp2Tel_NumberType, SourceNumberMapIp2Tel_NumberPlan, SourceNumberMapIp2Tel_RemoveFromLeft, SourceNumberMapIp2Tel_RemoveFromRight, SourceNumberMapIp2Tel_LeaveFromRight, SourceNumberMapIp2Tel_Prefix2Add, SourceNumberMapIp2Tel_Suffix2Add, SourceNumberMapIp2Tel_IsPresentationRestricted;<br>[\SourceNumberMapIp2Tel]<br>For a detailed description of the table, see "Configuring Source/Destination Number Manipulation" on page 393. |
| `prfm-ip-to-tel-src-map`<br>[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).<br>▪ **[0]** = Disable (default)<br>▪ **[1]** = Enable |
| **Source Phone Number Manipulation for Tel-to-IP Calls Table** | |

| Parameter | Description |
|---|---|
| Source Phone Number Manipulation Table for Tel-to-IP Calls<br>`configure voip > gw manipulations SourceNumberMapTel2Ip`<br>[SourceNumberMapTel2IP] | This table parameter manipulates the source phone number for Tel-to-IP calls. The format of the ini file table parameter is as follows:<br>[SourceNumberMapTel2Ip]<br>FORMAT SourceNumberMapTel2Ip_Index = SourceNumberMapTel2Ip_ManipulationName, SourceNumberMapTel2Ip_DestinationPrefix, SourceNumberMapTel2Ip_SourcePrefix, SourceNumberMapTel2Ip_NumberType, SourceNumberMapTel2Ip_NumberPlan, SourceNumberMapTel2Ip_RemoveFromLeft, SourceNumberMapTel2Ip_RemoveFromRight, SourceNumberMapTel2Ip_LeaveFromRight, SourceNumberMapTel2Ip_Prefix2Add, SourceNumberMapTel2Ip_Suffix2Add, SourceNumberMapTel2Ip_IsPresentationRestricted, SourceNumberMapTel2Ip_SrcTrunkGroupID;<br>[\SourceNumberMapTel2Ip]<br>For a detailed description of the table, see "Configuring Source/Destination Number Manipulation" on page 393. |
| **Redirect Number Tel-to-IP Table** | |
| Redirect Number Tel -> IP<br>`configure voip > gw manipulations redirect-number-map-tel2ip`<br>[RedirectNumberMapTel2IP] | This table parameter manipulates the Redirect Number for Tel-to-IP calls. The format of the ini file table parameter is as follows:<br>[RedirectNumberMapTel2Ip]<br>FORMAT RedirectNumberMapTel2Ip_Index = RedirectNumberMapTel2Ip_ManipulationName, RedirectNumberMapTel2Ip_DestinationPrefix, RedirectNumberMapTel2Ip_RedirectPrefix, RedirectNumberMapTel2Ip_NumberType, RedirectNumberMapTel2Ip_NumberPlan, RedirectNumberMapTel2Ip_RemoveFromLeft, RedirectNumberMapTel2Ip_RemoveFromRight, RedirectNumberMapTel2Ip_LeaveFromRight, RedirectNumberMapTel2Ip_Prefix2Add, RedirectNumberMapTel2Ip_Suffix2Add, RedirectNumberMapTel2Ip_IsPresentationRestricted, RedirectNumberMapTel2Ip_SrcTrunkGroupID;<br>[\RedirectNumberMapTel2Ip]<br>For a description of the table, see "Configuring Redirect Number Manipulation" on page 403. |
| **Phone Context Table** | |
| Phone Context Table<br>`configure voip > gw manipulations phone-context-table`<br>[PhoneContext] | Defines the Phone Context table. The parameter maps NPI and TON to the SIP 'phone-context' parameter, and vice versa.<br>The format for the parameter is as follows:<br>[PhoneContext]<br>FORMAT PhoneContext_Index = PhoneContext_Npi, PhoneContext_Ton, PhoneContext_Context;<br>[\PhoneContext]<br>For example:<br>PhoneContext 0 = 0,0,unknown.com<br>PhoneContext 1 = 1,1,host.com<br>PhoneContext 2 = 9,1,na.e164.host.com |

| Parameter | Description |
|---|---|
| | For a detailed description of the table, see "Configuring NPI/TON-SIP Phone-Context Mapping Rules" on page 406. |
| Add Phone Context As Prefix<br>`add-ph-cntxt-as-pref`<br>[AddPhoneContextAsPrefix] | Determines whether the received Phone-Context parameter is added as a prefix to the outgoing Called and Calling numbers.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable |

## 45.11 IP Media Parameters

The IP media parameters are described in the table below.

**Table 45-64: IP Media Parameters**

| Parameter | Description |
|---|---|
| IPMedia Detectors<br>`IPM-detectors-enable`<br>[EnableDSPIPMDetectors] | Enables the device's DSP detectors for detection features such as AMD.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ The DSP Detectors feature is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see "Software License Key" on page 510.<br>▪ When enabled (1), the number of available channels is reduced. |
| Number of Media Channels<br>`media-channels`<br>[MediaChannels] | Defines the maximum number of DSP channels allocated for various functionalities such as transcoding, IP conferencing.<br>The default is 0.<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ Other DSP channels can be used for PSTN interfaces. |
| Conferencing Parameters | |
| Conference ID<br>[ConferenceID] | Defines the Conference Identification string.<br>The valid value is a string of up to 16 characters. The default is "conf".<br>**Note:** To join a conference, the INVITE URI must include the Conference ID string preceded by the number of the participants in the conference and terminated by a unique number. For example:<br>INVITE sip:4conf1234@10.1.10.10<br>INVITE messages with the same URI join the same conference. |
| Automatic Gain Control (AGC) Parameters | |

| Parameter | Description |
|---|---|
| Enable AGC<br>`AGC-enable`<br>[EnableAGC] | Enables the AGC mechanism. The AGC mechanism adjusts the level of the received signal to maintain a steady (configurable) volume level.<br>▪ [0] Disable (default)<br>▪ [1] Enable<br>**Notes:**<br>▪ The parameter can also be configured for a Tel Profile (in the Tel Profile table).<br>▪ For a description of AGC, see Automatic Gain Control (AGC) on page 206. |
| AGC Slope<br>`AGC-gain-slope`<br>[AGCGainSlope] | Determines the AGC convergence rate:<br>▪ [0] 0 = 0.25 dB/sec<br>▪ [1] 1 = 0.50 dB/sec<br>▪ [2] 2 = 0.75 dB/sec<br>▪ [3] 3 = 1.00 dB/sec (default)<br>▪ [4] 4 = 1.25 dB/sec<br>▪ [5] 5 = 1.50 dB/sec<br>▪ [6] 6 = 1.75 dB/sec<br>▪ [7] 7 = 2.00 dB/sec<br>▪ [8] 8 = 2.50 dB/sec<br>▪ [9] 9 = 3.00 dB/sec<br>▪ [10] 10 = 3.50 dB/sec<br>▪ [11] 11 = 4.00 dB/sec<br>▪ [12] 12 = 4.50 dB/sec<br>▪ [13] 13 = 5.00 dB/sec<br>▪ [14] 14 = 5.50 dB/sec<br>▪ [15] 15 = 6.00 dB/sec<br>▪ [16] 16 = 7.00 dB/sec<br>▪ [17] 17 = 8.00 dB/sec<br>▪ [18] 18 = 9.00 dB/sec<br>▪ [19] 19 = 10.00 dB/sec<br>▪ [20] 20 = 11.00 dB/sec<br>▪ [21] 21 = 12.00 dB/sec<br>▪ [22] 22 = 13.00 dB/sec<br>▪ [23] 23 = 14.00 dB/sec<br>▪ [24] 24 = 15.00 dB/sec<br>▪ [25] 25 = 20.00 dB/sec<br>▪ [26] 26 = 25.00 dB/sec<br>▪ [27] 27 = 30.00 dB/sec<br>▪ [28] 28 = 35.00 dB/sec<br>▪ [29] 29 = 40.00 dB/sec<br>▪ [30] 30 = 50.00 dB/sec<br>▪ [31] 31 = 70.00 dB/sec |
| AGC Redirection<br>`AGC-redirection`<br>[AGCRedirection] | Determines the AGC direction.<br>▪ [0] 0 = (Default) AGC works on signals from the TDM side.<br>▪ [1] 1 = AGC works on signals from the IP side. |

| Parameter | Description |
|---|---|
| AGC Target Energy<br>`AGC-target-energy`<br>[AGCTargetEnergy] | Defines the signal energy value (dBm) that the AGC attempts to attain.<br>The valid range is 0 to -63 dBm. The default is -19 dBm. |
| AGC Minimum Gain<br>`AGC-min-gain`<br>[AGCMinGain] | Defines the minimum gain (in dB) by the AGC when activated.<br>The range is 0 to -31. The default is -20.<br>**Note:** For the parameter to take effect, a device reset is required. |
| AGC Maximum Gain<br>`AGC-max-gain`<br>[AGCMaxGain] | Defines the maximum gain (in dB) by the AGC when activated.<br>The range is 0 to 18. The default is 15.<br>**Note:** For the parameter to take effect, a device reset is required. |
| Disable Fast Adaptation<br>`AGC-disable-fast-adaptation`<br>[AGCDisableFastAdaptation] | Enables the AGC Fast Adaptation mode.<br>▪ [0] = Disable (default)<br>▪ [1] = Enable<br>**Note:** For the parameter to take effect, a device reset is required. |
| **Answering Machine Detector (AMD) Parameters**<br>For more information on AMD, see "Answering Machine Detection (AMD)" on page 203. | |
| Answer Machine Detector Sensitivity Parameter Suit<br>`amd-sensitivity-parameter-suit`<br>[AMDSensitivityParameterSuit] | Global parameter that defines the AMD Parameter Suite to use. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_AMDSensitivityParameterSuit). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| Answer Machine Detector Sensitivity Level<br>`amd-sensitivity-level`<br>[AMDSensitivityLevel] | Global parameter that defines the AMD detection sensitivity level of the selected AMD Parameter Suite. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_AMDSensitivityLevel). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| AMD Sensitivity File<br>[AMDSensitivityFileName] | Defines the name of the AMD Sensitivity file that contains the AMD Parameter Suites.<br>**Notes:**<br>▪ This file must be in binary format (.dat). You can use the DConvert utility to convert the original file format from XML to .dat.<br>▪ You can load this file using the Web interface (see "Loading Auxiliary Files" on page 493). |

| Parameter | Description |
| --- | --- |
| [AMDSensitivityFileUrl] | Defines the URL path to the AMD Sensitivity file for downloading from a remote server. |
| [AMDMinimumVoiceLength] | Defines the AMD minimum voice activity detection duration (in 5-ms units). Voice activity duration below this threshold is ignored and considered as non-voice.<br>The valid value range is 10 to 100. The default is 42 (i.e., 210 ms). |
| [AMDMaxGreetingTime] | Global parameter that defines the maximum duration that the device can take to detect a greeting message. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_AMDMaxGreetingTime). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| [AMDMaxPostGreetingSilenceTime] | Global parameter that defines the maximum duration of silence from after the greeting time is over (defined by AMDMaxGreetingTime) until the device's AMD decision. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_AMDMaxPostSilenceGreetingTime). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| [AMDTimeout] | Defines the timeout (in msec) between receiving Connect messages from the Tel side  and sending AMD results.<br>The valid range is 1 to 30,000. The default is 2,000 (i.e., 2 seconds). |
| AMD Beep Detection Mode<br>`amd-beep-detection`<br>[AMDBeepDetectionMode] | Determines the AMD beep detection mode. This mode detects the beeps played at the end of an answering machine message, by using the X-Detect header extension. The device sends a SIP INFO message containing the field values Type=AMD and SubType=Beep. This feature allows users of certain third-party, Application server to leave a voice message after an answering machine plays the “beep”.<br>▪ [0] Disabled (default)<br>▪ [1] Start After AMD<br>▪ [2] Start Immediately |
| Answer Machine Detector Beep Detection Timeout<br>`amd-beep-detection-timeout`<br>[AMDBeepDetectionTimeout] | Defines the AMD beep detection timeout (i.e., the duration that the beep detector functions from when detection is initiated). This is used for detecting beeps at the end of an answering machine message.<br>The valid value is in units of 100 milliseconds, from 0 to 1638. The default is 200 (i.e., 20 seconds). |
| Answer Machine Detector Beep Detection Sensitivity | Defines the AMD beep detection sensitivity for detecting beeps at the end of an answering machine message. |

| Parameter | Description |
|---|---|
| amd-beep-detection-sensitivity<br>[AMDBeepDetectionSensitivity] | The valid value is 0 to 3, where 0 (default) is the least sensitive. |
| AMD mode<br>amd-mode<br>[AMDmode] | Global parameter that enables the device to disconnect the IP-to-Tel call upon detection of an answering machine on the Tel side. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_AmdMode). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 366.<br>**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| Energy Detector Parameters | |
| Enable Energy Detector<br>energy-detector-enable<br>[EnableEnergyDetector] | Enables the Energy Detector feature. This feature generates events (notifications) when the signal received from the PSTN is higher or lower than a user-defined threshold (defined by the EnergyDetectorThreshold parameter).<br>▪ [0] Disable (default)<br>▪ [1] Enable |
| Energy Detector Quality Factor<br>energy-detector-sensitivity<br>[EnergyDetectorQualityFactor] | Defines the Energy Detector's sensitivity level.<br>The valid range is 0 to 10, where 0 is the lowest sensitivity and 10 the highest sensitivity. The default is 4. |
| Energy Detector Threshold<br>energy-detector-threshold<br>[EnergyDetectorThreshold] | Defines the Energy Detector's threshold. A signal below or above this threshold invokes an 'Above' or 'Below' event.<br>The threshold is calculated as follows:<br>Actual Threshold = -44 dBm + (EnergyDetectorThreshold * 6)<br>The valid value range is 0 to 7. The default is 3 (i.e., -26 dBm). |

# 45.12 Services

## 45.12.1 RADIUS and LDAP Parameters

### 45.12.1.1 General Parameters

The general RADIUS and LDAP parameters are described in the table below.

**Table 45-65: General RADIUS and LDAP Parameters**

| Parameter | Description |
|---|---|
| Use Local Users Database | Defines when the device uses its local management-users database (Web Users table) or an LDAP/RADIUS server for authenticating the login credentials (username-password) of users |

| Parameter | Description |
|---|---|
| `configure system > mgmt-auth > use-local-users-db`<br>[MgmtUseLocalUsersDatabase] | when logging into the device's management interface (e.g., Web or CLI).<br>▪ **[0]** When No Auth Server Defined = (Default) The device authenticates the users usng the Web Users table in the following scenarios:<br>✓ If no LDAP/RADIUS server is configured.<br>✓ If an LDAP/RADIUS server is configured, but connectivity with the server is down. If there is connectivity with the server, the device uses the server to authenticate the user.<br>▪ **[1]** Always = The device first attempts to authenticate the user using the Web Users table. If no user is found (based on the username-password combination), it attempts to authenticate the user using the LDAP/RADIUS server. |
| Behavior upon Authentication Server Timeout<br>`configure system > mgmt-auth > timeout-behavior`<br>[MgmtBehaviorOnTimeout] | Defines the device's response when a connection timeout occurs with the LDAP/RADIUS server.<br>▪ **[0]** Deny Access = User is denied access to the management platform.<br>▪ **[1]** Verify Access Locally = (Default) Device verifies the user's credentials in its Web Users table (local database).<br>**Note:** The parameter is applicable to LDAP- and RADIUS-based management-user login authentication. |
| Default Access Level<br>`default-access-level`<br>[DefaultAccessLevel] | Defines the default access level for the device when the LDAP/RADIUS response doesn't include an access level attribute for determining the user's management access level.<br>The valid range is 0 to 255. The default is 200 (i.e., Security Administrator).<br>**Note:** The parameter is applicable to LDAP- or RADIUS-based management-user login authentication and authorization. |

### 45.12.1.2 RADIUS Parameters

The RADIUS parameters are described in the table below.

**Table 45-66: RADIUS Parameters**

| Parameter | Description |
|---|---|
| **General RADIUS Parameters** | |
| Enable RADIUS Access Control<br>`enable`<br>[EnableRADIUS] | Enables the RADIUS application.<br>▪ **[0]** Disable (Default)<br>▪ **[1]** Enable<br>**Note:** For the parameter to take effect, a device reset is required. |
| [RadiusTrafficType] | Defines the device's network interface for communicating (RADIUS traffic) with the RADIUS server(s).<br>▪ [0] OAMP (default)<br>▪ [1] Control |

| Parameter | Description |
|---|---|
| | **Note:** If set to Control, only one Control interface must be configured in the Interface table; otherwise, RADIUS communication will fail. |
| RADIUS VSA Vendor ID<br>`configure system > radius > vsa-vendor-id`<br>[RadiusVSAVendorID] | Defines the vendor ID that the device accepts when parsing a RADIUS response packet.<br>The valid range is 0 to 0xFFFFFFFF. The default is 5003. |
| [MaxRADIUSSessions] | Defines the number of concurrent calls that can communicate with the RADIUS server (optional).<br>The valid range is 0 to 240. The default is 240. |
| RADIUS Packets Retransmission<br>[RADIUSRetransmission] | Defines the number of RADIUS retransmission retries when no response is received from the RADIUS server. See also the RadiusTo parameter.<br>The valid range is 1 to 10. The default is 1. |
| RADIUS Response Time Out<br>[RadiusTO] | Defines the time interval (in seconds) that the device waits for a response before it performs a RADIUS retransmission. See also the RADIUSRetransmission parameter.<br>The valid range is 1 to 30. The default is 2. |
| **RADIUS Accounting Parameters** | |
| RADIUS Accounting Type<br>`radius-accounting`<br>[RADIUSAccountingType] | Determines when the RADIUS accounting messages are sent to the RADIUS accounting server.<br>▪ **[0]** At Call Release = (Default) Sent at call release only.<br>▪ **[1]** At Connect & Release = Sent at call connect and release.<br>▪ **[2]** At Setup & Release = Sent at call setup and release. |
| AAA Indications<br>`aaa-indications`<br>[AAAIndications] | Determines the Authentication, Authorization and Accounting (AAA) indications.<br>▪ **[0]** None = (Default) No indications.<br>▪ **[3]** Accounting Only = Only accounting indications are used. |
| **RADIUS User Authentication Parameters** | |
| Use RADIUS for Web/Telnet Login<br>`enable-mgmt-login`<br>[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.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Notes:**<br>▪ For RADIUS login authentication to function, you must also configure the EnableRADIUS parameter to 1 (Enable).<br>▪ RADIUS authentication requires HTTP basic authentication, where the username and password are transmitted in clear text over the network. Therefore, it's recommended to set the HTTPSOnly parameter to 1 to force the use of HTTPS, since the transport is encrypted. |

| Parameter | Description |
|---|---|
| Password Local Cache Mode<br>`local-cache-mode`<br>[RadiusLocalCacheMode] | Defines the device's mode of operation regarding the timer (configured by the parameter RadiusLocalCacheTimeout) that determines the validity of the username and password (verified by the RADIUS server).<br>▪ **[0]** Absolute Expiry Timer = When you access a Web page, the timeout doesn't reset, instead it continues decreasing.<br>▪ **[1]** Reset Timer Upon Access = (Default) Upon each access to a Web page, the timeout always resets (reverts to the initial value configured by RadiusLocalCacheTimeout). |
| Password Local Cache Timeout<br>`local-cache-timeout`<br>[RadiusLocalCacheTimeout] | Defines the time (in seconds) the locally stored username and password (verified by the RADIUS server) are valid. When this time expires, the username and password become invalid and a must be re-verified with the RADIUS server.<br>The valid range is 1 to 0xFFFFFF. The default is 300 (5 minutes).<br>▪ **[-1]** = Never expires.<br>▪ **[0]** = Each request requires RADIUS authentication. |
| RADIUS VSA Access Level Attribute<br>`vsa-access-level`<br>[RadiusVSAAccessAttribute] | Defines the code that indicates the access level attribute in the Vendor Specific Attributes (VSA) section of the received RADIUS packet.<br>The valid range is 0 to 255. The default is 35. |

### 45.12.1.3 LDAP Parameters

The Lightweight Directory Access Protocol (LDAP) parameters are described in the table below.

**Table 45-67: LDAP Parameters**

| Parameter | Description |
|---|---|
| LDAP Service<br>`configure voip/ldap/enable`<br>[LDAPServiceEnable] | Enables the LDAP feature.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Note:** For the parameter to take effect, a device reset is required. |
| LDAP Authentication Filter<br>`configure voip > ldap > auth-filter`<br>[LDAPAuthFilter] | Defines the LDAP search filter attribute for searching the login username in the directory's subtree for LDAP-based user authentication and authorization.<br>You can use the dollar ($) sign to represent the username. For example, if the parameter is set to "(sAMAccountName=$)" and the user logs in with the username "SueM", the LDAP query is run for sAMAccountName=SueM. |
| Use LDAP for Web/Telnet Login<br>`configure voip > ldap > enable-mgmt-login`<br>[MgmtLDAPLogin] | Enables LDAP-based management-user login authentication and authorization.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Note:** For the parameter to take effect, a device reset is required. |
| [LDAPDebugMode] | Determines whether to enable the LDAP task debug messages. This is used for providing debug information regarding LDAP tasks.<br>The valid value range is 0 to 3. The default is 0. |
| LDAP Numeric Attribute<br>`configure voip > sip-definition advanced-settings > ldap-numeric-attr`<br>[LDAPNumericAttributes] | Defines up to five LDAP Attributes (separated by commas) for which the device employs LDAP query searches in the AD for numbers that may have characters between the digits.<br>For more information, see Enabling LDAP Searches for Numbers with Characters on page 253. |
| MS LDAP OCS Number attribute name<br>`ldap-ocs-nm-attr`<br>[MSLDAPOCSNumAttributeName] | Defines the name of the attribute that represents the user's Lync number in the Microsoft AD database.<br>The valid value is a string of up to 49 characters. The default is "msRTCSIP-PrimaryUserAddress". |
| MS LDAP PBX Number attribute name<br>`ldap-pbx-nm-attr`<br>[MSLDAPPBXNumAttributeName] | Defines the name of the attribute that represents the user PBX number in the Microsoft AD database.<br>The valid value is a string of up to 49 characters. The default is "telephoneNumber". |
| MS LDAP MOBILE Number attribute name<br>`ldap-mobile-nm-attr` | Defines the name of the attribute that represents the user Mobile number in the Microsoft AD database. |

| Parameter | Description |
|---|---|
| [MSLDAPMobileNumAttributeName] | The valid value is a string of up to 49 characters. The default is "mobile". |
| `ldap-private-nm-attr`<br>[MSLDAPPrivateNumAttributeName] | Defines the name of the attribute that represents the user's private number in the AD. If this value equals the value of the MSLDAPPrimaryKey or MSLDAPSecondaryKey parameter, then the device queries the AD for the destination number in this private attribute name; otherwise, the parameter is not used as a search key.<br>The default is "msRTCSIP-PrivateLine". |
| MS LDAP DISPLAY Name Attribute Name<br>`ldap-display-nm-attr`<br>[MSLDAPDisplayNameAttributeName] | Defines the attribute name that represents the Calling Name in the AD for LDAP queries based on calling number.<br>The valid value is a string of up to 49 characters. The default is "displayName". |
| `ldap-primary-key`<br>[MSLDAPPrimaryKey] | Defines the name of the attribute used as a query search key for the destination number in the AD. This is used instead of the "PBX" attribute name (configured by the MSLDAPPBXNumAttributeName parameter).<br>The default is not configured. |
| `ldap-secondary-key`<br>[MSLDAPSecondaryKey] | Defines the name of the attribute used as the second query search key for the destination number in the AD, if the primary search key or PBX search is not found. |
| LDAP Cache Service<br>`cache`<br>[LDAPCacheEnable] | Enables the LDAP cache service.<br>▪ **[0]** Disable (default)<br>▪ **[1]** Enable<br>**Notes:**<br>▪ For the parameter to take effect, a device reset is required.<br>▪ For more information on LDAP caching, see "Configuring the Device's LDAP Cache" on page 245. |
| **LDAP Configuration Table** | |
| LDAP Configuration Table<br>`configure voip > ldap > ldap-configuration`<br>[LdapConfiguration] | Defines LDAP servers.<br>The format of the ini file table parameter is as follows:<br>[ LdapConfiguration ]<br>FORMAT LdapConfiguration_Index = LdapConfiguration_Group, LdapConfiguration_LdapConfServerIp, LdapConfiguration_LdapConfServerPort, LdapConfiguration_LdapConfServerMaxRespondTime, LdapConfiguration_LdapConfServerDomainName, LdapConfiguration_LdapConfPassword, LdapConfiguration_LdapConfBindDn, LdapConfiguration_Interface, LdapConfiguration_MngmAuthAtt, LdapConfiguration_useTLS, LdapConfiguration_ConnectionStatus;<br>[ \LdapConfiguration ]<br>For a description of the table, see "Configuring LDAP Servers" on page 236. |

| Parameter | Description |
|---|---|
| **LDAP Server Search Base DN Table** | |
| LDAP Server Search Base DN Table<br>`configure voip > ldap > ldap-servers-search-dns`<br>[LdapServersSearchDNs] | Defines the full base path (i.e., distinguished name / DN) to the objects in the AD where the query is done, per LDAP server.<br>The format of the ini file table parameter is as follows:<br>[ LdapServersSearchDNs ]<br>FORMAT LdapServersSearchDNs_Index = LdapServersSearchDNs_Base_Path, LdapServersSearchDNs_LdapConfigurationIndex, LdapServersSearchDNs_SearchDnInternalIndex;<br>[ \LdapServersSearchDNs ]<br>For a detailed description of the table, see "Configuring LDAP DNs (Base Paths) per LDAP Server" on page 240. |
| **Management LDAP Groups Table** | |
| Management LDAP Groups Table<br>`configure voip > ldap > mgmt-ldap-groups`<br>[MgmntLDAPGroups] | Defines the users group attribute in the AD and corresponding management access level.<br>The format of the ini file table parameter is as follows:<br>[ MgmntLDAPGroups ]<br>FORMAT MgmntLDAPGroups_Index = MgmntLDAPGroups_LdapConfigurationIndex, MgmntLDAPGroups_GroupIndex, MgmntLDAPGroups_Level, MgmntLDAPGroups_Group;<br>[ \MgmntLDAPGroups ]<br>For a description of the table, see "Configuring Access Level per Management Groups Attributes" on page 242. |
| **LDAP Server Groups Table** | |
| LDAP Server Groups Table<br>`config-voip > ldap > ldap-servers-group`<br>[LDAPServersGroup] | Defines LDAP Server Groups.<br>The format of the ini file table parameter is as follows:<br>[ LdapServersGroup ]<br>FORMAT LdapServersGroup_Index = LdapServersGroup_Name, LdapServersGroup_ServerType, LdapServersGroup_SearchMethod, LdapServersGroup_CacheEntryTimeout, LdapServersGroup_CacheEntryRemovalTimeout, LdapServersGroup_SearchDnsMethod;<br>[ \LdapServersGroup ]<br>For a description of the table, see "Configuring LDAP Server Groups" on page 234. |

## 45.12.2 Least Cost Routing Parameters

The Least Cost Routing (LCR) parameters are described in the table below.

**Table 45-68: LCR Parameters**

| Parameter | Description |
|---|---|
| Cost Group Table<br>`configure voip > services least-cost-routing cost-group`<br>[CostGroupTable] | Defines the Cost Groups for LCR, where each Cost Group is configured with a name, fixed call connection charge, and a call rate (charge per minute).<br>[ CostGroupTable ]<br>FORMAT CostGroupTable_Index = CostGroupTable_CostGroupName, CostGroupTable_DefaultConnectionCost, CostGroupTable_DefaultMinuteCost;<br>[ \CostGroupTable ]<br>For example: CostGroupTable 2 = "Local Calls", 2, 1;<br>For a description of the table, see "Configuring Cost Groups" on page 263. |
| Cost Group > Time Band Table<br>`configure voip > services least-cost-routing cost-group-time-bands`<br>[CostGroupTimebands] | Defines time bands and associates them with Cost Groups.<br>[CostGroupTimebands]<br>FORMAT CostGroupTimebands_TimebandIndex = CostGroupTimebands_StartTime, CostGroupTimebands_EndTime, CostGroupTimebands_ConnectionCost, CostGroupTimebands_MinuteCost;<br>[\CostGroupTimebands]<br>For a description of the table, see "Configuring Time Bands for Cost Groups" on page 264. |

## 45.12.3 Call Setup Rules Parameters

The Call Setup Rules parameters are described in the table below.

**Table 45-69: Call Setup Rules Parameters**

| Parameter | Description |
|---|---|
| Call Setup Rules<br>`configure voip/services call-setup-rules`<br>[CallSetupRules] | Defines Call Setup Rules that the device runs at call setup for LDAP-based routing and other advanced routing logic requirements including manipulation.<br>[ CallSetupRules ]<br>FORMAT CallSetupRules_Index = CallSetupRules_RulesSetID, CallSetupRules_QueryTarget, CallSetupRules_AttributesToQuery, CallSetupRules_AttributesToGet, CallSetupRules_RowRole, CallSetupRules_Condition, CallSetupRules_ActionSubject, CallSetupRules_ActionType, CallSetupRules_ActionValue;<br>[ \CallSetupRules ]<br>For a description of the table, see "Configuring Call Setup Rules" on page 283. |

## 45.12.4 HTTP-based Services

The HTTP-based service parameters are described in the table below.

**Table 45-70: HTTP-based Service Parameters**

| Parameter | Description |
|---|---|
| HTTP Remote Services<br>[HTTPRemoteServices] | Defines HTTP-based services.<br>The format of the ini file table parameter is as follows:<br>[HTTPRemoteServices]<br>FORMAT HTTPRemoteServices_Index = HTTPRemoteServices_Name, HTTPRemoteServices_Path, HTTPRemoteServices_HTTPType, HTTPRemoteServices_Policy, HTTPRemoteServices_LoginNeeded, HTTPRemoteServices_PersistentConnection, HTTPRemoteServices_NumOfSockets, HTTPRemoteServices_AuthUserName, HTTPRemoteServices_AuthPassword, HTTPRemoteServices_TLSContext, HTTPRemoteServices_VerifyCertificate, HTTPRemoteServices_TimeOut, HTTPRemoteServices_KeepAliveTimeOut, HTTPRemoteServices_ServiceStatus;<br>[\HTTPRemoteServices]<br>For a description of the table, see "Configuring HTTP Services" on page 267. |
| HTTP Remote Hosts<br>[HTTPRemoteHosts] | Defines remote HTTP hosts per HTTP-based service.<br>The format of the ini file table parameter is as follows:<br>[HTTPRemoteHosts]<br>FORMAT HTTPRemoteHosts_Index = HTTPRemoteHosts_HTTPRemoteServiceIndex, HTTPRemoteHosts_RemoteHostIndex, HTTPRemoteHosts_Name, HTTPRemoteHosts_Address, HTTPRemoteHosts_Port, HTTPRemoteHosts_Interface, HTTPRemoteHosts_HTTPTransportType, HTTPRemoteHosts_HostStatus;<br>[\HTTPRemoteHosts]<br>For a description of the table, see "Configuring Remote HTTP Hosts" on page 271. |
| Topology Status<br>[RoutingServerGroupStatus] | Enables the reporting of the device's topology status (using the REST TopologyStatus API command) to HTTP remote hosts.<br>▪ [0] Disable (default)<br>▪ [1] Enable<br>For more information, see "Configuring HTTP Services" on page 267. |
| GW Routing Server<br>configure voip > gw routing general-setting > gw-routing-server<br>[GWRoutingServer] | Enables routing by a Routing server.<br>▪ [0] Disable = (Default)<br>▪ [1] Enable<br>For more information, see Centralized Third-Party Routing Server on page 272.<br>**Note:** The parameter is applicable only to the Gateway application. |

## 45.12.5 HTTP Proxy Parameters

The HTTP Proxy service parameters are described in the table below.

**Table 45-71: HTTP Proxy Service Parameters**

| Parameter | Description |
|---|---|
| HTTP Proxy Application<br>`configure system > http-proxy > http-proxy-app`<br>[HTTPProxyApplication] | Enables the HTTP Proxy application.<br>▪ [0] Disable (default)<br>▪ [1] Enable<br>**Note:** For the parameter to take effect, a device reset is required. |
| HTTP Interfaces Table<br>`configure system > http-proxy > http-interface`<br>[HTTPInterface] | Defines local listening interfaces for receiving HTTP/S requests from Web clients for HTTP/S-based services.<br>The format of the ini file table parameter is as follows:<br>[ HTTPInterface ]<br>FORMAT HTTPInterface_Index = HTTPInterface_InterfaceName, HTTPInterface_NetworkInterface, HTTPInterface_Protocol, HTTPInterface_Port, HTTPInterface_TLSContext, HTTPInterface_VerifyCert;<br>[ \HTTPInterface ]<br>For a description of the table, see 'Configuring HTTP Interfaces' on page 276. |
| HTTP Proxy Services Table<br>`configure system > http-proxy > http-proxy-serv`<br>[HTTPProxyService] | Defines HTTP Proxy based services.<br>The format of the ini file table parameter is as follows:<br>[ HTTPProxyService ]<br>FORMAT HTTPProxyService_Index = HTTPProxyService_ServiceName, HTTPProxyService_ListeningInterface, HTTPProxyService_URLPrefix, HTTPProxyService_KeepAliveMode;<br>[ \HTTPProxyService ]<br>For a description of the table, see 'Configuring HTTP Proxy Services' on page 278. |
| HTTP Proxy Hosts Table<br>`configure system > http-proxy > http-proxy-host`<br>[HTTPProxyHost] | Defines HTTP Proxy hosts. The table is a "child" of the HTTP Proxy Services table (HTTPProxyService). An HTTP Proxy Host represents the HTTP-based managed equipment (e.g., IP Phone).<br>The format of the ini file table parameter is as follows:<br>[ HTTPProxyHost ]<br>FORMAT HTTPProxyHost_Index = HTTPProxyHost_HTTPProxyServiceId, HTTPProxyHost_HTTPProxyHostId, HTTPProxyHost_NetworkInterface, HTTPProxyHost_IpAddress, HTTPProxyHost_Protocol, HTTPProxyHost_Port, HTTPProxyHost_TLSContext, HTTPProxyHost_VerifyCert;<br>[ \HTTPProxyHost ]<br>For a description of the table, see 'Configuring HTTP Proxy Hosts' on page 279. |

| Parameter | Description |
|---|---|
| EMS Services Table<br>`configure system > http-proxy > ems-serv`<br>[EMSService] | Defines an HTTP-based EMS Service so that the device can act as an HTTP Proxy that enables AudioCodes EMS to manage AudioCodes equipment (such as IP Phones) over HTTP when the equipment is located behind NAT (e.g., in the LAN) and EMS is located in a public domain (e.g., in the WAN).<br>The format of the ini file table parameter is as follows:<br>[ EMSService ]<br>FORMAT EMSService_Index = EMSService_ServiceName, EMSService_PrimaryServer, EMSService_SecondaryServer, EMSService_DeviceLoginInterface, EMSService_EMSInterface;<br>[ \EMSService ]<br>For a description of the table, see 'Configuring an HTTP-based EMS Service' on page 281. |

**This page is intentionally left blank.**

# 46 Channel Capacity

The table below lists the maximum capacity figures for SIP signaling, media sessions, and registered users.

**Table 46-1: Maximum Signaling, Media Sessions and Registered Users**

| Product | Signaling Sessions | Media Sessions | | | Registered Users |
|---|---|---|---|---|---|
| | | RTP-to-TDM | SRTP-TDM | Codec Transcoding | |
| MP-1288 | 288 | | 288 | - | - |

The channel capacity based on coder and FXS blade number is shown in the table below.

**Table 46-2: Channel Capacity per Coder and FXS Blade**

| Coder | Capacity | |
|---|---|---|
| | Single FXS Blade | Fully Populated (4 x FXS Blades) |
| Basic: G.711, G.729A/B, G.723.1, G.726 / G.727 ADPCM | 72 | 288 |
| G.722 | 72 | 288 |
| AMR-NB | 72 | 288 |
| Opus-NB | 60 | 240 |

**Note:** Installation and use of voice coders is subject to obtaining the appropriate license and royalty payments.

**This page is intentionally left blank.**

# 47 Technical Specifications

The device's technical specifications are listed in the table below.

**Notes:**

- All specifications in this document are subject to change without prior notice.
- The compliance and regulatory information can be downloaded from AudioCodes Web site at http://www.audiocodes.com/library.

**Table 47-1: Technical Specifications**

<table>
<tr><td colspan="4">System</td></tr>
<tr><td>Telephony Capacity</td><td colspan="3">Up to 288 ports in 4 FXS Blades (each FXS Blade supports 72 ports) Three available capacity options: 288, 216 and 144 ports</td></tr>
<tr><td>Hardware Elements</td><td>Single System Controller (SC)<br>4 FXS Blades with analog interfaces (hot-swappable*)</td><td colspan="2">1+1 Power Supplies<br>Fan Module (front-to-rear air flow)</td></tr>
<tr><td colspan="4">Signaling</td></tr>
<tr><td>Control</td><td colspan="3">SIP (RFC 3261), mature & broadly deployed SIP stack</td></tr>
<tr><td>Message Manipulation</td><td colspan="3">Ability to add/modify/delete SIP headers and message body using advanced regular expressions (regex)</td></tr>
<tr><td colspan="4">SIP Routing</td></tr>
<tr><td>Routing Methods</td><td colspan="3">Request URL, IP Address, FQDN, ENUM, advanced LDAP, third-party routing control through REST API</td></tr>
<tr><td>Redundancy</td><td colspan="3">Detection of proxy failures and subsequent routing to alternative proxies</td></tr>
<tr><td>Routing Features</td><td colspan="3">Least-cost routing, call forking, load balancing, emergency call detection and prioritization</td></tr>
<tr><td colspan="4">Voice Capabilities</td></tr>
<tr><td>Voice over Packet</td><td colspan="3">G.168-2004 compliant Echo Cancellation, Packet Loss Concealment, Dynamic Programmable Jitter Buffer, Silence Suppression/Comfort Noise Generation, RTP redundancy, broken connection detection</td></tr>
<tr><td>Voice Compression</td><td colspan="3">G.711, G.723.1, G.726 ADPCM, G.727 ADPCM, G.729A/B, G.722, AMR-NB, Opus-NB</td></tr>
<tr><td>Fax-over-IP</td><td colspan="3">Bypass, T.38 and T.38v3</td></tr>
<tr><td>3-Way Conference</td><td colspan="3">3-way conference with local mixing across all FXS Blades</td></tr>
<tr><td>In-band Signaling</td><td colspan="3">DTMF (TIA 464B), User-defined and call progress tones</td></tr>
<tr><td>Out-of-Band Signaling</td><td colspan="3">DTMF Relay (RFC 2833), DTMF via SIP INFO/NOTIFY</td></tr>
<tr><td colspan="4">Network Protocols</td></tr>
<tr><td>IP Transport</td><td colspan="3">IPv4, IPv6 for media and control, RTP/RTCP per IETF RFC 3550</td></tr>
<tr><td colspan="4">Security</td></tr>
<tr><td>Media</td><td>SRTP</td><td>Control</td><td>TLS/SIPS</td></tr>
<tr><td>Management</td><td colspan="3">HTTPS, SSH, SNMPv3, Access List, RADIUS Web and Telnet authorization</td></tr>
</table>

| | | | |
|---|---|---|---|
| **Voice Quality and SLA** | | | |
| Survivability* | Ensures call continuity between LAN SIP clients upon connectivity failure. Support 300 registered users | | |
| Packet Marking | 802.1p/Q VLAN tagging, DiffServ, TOS | | |
| Voice Enhancement | RTCP-XR, acoustic echo cancellation, replacing voice profile due to impairment detection | | |
| Test Agent | Ability to remotely verify connectivity, voice quality and SIP message flow between SIP UAs | | |
| **Management** | | | |
| OAM&P | Browser-based GUI, CLI, SNMP, EMS, INI configuration file, REST API | | |
| Automatic Configuration | DHCP, TFTP and HTTP for automatic installation | | |
| **Physical Interfaces** | | | |
| Telephone Interfaces | Up to 288 FXS ports | | |
| Lifeline | Automatic switching to PSTN via 3 dedicated lifeline interfaces per FXS Blade | | |
| Network Interfaces | Dual Redundant 10/100/1000 Base-T Ethernet ports<br>Dual Redundant Small Form-Factor Pluggable (SFP)-based connectivity*<br>**Note:** Hardware installation selectable option | | |
| Console | RJ-45 serial interface for local management | | |
| USB Interface | USB 2.0 for supporting external USB dongle* | | |
| **Power** | | | |
| AC Input Voltage | 100 - 240 V AC | DC Input Voltage* | -48 V DC |
| Max. AC Input Current | 10 A | AC Input Frequency | 50/60 Hz |
| Redundant Power Supply | Optional, dual feed, redundant Power Supply modules | | |
| Max Power Consumption | FXS Interfaces | Short Haul (W) | Long Haul (W) |
| | 288 | 450 | 950 |
| | 216 | 400 | 770 |
| | 144 | 350 | 600 |
| **Physical** | | | |
| Width | 17.13 inches (435.2 mm) | Height | 5.16 inches (131.2 mm) |
| Depth | 17.75 inches (451 mm) | Weight | 21 Kg (fully populated system) |
| Mounting | 3U, 19-inch rack | | |

| | | | |
|---|---|---|---|
| **Environment** | | | |
| Temperature | Operational Temp.: 0 to 40°C (41 to 104°F) | Storage Temp.: -40 to 70°C (-40 to 158°F) | Humidity: 5 to 90% non-condensing |
| Over-voltage protection and surge immunity | ITU-T K.21 (basic) compliant.<br>Note: Routing of FXS telephony cables outdoors can be done only in conjunction with AudioCodes-approved primary surge protector and proper installation and grounding. | | |
| **FXS Port Specifications** | | | |
| Interface Type | FXS connection via 50-pin CHAMP connector | | |
| FXS Signaling Formats | In-band signaling DTMF (TIA 464B), Out-of-band pulse signalling* | | |
| FXS Loop Impedance | Up to 1500 ohm (including phone impedance) | | |
| Off-hook Loop Current | 25 mA max. on all ports (35 mA max. on two ports per FXS Blade for emergency / elevator phones)* | | |
| Ring Voltage (Sine) | 54 Vrms (80 Vrms on two ports per FXS Blade for emergency / elevator phones)<br>**Notes:** Balanced ringing only, Enables simultaneous ringing of 288 phones (72 per FXS Blade given REN 3 load) | | |
| Ring Frequency | 25-100 Hz | | |
| Maximum Ringer Load | Ringer Equivalency Number (REN) 3 | | |
| 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 | | |
| Message Waiting Indication (MWI) | DC voltage generation (TIA/EIA-464-B), V23 FSK data, Stutter dial tone | | |
| **Regulatory Compliance** | | | |
| EMC | EN55022 Class A , CFR Part 15 Class A, EN55024, EN61000-3-3, EN61000-3-2, VCCI Class X1 (equal to class B) | | |
| Safety | EN60950-1, UL60950-1 | | |

**International Headquarters**
1 Hayarden Street,
Airport City
Lod 7019900, Israel
Tel: +972-3-976-4000
Fax: +972-3-976-4040

**AudioCodes Inc.**
27 World's Fair Drive,
Somerset, NJ 08873
Tel: +1-732-469-0880
Fax: +1-732-469-2298

**Contact us:** www.audiocodes.com/contact
**Website:** www.audiocodes.com



Document #: LTRT-28052