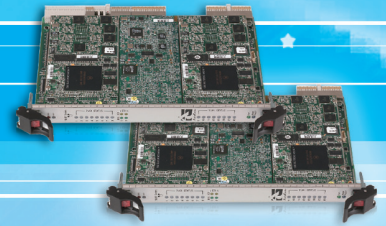


# Product Reference Manual

MediaPack™ Media Gateways  
Mediant™ Media Gateways  
IPmedia™ Media Servers



**SIP**

**Version 5.6**

December 2008

Document # LTRT-52303





---

## Table of Contents

---

<b>1</b>	<b>Introduction.....</b>	<b>15</b>
<b>2</b>	<b>Device Initialization .....</b>	<b>17</b>
2.1	Startup Process.....	17
2.2	Device Firmware .....	18
2.3	Configuration Parameters and Files.....	19
2.4	Using BootP / DHCP .....	19
2.4.1	BootP/DHCP Server Parameters.....	20
2.4.2	DHCP Host Name Support .....	20
2.4.3	Microsoft DHCP/BootP Server.....	21
2.4.4	Using BootP .....	22
2.4.4.1	Upgrading the Device .....	22
2.4.4.2	Vendor Specific Information Field.....	22
2.4.4.3	Selective BootP .....	24
2.5	Automatic Update Mechanism .....	24
<b>3</b>	<b>Command-Line Interface Based Management.....</b>	<b>27</b>
3.1	Starting a CLI Management Session.....	27
3.2	CLI Navigation Concepts.....	28
3.3	Commands .....	28
3.3.1	General Commands.....	29
3.3.2	Configuration Commands .....	32
3.3.3	Management Commands.....	33
3.3.4	PSTN Commands .....	33
<b>4</b>	<b>SNMP-Based Management .....</b>	<b>35</b>
4.1	SNMP Standards and Objects .....	35
4.1.1	SNMP Message Standard .....	35
4.1.2	SNMP MIB Objects .....	36
4.1.3	SNMP Extensibility Feature .....	37
4.2	Carrier-Grade Alarm System.....	37
4.2.1	Active Alarm Table.....	37
4.2.2	Alarm History.....	38
4.3	Topology MIB - Objects.....	38
4.3.1	Physical Entity - RFC 2737 .....	38
4.3.2	IF-MIB - RFC 2863.....	38
4.4	Cold Start Trap .....	43
4.5	Performance Measurements .....	43
4.5.1	Total Counters.....	44
4.6	TrunkPack-VoP Series Supported MIBs .....	44
4.7	Traps .....	49
4.8	SNMP Interface Details .....	53
4.8.1	SNMP Community Names .....	53
4.8.1.1	Configuring Community Strings via the Web.....	53
4.8.1.2	Configuring Community Strings via the ini File.....	53
4.8.1.3	Configuring Community Strings via SNMP.....	53
4.8.2	SNMPv3 USM Users.....	55

4.8.2.1	Configuring SNMPv3 Users via the ini File .....	56
4.8.2.2	Configuring SNMPv3 Users via SNMP .....	57
4.8.3	Trusted Managers .....	58
4.8.3.1	Configuring Trusted Managers via ini File .....	58
4.8.3.2	Configuring Trusted Managers via SNMP .....	58
4.8.4	SNMP Ports .....	59
4.8.5	Multiple SNMP Trap Destinations .....	59
4.8.5.1	Configuring Trap Managers via Host Name .....	60
4.8.5.2	Configuring Trap Managers via the ini File .....	60
4.8.5.3	Configuring Trap Managers via SNMP .....	61
4.8.5.4	SNMP Manager Backward Compatibility .....	62
4.9	Dual Module Interface .....	62
4.10	SNMP NAT Traversal .....	63
4.11	Media Server Configuration .....	64
4.12	Systems .....	64
4.13	High Availability Systems .....	65
4.14	Configuring Clock Synchronization .....	65
4.15	SNMP Administrative State Control .....	66
4.15.1	Node Maintenance .....	66
4.15.2	Graceful Shutdown .....	66
4.16	AudioCodes' Element Management System .....	67
4.17	SNMP Traps .....	67
4.17.1	Alarm Traps .....	67
4.17.1.1	Component: Chassis#0 .....	68
4.17.1.2	Component: Chassis#0/TimingManager#0 .....	70
4.17.1.3	Component: Interfaces#0/Sonet#<m> .....	72
4.17.1.4	Component: System#0<n> and Board#0<n> .....	74
4.17.1.5	Component: System#0 .....	78
4.17.1.6	Component: AlarmManager#0 .....	79
4.17.1.7	Component: AudioStaging#0 .....	79
4.17.1.8	Component: AnalogPorts#0 .....	80
4.17.1.9	Component: SS7#0 .....	81
4.17.1.10	Component: System#0/Module#<m> .....	85
4.17.1.11	Component: Interfaces#0/Trunk#<m> .....	88
4.17.2	Log Traps (Notifications) .....	90
4.17.3	Other Traps .....	92
4.17.4	Trap Varbinds .....	93
4.17.5	Customizing Trap's Enterprise OID .....	94
<b>5</b>	<b>Automatic Device Configuration .....</b>	<b>95</b>
5.1	Automatic Configuration Options .....	95
5.1.1	Local Configuration Server with BootP/TFTP .....	95
5.1.2	DHCP-based Configuration Server .....	96
5.1.3	Configuration using DHCP Option 67 .....	96
5.1.4	TFTP Configuration using DHCP Option 66 .....	97
5.1.5	HTTP-based Automatic Updates .....	97
5.1.6	Configuration using FTP or NFS .....	98
5.1.7	Configuration using AudioCodes EMS .....	98
5.2	Loading Files Securely (Disabling TFTP) .....	99
<b>6</b>	<b>Security .....</b>	<b>101</b>
6.1	IPSec and IKE .....	101
6.1.1	IKE .....	102
6.1.2	IPSec .....	102

6.1.3	IPSec and IKE Configuration Table's Confidentiality .....	103
6.1.4	Dead Peer Detection (RFC 3706) .....	104
6.1.5	Certificate Revocation Checking .....	104
6.1.6	Certificate Chain .....	105
6.2	Secure Shell .....	105
6.3	SSL / TLS .....	107
6.3.1	SIP Over TLS (SIPS) .....	108
6.3.2	Secured HTTPS Web Interface Configuration .....	108
6.3.3	Secured Telnet .....	108
6.4	SRTP .....	109
6.5	RADIUS Login Authentication .....	110
6.5.1	Setting Up a RADIUS Server .....	111
6.5.2	Configuring RADIUS Support .....	112
6.6	Internal Firewall .....	114
6.7	Network Port Usage .....	116
6.8	Recommended Practices .....	117
6.9	Legal Notice .....	117
<b>7</b>	<b>RTP Control Protocol Extended Reports (RTCP-XR) .....</b>	<b>119</b>
<b>8</b>	<b>RTP / RTCP Payload Types and Port Allocation .....</b>	<b>121</b>
8.1	Payload Types Defined in RFC 3551 .....	121
8.2	Defined Payload Types .....	122
8.3	Default RTP / RTCP / T.38 Port Allocation .....	123
<b>9</b>	<b>CAS Protocol Table .....</b>	<b>125</b>
9.1	Constructing CAS Protocol Tables for CAS-Terminated Protocols .....	125
9.2	Protocol Table Elements .....	125
9.2.1	INIT Variables .....	125
9.2.2	Actions .....	126
9.2.3	Functions .....	126
9.2.4	States .....	126
9.3	Reserved Words .....	128
9.4	State Line Structure .....	128
9.5	Action / Event .....	128
9.5.1	User Command Oriented Action / Event .....	129
9.5.2	CAS Change Oriented Events .....	129
9.5.3	Timer Oriented Events .....	130
9.5.4	Counter Oriented Events .....	130
9.5.5	IBS Oriented Events .....	130
9.5.6	DTMF/MF Oriented Events .....	131
9.5.7	Operator Service Events (up to GR-506) .....	133
9.6	Function .....	134
9.7	Parameters .....	134
9.8	Next State .....	136
9.9	Changing the Script File .....	137
9.9.1	MFC-R2 Protocol .....	137

<b>10 SS7 Tunneling.....</b>	<b>139</b>
10.1 MTP2 Tunneling Technology .....	140
10.2 SS7 Characteristics .....	140
10.3 SS7 Parameters .....	141
10.4 SS7 MTP2 Tunneling ini File Example .....	146
10.5 Configuring SS7 Tunneling .....	149
10.5.1 Configuring MTP2 Attributes .....	149
10.5.2 Configuring SS7 Signaling Node Timers .....	152
10.5.3 Configuring Link-Set Timers .....	155
10.5.4 Configuring Links .....	157
10.5.5 Configuring SS7 Signaling Nodes .....	159
10.5.6 Configuring MTP3 Redundancy .....	161
10.5.7 Configuring Static Routing Context .....	162
10.5.8 Configuring Sigtran Group IDs .....	163
10.5.9 Configuring Sigtran Interface IDs .....	165
<b>11 Accessory Programs and Tools .....</b>	<b>167</b>
11.1 BootP/TFTP Server Configuration Utility .....	167
11.1.1 When to Use the BootP/TFTP .....	167
11.1.2 An Overview of BootP .....	167
11.1.3 Key Features .....	168
11.1.4 Specifications .....	168
11.1.5 Installation .....	168
11.1.6 Loading the cmp File - Booting the Device .....	169
11.1.7 BootP/TFTP Application User Interface .....	169
11.1.8 Toolbar Buttons in the Main Screen .....	170
11.1.9 Log Window .....	170
11.1.10 Setting the Preferences .....	172
11.1.10.1 BootP Preferences .....	172
11.1.10.2 TFTP Preferences .....	173
11.1.11 Configuring the BootP Clients .....	174
11.1.11.1 Client Parameters .....	174
11.1.11.2 Using Command Line Switches .....	175
11.1.11.3 Adding Clients .....	177
11.1.11.4 Editing Client Parameters .....	178
11.1.11.5 Deleting Clients .....	178
11.1.11.6 Testing the Client .....	178
11.1.12 Managing Client Templates .....	179
11.2 TrunkPack Downloadable Conversion Utility .....	180
11.2.1 Converting a CPT ini File to a Binary dat File .....	181
11.2.2 Creating a Loadable Voice Prompts File .....	183
11.2.3 Creating a Loadable CAS Protocol Table File .....	184
11.2.4 Creating a Dial Plan File .....	186
11.2.5 Encoding / Decoding an ini File .....	187
11.2.6 Creating a Loadable Prerecorded Tones File .....	188
11.3 Call Progress Tones Wizard .....	190
11.3.1 Installation .....	190
11.3.2 Initial Settings .....	191
11.3.3 Recording Screen - Automatic Mode .....	192
11.3.4 Recording Screen - Manual Mode .....	194
11.3.5 Call Progress Tones ini and dat Files .....	195
11.3.6 Adding a Reorder Tone to the CPT File .....	196
<b>12 Diagnostics .....</b>	<b>197</b>
12.1 Self-Testing .....	197

12.2	Analog Line Testing.....	198
12.3	Syslog Support .....	199
12.3.1	Syslog Servers.....	200
12.3.2	Enabling the Syslog Server.....	201
12.4	Debug Recording (DR).....	201
12.4.1	Collecting DR Messages.....	202
12.4.2	Activating DR .....	202
12.4.3	DR Command Reference.....	203
<b>13</b>	<b>Glossary .....</b>	<b>207</b>



---

## List of Figures

---

Figure 2-1: Startup Process.....	18
Figure 6-1: IPSec Encryption.....	101
Figure 6-2: Certificate Chain Hierarchy .....	105
Figure 10-1: M2UA Architecture .....	139
Figure 10-2: M2TN Architecture .....	139
Figure 10-3: Protocol Architecture for MTP2 Tunneling .....	140
Figure 10-4: MTP2 Attributes Page .....	150
Figure 10-5: SS7 Signaling Node Timers Page.....	152
Figure 10-6: SS7 Link-set Timers Page.....	155
Figure 10-7: Links Page.....	157
Figure 10-8: SS7 Signaling Nodes Page.....	159
Figure 10-9: MTP3 Redundancy Configuration Page.....	161
Figure 10-10: Static Routing Context Table Page .....	162
Figure 10-11: Sigtran Group IDs Page .....	163
Figure 10-12: Sigtran Interface IDs Page .....	165
Figure 11-1: Main Screen .....	169
Figure 11-2: Reset Screen.....	170
Figure 11-3: Preferences Screen.....	172
Figure 11-4: BootP Client Configuration Screen .....	174
Figure 11-5: Templates Screen .....	179
Figure 11-6: TrunkPack Downloadable Conversion Utility Main Screen.....	181
Figure 11-7: Call Progress Tones Screen .....	182
Figure 11-8: Voice Prompts Screen.....	183
Figure 11-9: File Data Window .....	184
Figure 11-10: Call Associated Signaling (CAS) Screen .....	185
Figure 11-11: Dial Plan Screen.....	186
Figure 11-12: Encode / Decode ini File(s) Screen.....	187
Figure 11-13: Prerecorded Tones Screen .....	189
Figure 11-14: File Data Window .....	190
Figure 11-15: Initial Settings Screen.....	191
Figure 11-16: Recording Screen - Automatic Mode .....	192
Figure 11-17: Recording Screen after Automatic Detection .....	193
Figure 11-18: Recording Screen - Manual Mode .....	194
Figure 12-1: AudioCodes' Proprietary Syslog Server.....	201



## List of Tables

Table 2-1: Vendor Specific Information Field .....	23
Table 2-2: Structure of the Vendor Specific Information Field.....	24
Table 3-1: Summary of CLI Commands .....	28
Table 3-2: General CLI Commands.....	29
Table 3-3: Configuration CLI Commands .....	32
Table 3-4: CLI Management Command .....	33
Table 3-5: PSTN CLI Command.....	33
Table 4-1: DS1 Digital Interfaces.....	39
Table 4-2: BRI Interfaces (Applicable to Mediant 1000 & Mediant 600) .....	39
Table 4-3: Ethernet (Gigabit for 3000 Series) Interface.....	40
Table 4-4: SONET /SDH Interfaces (3000 Series Only).....	41
Table 4-5: DS3 Interfaces (3000 Series Only).....	42
Table 4-6: Proprietary Traps.....	49
Table 4-7: SNMP Predefined Groups.....	53
Table 4-8: SNMPv3 Security Levels.....	55
Table 4-9: SNMPv3 Predefined Groups .....	55
Table 4-10: SNMPv3 Table Columns Description .....	56
Table 4-11: acFanTrayAlarm Alarm Trap (Applicable Only to 3000 Series and Mediant 1000) .....	68
Table 4-12: acPowerSupplyAlarm Alarm Trap (Applicable Only to 3000 Series and Mediant 1000) ...	68
Table 4-13: acUserInputAlarm Alarm Trap.....	69
Table 4-14: acPEMAlarm Alarm Trap (Applicable Only to 3000 Series).....	69
Table 4-15: acHwFailureAlarm Alarm Trap (Applicable Only to Mediant 1000 and Mediant 600).....	70
Table 4-16: acTMInconsistentRemoteAndLocalPLLStatus Alarm .....	70
Table 4-17: acTMReferenceStatus Alarm .....	71
Table 4-18: acTMReferenceChange Alarm.....	71
Table 4-19: AcSonetSectionLOFAlarm Alarm Trap.....	72
Table 4-20: AcSonetSectionLOSAAlarm Alarm Trap .....	72
Table 4-21: AcSonetLineAISAlarm Alarm Trap .....	73
Table 4-22: AcSonetLineRDIAAlarm Alarm Trap.....	73
Table 4-23: acBoardFatalError Alarm Trap .....	74
Table 4-24: acBoardConfigurationError Alarm Trap.....	74
Table 4-25: acBoardTemperatureAlarm Alarm Trap (Applicable to 2000 and 3000 Series - Except Mediant 3000 HA) .....	75
Table 4-26: acBoardEvResettingBoard Alarm Trap .....	75
Table 4-27: acBoardEthernetLinkAlarm Alarm Trap (Applicable only to 2000 Series, Mediant 1000, and MediaPack) .....	76
Table 4-28: acBoardCallResourcesAlarm Alarm Trap .....	76
Table 4-29: acBoardControllerFailureAlarm Alarm Trap .....	77
Table 4-30: acBoardOverloadAlarm Alarm Trap .....	77
Table 4-31: acFeatureKeyError Alarm Trap (Applicable only to Digital devices) .....	77
Table 4-32: acSAMissingAlarm Alarm Trap (Applicable only to the 3000 Series Devices) .....	78
Table 4-33: acHitlessUpdateStatus Alarm Trap .....	78
Table 4-34: acActiveAlarmTableOverflow Alarm Trap .....	79
Table 4-35: acAudioProvisioningAlarm Alarm Trap.....	79
Table 4-36: acAnalogPortSPIOutOfService Alarm Trap.....	80
Table 4-37: acAnalogPortHighTemperature Alarm Trap.....	80
Table 4-38: acSS7LinkStateChangeAlarm Trap .....	81
Table 4-39: acSS7LinkCongestionStateChangeAlarm Trap .....	82
Table 4-40: acSS7LinkInhibitStateChangeAlarm Trap.....	82
Table 4-41: acSS7LinkBlockStateChangeAlarm Trap.....	83
Table 4-42: acSS7LinkSetStateChangeAlarm Trap .....	83
Table 4-43: acSS7RouteSetStateChangeAlarm Trap .....	84
Table 4-44: acSS7SNSetStateChangeAlarm Trap.....	84
Table 4-45: acSS7RedundancyAlarm Trap.....	85
Table 4-46: acHASystemConfigMismatchAlarm Trap .....	85
Table 4-47: acHASystemFaultAlarm Trap.....	86

Table 4-48: acHASystemSwitchOverAlarm Trap .....	87
Table 4-49: acBoardTemperatureAlarm Trap .....	87
Table 4-50: acBoardEthernetLinkAlarm Trap .....	88
Table 4-51: acTrunksAlarmNearEndLOS Alarm Trap .....	88
Table 4-52: acTrunksAlarmNearEndLOF Alarm Trap .....	89
Table 4-53: acTrunksAlarmRcvAIS Alarm Trap .....	89
Table 4-54: acTrunksAlarmFarEndLOF Alarm Trap .....	90
Table 4-55: acKeepAlive Log Trap .....	90
Table 4-56: acPerformanceMonitoringThresholdCrossing Log Trap .....	91
Table 4-57: acHTTPDownloadResult Log Trap .....	91
Table 4-58: acDialPlanFileReplaced Log Trap (Applicable Only to Digital Devices, Except IPmedia 3000/IPM-8410) .....	91
Table 4-59: acHitlessUpdateStatus Log Trap (Applicable Only to 3000 Series Devices) .....	92
Table 4-60: coldStart Trap .....	92
Table 4-61: authenticationFailure Trap .....	92
Table 4-62: acBoardEvBoardStarted Trap .....	92
Table 4-63: AcDChannelStatus Trap (Applicable Only to Digital Devices) .....	93
Table 6-1: Default TCP/UDP Network Port Numbers .....	116
Table 7-1: RTCP-XR Published VoIP Metrics .....	119
Table 8-1: Packet Types Defined in RFC 3551 .....	121
Table 8-2: Defined Payload Types .....	122
Table 8-3: Local UDP Port Offsets .....	123
Table 9-1: ST_DIAL: Table Elements .....	126
Table 9-2: User Command Orientated Action / Event .....	129
Table 9-3: CAS Change Orientated Events .....	129
Table 9-4: Time-Orientated Events .....	130
Table 9-5: Counter Orientated Events .....	130
Table 9-6: IBS Orientated Events .....	130
Table 9-7: DTMF / MF Orientated Events .....	131
Table 9-8: Actions / Events Causing MFC-R2 Table to Send Correct MF Tone to Backward Direction .....	132
Table 9-9: Operator Service Events (Up to GR-506) .....	133
Table 9-10: Available User Functions and Corresponding Parameters .....	135
Table 9-11: Parameters Associated with Sending Digits .....	135
Table 10-1: SS7 Parameters .....	141
Table 10-2: MTP2 Parameters .....	150
Table 10-3: SS7 Signaling Node Timers Parameters .....	153
Table 10-4: SS7 Link-Set Timers Parameters .....	156
Table 10-5: SS7 Links Parameters .....	157
Table 10-6: SS7 Signaling Nodes Parameters .....	159
Table 10-7: MTP3 Redundancy Parameters .....	161
Table 10-8: SS7 Static Routing Context Parameters .....	162
Table 10-9: Sigtran Group IDs Parameters .....	164
Table 10-10: Sigtran Interface IDs Parameters .....	166
Table 11-1: Command Line Switch Descriptions .....	176
Table 12-1: Client Setup Commands .....	203
Table 12-2: Trace Rules .....	204
Table 12-3: DR Activation .....	206
Table 13-1: Glossary of Terms .....	207

## Notice

This document provides a reference guide for the following AudioCodes SIP-based Voice over IP (VoIP) products:

- Media Gateway Systems: MediaPack Series, Mediant 600, Mediant 1000, Mediant 2000, Mediant 3000.
- Media Server Systems: IPmedia 2000 and IPmedia 3000.
- cPCI Blades: TP-6310, IPM-6310, TP-8410, and IPM-8410.

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

© Copyright 2008 AudioCodes Ltd. All rights reserved.

This document is subject to change without notice.

Date Published: November-23-2008



**Tip:** When viewing this manual on CD, Web site or on any other electronic copy, all cross-references are hyperlinked. Click on the page or section numbers (shown in blue) to reach the individual cross-referenced item directly. To return back to the point from where you accessed the cross-reference, press the **ALT** and **←** keys

## Trademarks

AudioCodes, AC, Ardito, AudioCoded, NetCoder, TrunkPack, VoicePacketizer, MediaPack, Stretto, Mediant, VolPerfect and IPmedia, OSN, Open Solutions Network, What's Inside Matters, Your Gateway To VoIP, 3GX and Nuera, Netrake, InTouch, CTI<sup>2</sup> and CTI Squared are trademarks or registered trademarks of AudioCodes Limited. All other products or trademarks are property of their respective owners.

## WEEE EU Directive

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

## Customer Support

Customer technical support and service are provided by AudioCodes' Distributors, Partners, and Resellers from whom the product was purchased. For Customer support for products purchased directly from AudioCodes, contact [support@audiocodes.com](mailto:support@audiocodes.com).

## Abbreviations and Terminology

Each abbreviation, unless widely used, is spelled out in full when first used. Only industry-standard terms are used throughout this manual. Hexadecimal notation is indicated by 0x preceding the number.

## Related Documentation

Document #	Manual Name
LTRT-656xx (where xx denotes the document version)	MP-11x & MP-124 SIP Release Notes
LTRT-654xx	MP-11x & MP-124 SIP User's Manual
LTRT-598xx	MP-11x & MP-124 SIP-MGCP Installation Manual
LTRT-529xx	MP-11x SIP Track Guide
LTRT-532xx	MP-124 SIP Track Guide
LTRT-831xx	Mediant 1000 & Mediant 600 SIP Release Notes
LTRT-833xx	Mediant 1000 & Mediant 600 SIP User's Manual
LTRT-856xx	Mediant 600 SIP Installation Manual
LTRT-835xx	Mediant 1000 SIP-MEGACO Installation Manual
LTRT-690xx	Mediant 3000 & Mediant 2000 & TP Series SIP Release Notes
LTRT-688xx	Mediant 2000 & TP-1610 SIP User's Manual
LTRT-701xx	Mediant 2000 & IPmedia 2000 SIP-MGCP-MEGACO Installation Manual
LTRT-590xx	IPmedia Series & IPM Series SIP Release Notes
LTRT-588xx	IPmedia 2000 & IPM-1610 SIP User's Manual
LTRT-898xx	IPmedia 3000 & IPM-6310 & IPM-8410 SIP User's Manual
LTRT-897xx	Mediant 3000 & TP-6310 & TP-8410 SIP User's Manual
LTRT-947xx	Mediant 3000 & IPmedia 3000 Installation Manual
LTRT-665xx	CPE SIP Configuration Guide for IP Voice Mail



**Note:** Throughout this manual and unless otherwise specified, the following terms are used to refer to AudioCodes' products:

- **Device:** refers to all the AudioCodes' products listed in the Notice bulletin above.
- **3000 Series:** refers to Mediant 3000, IPmedia 3000, TP-6310, IPM-6310, TP-8410, and IPM-8410.
- **IPmedia Series:** refers to IPmedia 3000, IPM-6310, and IPM-8410, as well as IPmedia 2000 and IPM-1610.
- **8410 Blade Series:** refers to TP-8410 and IPM-8410 blades.
- **6310 Blade Series:** refers to TP-6310 and IPM-6310 blades.
- **2000 Series:** refers to Mediant 2000, TP-1610, IPmedia 2000, and IPM-1610.
- **Digital:** refers to all products except MediaPack.
- **Analog:** refers to the MediaPack series and Mediant 1000 (analog interface).
- **MediaPack:** refers to MP-118, MP-114, MP-112, and MP-124.



**Note:** The terms *IP-to-Tel* and *Tel-to-IP* refer to the direction of the call relative to the AudioCodes device: *IP-to-Tel* refers to calls received from the IP network and destined to the PSTN (i.e., telephone connected directly or indirectly to the device); *Tel-to-IP* refers to calls received from the PSTN and destined for the IP network.

## Reader's Notes

# 1 Introduction

This manual provides you with supplementary information on AudioCodes SIP-based, Voice-over-IP (VoIP) devices. This information is complementary to the information provided by the device's *User's Manual* and includes, for example, detailed descriptions on various supported features, AudioCodes proprietary applications, advanced configuration methods, and so on.

This manual relates to the following AudioCodes VoIP devices:

## ■ 3000 Series:

- Media Gateway series:
  - ◆ Mediant 3000 gateway hosting a single or dual (High Availability) TP-8410 blade
  - ◆ Mediant 3000 gateway hosting a single or dual (High Availability) TP-6310 blade
  - ◆ Standalone TP-8410 cPCI blade
  - ◆ Standalone TP-6310 cPCI blade
- Media Server series:
  - ◆ IPmedia 3000 media server hosting a single IPM-8410 blade
  - ◆ IPmedia 3000 media server hosting a single IPM-6310 blade
  - ◆ Standalone IPM-8410 cPCI blade
  - ◆ Standalone IPM-6310 cPCI blade

## ■ 2000 Series:

- Media Gateway series:
  - ◆ Mediant 2000 gateway (with TP-1610 cPCI blade)
- Media Server series:
  - ◆ IPmedia 2000 media server (with IPM-1610 cPCI blade)

## ■ Mediant 1000 media gateway

## ■ Mediant 600 media gateway

## ■ MediaPack Series gateways

Please refer to the notes in the previous section 'Notices' for the naming conventions used throughout this manual. For information on how to fully configure the device, please refer to the relevant device's *User's Manual*.



**Note:** This manual is not applicable to the Mediant 1000 MSBG device.



## Reader's Notes

## 2 Device Initialization

This section describes the device's initialization process, including the different methods for initial configuration.

### 2.1 Startup Process

The startup process (illustrated in the following figure) begins when the device is reset. The device resets either by a manual (physical) reset, using the Web interface, using SNMP, or when there is a device irregularity. The startup process ends when the operational software is running. In the startup process, the device obtains its IP address, and software and configuration files.

After the device powers up or after it's physically reset, it broadcasts a BootRequest message to the network. If it receives a reply (from a BootP server), it changes its network parameters (IP address, subnet mask and default gateway address) to the values provided. If there is no reply from a BootP server and if DHCP is enabled (DHCPEnable = 1), the device initiates a standard DHCP procedure to configure its network parameters.

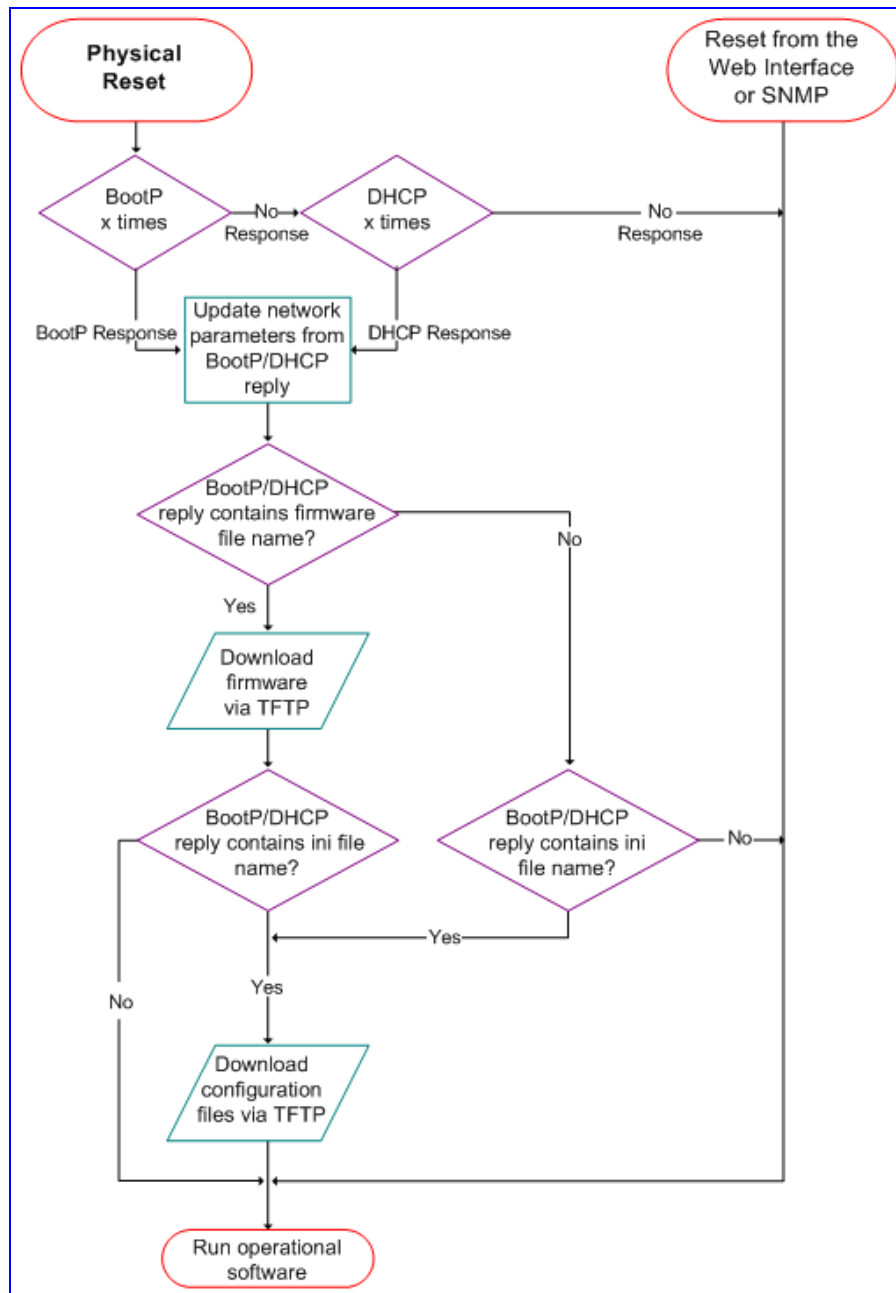
After changing the network parameters, the device attempts to load the device's operational firmware (*cmp*) and various configuration files from the TFTP server's IP address, received from the BootP/DHCP servers. If a TFTP server's IP address isn't received, the device attempts to load the *cmp* file and / or configuration files from a preconfigured TFTP server (refer to "Automatic Update Mechanism" on page 24). Thus, the device can obtain its network parameters from BootP or DHCP servers, and its software and configuration files from a different TFTP server (preconfigured in the *ini* configuration file).

If BootP/DHCP servers are not located or when the device is reset using the Web interface or SNMP, it retains its network parameters and attempts to load the *cmp* file and / or configuration files from a preconfigured TFTP server. If a preconfigured TFTP server doesn't exist, the device operates using the existing software and configuration files in its non-volatile memory.



#### Notes:

- After the operational software runs and if DHCP is configured, the device attempts to renew its lease with the DHCP server.
- Though DHCP and BootP servers are very similar in operation, the DHCP server includes some differences that could prevent its operation with BootP clients. However, many DHCP servers such as Windows™ NT DHCP server are backward compatible with BootP protocol and can be used for device configuration.
- By default, the duration between BootP/DHCP requests is one second (configured by the BootPDelay *ini* file parameter). By default, the number of requests is three (configured by the BootPRetries *ini* file parameter). Both parameters can also be set using the BootP command line switches (refer to Section "Using Command Line Switches" on page 175).

**Figure 2-1: Startup Process**


## 2.2 Device Firmware

The device runs two distinct software programs:

- Boot firmware:** Boot-loader firmware (also known as flash software), which resides on the device's non-volatile memory. When the device is reset, Boot firmware is initialized and the operational software is loaded from a TFTP server or integral non-volatile memory. Boot firmware is also responsible for obtaining the device's IP parameters and *ini* file name (used to obtain the device's configuration parameters) using integral BootP or DHCP clients. The Boot firmware version can be viewed in the Web interface (refer to the 'Device Information' page in the device's *User's Manual*). The last step the Boot firmware performs is to invoke the operational firmware.

- **Operational firmware file:** The operational firmware, in the form of a *cmp* file (the software image file) is supplied in the software package contained on the CD accompanying the device. This *cmp* file contains the device's main software, providing all the features described in this manual. The *cmp* file is usually burnt on the device's non-volatile memory so that it does not need to be externally loaded each time the device is reset.

## 2.3 Configuration Parameters and Files

The device's configuration is located in two types of files:

- **Initialization file:** An initialization (*ini*) text file containing the device's configuration parameters (referred to as *ini* file parameters and *ini* file table parameters). This file carries the file name extension \*.ini.
- **Auxiliary files:** Contains the raw data used for various tasks such as Call Progress Tones. These files carry the file name extension \*.dat.

These files are stored in the device's non-volatile memory (i.e., flash) and are set to factory defaults when shipped to the customer. The device starts up initially with this default configuration. Subsequently, these files can be modified and reloaded (to flash memory) using any of the following methods:

- BootP/TFTP during startup process (refer to "Using BootP / DHCP" on page 19).
- Device's Embedded Web server (refer to the device's *User's Manual*).
- Automatic Update facility (refer to "Automatic Update Facility on page 39" on page 24).



### Notes:

- When configuring the device using the Web interface, loading an *ini* file is unnecessary. There is also no need for a TFTP server.
- When configuring the device using SNMP, the only configuration in the *ini* file is the IP address for the SNMP traps.
- For information on the structure of the *ini* file, refer to the device's *User's Manual*.

## 2.4 Using BootP / DHCP

The device uses the Bootstrap Protocol (BootP) and the Dynamic Host Configuration Protocol (DHCP) to automatically obtain its networking parameters and configuration after it is reset. BootP and DHCP are also used to provide the IP address of a TFTP server on the network and files (*cmp* and *ini*) to be loaded into memory.

DHCP is a communication protocol that automatically assigns IP addresses from a central point. BootP is a protocol that enables a device to discover its own IP address. Both protocols have been extended to enable the configuration of additional parameters specific to the device.



**Note:** BootP is typically used to initially configure the device. Thereafter, BootP is no longer required as all parameters can be stored in the device's non-volatile memory and used when BootP is inaccessible. BootP can be used later to change the device's IP address. For a description on using the BootP application, refer to "BootP/TFTP Configuration Utility" on page 167.

## 2.4.1 BootP/DHCP Server Parameters

BootP and DHCP can be used to provision the following parameters (included in the BootP/DHCP reply):

- **IP Address, Subnet Mask:** Mandatory parameters sent to the device every time a BootP/DHCP process occurs.
- **Default Gateway IP Address:** Optional parameter sent to the device only if configured in the BootP/DHCP server.
- **TFTP Server IP Address:** Optional parameter containing the IP address of the TFTP server from which the software (*cmp*) and *ini* files are loaded.
- **DNS Server IP Address (Primary and Secondary):** Optional parameters containing the IP addresses of the primary and secondary DNS servers. These parameters are available only for DHCP.
- **Syslog Server IP Address:** Optional parameter sent to the device only if configured. This parameter is available only for DHCP.
- **SIP Server IP Address:** Two optional parameters (primary and secondary SIP server) sent to the device only if configured. These parameters are available only for DHCP.
- **Firmware File Name:** Optional parameter containing the name of the firmware file to be loaded to the device using TFTP.
- **Configuration ini File Name:** Optional parameter containing the name of the *ini* file (proprietary configuration file with the extension \*.ini) to be loaded to the device using TFTP. When the device detects that this parameter field is defined in BootP, it initiates a TFTP process to load the file to the device. The new configuration contained in the *ini* file can be stored in the device's integral non-volatile memory. Whenever the device is reset and no BootP reply is received, or the *ini* file name is missing in the BootP reply, the device uses the previously stored *ini* file.

## 2.4.2 DHCP Host Name Support

When the device is configured to use DHCP (in the Web interface's 'Application Settings' page or using the *ini* file parameter DHCPEnable = 1), it attempts to contact the local DHCP server to obtain the networking parameters (IP address, subnet mask, default gateway, primary/secondary DNS server, and two SIP server addresses). These network parameters have a 'time limit'. After the time limit expires, the device must 'renew' its lease from the DHCP server.

To detect the device's IP address, follow one of the procedures below:

- 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 equal to the last six digits of the MAC address converted to decimal representation. If the DHCP server registers this host name to a DNS server, you can access the device (through a Web browser) using a URL of `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`.
- After physically resetting the device, its IP address is displayed in the 'Client Info' column in the BootP/TFTP configuration utility (refer to "BootP/TFTP Configuration Utility" on page 167).
- Use a serial communication software (refer to 'Assigning an IP Address Using the CLI' in the device's Fast Track Guide).
- Contact your System Administrator.

**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 happens while calls are in progress, they are not automatically rerouted to the new network address (since this function is beyond the scope of a VoIP device). Therefore, administrators are advised to configure DHCP servers to allow renewal of IP addresses.
- If, during operation, the device's IP address is changed as a result of a DHCP renewal, the device is automatically reset.
- If the device's network cable is disconnected and reconnected, a DHCP renewal is performed (to verify that the device is still connected to the same network). When DHCP is enabled, 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, DHCP 'option 60' is contained. If the device is reset from the Web interface or SNMP, only a single DHCP sequence containing 'option 60' is sent. If DHCP procedure is used, the new device IP address, allocated by the DHCP server, must be detected.

### 2.4.3 Microsoft DHCP/BootP Server

The device can be configured using any third-party BootP server, including Microsoft™ Windows™ DHCP server, to provide the device with an IP address and other initial parameter configurations.

To configure the Windows DHCP Server for assigning an IP address to BootP clients (i.e., device>s), add a Reservation for each BootP client. The Reservation builds an association between the MAC address (12 digits) provided in the accompanying device documentation and the IP address. Windows Server provides the IP address based on the device's MAC address in the BootP request frame. For information on how to add a reservation, view the "Managing Client Reservations Help" topic in the DHCP console.

To configure Windows DHCP server to provide Boot File information to BootP clients, edit the BootP Table in the DHCP console. The BootP Table must be enabled from the **Action > Properties** dialog box (select the option 'Show the BootP Table Folder' and then click **OK**). For information on editing the BootP Table, view the "Manage BOOTP and remote access clients" Help topic in the DHCP console.

The following parameters must be specified:

- Local IP address - the device's IP address
- Subnet mask
- Gateway IP address - default Gateway IP address
- BootP File name - Optional (refer to the following Note)



**Note:** The BootP File field is typically not used. This field is only used for software upgrade.

## 2.4.4 Using BootP



**Note:** For a description on using the BootP application, refer to "BootP/TFTP Configuration Utility" on page 167.

### 2.4.4.1 Upgrading the Device

When upgrading the device (loading new software to the device) using the BootP/TFTP configuration utility:

- From version 4.4 to version 4.4 or to any higher version, the device retains its configuration (*ini* file). However, the auxiliary files (CPT, logo, etc.) may be erased.
- From version 4.6 to version 4.6 or to any higher version, the device retains its configuration (*ini* file) and auxiliary files (CPT, logo, etc.).

You can also use the Web interface's Software Upgrade Wizard to upgrade the device (refer to the device's *User's Manual*).



**Note:** To save the *cmp* file to the device's non-volatile memory, use the **-fb** command line switch. If the file is not saved, the device reverts to the old software version after the next reset. For information on using command line switches, refer to "Using Command Line Switches" on page 175.

### 2.4.4.2 Vendor Specific Information Field

The device uses the Vendor Specific Information field in the BootP request to provide device-related initial startup information. The BootP/TFTP utility displays this information in the Log window's 'Client Info' column (refer to "BootP/TFTP Configuration Utility" on page 167).



**Note:** This option is not available on DHCP servers.

The Vendor Specific Information field is disabled by default. To enable this feature, configure the *ini* file parameter ExtBootPReqEnable (refer to the device's *User's Manual*) or use the **-be** command line switch (refer to "BootP/TFTP Configuration Utility" on page 167).



The following table details the Vendor Specific Information field according to device:

**Table 2-1: Vendor Specific Information Field**

Tag #	Description	Value	Length
<b>220</b>	Device Type	<ul style="list-style-type: none"> <li>▪ <b>#02</b> = TP-1610; IPM-1610; Mediant 2000; IPmedia 2000</li> <li>▪ <b>#08</b> = TP-6310; IPM-6310</li> <li>▪ <b>#09</b> = IPmedia 3000; Mediant 3000; Mediant 1000; Mediant 600</li> <li>▪ <b>#13</b> = MP-124</li> <li>▪ <b>#14</b> = MP-118</li> <li>▪ <b>#15</b> = MP-114</li> <li>▪ <b>#16</b> = MP-112</li> <li>▪ <b>#24</b> = TP-8410; IPM-8410</li> </ul>	1
<b>221</b>	Current IP Address	XXX.XXX.XXX.XXX	4
<b>222</b>	Burned Boot Software Version	X.XX	4
<b>223</b>	Burned <i>cmp</i> Software Version	XXXXXXXXXXXX	12
<b>224</b>	Geographical Address	0-31	1
<b>225</b>	Chassis Geographical Address	0-31	1
<b>228</b>	Indoor / Outdoor	<ul style="list-style-type: none"> <li>▪ <b>#0</b> = Indoor</li> <li>▪ <b>#1</b> = Outdoor</li> </ul> <b>Notes:</b> <ul style="list-style-type: none"> <li>▪ Applicable only to analog interfaces (Mediant 1000 and MediaPack).</li> <li>▪ Indoor is applicable only to FXS interfaces; Outdoor is applicable only to FXO interfaces.</li> </ul>	1
<b>230</b>	Analog Channels	<ul style="list-style-type: none"> <li>▪ <b>#2</b> = MediaPack only</li> <li>▪ <b>#4, #8, #24</b> = Mediant 1000 and MediaPack</li> <li>▪ <b>#12, #16, #20</b> = Mediant 1000 only</li> </ul> <b>Note:</b> Applicable only to analog interfaces (Mediant 1000 and MediaPack).	1

The following table shows an example of the structure of the vendor specific information field:

**Table 2-2: Structure of the Vendor Specific Information Field**

Vendor-Specific Information Code	Length Total	Tag Num	Length	Value	Tab Num	Length	Value	Tag Num	Length	Value (1)	Value (2)	Value (3)	Value (4)	Tag End
42	12	220	1	2	225	1	1	221	4	10	2	70	1	255

### 2.4.4.3 Selective BootP

The Selective BootP mechanism allows the integral BootP client to filter out unsolicited BootP replies. This can be beneficial for environments where more than one BootP server exists and only one BootP server is used to configure AudioCodes devices. The command line switch **-bs** is used to activate this feature (refer to "Using Command Line Switches" on page 175).

## 2.5 Automatic Update Mechanism

The device can automatically update its *cmp*, *ini*, and auxiliary files. These files can be stored on any standard Web, FTP, or NFS server and can be loaded periodically to the device using HTTP, HTTPS, FTP, or NFS. This mechanism can be used even for devices that are installed behind NAT and firewalls. The Automatic Update mechanism is applied per file. For a detailed description on automatic configuration, refer to "Automatic Device Configuration" on page 95.

The Automatic Update mechanism is activated by the following:

- Upon device start-up (refer to the "Startup Process" on page 17).
- At a user-defined time of day (e.g., 18:00), using the *ini* file parameter *AutoUpdatePredefinedTime*. This option is disabled by default.
- At fixed intervals (e.g., every 60 minutes), using the *ini* file parameter *AutoUpdateFrequency*. This option is disabled by default.
- Upon start-up, but before the device is operational, if the Secure Startup feature is enabled (refer to "Loading Files Securely (Disabling TFTP)" on page 99).

**Notes:**

- The Automatic Update mechanism assumes that the external 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, adjust the update frequency (using the parameter `AutoUpdateFrequency`).
- When HTTP or HTTPS is used, the device queries the Web server(s) for the requested files. The *ini* file is loaded only if it was modified since the last automatic update. The *cmp* file is loaded only if its version is different from the version currently stored on the device's non-volatile memory. All other auxiliary files (e.g., CPT) are updated only once. To update a previously loaded auxiliary file, you must update the parameter containing its URL.
- To load different configurations (*ini* files) for specific devices, add the string '<MAC>' to the URL. This mnemonic is replaced with the device's hardware MAC address, resulting in an *ini* file name request that contains the device's MAC address.
- To automatically update the *cmp* file, use the parameter `CmpFileURL` to specify its name and location. As a precaution (to protect the device from an accidental update), by default, the Automatic Update mechanism doesn't apply to the *cmp* file. Therefore, to enable it set the parameter `AutoUpdateCmpFile` to 1.
- By default, when using the Auto Update mechanism to load an *ini* file, a device reset is not performed automatically. If the *ini* file contains a tables or parameters which are not applied on-the-fly (i.e., a device reset is required), the *ini* file must include `ResetNow = 1` to initiate a device reset.
- By default, when using the Auto Update mechanism to load an *ini* file, all parameters that are not included in the file are set to their default values. However, it is possible to configure only certain parameters while retaining the settings of all the other device parameters. To achieve this, the *ini* file must include the parameter `SetDefaultOnINIFileProcess = 0`.

The following *ini* file example can be used to activate the Automatic Update mechanism.

```
# DNS is required for specifying domain names in URLs
DnsPriServerIP = 10.1.1.11
# Load an extra configuration ini file using HTTP
IniFileURL = 'http://webserver.corp.com/AudioCodes/inifile.ini'
# Load Call Progress Tones file using HTTPS
CptFileUrl = 'https://10.31.2.17/usa tones.dat'
# Load Voice Prompts file using FTPS with user 'root' and password 'wheel'
VPFileUrl = 'ftps://root:wheel@ftpserver.corp.com/vp.dat'
# Update every day at 03:00 AM
AutoUpdatePredefinedTime = '03:00'
# Note: The cmp file isn't updated since it's disabled by default
(AutoUpdateCmpFile).
```

The following example illustrates how to utilize Automatic Updates for deploying a device with minimum manual configuration.

➤ **To utilize Automatic Updates for deploying the device with minimal manual configuration, take these 6 steps:**

1. Setup a Web server (e.g., <http://www.corp.com>) where all configuration files are located.
2. For each device, pre-configure the following parameter (DHCP / DNS are assumed):

```
IniFileURL = 'http://www.corp.com/master configuration.ini'
```

3. Create a file named *master\_configuration.ini* with the following text:

```
# Common configuration for all devices
# -----
CptFileURL = 'http://www.corp.com/call_progress.dat'
# Check for updates every 60 minutes
AutoUpdateFrequency = 60
# Additional configuration per device
# -----
# Each device loads a file named after its MAC address,
# (e.g., config_00908F033512.ini)
IniFileURL = 'http://www.corp.com/config <MAC>.ini'
# Reset the device after configuration is updated.
# The device resets after all of the files are processed.
ResetNow = 1
```

You can modify the *master\_configuration.ini* file (or any of the *config\_<MAC>.ini* files) at any time. The device queries for the latest version every 60 minutes and applies the new settings immediately.

4. 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> for the Automatic Update mechanism.
5. The configuration URL can be provided using the Voice Configuration Menu (refer to the *Fast Track Guide*). (Applicable only to Analog devices.)
6. To load configuration files from an NFS server, the NFS file system parameters should be defined in the configuration *ini* file. The following is an example of an *ini* file for loading files from NFS servers using NFS version 2.

```
# Define NFS servers for Automatic Update
[ NFSServers ]
FORMAT NFSServers Index = NFSServers HostOrIP,
NFSServers RootPath, NFSServers NfsVersion;
NFSServers 1 = 10.31.2.10, /usr/share, 2 ;
NFSServers 2 = 192.168.100.7, /d/shared, 2 ;
[ \NFSServers ]
CptFileUrl = 'file://10.31.2.10/usr/share/public/usa tones.dat'
VpFileUrl =
'file://192.168.100.7/d/shared/audiocodes/voiceprompt.dat'
```

## 3 Command-Line Interface Based Management

The command line interface (CLI) is available through a Telnet or an Secure SHell (SSH) session with the device's management interface. It mainly allows you to view various information regarding device setup and performance.

### 3.1 Starting a CLI Management Session

The procedure below describes how to start a CLI session.

➤ **To start a CLI management session, take these 2 steps:**

1. Enable CLI (Telnet or SSH) using one of the following methods:
  - **Web interface:** In the 'Application Settings' page (**Configuration** tab > **Network Settings** menu > **Application Settings** page item), set the parameter 'Embedded Telnet Server' to 'Enable (Unsecured)' or 'Enable Secured (SSL)' (refer to the device's *User's Manual*).
  - **ini file:** Configure the following *ini* file parameters as shown below:
    - ◆ TelnetServerEnable = 1
    - ◆ SSHServerEnable = 1
  - **SNMP:** set the objects acSysTelnetSSHServerEnable and acSysTelnetServerEnable to 'enable' (1).
2. Establish a Telnet or SSH session with the device's OAMP IP address, using the system's user name and password (as shown below).
  - User name: Admin
  - Password: Admin

A Telnet or SSH client application must be running on the management PC. Most operating systems, including Microsoft Windows, include a built-in Telnet client, which can be activated from the command prompt. SSH, however, must be installed separately. See the following link for a discussion of available SSH client implementations: [http://en.wikipedia.org/wiki/Comparison\\_of\\_SSH\\_clients](http://en.wikipedia.org/wiki/Comparison_of_SSH_clients).

After logging in, the current directory (root), available commands (**SHoW**, **PING**), available subdirectories, and a welcome message are displayed at the CLI prompt:

```
login: Admin
password:
AudioCodes device ready. Type "exit" to close the connection.
MGmt/ CONFIguration/ IPNetworking/ TPApp/ BSP/
SHoW PING
/>
```


**Notes:**

- By default, CLI access is disabled for security.
- The user name and password are case-sensitive.
- Only the primary User Account (which has Security Administration access level - 200) can be used to access the device using Telnet/CLI. This user is defined in the device's Web interface.
- The CLI user name and password can be changed by the device's administrator. Multiple users can be defined.

## 3.2 CLI Navigation Concepts

Commands are organized in subdirectories. When the CLI session starts, you are located in the 'root' directory, which contains only two commands: **SH**ow and **P**ING. To access a subdirectory, type its name, and then press <Enter>. To move back one directory, type two periods (..), and then press <Enter>. Alternatively, if you know the full path to a command inside one of the subdirectories, the short format can be used to run it directly. For example, the **PER**formance command in the MGmt subdirectory may be run directly by typing:

```
/mg/perf
```

The CLI commands can be entered in an abbreviated format by typing only the letters shown in upper case (i.e., capital letters). For example, the **CH**angePassWord command can be entered by typing **chpw**.

## 3.3 Commands

The following table summarizes the CLI commands and their options.

**Table 3-1: Summary of CLI Commands**

Purpose	Commands	Description
Help	<b>h</b>	Displays the help for a specific command, action, or parameter.
Navigation	<b>cd</b>	Enters another directory.
	<b>cd root</b>	Navigates to the root directory (/).
	<b>..</b>	Goes up one level.
	<b>exit</b>	Terminates the CLI session.
Status	<b>show</b>	Displays the device's operational status.
	<b>ping</b>	Sends Internet Control Message Protocol (ICMP) echo request packets from the device to a defined IP address.
Configuration	<b>/conf/scp</b>	Sets a value for the specific parameter.
	<b>/conf/rfs</b>	Restores factory defaults.
	<b>/conf/sar</b>	Restarts the device.

### 3.3.1 General Commands

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

**Table 3-2: General CLI Commands**

Command	Short Format	Arguments	Description
<b>SHow</b>	sh	info   mgcp   tdm   dsp   ip   log	Displays operational data. The individual sub-commands are documented below.  <b>Note:</b> The <i>mgcp</i> argument is not applicable to SIP devices.
<b>SHow INFO</b>	sh info	-	Displays device hardware information, versions, uptime, temperature reading, and the last reset reason.
<b>SHow HW</b>	sh hw	--	Displays Mediant 3000 system information: power status, High-Availability status, and fan information.
<b>SHow TDM</b>	sh tdm	status   perf   summary	Displays the alarm status and performance statistics for E1/T1 trunks.
<b>SHow DSP</b>	sh dsp	status   perf	Displays status and version for each DSP device, along with overall performance statistics.
<b>SHow IP</b>	sh ip	conf   perf   route	Displays IP interface status and configuration, along with performance statistics.  <b>Note:</b> The display format may change according to the configuration.
<b>SHow LOG</b>	sh log	[stop]	Displays (or stops displaying) Syslog messages in the CLI session.
<b>PING</b>	ping	[-n count] [-l size] [-w timeout] [-p cos] ip-address	Sends ICMP echo request packets to a specified IP address. <ul style="list-style-type: none"> <li>count: number of packets to send.</li> <li>size: payload size in each packet.</li> <li>timeout: time (in seconds) to wait for a reply to each packet.</li> <li>cos: Class-of-Service (as per 802.1p) to use.</li> </ul>

**Example:**

```

/>sh ?
Usage:
  SHow INFO      Displays general device information
  SHow TDM       Displays PSTN-related information
  SHow DSP       Displays DSP resource information
  SHow IP        Displays information about IP interfaces

/>sh info
Board type: gateway SDH, firmware version 5.20.000.017
Uptime: 0 days, 0 hours, 3 minutes, 54 seconds
Memory usage: 63%

```



```

Temperature reading: 39 C
Last reset reason:
Board was restarted due to issuing of a reset from Web interface
Reset Time : 7.1.2000 21.51.13
/>sh tdm status
Trunk 00: Active
Trunk 01: Active
Trunk 02: Active
Trunk 03: Active
Trunk 04: Active
Trunk 05: Active
Trunk 06: Active
Trunk 07: Active
Trunk 08: Active
Trunk 09: Active
Trunk 10: Active
Trunk 11: Active
Trunk 12: Active
Trunk 13: Active
Trunk 14: Active
Trunk 15: Not Configured
Trunk 16: Not Configured
Trunk 17: Not Configured
Trunk 18: Not Configured
Trunk 19: Not Configured
Trunk 20: Not Configured
Trunk 21: Not Configured
/>sh tdm perf
DS1 Trunk Statistics (statistics for 948 seconds):
Trunk #      B-Channel      Call count  RTP packet  RTP packet  Activity
            utilization      Tx          Rx          Seconds
0            1            1          2865         0           57
1            0            0           0         0            0
2           20           20        149743         0          3017
3            0            0           0         0            0
4            0            0           0         0            0
5            0            0           0         0            0
6            0            0           0         0            0
7            0            0           0         0            0
8            0            0           0         0            0
9            0            0           0         0            0
10           0            0           0         0            0
11           0            0           0         0            0
12           0            0           0         0            0
13           0            0           0         0            0
14           0            0           0         0            0
/>sh dsp status
DSP firmware: 491096AE8 Version:0540.03 Used=0 Free=480 Total=480
DSP device 0: Active      Used=16   Free= 0   Total=16
DSP device 1: Active      Used=16   Free= 0   Total=16
DSP device 2: Active      Used=16   Free= 0   Total=16
DSP device 3: Active      Used=16   Free= 0   Total=16
DSP device 4: Active      Used=16   Free= 0   Total=16
DSP device 5: Active      Used=16   Free= 0   Total=16
DSP device 6: Inactive
DSP device 7: Inactive
DSP device 8: Inactive
DSP device 9: Inactive
DSP device 10: Inactive
DSP device 11: Inactive
DSP device 12: Active     Used=16   Free= 0   Total=16
DSP device 13: Active     Used=16   Free= 0   Total=16
DSP device 14: Active     Used=16   Free= 0   Total=16
DSP device 15: Active     Used=16   Free= 0   Total=16
DSP device 16: Active     Used=16   Free= 0   Total=16
DSP device 17: Active     Used=16   Free= 0   Total=16
DSP device 18: Inactive
PSEC - DSP firmware: AC491IPSEC Version: 0540.03
CONFERENCE - DSP firmware: AC491256C Version: 0540.03

```

```

/>sh dsp perf
DSP Statistics (statistics for 968 seconds):
Active DSP resources: 480
Total DSP resources: 480
DSP usage %: 100
/>sh ip perf
Networking Statistics (statistics for 979 seconds):
IP KBytes TX: 25
IP KBytes RX: 330
IP KBytes TX per second: 0
IP KBytes RX per second: 1
IP Packets TX: 1171
IP Packets RX: 5273
IP Packets TX per second: 3
IP Packets RX per second: 12
Peak KByte/s TX in this interval: 18
Peak KByte/s RX in this interval: 4
Discarded packets: 186
DHCP requests sent: 0
IPSec Security Associations: 0
/>/mg/perf reset
Done.
/>sh ip perf
Networking Statistics (statistics for 2 seconds):
IP KBytes TX: 2
IP KBytes RX: 4
IP KBytes TX per second: 0
IP KBytes RX per second: 1
IP Packets TX: 24
IP Packets RX: 71
IP Packets TX per second: 3
IP Packets RX per second: 12
Peak KByte/s TX in this interval: 18
Peak KByte/s RX in this interval: 4
Discarded packets: 0
DHCP requests sent: 0
IPSec Security Associations: 0
/>sh ip conf
Interface  IP Address          Subnet Mask          Default Gateway
-----
OAM        10.4.64.13           55.255.0.0           10.4.0.1
Media      10.4.64.13           255.255.0.0          10.4.0.1
Control    10.4.64.13           255.255.0.0          10.4.0.1
MAC address: 00-90-8f-04-5c-e9
/>sh ip route
Destination      Mask                Gateway             Intf  Flags
-----
0.0.0.0          0.0.0.0             10.4.0.1            OAM   A S
10.4.0.0         255.255.0.0         10.4.64.13          OAM   A L
127.0.0.0        255.0.0.0           127.0.0.1           AR    S
127.0.0.1        255.255.255.255     127.0.0.1           A L   H
Flag legend: A=Active R=Reject L=Local S=Static E=rEdirect
M=Multicast
               B=Broadcast H=Host I=Invalid
End of routing table, 4 entries displayed.
/>ping 10.31.2.10
Ping process started for address 10.31.2.10. Process ID - 27.
Reply from 10.31.2.10: bytes=0 time<0ms
Reply from 10.31.2.10: bytes=0 time<0ms
Reply from 10.31.2.10: bytes=0 time<0ms
Reply from 10.31.2.10: bytes=0 time<0ms
Ping statistics for 10.31.2.10:
Packets: Sent = 4, Received = 4, Lost 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

### 3.3.2 Configuration Commands

The commands under the 'CONFIGuration' directory, query and modify the current device configuration. The following commands are available:

**Table 3-3: Configuration CLI Commands**

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

### 3.3.3 Management Commands

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

**Table 3-4: CLI Management Command**

Command	Short Format	Arguments	Description
<b>/MGmt/PERFormance</b>	/mg/perf	basic   control   dsp   net   ds1   ss7   reset	Displays performance statistics. The <i>reset</i> argument clears all statistics to zero.

### 3.3.4 PSTN Commands

The commands under the 'PSTN' directory allows you to perform various PSTN actions.

**Table 3-5: PSTN CLI Command**

Command	Short Format	Arguments	Description
<b>PstnLoopCommands</b>	PS/PH/PLC	<TrunkId> <LoopCode> <BChannel>	Activates a loopback on a specific trunk and B-channel. For loopback on the entire trunk, set BChanne=(-1). LoopCode: <ul style="list-style-type: none"> <li>0 = NO_LOOPS</li> <li>1 = REMOTE_LOOP (whole trunk only)</li> <li>2 = LINE_PAYLOAD_LOOP (whole trunk only)</li> <li>3 = LOCAL_ALL_CHANNELS_LOOP (whole trunk only)</li> <li>4 = LOCAL_SINGLE_CHANNEL_LOOP</li> <li>10 = PRBS_START (whole trunk only)</li> <li>11 = PRBS_STOP (whole trunk only)</li> </ul>

**Reader's Notes**

## 4 SNMP-Based Management

Simple Network Management Protocol (SNMP) is a standards-based network control protocol for managing elements in a network. The SNMP Manager (usually implemented by a network Management System (NMS) or an Element Management System (EMS)) connects to an SNMP Agent (embedded on a remote Network Element (NE)) to perform network element Operation, Administration, Maintenance, and Provisioning (OAMP).

Both the SNMP Manager and the NE refer to the same database to retrieve information or configure parameters. This database is referred to as the Management Information Base (MIB), and is a set of statistical and control values. Apart from the standard MIBs documented in IETF RFCs, SNMP additionally enables the use of proprietary MIBs, containing non-standard information set (specific functionality provided by the Network Element).

Directives, issued by the SNMP Manager to an SNMP Agent, consist of the identifiers of SNMP variables (referred to as MIB object identifiers or MIB variables) along with instructions to either get the value for that identifier, or set the identifier to a new value (configuration). The SNMP Agent can also send unsolicited events towards the EMS, called SNMP traps.

The definitions of MIB variables supported by a particular agent are incorporated in descriptor files, written in Abstract Syntax Notation (ASN.1) format, made available to EMS client programs so that they can become aware of MIB variables and their usage.

The device contains an embedded SNMP Agent supporting both general network MIBs (such as the IP MIB), VoP-specific MIBs (such as RTP) and AudioCodes' proprietary MIBs (acBoard, acGateway, acAlarm, and other MIBs) enabling a deeper probe into the interworking of the device. All supported MIB files are supplied to customers as part of the release.

### 4.1 SNMP Standards and Objects

This section discusses the SNMP standards and SNMP objects.

#### 4.1.1 SNMP Message Standard

Four types of SNMP messages are defined:

- **Get:** A request that returns the value of a named object.
- **Get-Next:** A request that returns the next name (and value) of the "next" object supported by a network device given a valid SNMP name.
- **Set:** A request that sets a named object to a specific value.
- **Trap:** A message generated asynchronously by network devices. It notifies the network manager of a problem apart from the polling of the device.

Each of these message types fulfills a particular requirement of network managers:

- **Get Request:** Specific values can be fetched via the "get" request to determine the performance and state of the device. Typically, many different values and parameters can be determined via SNMP without the overhead associated with logging into the device, or establishing a TCP connection with the device.
- **Get Next Request:** Enables the SNMP standard network managers to "walk" through all SNMP values of a device (via the "get-next" request) to determine all names and values that a device supports.

- **Get-Bulk:** Extends the functionality of GETNEXT by allowing multiple values to be returned for selected items in the request.
- This is accomplished by beginning with the first SNMP object to be fetched, fetching the next name with a "get-next", and repeating this operation.
- **Set Request:** The SNMP standard provides a action method for a device (via the "set" request) to accomplish activities such as disabling interfaces, disconnecting users, clearing registers, etc. This provides a way of configuring and controlling network devices via SNMP.
- **Trap Message:** The SNMP standard furnishes a mechanism for a device to "reach out" to a network manager on their own (via the "trap" message) to notify or alert the manager of a problem with the device. This typically requires each device on the network to be configured to issue SNMP traps to one or more network devices that are awaiting these traps.

The above message types are all encoded into messages referred to as "Protocol Data Units" (PDUs) that are interchanged between SNMP devices.

## 4.1.2 SNMP MIB Objects

The SNMP MIB is arranged in a tree-structure, similar to a disk directory structure of files. The top level SNMP branch begins with the ISO "internet" directory, which contains four main branches:

- **"mgmt" SNMP branch:** Contains the standard SNMP objects usually supported (at least in part) by all network devices.
- **"private" SNMP branch:** Contains those "extended" SNMP objects defined by network equipment vendors.
- **"experimental" and "directory" SNMP branches:** Also defined within the "internet" root directory, are usually devoid of any meaningful data or objects.

The "tree" structure described above is an integral part of the SNMP standard, though the most pertinent parts of the tree are the "leaf" objects of the tree that provide actual management data regarding the device. Generally, SNMP leaf objects can be partitioned into two similar but slightly different types that reflect the organization of the tree structure:

- **Discrete MIB Objects:** Contain one precise piece of management data. These objects are often distinguished from "Table" items (below) by adding a ".0" (dot-zero) extension to their names. The operator must merely know the name of the object and no other information.
- **Table MIB Objects:** Contain multiple pieces of management data. These objects are distinguished from "Discrete" items (above) by requiring a "." (dot) extension to their names that uniquely distinguishes the particular value being referenced. The "." (dot) extension is the "instance" number of an SNMP object. For "Discrete" objects, this instance number is zero. For "Table" objects, this instance number is the index into the SNMP table. SNMP tables are special types of SNMP objects, which allow parallel arrays of information to be supported. Tables are distinguished from scalar objects, such that tables can grow without bounds. For example, SNMP defines the "ifDescr" object (as a standard SNMP object) that indicates the text description of each interface supported by a particular device. Since network devices can be configured with more than one interface, this object can only be represented as an array.

By convention, SNMP objects are always grouped in an "Entry" directory, within an object with a "Table" suffix. (The "ifDescr" object described above resides in the "ifEntry" directory contained in the "ifTable" directory).



### 4.1.3 SNMP Extensibility Feature

One of the principal components of an SNMP manager is a MIB Compiler, which allows new MIB objects to be added to the management system. When a MIB is compiled into an SNMP manager, the manager is made "aware" of new objects that are supported by agents on the network. The concept is similar to adding a new schema to a database.

Typically, when a MIB is compiled into the system, the manager creates new folders or directories that correspond to the objects. These folders or directories can typically be viewed with a "MIB Browser", which is a traditional SNMP management tool incorporated into virtually all network management systems.

The act of compiling the MIB allows the manager to know about the special objects supported by the agent and access these objects as part of the standard object set.

## 4.2 Carrier-Grade Alarm System

The basic alarm system has been extended to a carrier-grade alarm system. A carrier-grade alarm system provides a reliable alarm reporting mechanism that takes into account element management system (EMS) outages, network outages, and transport mechanism such as SNMP over UDP.

A carrier-grade alarm system is characterized by the following:

- The device allows an EMS to determine which alarms are currently active in the device. That is, the device maintains an active alarm table.
- The device allows an EMS to detect lost alarms and clear notifications. [sequence number in trap, current sequence number MIB object]
- The device allows an EMS to recover lost alarm raise and clear notifications [maintains a log history]
- The device sends a cold start trap to indicate that it is starting. This allows the EMS to synchronize its view of the device's active alarms.

When the SNMP alarm traps are sent, the carrier-grade alarm system does not add or delete alarm traps as part of the feature. This system provides the mechanism for viewing of history and current active alarm information.

### 4.2.1 Active Alarm Table

The device maintains an active alarm table to allow an EMS to determine which alarms are currently active in the device. Two views of the active alarm table are supported by the agent:

- `acActiveAlarmTable` in the enterprise `AcAlarm`
- `alarmActiveTable` and `alarmActiveVariableTable` in the IETF standard `AcAlarm` MIB (rooted in the MIB tree)

The `acActiveAlarmTable` is a simple, one-row per alarm table that is easy to view with a MIB browser.

The Alarm MIB is currently a draft standard and therefore, has no OID assigned to it. In the current software release, the MIB is rooted in the experimental MIB subtree. In a future release, after the MIB has been ratified and an OID assigned to it, it is to be moved to the official OID.

## 4.2.2 Alarm History

The device maintains a history of alarms that have been raised and traps that have been cleared to allow an EMS to recover any lost raise or clear traps. Two views of the alarm history table are supported by the agent:

- **acAlarmHistoryTable** in the enterprise AcAlarm - a simple, one-row per alarm table, that is easy to view with a MIB browser.
- **nImLogTable** and **nImLogVariableTable** in the standard NOTIFICATION-LOG-MIB

## 4.3 Topology MIB - Objects



**Note:** This subsection is applicable only to AudioCodes' 3000 Series, Mediant 1000, and Mediant 600 devices.

### 4.3.1 Physical Entity - RFC 2737

The following groups are supported:

- **entityPhysical group:** Describes the physical entities managed by a single agent.
- **entityMapping group:** Describes the associations between the physical entities, logical entities, interfaces, and non-interface ports managed by a single agent.
- **entityGeneral group:** Describes general system attributes shared by potentially all types of entities managed by a single agent.
- **entityNotifications group:** Contains status indication notifications.

### 4.3.2 IF-MIB - RFC 2863

The following interface types are presented in the ifTable:

- **ethernetCsmacd(6):** for all Ethernet-like interfaces, regardless of speed, as per RFC 3635 (Gigabit Ethernet for 3000 Series devices)
- **ds1(18):** DS1-MIB
- **voiceFXO(101):** Voice Foreign Exchange Office. (Applicable only to Mediant 1000.)
- **voiceFXS(102):** Voice Foreign Exchange Station. (Applicable only to Mediant 1000.)
- **sonet(39):** SONET-MIB. (Applicable only to the 3000 Series.)
- **ds3(30):** DS3-MIB. (Applicable only to the 3000 Series.)

The numbers in the brackets above refer to the IANA's interface-number.

For each interface type, the following objects are supported:

**Table 4-1: DS1 Digital Interfaces**

ifTable	Value
ifDescr	Digital DS1 interface.
ifType	ds1(18).
ifMtu	Constant zero.
ifSpeed	DS1 = 1544000, or E1 = 2048000, according to dsx1LineType
ifPhysAddress	The value of the Circuit Identifier [dsx1CircuitIdentifier]. If no Circuit Identifier has been assigned this object should have an octet string with zero length.
ifAdminStatus	Trunk's Lock & Unlock during run time. In initialization process we need to refer the Admin-Status parameter.
ifOperStatus	Up or Down, according to the operation status.
ifLastChange	The value of sysUpTime at the time the interface entered its current operational state.
ifXTable	Value
ifName	Digital# acTrunkIndex
ifLinkUpDownTrapEnable	Set to enabled(1)
ifHighSpeed	Speed of line in Megabits per second: 2
ifConnectorPresent	Set to true(1) normally, except for cases such as DS1/E1 over AAL1/ATM where false(2) is appropriate
ifCounterDiscontinuityTime	Always zero.

**Table 4-2: BRI Interfaces (Applicable to Mediant 1000 & Mediant 600)**

ifTable	Value
ifDescr	BRI interface
ifType	isdns(75)
ifMtu	Constant zero
ifSpeed	144000
ifPhysAddress	Octet string with zero length
ifAdminStatus	Trunk's Lock & Unlock during run time. In initialization process, refer to the Admin-Status parameter.
ifOperStatus	Up or Down according to the operation status.
ifLastChange	The value of sysUpTime at the time the interface entered its current operational state.
ifXTable	Value
ifName	BRI port no. #

ifTable	Value
ifDescr	BRI interface
ifLinkUpDownTrapEnable	Set to enabled (1)
ifHighSpeed	Speed of line in megabits per second.
ifPromiscuousMode	Non promiscuous mode (1)
ifConnectorPresent	Set to true (1) normally
ifCounterDiscontinuityTime	Always zero

**Table 4-3: Ethernet (Gigabit for 3000 Series) Interface**

ifTable & ifXTable	Value
ifIndex	Constructed as defined in the device's Index format.
ifDescr	Ethernet interface.
ifType	ethernetCsmacd(6)
ifMtu	1500
ifSpeed	acSysEthernetFirstPortSpeed in bits per second (Applicable only to Mediant 1000) 0 since it's GBE - refer to ifHighSpeed (Applicable only to 3000 Series).
ifPhysAddress	00-90-8F plus acSysIdSerialNumber in hex. Will be same for both dual ports.
ifAdminStatus	Always UP. [Read Only] - Write access is not required by the standard. Support for 'testing' is not required.
ifOperStatus	Up or Down corresponding to acAnalogFxsFxoType where Unknown is equal to Down.
ifLastChange	The value of sysUpTime at the time the interface entered its current operational state.
ifInOctets	The number of octets in valid MAC frames received on this interface, including the MAC header and FCS. This does include the number of octets in valid MAC Control frames received on this interface.
ifInUcastPkts	As defined in IfMIB.
ifInDiscards	As defined in IfMIB.
ifInErrors	The sum for this interface of dot3StatsAlignmentErrors, dot3StatsFCSErrors, dot3StatsFrameTooLongs, and dot3StatsInternalMacReceiveErrors.
ifInUnknownProtos	As defined in IfMIB.
ifOutOctets	The number of octets transmitted in valid MAC frames on this interface, including the MAC header and FCS. This does include the number of octets in valid MAC Control frames transmitted on this interface.
ifOutUcastPkts	As defined in IfMIB.
ifOutDiscards	As defined in IfMIB.
ifOutErrors	The sum for this interface of: dot3StatsSQETestErrors,

ifTable & ifXTable	Value
	dot3StatsLateCollisions, dot3StatsExcessiveCollisions, dot3StatsInternalMacTransmitErrors and dot3StatsCarrierSenseErrors.
<b>ifName</b>	Ethernet (Gigabit for 3000 Series) port #1 or# 2
<b>ifInMulticastPkts</b>	As defined in IfMIB.
<b>ifInBroadcastPkts</b>	As defined in IfMIB.
<b>ifOutMulticastPkts</b>	As defined in IfMIB.
<b>ifOutBroadcastPkts</b>	As defined in IfMIB.
<b>ifHCInOctets</b> <b>ifHCOctets</b>	64-bit versions of counters. Required for ethernet-like interfaces that are capable of operating at 20 Mb/s or faster, even if the interface is currently operating at less than 20 Mb/s.
<b>ifHCInUcastPkts</b> <b>ifHCInMulticastPkts</b> <b>ifHCInBroadcastPkts</b> <b>ifHCOctetsUcastPkts</b> <b>ifHCOctetsMulticastPkts</b> <b>ifHCOctetsBroadcastPkts</b>	64-bit versions of packet counters. Required for ethernet-like interfaces that are capable of operating at 640 Mb/s or faster, even if the interface is currently operating at less than 640 Mb/s. Therefore, will be constant zero.
<b>ifLinkUpDownTrapEnable</b>	Refer to [RFC 2863]. Default is 'enabled'
<b>ifHighSpeed</b>	<b>3000 Series:</b> 1000 <b>Mediant 1000:</b> 10 or 100 according to acSysEthernetFirstPortSpeed
<b>ifPromiscuousMode</b>	Constant False. [R/O]
<b>ifConnectorPresent</b>	Constant True.
<b>ifAlias</b>	An 'alias' name for the interface as specified by a network manager (NVM)
<b>ifCounterDiscontinuityTime</b>	As defined in IfMIB.

Table 4-4: SONET /SDH Interfaces (3000 Series Only)

ifTable & ifXTable	Value
<b>ifDescr</b>	SONET/SDH interface. Module #n Port #n
<b>ifType</b>	sonet(39).
<b>ifMtu</b>	Constant zero.

ifTable & ifXTable	Value
<b>ifSpeed</b>	155520000
<b>ifPhysAddress</b>	The value of the Circuit Identifier. If no Circuit Identifier has been assigned this object should have an octet string with zero length.
<b>ifAdminStatus</b>	Read-only access -- Always UP.
<b>ifOperStatus</b>	The value testing(3) is not used. This object assumes the value down(2), if the objects sonetSectionCurrentStatus and sonetLineCurrentStatus have any other value than sonetSectionNoDefect(1) and sonetLineNoDefect(1), respectively.
<b>ifLastChange</b>	The value of sysUpTime at the time the interface entered its current operational state.
<b>ifName</b>	SONET /SDH port no. n
<b>ifLinkUpDownTrapEnable</b>	Set to enabled(1)
<b>ifHighSpeed</b>	Speed of line in Megabits per second: 155
<b>ifConnectorPresent</b>	Set to true(1) normally, except for cases such as DS1/E1 over AAL1/ATM where false(2) is appropriate
<b>ifCounterDiscontinuityTime</b>	Always zero.

**Table 4-5: DS3 Interfaces (3000 Series Only)**

ifTable & ifXTable	Value
<b>ifDescr</b>	DS3 interface, Module no.#d, Port no.#d
<b>ifType</b>	Ds3(30).
<b>ifMtu</b>	Constant zero.
<b>ifSpeed</b>	44736000
<b>ifPhysAddress</b>	The value of the Circuit Identifier. If no Circuit Identifier has been assigned this object should have an octet string with zero length.
<b>ifAdminStatus</b>	Read-only access -- Always UP.
<b>ifOperStatus</b>	The value testing(3) is not used. This object assumes the value down(2), if the objects dsx3LineStatus has any other value than dsx3NoAlarm(1).
<b>ifLastChange</b>	The value of sysUpTime at the time the interface entered its current operational state.
<b>ifName</b>	DS3 port no. n
<b>ifLinkUpDownTrapEnable</b>	Set to enabled(1)
<b>ifHighSpeed</b>	Speed of line in Megabits per second: 45
<b>ifConnectorPresent</b>	Set to true(1)
<b>ifCounterDiscontinuityTime</b>	Always zero.

## 4.4 Cold Start Trap

The device technology supports a cold start trap to indicate that the device is starting. This allows the EMS to synchronize its view of the device's active alarms. In fact, two different traps are sent at start-up:

- **Standard coldStart trap:** iso(1).org(3).dod(6).internet(1). snmpV2(6). snmpModules(3). snmpMIB(1). snmpMIBObjects(1). snmpTraps(5). coldStart(1) sent at system initialization.
- **Enterprise acBoardEvBoardStarted:** generated at the end of system initialization. This is more of an "application-level" cold start sent after all the initializing process is over and all the modules are ready.

## 4.5 Performance Measurements

Performance measurements are available for a third-party performance monitoring system through an SNMP interface. These can be polled at scheduled intervals by an external poller or utility in the management server or other off-board systems.

The device provides performance measurements in the form of two types:

- **Gauges:** Gauges represent the current state of activities on the device. Gauges unlike counters can decrease in value and like counters, can increase. The value of a gauge is the current value or a snapshot of the current activity on the device at that moment.
- **Counters:** Counters always increase in value and are cumulative. Counters, unlike gauges, never decrease in value unless the server is reset and then the counters are zeroed.

The device performance measurements are provided by several proprietary MIBs (located under the acPerformance subtree):

**iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).AudioCodes(5003).acPerformance(10).**

The performance monitoring MIBs all have an identical structure, which includes two major subtrees:

- **Configuration:** allows configuration of general attributes of the MIB and specific attributes of the monitored objects
- **Data**

The monitoring results are presented in tables. There are one or two indices in each table. If there are two indices, the first is a sub-set in the table (e.g., trunk number) and the second (or a single where there is only one) index represents the interval number (present - 0, previous - 1, and the one before - 2).

The MIBs include:

- **acPMMedia:** media-related (voice) monitoring such as RTP and DSP.
- **acPMControl:** Control Protocol-related monitoring such as connections, commands.
- **acPMAnalog:** Analog channels off-hook state. (Applicable only to Analog devices.)
- **acPMPSTN:** PSTN-related monitoring such as channel use, trunk utilization. (Applicable only to Digital devices.)
- **acPMSystem:** general (system-related) monitoring.
- **acPMMediaServer:** for Media Server specific monitoring. (Applicable only to 3000 Series devices.)

The log trap `acPerformanceMonitoringThresholdCrossing` (non-alarm) is sent every time the threshold of a Performance Monitored object is crossed. The severity field is 'indeterminate' when the crossing is above the threshold and 'cleared' when it goes back under the threshold. The 'source' varbind in the trap indicates the object for which the threshold is being crossed.

### 4.5.1 Total Counters

The counter's attribute 'total' accumulates counter values since the device's most recent restart. The user can reset the total's value by setting the Reset-Total object.

Each MIB module has its own Reset Total object, as follows:

- **PM-Analog:** `acPMAAnalogConfigurationResetTotalCounters` (Applicable only to Analog devices)
- **PM-Control:** `acPMControlConfigurationResetTotalCounters`
- **PM-Media:** `acPMMediaConfigurationResetTotalCounters`
- **PM-PSTN:** `acPMPSTNConfigurationResetTotalCounters` (Applicable only to Digital devices, except IPmedia 3000/IPM-8410)
- **PM-System:** `acPMSystemConfigurationResetTotalCounters`

## 4.6 TrunkPack-VoP Series Supported MIBs

The device contains an embedded SNMP agent supporting the listed MIBs below. A description in HTML format for all supported MIBs can be found in the MIBs directory in the release package.

- **The Standard MIB (MIB-2):** The various SNMP values in the standard MIB are defined in RFC 1213. The standard MIB includes various objects to measure and monitor IP activity, TCP activity, UDP activity, IP routes, TCP connections, interfaces, and general system description.
  - The standard `icmpStatsTable` and `icmpMsgStatsTable` under MIB-2 support ICMP statistics for both IPv4 and IPv6.
  - The `inetCidrRouteTable` (from the standard IP-FORWARD-MIB) supports both IPv4 and IPv6.
  - `sysDescr` - support for 8410 Series Blades:
    - ◆ "MG3K-8410" for TP-8410 with Mediant 3000
    - ◆ "MS3K-8410-VIDEO" for video on IPM-8410 with IPmedia 3000

The `sysDescr` field is a textual description of the entity. This value includes the full name and version identification of the system's hardware type, software operating system, and networking software. This field must only contain printable ASCII characters
  - `entPhysicalTable` - support for 8410 Series Blades.



**Note:** For Mediant 3000/TP-6310 and Mediant 2000/TP-1610: In the `ipCidrRouteIfIndex`, the IF MIB indices are not referenced. Instead, the index used is related to one of the IP interfaces in the blade: (1) OAMP, (2) Media, (3) Control. When there is only one interface, the only index is OAMP (1). Refer to Getting Started with VLANs and Multiple IPs in the device's User's Manual.



- **System MIB (under MIB-2):** The standard system group: sysDescr, sysObjectID, sysUpTime, sysContact, sysName, sysLocation, and sysServices. You can replace the value of sysObjectID.0 with variable value using the *ini* file parameter that calls SNMPSysOid. This parameter is polled during the startup and overwrites the standard sysObjectID. SNMPSysName is an administratively assigned name for this managed node. By convention, this is the node's fully-qualified domain name. If the name is unknown, the value is the zero-length string.
- **RTP MIB:** The RTP MIB is supported according to RFC 2959. It contains objects relevant to the RTP streams generated and terminated by the device and to the RTCP information related to these streams.



**Note:** The inverse tables are not supported.

- **Notification Log MIB:** Standard MIB (RFC 3014 - iso.org.dod.internet.mgmt.mib-2) supported as part of AudioCodes' implementation of Carrier Grade Alarms.
- **Alarm MIB:** IETF MIB (RFC 3877) supported as part of the implementation of Carrier Grade Alarms. This MIB is a new standard and therefore, is under the audioCodes.acExperimental branch.
- **SNMP Target MIB:** (RFC 2273) allows for configuration of trap destinations and trusted managers.
- **SNMP MIB:** (RFC 3418) allows support for the coldStart and authenticationFailure traps.
- **SNMP Framework MIB:** (RFC 3411).
- **SNMP Usm MIB:** (RFC 3414) implements the user-based Security Model.
- **SNMP Vacm MIB:** (RFC 3415) implements the view-based Access Control Model.
- **SNMP Community MIB:** (RFC 3584) implements community string management.
- **ipForward MIB:** (RFC 2096) - fully supported.
- **RTCP-XR:** (RFC) implements the following partial support:
  - The rtpXrCallQualityTable is fully supported.
  - In the rtpXrHistoryTable, support of the RCQ objects is provided only with no more than 3 intervals, 15 minutes long each.
  - Supports the rtpXrVoipThresholdViolation trap.
- **ds1 MIB:** supports the following (Applicable only to Digital devices, except IPmedia 3000/8410):
  - dsx1ConfigTable: partially supports the following objects with SET and GET applied:
    - ◆ dsx1LineCoding
    - ◆ dsx1LoopbackConfig
    - ◆ dsx1LineStatusChangeTrapEnable
    - ◆ dsx1CircuitIdentifier

All other objects in this table support GET only.

- dsx1CurrentTable
- dsx1IntervalTable
- dsx1TotalTable
- dsx1LineStatusChange trap

■ **ds3 MIB:** (RFC 3896) supports the following (Applicable only to the 3000 Series, except IPmedia 3000/8410):

- dsx3ConfigTable: refer to the MIB version supplied by AudioCodes for limits on specific objects. The table includes the following objects:
  - ◆ TimerElapsed
  - ◆ ValidIntervals
- dsx3LineStatusChange  
The following tables (RFC 2496) are supported
  - ◆ dsx3CurrentTable
  - ◆ dsx3IntervalTable
  - ◆ dsx3TotalTable

Proprietary MIB objects that are connected to the SONET/SDH configuration:

■ **In the acSystem MIB** (Applicable only to the 3000 Series):

- ◆ acSysTransmissionType: sets the transmission type to optical or DS3 (T3).

■ **SONET MIB:** (RFC 3592) implements the following partial support (Applicable only to the 3000 Series):

- In the SonetMediumTable, the following objects are supported:
  - ◆ SonetMediumType
  - ◆ SonetMediumLineCoding
  - ◆ SonetMediumLineType
  - ◆ SonetMediumCircuitIdentifier
  - ◆ sonetMediumLoopbackConfig
- In the SonetSectionCurrentTable, the following objects are supported:
  - ◆ IsonetSectionCurrentStatus
  - ◆ sonetSectionCurrentESs
  - ◆ sonetSectionCurrentSESs
  - ◆ sonetSectionCurrentSEFSs
  - ◆ sonetSectionCurrentCVs
- In the SonetLineCurrentTable, the following objects are supported:
  - ◆ sonetLineCurrentStatus
  - ◆ sonetLineCurrentESs
  - ◆ sonetLineCurrentSESs
  - ◆ sonetLineCurrentCVs
  - ◆ sonetLineCurrentUASs
- sonetSectionIntervalTable

- sonetLineIntervalTable

The following proprietary MIB objects are associated with the SONET/SDH configuration (Applicable only to the 3000 Series):

■ **Traps (all defined in the AcBoard MIB):**

- acSonetSectionLOFAlarm
- acSonetSectionLOSAlarm
- acSonetLineAISAlarm
- acSonetLineRDIAAlarm
- acSonetIfHwFailureAlarm

(Refer to the MIB for more details.)

■ **In the acPSTN MIB:**

- acSonetSDHTable: currently has one entry (acSonetSDHFbrGrpMappingType) for selecting a low path mapping type. Relevant only for PSTN applications. (Refer to the MIB for more details.)

■ **In the acSystem MIB:**

- acSysTransmissionType: sets the transmission type to optical or DS3 (T3).

In addition to the standard MIBs, the complete product series contains proprietary MIBs:

- **AC-TYPES MIB:** lists the known types defined by the complete product series. This is referred to by the sysObjectID object in the MIB-II.

In version 4.8, the SR-COMMUNITY-MIB was changed to the standard snmpCommunity MIB.

In version 5.0, support was added for the standard SNMP-USER-BASED-SM-MIB.

- **AcBoard MIB:** This proprietary MIB contains objects related to configuration of the device and channels as well as to run-time information. Through this MIB, users can set up the device's configuration parameters, reset the device, monitor the device's operational robustness and quality of service during run-time and receive traps.



**Note:** The AcBoard MIB is being phased out and is being replaced by an updated proprietary MIBs.

The AcBoard MIB includes the following groups:

- **channelStatus**
- **acTrap**

Each AudioCodes proprietary MIB contains a Configuration subtree for configuring the related parameters. In some, there also are Status and Action subtrees.

- **AcAnalog MIB** (Applicable only to Analog devices)
- **acControl MIB**
- **acMedia MIB**
- **acSystem MIB**
- **acSysEthernetStatusTable** - Ethernet relevant information. (Applicable only to

Mediant 3000 with TP-8410 Blade)

- **acSysModuleTable** (Applicable only to 8410 Blade Series)
- **acIPMediaChannelsresourcesTable** - IPMedia channels information such as Module ID and DSP Channels Reserved (Applicable only to Mediant 1000)
- **acPSTN MIB** (Applicable only to Digital devices)
- **acSS7 MIB** (Applicable only to the 3000 Series and the 2000 Series devices)
- **acGateway MIB:** This proprietary MIB contains objects related to configuration of the SIP device. This MIB complements the other AudioCodes proprietary MIBs.

The acGateway MIB includes the following groups:

- **Common:** parameters common to both SIP and H.323.
- **SIP:** SIP only parameters.
- **AcAlarm:** This is a proprietary carrier-grade alarm MIB. It is a simpler implementation of the notificationLogMIB and the IETF suggested alarmMIB (both also supported in all AudioCodes devices).

The acAlarm MIB has the following groups:

- **ActiveAlarm:** straight forward (single indexed) table listing all currently active Alarms together with their bindings (the Alarm bindings are defined in acAlarm. acAlarmVarbinds and also in acBoard.acTrap. acBoardTrapDefinitions. oid\_1\_3\_6\_1\_4\_1\_5003\_9\_10\_1\_21\_2\_0).
- **acAlarmHistory:** straight forward (single indexed) table listing all recently raised Alarms together with their bindings (the Alarm bindings are defined in acAlarm. acAlarmVarbinds and also in acBoard.acTrap. acBoardTrapDefinitions. oid\_1\_3\_6\_1\_4\_1\_5003\_9\_10\_1\_21\_2\_0).

The table size can be altered via:

notificationLogMIB.notificationLogMIBObjects.nlmConfig.nlmConfigGlobalEntryLimit or  
notificationLogMIB.notificationLogMIBObjects.nlmConfig.nlmConfigLogTable.nlmConfigLogEntry.nlmConfigLogEntryLimit.

The table size (i.e., number of contained alarms) can be as follows:

- Digital devices: Any value between 10 and 1,000 (default is 500)
- MediaPack devices: Any value between 10 and 100 (default is 100)



#### Notes:

- A detailed explanation of each parameter can be viewed in the MIB Description field.
- Not all groups in the MIB are implemented. Refer to version release notes.
- MIB Objects that are marked as 'obsolete' are not implemented.
- When a parameter is Set to a new value via SNMP, the change may affect device functionality immediately or may require that the device be soft reset for the change to take effect. This depends on the parameter type.
- The current (updated) device configuration parameters are configured on the device provided the user doesn't load an *ini* file to the device after reset. Loading an *ini* file after reset overrides the updated parameters.

## 4.7 Traps

Full proprietary trap definitions and trap Varbinds are found in AcBoard MIB and AcAlarm MIB. For a detailed inventory of traps, refer to "SNMP Traps" on page 67.



**Note:** All traps are sent from the SNMP port (default 161). This is part of the NAT traversal solution.

The following proprietary traps are supported by the device:

**Table 4-6: Proprietary Traps**

Trap	Description
<b>acBoardFatalError</b>	Sent whenever a fatal device error occurs.
<b>acBoardConfigurationError</b>	Sent when the device's settings are invalid. The trap contains a message stating/detailing/explaining the invalid setting.
<b>acBoardTemperatureAlarm</b>	Sent when the device exceeds its temperature limits. <b>Note:</b> Applicable only to 2000 and 3000 Series devices.
<b>acBoardEvResettingBoard</b>	Sent after the device resets.
<b>acBoardEvBoardstarted</b>	Sent after the device is successfully restored and initialized following reset.
<b>acFeatureKeyError</b>	Sent to relay Feature Key errors etc.
<b>acgwAdminStateChange</b>	Sent when Graceful Shutdown commences and ends.
<b>acBoardEthernetLinkAlarm</b>	Sent when the Ethernet link(s) is down.
<b>acActiveAlarmTableOverflow</b>	Sent when an active alarm cannot be entered into the Active Alarm table because the table is full.
<b>acAudioProvisioningAlarm</b>	Sent if the device is unable to provision its audio.
<b>acOperationalStateChange</b>	Sent if the operational state of the node goes to disabled; cleared when the operational state of the node goes to enabled.
<b>acKeepAlive</b>	Part of the NAT traversal mechanism. If the STUN application in the device detects a NAT, this trap is sent on a regular time laps - 9/10 of the acSysSTUNBindingLifeTime object. The AdditionalInfo1 varbind has the MAC address of the device.
<b>acBoardCallResourcesAlarm</b>	Sent when no free channels are available.
<b>acBoardControllerFailureAlarm</b>	Sent when the Proxy is not found or registration fails. Internal routing table may be used for routing.

Trap	Description
<b>acBoardOverloadAlarm</b>	Sent when there is an overload in one or some of the system's components.
<b>acNATTraversalAlarm</b>	Sent when the NAT is placed in front of a device and is identified as a symmetric NAT. It is cleared when a non-symmetric NAT or no NAT replace the symmetric one.
<b>acEnhancedBITStatus</b>	Sent for the status of the BIT (Built In Test). The information in the trap contains blade hardware elements being tested and their status. The information is presented in the additional info fields.
<b>acPerformanceMonitoringThresholdCrossing</b>	Sent every time the threshold of a Performance Monitored object is crossed. The severity field is 'indeterminate' when the crossing is above the threshold and 'cleared' when it goes back under the threshold. The 'source' varbind in the trap indicates the object for which the threshold is being crossed.
<b>acHTTPDownloadResult</b>	Sent upon success or failure of the HTTP Download action.
<b>acSS7LinkStateChangeAlarm</b>	Sent when the operational state of the SS7 link becomes BUSY. The alarm is cleared when the operational state of the link becomes SERVICE or OFFLINE. <b>Note:</b> Applicable only to the 3000 Series and 2000 Series devices.
<b>acSS7LinkInhibitStateChangeAlarm</b>	Sent when the SS7 link becomes inhibited (local or remote). The alarm is cleared when the link becomes uninhibited - local AND remote. Note that this alarm is raised for any change in the remote or local inhibition status. <b>Note:</b> Applicable only to the 3000 Series and 2000 Series devices.
<b>acSS7LinkBlockStateChangeAlarm</b>	Sent when the SS7 link becomes blocked (local or remote). The alarm is cleared when the link becomes unblocked - local AND remote. Note that this alarm is raised for any change in the remote or local blocking status. <b>Note:</b> Applicable only to the 3000 Series and 2000 Series devices.
<b>acSS7LinkCongestionStateChangeAlarm</b>	Sent when if the SS7 link becomes congested (local or remote). The alarm is cleared when the link becomes uncongested - local AND remote. Note that this alarm is raised for any change in the remote or local congestion status. <b>Note:</b> Applicable only to the 3000 Series and 2000 Series devices.

Trap	Description
<b>acSS7LinkSetStateChangeAlarm</b>	Sent when the operational state of the SS7 linkset becomes BUSY. The alarm is cleared when the operational state of the linkset becomes SERVICE or OFFLINE. <b>Note:</b> Applicable only to the 3000 Series and 2000 Series devices.
<b>acSS7RouteSetStateChangeAlarm</b>	Sent when the operational state of the SS7 routeset becomes BUSY. The alarm is cleared when the operational state of the routeset becomes SERVICE or OFFLINE. <b>Note:</b> Applicable only to the 3000 Series and 2000 Series devices.
<b>acSS7SNSetStateChangeAlarm</b>	Sent when the operational state of the SS7 node becomes BUSY. The alarm is cleared when the operational state of the node becomes IN-SERVICE or OFFLINE. <b>Note:</b> Applicable only to the 3000 Series and 2000 Series devices.
<b>acFanTrayAlarm</b>	Sent when a fault occurs in the fan tray or a fan tray is missing. <b>Note:</b> Applicable only to the 3000 Series devices.
<b>acPowerSupplyAlarm</b>	Sent when a fault occurs in one of the power supply (PS) modules or a PS module is missing. <b>Note:</b> Applicable only to the 3000 Series devices.
<b>acPEMAlarm</b>	Sent when a fault occurs in one of the PEM modules or a PEM module is missing. <b>Note:</b> Applicable only to the 3000 Series devices.
<b>acSAMissingAlarm</b>	Sent when the SA module is missing or non operational. <b>Note:</b> Applicable only to the 3000 Series devices.
<b>acUserInputAlarm</b>	Sent when the input dry contact is short circuited; cleared when the circuit is reopened. <b>Note:</b> Applicable only to the 3000 Series devices.
<b>acHASystemFaultAlarm</b>	Sent when the High Availability (HA) system is faulty (i.e., no HA functionality). <b>Note:</b> Applicable only to Mediant 3000 HA devices.
<b>acHASystemConfigMismatchAlarm</b>	Sent when the configuration of the modules in the HA system is not identical, causing instability. <b>Note:</b> Applicable only to Mediant 3000 HA devices.
<b>acHASystemSwitchOverAlarm</b>	Sent when a switchover from the active to the redundant module has occurred. <b>Note:</b> Applicable only to Mediant 3000 HA devices.
<b>acSonetSectionLOFAlarm</b>	SONET section Loss of Frame alarm. <b>Note:</b> Applicable only to the 3000 Series devices.

Trap	Description
<b>acSonetSectionLOSAAlarm</b>	SONET section Loss of Signal alarm. <b>Note:</b> Applicable only to the 3000 Series devices.
<b>acSonetLineAISAlarm</b>	SONET Line AIS alarm. <b>Note:</b> Applicable only to the 3000 Series devices.
<b>acSonetLineRDIAAlarm</b>	SONET Line RDI alarm. <b>Note:</b> Applicable only to the 3000 Series devices.
<b>acDChannelStatus</b>	Non-alarm trap sent at the establishment, re-establishment or release of LAPD link with its peer connection occurs. The trap is sent with one of the following textual descriptions: <ul style="list-style-type: none"> <li>▪ D-channel synchronized</li> <li>▪ D-channel not-synchronized</li> </ul> <b>Note:</b> Applicable only to the Digital devices.
<b>acGWSASEmergencyModeAlarm</b>	Sent by the Stand-Alone Survivability (SAS) application when switching from "Normal" mode to "Emergency" mode. This alarm is cleared once the SAS returns to "Normal" mode. <b>Note:</b> Applicable only to MediaPack, Mediant 1000, and Mediant 2000.
<b>acTMInconsistentRemoteAndLocalPLLStatus</b>	Inconsistent Remote and Local PLL status. <b>Note:</b> Applicable only to Mediant 3000.
<b>acTMReferenceStatus</b>	Timing manager reference status. <b>Note:</b> Applicable only to Mediant 3000.
<b>acTMReferenceChange</b>	Timing manager reference change. <b>Note:</b> Applicable only to Mediant 3000.

In addition to the traps listed in the table above, the device also supports the following standard traps:

- **authenticationFailure**
- **coldStart**
- **linkDown**
- **linkup**
- **entConfigChange**
- **dsx1LineStatusChange** (Applicable only to Digital devices)
- **dsx3LineStatusChange** (Applicable only to the 3000 Series devices)



## 4.8 SNMP Interface Details

This subsection describes details of the SNMP interface needed when developing an Element Management System (EMS) for any of the TrunkPack-VoP Series products, or to manage a device with a MIB browser.

There are several alternatives for SNMP security:

- SNMPv2c community strings
- SNMPv3 User-based Security Model (USM) users
- SNMP encoded over IPSec (for more details, refer to "IPSec and IKE" on page 101)
- Various combinations of the above

Currently, both SNMP and *ini* file commands and downloads are not encrypted. For *ini* file encoding, refer to the device's *User's Manual*.

### 4.8.1 SNMP Community Names

By default, the device uses a single, read-only community string of "public" and a single read-write community string of "private". Up to five read-only community strings and up to five read-write community strings, and a single trap community string can be configured. Each community string must be associated with one of the following predefined groups:

**Table 4-7: SNMP Predefined Groups**

Group	Get Access	Set Access	Sends Traps
ReadGroup	Yes	No	Yes
ReadWriteGroup	Yes	Yes	Yes
TrapGroup	No	No	Yes

#### 4.8.1.1 Configuring Community Strings via the Web

For detailed information on configuring community strings via the Web interface, refer to the device's *User's Manual*.

#### 4.8.1.2 Configuring Community Strings via the ini File

The following *ini* file parameters are used to configure community strings:

- SNMPREADONLYCOMMUNITYSTRING\_<x> = '#####'
- SNMPREADWRITECOMMUNITYSTRING\_<x> = '#####'

Where <x> is a number from 0 through 4. Note that the '#' character represents any alphanumeric character. The maximum length of the string is 20 characters.

#### 4.8.1.3 Configuring Community Strings via SNMP

To configure community strings, the EMS must use the standard snmpCommunityMIB. To configure the trap community string, the EMS must also use the snmpTargetMIB.

➤ **To add a read-only v2user community string, take these 2 steps:**

1. Add a new row to the snmpCommunityTable with CommunityName v2user.
2. Add a row to the vacmSecurityToGroupTable for SecurityName v2user, GroupName ReadGroup and SecurityModel snmpv2c.

➤ **To delete the read-only v2user community string, take these 3 steps:**

1. If v2user is being used as the trap community string, follow the procedure for changing the trap community string. (See below.)
2. Delete the snmpCommunityTable row with CommunityName v2user.
3. Delete the vacmSecurityToGroupTable row for SecurityName v2user, GroupName ReadGroup and SecurityModel snmpv2c.

➤ **To add a read-write v2admin community string, take these 2 steps:**

1. Add a new row to the snmpCommunityTable with CommunityName v2admin.
2. Add a row to the vacmSecurityToGroupTable for SecurityName v2admin, GroupName ReadWriteGroup and SecurityModel snmpv2c.

➤ **To delete the read-write v2admin community string, take these 2 steps:**

1. If v2admin is being used as the trap community string, follow the procedure for changing the trap community string. (See below.)
2. Delete the snmpCommunityTable row with a CommunityName of v2admin and GroupName of ReadWriteGroup.

➤ **To change the only read-write community string from v2admin to v2mgr, take these 4 steps:**

1. Follow the procedure above to add a read-write community string to a row for v2mgr.
2. Set up the EM such that subsequent set requests use the new community string, v2mgr.
3. If v2admin is being used as the trap community string, follow the procedure to change the trap community string. (See below.)
4. Follow the procedure above to delete a read-write community name in the row for v2admin.

The following procedure assumes that a row already exists in the snmpCommunityTable for the new trap community string. The trap community string can be part of the TrapGroup, ReadGroup, or ReadWriteGroup. If the trap community string is used solely for sending traps (recommended), then it should be made part of the TrapGroup.

➤ **To change the trap community string, take these 3 steps:**

1. Add a row to the vacmSecurityToGroupTable with these values: SecurityModel=2, SecurityName=the new trap community string, GroupName=TrapGroup, ReadGroup or ReadWriteGroup. The SecurityModel and SecurityName objects are row indices.



**Note:** You must add GroupName and RowStatus on the same set.

2. Modify the SecurityName field in the appropriate row of the snmpTargetParamsTable.
3. Remove the row from the vacmSecurityToGroupTable with SecurityName=the old trap community string.

## 4.8.2 SNMPv3 USM Users

You can configure up to 10 User-based Security Model (USM) users (referred to as *SNMPv3 user*). Each SNMPv3 user can be configured for one of the following security levels:

**Table 4-8: SNMPv3 Security Levels**

Security Levels	Authentication	Privacy
noAuthNoPriv(1)	none	none
authNoPriv(2)	MD5 or SHA-1	none
authPriv(3)	MD5 or SHA-1	DES, 3DES, AES128, AES192, or AES256

Each SNMPv3 user must be associated with one of the predefined groups listed in the following table:

**Table 4-9: SNMPv3 Predefined Groups**

Group	Get Access	Set Access	Sends Traps	Security Level
ReadGroup1	Yes	No	Yes	noAuthNoPriv(1)
ReadWriteGroup1	Yes	Yes	Yes	noAuthNoPriv(1)
TrapGroup1	No	No	Yes	noAuthNoPriv(1)
ReadGroup2	Yes	No	Yes	authNoPriv(2)
ReadWriteGroup2	Yes	Yes	Yes	authNoPriv(2)
TrapGroup2	No	No	Yes	authNoPriv(2)
ReadGroup3	Yes	No	Yes	authPriv(3)
ReadWriteGroup3	Yes	Yes	Yes	authPriv(3)
TrapGroup3	No	No	Yes	authPriv(3)

### 4.8.2.1 Configuring SNMPv3 Users via the ini File

Use the SNMPUsers *ini* file table parameter to add, modify, and delete SNMPv3 users. The SNMPUsers *ini* table is a hidden parameter. Therefore, when you load the *ini* file to the device using the Web interface, the table is not included in the generated file.

**Table 4-10: SNMPv3 Table Columns Description**

Parameter	Description	Default
Row number	Table index. Its valid range is 0 to 9.	N/A
SNMPUsers_Username	Name of the v3 user. Must be unique. The maximum length is 32 characters.	N/A
SNMPUsers_AuthProtocol	Authentication protocol to be used for this user. Possible values are 0 (none), 1 (MD5), 2 (SHA-1)	0
SNMPUsers_PrivProtocol	Privacy protocol to be used for this user. Possible values are 0 (none), 1 (DES), 2 (3DES), 3 (AES128), 4 (AES192), 5 (AES256)	0
SNMPUsers_AuthKey	Authentication key.	""
SNMPUsers_PrivKey	Privacy key.	""
SNMPUsers_Group	The group that this user is associated with. Possible values are 0 (read-only group), 1 (read-write group), and 2 (trap group). The actual group will be ReadGroup<sl>, ReadWriteGroup<sl> or TrapGroup<sl> where <sl> is the SecurityLevel (1=noAuthNoPriv, 2=authNoPriv, 3=authPriv)	0

Keys can be entered in the form of a text password or in the form of a localized key in hex format. If using a text password, then it should be at least 8 characters in length. Below is an example showing the format of a localized key:

26:60:d8:7d:0d:4a:d6:8c:02:73:dd:22:96:a2:69:df

The following sample configuration creates three SNMPv3 USM users.

```
[ SNMPUsers ]
FORMAT SNMPUsers_Index = SNMPUsers_Username,
SNMPUsers_AuthProtocol, SNMPUsers_PrivProtocol, SNMPUsers_AuthKey,
SNMPUsers_PrivKey, SNMPUsers_Group;
SNMPUsers 0 = v3user, 0, 0, -, -, 0;
SNMPUsers 1 = v3admin1, 1, 0, myauthkey, -, 1;
SNMPUsers 2 = v3admin2, 2, 1, myauthkey, myprivkey, 1;
[ \SNMPUsers ]
```

The example above creates three SNMPv3 users:

- The user v3user is set up for a security level of noAuthNoPriv(1) and is associated with ReadGroup1.
- The user v3admin1 is setup for a security level of authNoPriv(2), with authentication protocol MD5. The authentication text password is "myauthkey" and the user is associated with ReadWriteGroup2.
- The user v3admin2 is setup for a security level of authPriv(3), with authentication protocol SHA-1 and privacy protocol DES. The authentication text password is "myauthkey", the privacy text password is "myprivkey", and the user is associated with ReadWriteGroup3.

### 4.8.2.2 Configuring SNMPv3 Users via SNMP

To configure SNMPv3 users, the EMS must use the standard snmpUsmMIB and the snmpVacmMIB.

➤ **To add a read-only, noAuthNoPriv SNMPv3 user, v3user, take these 3 steps:**

1. Clone the row with the same security level. After the clone step, the status of the row will be notReady(3).
2. Activate the row. That is, set the row status to active(1).
3. Add a row to the vacmSecurityToGroupTable for SecurityName v3user, GroupName ReadGroup1 and SecurityModel usm(3).



**Note:** A row with the same security level (noAuthNoPriv) must already exist in the usmUserTable. (see the usmUserTable for details).

➤ **To delete the read-only, noAuthNoPriv SNMPv3 user, v3user, take these 3 steps:**

1. If v3user is associated with a trap destination, follow the procedure for associating a different user to that trap destination. (See below.)
2. Delete the vacmSecurityToGroupTable row for SecurityName v3user, GroupName ReadGroup1 and SecurityModel usm.
3. Delete the row in the usmUserTable for v3user.

➤ **To add a read-write, authPriv SNMPv3 user, v3admin1, take these 4 steps:**

1. Clone the row with the same security level.
2. Change the authentication key and privacy key.
3. Activate the row. That is, set the row status to active(1).
4. Add a row to the vacmSecurityToGroupTable for SecurityName v3admin1, GroupName ReadWriteGroup3 and SecurityModel usm(3).



**Note:** A row with the same security level (authPriv) must already exist in the usmUserTable (see the usmUserTable for details).

- **To delete the read-write, authPriv SNMPv3 user, v3admin1, take these 3 steps:**
  1. If v3admin1 is associated with a trap destination, follow the procedure for associating a different user to that trap destination. (See below.)
  2. Delete the vacmSecurityToGroupTable row for SecurityName v3admin1, GroupName ReadWriteGroup1 and SecurityModel usm.
  3. Delete the row in the usmUserTable for v3admin1.

### 4.8.3 Trusted Managers

By default, the SNMP agent accepts Get and Set requests from any IP address, as long as the correct community string is used in the request. Security can be enhanced implementing *Trusted Managers*. A Trusted Manager is an IP address from which the SNMP agent accepts and processes Get and Set requests. An element management can be used to configure up to five Trusted Manager.

The concept of Trusted Managers is considered to be a weak form of security and therefore is not a required part of SNMPv3 security, which uses authentication and privacy. Trusted Managers for the devices' SNMP agent are applicable only for SNMPv2c users. An exception to this is when the community string is not the default string ('public'/'private'), at which time Trusted Managers are applicable for SNMPV2c users alongside SNMPv3 users.



**Note:** If trusted managers are defined, then all community strings works from all trusted managers, i.e., there is no way to associate a community string with specific trusted managers.

#### 4.8.3.1 Configuring Trusted Managers via ini File

To set the Trusted Managers table from start up, write the following in the *ini* file:

```
SNMPTRUSTEDMGR_X = D.D.D.D
```

Where X is any integer between 0 and 4 (0 sets the first table entry, 1 sets the second and so on), and D is an integer between 0 and 255.

#### 4.8.3.2 Configuring Trusted Managers via SNMP

To configure Trusted Managers, the EMS must use the SNMP-COMMUNITY-MIB and snmpCommunityMIB and the snmpTargetMIB.

The procedure below assumes the following: at least one configured read-write community; currently no Trusted Managers; TransportTag for columns for all snmpCommunityTable rows are currently empty.

- **To add the first Trusted Manager, take these 3 steps:**
  1. Add a row to the snmpTargetAddrTable with these values: Name=mgr0, TagList=MGR, Params=v2cparams.
  2. Add a row to the snmpTargetAddrExtTable table with these values: Name=mgr0, snmpTargetAddrTMask=255.255.255.255:0. The agent does not allow creation of a row in this table unless a corresponding row exists in the snmpTargetAddrTable.
  3. Set the value of the TransportTag field on each non-TrapGroup row in the snmpCommunityTable to MGR.

The procedure below assumes the following: at least one configured read-write community; currently one or more Trusted Managers; TransportTag for columns for all rows in the snmpCommunityTable are currently set to MGR. This procedure must be done from one of the existing Trusted Managers.

➤ **To add a subsequent Trusted Manager, take these 2 steps:**

1. Add a row to the snmpTargetAddrTable with these values: Name=mgrN, TagList=MGR, Params=v2cparams, where N is an unused number between 0 and 4.
2. Add a row to the snmpTargetAddrExtTable table with these values: Name=mgrN, snmpTargetAddrTMask=255.255.255.255:0.

An alternative to the above procedure is to set the snmpTargetAddrTMask column while you are creating other rows in the table.

The procedure below assumes the following: at least one configured read-write community; currently two or more Trusted Managers; taglist for columns for all rows in the snmpCommunityTable are currently set to MGR. This procedure must be done from one of the existing trusted managers, but not the one that is being deleted.

➤ **To delete a Trusted Manager (not the final one), take this step:**

- Remove the appropriate row from the snmpTargetAddrTable.

The change takes affect immediately. The deleted trusted manager cannot access the device. The agent automatically removes the row in the snmpTargetAddrExtTable.

The procedure below assumes the following: at least one configured read-write community; currently only one Trusted Manager; taglist for columns for all rows in the snmpCommunityTable are currently set to MGR. This procedure must be done from the final Trusted Manager.

➤ **To delete the final Trusted Manager, take these 2 steps:**

1. Set the value of the TransportTag field on each row in the snmpCommunityTable to the empty string.
2. Remove the appropriate row from the snmpTargetAddrTable.

The change takes affect immediately. All managers can now access the device. The agent automatically removes the row in the snmpTargetAddrExtTable.

## 4.8.4 SNMP Ports

The SNMP Request Port is 161 and Trap Port is 162. These port numbers for SNMP requests and responses can be changed by using the following *ini* file parameter:

**SNMPPort** = <port\_number>

The valid value is any valid UDP port number; the default is 161 (recommended).

## 4.8.5 Multiple SNMP Trap Destinations

An agent can send traps to up to five managers. For each manager you need to define the manager IP address and trap receiving port along with enabling the sending to that manager. You can also associate a trap destination with a specific SNMPv3 USM user. Traps are sent to this trap destination using the SNMPv3 format and the authentication and privacy protocol configured for that user.

To configure the Trap Managers table, use one of the following methods:

- Web interface (refer to the device's *User's Manual*)
- *ini* file (refer to "Configuring Trap Managers via the ini File" on page 60)
- SNMP (refer to "Configuring Trap Managers via SNMP" on page 61)

#### 4.8.5.1 Configuring Trap Managers via Host Name

One of the five available SNMP managers can be defined using the manager's host name (i.e., FQDN). This is currently supported using an *ini* file only (SNMPTrapManagerHostName).

When this parameter value is defined for this trap, the device at start up tries to resolve the host name. Once the name is resolved (i.e., the IP address is found), the resolved IP address replaces the last entry of the trap manager table (defined by the parameter SNMPManagerTableIP\_x) and the last trap manager entry of snmpTargetAddrTable in the snmpTargetMIB. The port is 162 (unless specified otherwise). The row is marked as 'used' and the sending is 'enabled'.

When using 'host name' resolution, any changes made by the user to this row in either MIBs are overwritten by the device when a resolving is redone (once an hour).



**Note:** Some traps may be lost until the name resolving is complete.

#### 4.8.5.2 Configuring Trap Managers via the ini File

In the *ini* file, parameters below can be set to enable or disable the sending of SNMP traps. Multiple trap destinations can be supported on the device by setting multiple trap destinations in the ini file.

- **SNMPManagerTrapSendingEnable\_<x>**: indicates whether or not traps are to be sent to the specified SNMP trap manager. A value of '1' means that it is enabled, while a value of '0' means disabled. The <x> represents a number 0, 1, or 2, which is the array element index. Currently, up to five SNMP trap managers is supported.
- **SNMPManagerTrapUser\_<x>**: indicates to send an SNMPv2 trap using the trap user community string configured with the SNMPTrapCommunityString parameter. You may instead specify an SNMPv3 user name.

Below is an example of entries in the *ini* file regarding SNMP. The device can be configured to send to multiple trap destinations.

```
; SNMP trap destinations
; The device maintains a table of trap destinations containing 5
; rows. The rows are numbered 0..4. Each block of 5 items below
; applies to a row in the table.
;
; To configure one of the rows, uncomment all 5 lines in that
; block. Supply an IP address and if necessary, change the port
; number.
;
; To delete a trap destination, set ISUSED to 0.
;
;SNMPManagerTableIP_0=
;SNMPManagerTrapPort 0=162
;SNMPManagerIsUsed 0=1
```



```

;SNMPManagerTrapSendingEnable 0=1
;SNMPManagerTrapUser 0=''
;
;SNMPManagerTableIP 1=
;SNMPManagerTrapPort 1=162
;SNMPManagerIsUsed_1=1
;SNMPManagerTrapSendingEnable 1=1
;SNMPMANAGERTRAPUSER 1=''
;
;SNMPManagerTableIP 2=
;SNMPManagerTrapPort 2=162
;SNMPManagerIsUsed_2=1
;SNMPManagerTrapSendingEnable 2=1
;SNMPManagerTrapUser 2=''
;
;SNMPManagerTableIP 3=
;SNMPManagerTrapPort 3=162
;SNMPManagerIsUsed_3=1
;SNMPManagerTrapSendingEnable 3=1
;SNMPManagerTrapUser 3=''
;
;SNMPMANAGERTABLEIP 4=
;SNMPManagerTrapPort 4=162
;SNMPManagerIsUsed_4=1
;SNMPManagerTrapSendingEnable 4=1
;SNMPManagerTrapUser 4=''

```

The 'trap manager host name' is configured via `SNMPTrapManagerHostName`. For example:

```
;SNMPTrapManagerHostName = 'myMananger.corp.MyCompany.com'
```



**Note:** The same information that is configurable in the *ini* file can also be configured via the `acBoardMIB`.

### 4.8.5.3 Configuring Trap Managers via SNMP

The `snmpTargetMIB` interface is available for configuring trap managers.



**Note:** The `acBoard MIB` is planned to become obsolete. The only relevant section in this MIB is the trap subtree `acTrap`.

#### ➤ To add an SNMPv2 trap destination, take this step:

- Add a row to the `snmpTargetAddrTable` with these values: Name=trapN, TagList=AC\_TRAP, Params=v2cparams, where N is an unused number between 0 and 4

All changes to the trap destination configuration take effect immediately.

#### ➤ To add an SNMPv3 trap destination, take these 2 steps:

1. Add a row to the `snmpTargetAddrTable` with these values: Name=trapN, TagList=AC\_TRAP, Params=usm<user>, where N is an unused number between 0 and 4, and <user> is the name of the SNMPv3 that this user is associated with.

2. If a row does not already exist for this combination of user and SecurityLevel, add a row to the snmpTargetParamsTable with these values: Name=usm<user>, MPMModel=3(SNMPv3), SecurityModel=3 (usm), SecurityName=<user>, SecurityLevel=M, where M is either 1(noAuthNoPriv), 2(authNoPriv) or 3(authPriv).

All changes to the trap destination configuration take effect immediately.

➤ **To delete a trap destination, take these 2 steps:**

- Remove the appropriate row from the snmpTargetAddrTable.
- If this is the last trap destination associated with this user and security level, you could also delete the appropriate row from the snmpTargetParamsTable.

➤ **To modify a trap destination, take this step:**

You can change the IP address and or port number for an existing trap destination. The same effect can be achieved by removing a row and adding a new row.

- Modify the IP address and/or port number for the appropriate row in the snmpTargetAddrTable.

➤ **To disable a trap destination, take this step:**

- Change TagList on the appropriate row in the snmpTargetAddrTable to the empty string.

➤ **To enable a trap destination, take this step:**

- Change TagList on the appropriate row in the snmpTargetAddrTable to 'AC\_TRAP'.
- Change TagList on the appropriate row in the snmpTargetAddrTable to "AC\_TRAP".

#### 4.8.5.4 SNMP Manager Backward Compatibility

With support of the Multi Manager Trapping feature, there is also a need to support the older acSNMPManagerIP MIB object, which is synchronized with the first index in the snmpManagers MIB table. This is translated in two new features:

- SET/GET to either of the two is currently identical. i.e., OID 1.3.6.1.4.1.5003.9.10.1.1.2.7 is identical to OID 1.3.6.1.4.1.5003.9.10.1.1.2.21.1.1.3 as far as the SET/GET are concerned.
- When setting ANY IP to the acSNMPManagerIP (this is the older parameter, not the table parameter), two more parameters are SET to ENABLE. snmpManagerIsUsed.0 and snmpManagerTrapSendingEnable.0 are both set to 1.

## 4.9 Dual Module Interface



**Note:** This subsection is applicable only to AudioCodes 2000 Series devices.

Dual module blades have a first and second module (the first is on the right side of the blade -- TP-1610 and IPM-1610 -- when looking at it from the front). Differentiation is based on the modules' serial numbers.

MIB object `acSysIdSerialNumber` always returns the serial number of the module on which the GET is performed. MIB object `acSysIdFirstSerialNumber` always returns the serial number of the first module.

If the module on which the GET is performed is the second module, the values in these two are different. If, on the other hand, the module is the first module, the value in the two objects is the same.

## 4.10 SNMP NAT Traversal

A NAT placed between the device and the element manager calls for traversal solutions:

- **Trap source port:** all traps are sent from the SNMP port (default is 161). A manager receiving these traps can use the binding information (in the UDP layer) to traverse the NAT back to the device.  
The trap destination address (port and IP) are as configured in the `snmpTargetMIB`.
- **acKeepAliveTrap:** this trap is designed to be a constant life signal from the device to the manager, allowing the manager NAT traversal at all times. The `acBoardTrapGlobalsAdditionalInfo1` varbind has the device's serial number.

The destination port (i.e., the manager port for this trap), can be set to be different than the port to which all other traps are sent. To do this, use the **acSysSNMPKeepAliveTrapPort** object in the `acSystem` MIB or the `KeepAliveTrapPort` *ini* file parameter.

The Trap is instigated in three ways:

- Via an *ini* file parameter (`SendKeepAliveTrap = 1`). This ensures that the trap is continuously sent. The frequency is set via the 9/10 of the `NATBindingDefaultTimeout` (or MIB object `acSysSTUNBindingLifeTime`) parameter.
- After the STUN client has discovered a NAT (any NAT).
- If the STUN client can not contact a STUN server.



**Note:** The two latter options require the STUN client be enabled (*ini* file parameter `EnableSTUN`). In addition, once the `acKeepAlive` trap is instigated it does not stop.

- The manager can view the NAT type in the MIB:  
`audioCodes(5003).acProducts(9).acBoardMibs(10).acSystem(10).acSystemStatus(2).acSysNetwork(6).acSysNAT(2).acSysNATType(1)`
- The manager also has access to the STUN client configuration:  
`audioCodes(5003).acProducts(9).acBoardMibs(10).acSystem(10).acSystemConfiguration(1).acSysNetworkConfig(3).acSysNATTraversal(6).acSysSTUN(21)`
- **acNATTraversalAlarm:** When the NAT is placed in front of a device that is identified as a symmetric NAT, this alarm is raised. It is cleared when a non-symmetric NAT or no NAT replaces the symmetric one.

## 4.11 Media Server Configuration



**Note:** This subsection is applicable only to AudioCodes IPmedia Series and Mediant 1000 devices.

Configuration for the device can be performed by using the SNMP interfaces in the acBoardMIB or setting of configuration parameters in the *ini* file. Access to the configuration parameters is also provided through the Web interface.

A default *ini* (or initialization) template has been defined, which configures the configuration parameters to settings that typically, do not require later modifications.

Configuration parameters in the acBoardMIB specific to services on the device include:

- **amsApsIpAddress:** IP address of the audio provisioning server
- **amsApsPort:** port number to use for the audio provisioning server
- **amsPrimaryLanguage:** primary language used for audio variables
- **amsSecondaryLanguage:** secondary language used for audio variables

## 4.12 Systems



**Note:** This subsection is applicable only to AudioCodes 3000 Series devices.

For the management of a system (a chassis with more than one type of module running), the acSystem/acSystemChassis subtree in the acSystem MIB should be used:

- The first few objects are scalars that are read-only objects for the dry-contacts' state.
- **acSysModuleTable:** A table containing mostly status information that describes the blade modules in the system. In addition, the table can be used to reset an entire system, reset a redundant module or perform switchover when the system is HA.
- **acSysFanTrayTable:** A status-only table with the fan tray's state. Objects in the table indicate the specific state of the individual fans within the fan tray.
- **acSysPowerSupplyTable:** A status-only table with the states of the two power supplies.
- **acSysPEMTable:** A status-only table with the states of the two PEMs (Power Entry Modules).

The above tables are complemented by the following alarm traps (as defined in the acBoard MIB. For more details, refer to "SNMP Traps" on page 67):

- **acFanTrayAlarm:** fault in the fan tray or fan tray missing.
- **acPowerSupplyAlarm:** fault in one of the power supply modules or PS module missing.
- **acPEMAlarm:** fault in the one of the PEM modules or PEM module missing.
- **acSAMissingAlarm:** SA module missing or non operational.
- **acUserInputAlarm:** the alarm is raised when the input dry contact is short circuited and cleared when the circuit is reopened.

## 4.13 High Availability Systems



**Note:** This subsection is applicable only to Mediant 3000 device.

For the management of the High Availability (HA) systems, use the acSysChassis MIB subtree (as in the above section). The acSysModuleTable gives the HA state of the system. This includes defining which modules are active and which are in standby mode (redundant). The table also enables to read some of the statuses of the redundant modules (such as SW version, HW version, temperature, license key list, etc.). Resetting the system, resetting the redundant module, and performing switchover are performed done using this table.

Complementing the above are the following alarm traps (as defined in the acBoard MIB):

- **acHASystemFaultAlarm:** the HA is faulty and therefore, there is no HA.
- **acHASystemConfigMismatchAlarm:** configuration to the modules in the HA system is uneven causing instability.
- **acHASystemSwitchOverAlarm:** a switchover from the active to the redundant module has occurred.

## 4.14 Configuring Clock Synchronization



**Note:** This subsection is applicable only to Mediant 3000.

The procedures below describe how to configure clock synchronization modes.

➤ **To configure line synchronization, perform the following steps:**

1. Set acSysTimingMode to lineSync.
2. Set acSysTDMClockSource to the interface (according to the hardware you are using) from which you wish to derive the clock.

3. Set TDMBusLocalReference to the reference trunk number.
4. Set acSysTDMClockPLLOutOfRange to the requested value.
5. Set acSysActionSetOnLineChangesApply to 1 in order to apply all changes.

➤ **To configure BITS Synchronization mode through SNMP:**

1. Set acSysTimingMode to external.
2. Set acSysTDMClockBitsReference (1 – Primary Clock Reference is BITs A. (Default) 2 – Primary Clock Reference is BITs B).
3. Set acSysTDMClockEnableFallback (manual(0), autoNon-Revertive(1), auto-Revertive(2) TDMBusEnableFallback sets the fallback clock method between primary to secondary BITS clock references.)
4. Set acSysTimingExternalIFTType to define the external BITS reference transmission type for both primary and secondary interfaces.
5. Set acSysTimingT1LineBuildOut / acSysTimingE1LineBuildOut.
6. Set acSysTimingValidationTime to the requested time range: 0-15 minutes.
7. Set acSysActionSetOnLineChangesApply to 1 in order to apply all changes.

## 4.15 SNMP Administrative State Control

### 4.15.1 Node Maintenance

Node maintenance for the device is provided via an SNMP interface. The acBoardMIB provides two parameters for graceful and forced shutdowns of the device (refer to the note in "Graceful Shutdown" on page 66). These parameters are in the acBoardMIB as the following:

- **acgwAdminState:** requests (sets) a shutdown (0), undo shutdown (2), or view (get) the device condition (0 = locked, 1 = shutting down, 2 = unlocked).
- **acgwAdminStateLockControl:** sets a time limit for the shutdown (in seconds) where 0 means shutdown immediately (forced), -1 means no time limit (graceful), and x, where x > 0 indicates a time limit in seconds (timed limit is considered a graceful shutdown)



**Note:** The acgwAdminStateLockControl must be set before the acgwAdminState.

### 4.15.2 Graceful Shutdown

acgwAdminState is a read-write MIB object. When a get request is sent for this object, the agent returns the current device administrative state.

The possible values received on a get request include:

- **locked(0):** the device is locked.
- **shuttingDown(1):** the device is in the process of performing a graceful lock.
- **unlocked(2):** the device is unlocked.

On a set request, the manager supplies the desired administrative state: either locked(0) or unlocked(2). When the device changes to either shuttingDown or locked state, an adminStateChange alarm is raised. When the device changes to an unlocked state, the adminStateChange alarm is cleared.

Before setting acgwAdminState to perform a lock, acgwAdminStateLockControl must be set first to control the type of lock that is performed. The possible values include:

- 1 = Perform a graceful lock: calls are allowed to complete, but no new calls are allowed from the device.
- 0 = Perform a force lock: calls are immediately terminated.
- Any number greater than 0: time in seconds before the graceful lock turns into a force lock.

## 4.16 AudioCodes' Element Management System

Using AudioCodes' Element Management System (EMS) is recommended for customers requiring large deployments (for example, multiple devices in globally distributed enterprise offices) that need to be managed by central personnel.

The EMS is not included in the device's supplied package. Contact AudioCodes for detailed information on AudioCodes' EMS solution for large VoIP deployments.

## 4.17 SNMP Traps

This subsection provides information on proprietary SNMP traps supported by the device. There is a separation between traps that are alarms and traps that are not (i.e., logs). All the traps have the same structure made up of the same 11 varbinds (Variable Binding), i.e., 1.3.6.1.4.1.5003.9.10.1.21.1. For a list of the varbinds, refer to "Trap Varbinds" on page 93.

The source varbind is composed of a string that details the device component from which the trap is being sent (forwarded by the hierarchy in which it resides). For example, an alarm from an SS7 link has the following string in its source varbind:

acBoard#1/SS7#0/SS7Link#6

In this example, the SS7 link number is specified as 6 and is part of the only SS7 module in the device that is placed in slot number 1 (in a chassis) and is the module to which this trap relates. For devices where there are no chassis options the slot number of the device is always 1.

### 4.17.1 Alarm Traps

The tables in the following subsections provide information on alarms that are raised as a result of a generated SNMP trap. The component name (described in each of the following headings) refers to the string that is provided in the acBoardTrapGlobalsSource trap varbind. To clear a generated alarm, the same notification type is sent but with the severity set to 'cleared'.

### 4.17.1.1 Component: Chassis#0



**Note:** This subsection is applicable only to AudioCodes' 3000 Series, Mediant 1000, and Mediant 600 devices.

The source varbind text for the alarm under this component is Chassis#0/FanTray#0.

**Table 4-11: acFanTrayAlarm Alarm Trap (Applicable Only to 3000 Series and Mediant 1000)**

<b>Alarm:</b>	acFanTrayAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.29
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	equipmentAlarm
<b>Probable Cause:</b>	heatingVentCoolingSystemProblem
<b>Alarm Text:</b>	Fan-Tray Alarm
<b>Status Changes:</b>	
<b>1. Condition:</b>	Fan-Tray is missing
<b>Alarm Status:</b>	Critical
<b>&lt;text&gt; Value:</b>	Fan-Tray Alarm. Fan-Tray is missing
<b>2. Condition:</b>	One or more fans in the Fan-Tray are faulty.
<b>Alarm Status:</b>	Major
<b>Corrective Action:</b>	Fan is faulty
<b>3. Condition:</b>	Fan tray is in place and fans are working.
<b>Alarm Status:</b>	Cleared

The source varbind text for the alarm under this component is Chassis#0/PowerSupply#<m>, where *m* is the power supply's slot number.

**Table 4-12: acPowerSupplyAlarm Alarm Trap (Applicable Only to 3000 Series and Mediant 1000)**

<b>Alarm:</b>	acPowerSupplyAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.30
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	equipmentAlarm
<b>Probable Cause:</b>	powerProblem
<b>Alarm Text:</b>	Power-Supply Alarm. Power-Supply is missing.
<b>Status Changes:</b>	
<b>1. Condition:</b>	The HA (High Availability) feature is active (applicable only to Mediant 3000) and one of the power supply units is faulty or missing.
<b>Alarm Status:</b>	Major
<b>2. Condition:</b>	PS unit is placed and working.
<b>Alarm Status:</b>	Cleared



The source varbind text for the alarm under this component is Chassis#0.

**Table 4-13: acUserInputAlarm Alarm Trap**

<b>Alarm:</b>	acUserInputAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.36
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	equipmentAlarm
<b>Probable Cause:</b>	inputDeviceError
<b>Alarm Text:</b>	User input Alarm. User's Input-Alarm turn on.
<b>Status Changes:</b>	
<b>1. Condition:</b>	Input dry contact is short circuited.
<b>Alarm Status:</b>	Critical
<b>2. Condition:</b>	Input dry contact circuit is reopened.
<b>Alarm Status:</b>	Cleared

The following trap is applicable only to the 3000 Series devices. The source varbind text for the alarm under this component is Chassis#0/PemCard#<m>, where *m* is the power entry module's (PEM) slot number.

**Table 4-14: acPEMAlarm Alarm Trap (Applicable Only to 3000 Series)**

<b>Alarm:</b>	acPEMAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.31
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	equipmentAlarm
<b>Probable Cause:</b>	underlyingResourceUnavailable
<b>Alarm Text:</b>	PEM Module Alarm.
<b>Status Changes:</b>	
<b>1. Condition:</b>	The HA (High Availability) feature is active (applicable only to Mediant 3000) and one of the PEM units is missing (PEM – Power Entry Module)
<b>Alarm status:</b>	Critical
<b>&lt;text&gt; Value:</b>	PEM card is missing.
<b>2. Condition:</b>	PEM card is placed and both DC wires are in.
<b>Alarm Status:</b>	Cleared

The following trap is applicable only to Mediant 1000 devices. The source varbind text for the alarm under this component is Chassis#0/module#<m>, where *m* is the module's number.

**Table 4-15: acHwFailureAlarm Alarm Trap (Applicable Only to Mediant 1000 and Mediant 600)**

<b>Alarm:</b>	acHwFailureAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.43
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	equipmentAlarm
<b>Probable Cause:</b>	equipmentMalfunction
<b>Alarm Text:</b>	Module Alarm: <text>
<b>Status Changes:</b>	
<b>1. Condition:</b>	The module is faulty or has been removed incorrectly.
<b>Alarm Status:</b>	Critical
<b>&lt;text&gt; Value:</b>	Faulty IF-Module
<b>Note:</b>	This alarm is not cleared. The device must be restarted to clear this alarm.
<b>2. Condition:</b>	Module mismatch - module and CPU board mismatch.
<b>Alarm Status:</b>	Major
<b>&lt;text&gt; Value:</b>	IF-Module Mismatch
<b>Note:</b>	This alarm is not cleared. The device must be restarted to clear this alarm.

#### 4.17.1.2 Component: Chassis#0/TimingManager#0



**Note:** This subsection is applicable only to Mediant 3000 High Availability.

**Table 4-16: acTMInconsistentRemoteAndLocalPLLStatus Alarm**

<b>Alarm:</b>	acTMInconsistentRemoteAndLocalPLLStatus
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.56
<b>Default Severity:</b>	Major
<b>Event Type:</b>	equipmentAlarm
<b>Probable Cause:</b>	underlyingResourceUnavailable
<b>Alarm Text:</b>	Timing Manager Alarm <text>
<b>1. Condition:</b>	The alarm is triggered when the system is in 1+1 status and redundant board PLL status is deferent than active board PLL status
<b>Alarm Status:</b>	Major
<b>&lt;text&gt; Value:</b>	Timing Manager Alarm. Local and Remote PLLs status is different.
<b>2. Condition:</b>	
<b>Alarm Status:</b>	Status remains major until a reboot. A clear trap is not sent.
<b>Corrective Action:</b>	Synchronize the timing module.

**Table 4-17: acTMReferenceStatus Alarm**

<b>Alarm:</b>	acTMReferenceStatus
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.57
<b>Default Severity:</b>	Major
<b>Event Type:</b>	equipmentAlarm
<b>Probable Cause:</b>	underlyingResourceUnavailable
<b>Alarm Text:</b>	Timing Manager Alarm <text>
<b>Status Changes:</b>	While primary and secondary clock references are down for more than 24 hours, the alarm will be escalated to critical.
<b>1. Condition:</b>	The alarm is triggered when the primary reference or secondary reference or both are down.
<b>Alarm Status:</b>	Major
<b>&lt;text&gt; Value:</b>	Timing Manager Alarm. PRIMARY REFERENCE DOWN/SECONDARY REFERENCE DOWN/ALL REFERENCES ARE DOWN
<b>2. Condition:</b>	
<b>Alarm Status:</b>	Status remains major until a reboot. A clear trap is not sent.
<b>Corrective Action:</b>	Synchronize the timing module.

**Table 4-18: acTMReferenceChange Alarm**

<b>Alarm:</b>	acTMReferenceChange
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.58
<b>Default Severity:</b>	Indeterminate
<b>Event Type:</b>	
<b>Probable Cause:</b>	
<b>Alarm Text:</b>	Timing Manager
<b>Status Changes:</b>	
<b>1. Condition:</b>	Log is sent on PLL status change.

### 4.17.1.3 Component: Interfaces#0/Sonet#<m>



**Note:** This subsection is applicable only to AudioCodes' 3000 Series with 6310 blade series devices.

The source varbind text for the alarms under this component is Interfaces#0/Sonet#<m>, where *m* is the Sonet IF number.

**Table 4-19: AcSonetSectionLOFAlarm Alarm Trap**

<b>Alarm:</b>	acSonetSectionLOFAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.38
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	lossOfFrame
<b>Alarm Text:</b>	SONET-Section LOF.
<b>Status Changes:</b>	
<b>1. Condition:</b>	LOF condition is present on SONET no.n
<b>Alarm Status:</b>	Critical
<b>&lt;text&gt; Value:</b>	LOF
<b>Note:</b>	The sonetSectionCurrentStatus field in the sonetSectionCurrentTable will have a value sonetSectionLOF (4).
<b>2. Condition:</b>	LOF condition is not present.
<b>Alarm Status:</b>	Cleared

**Table 4-20: AcSonetSectionLOSAAlarm Alarm Trap**

<b>Alarm:</b>	acSonetSectionLOSAAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.39
<b>Default Severity:</b>	critical
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	lossOfSignal
<b>Alarm Text:</b>	SONET-Section LOS.
<b>Status Changes:</b>	
<b>1. Condition:</b>	LOS condition is present on SONET no #n
<b>Alarm Status:</b>	Critical
<b>&lt;text&gt; Value:</b>	LOS
<b>Note:</b>	The sonetSectionCurrentStatus field in the sonetSectionCurrentTable will have a value sonetSectionLOS (2).
<b>2. Condition:</b>	AIS condition is present (LOS condition is not present)
<b>Alarm Status:</b>	Critical
<b>3. Condition:</b>	LOS condition is not present.
<b>Alarm Status:</b>	Cleared

**Table 4-21: AcSonetLineAISAlarm Alarm Trap**

<b>Alarm:</b>	acSonetLineAISAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.40
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	receiveFailure
<b>Alarm Text:</b>	SONET-Line AIS.
<b>Status Changes:</b>	
<b>1. Condition:</b>	AIS condition is present on SONET-Line #n.
<b>Alarm Status:</b>	Critical
<b>&lt;text&gt; Value:</b>	AIS
<b>Note:</b>	The sonetLineCurrentStatus field in the sonetLineCurrentTable will have a value sonetLineAIS (2).
<b>2. Condition:</b>	AIS condition is not present.
<b>Alarm Status:</b>	Cleared

**Table 4-22: AcSonetLineRDIALarm Alarm Trap**

<b>Alarm:</b>	acSonetLineRDIALarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.41
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	transmitFailure
<b>Alarm Text:</b>	SONET-Line RDI.
<b>Status Changes:</b>	
<b>1. Condition:</b>	RDI condition is present on SONET-Line #n.
<b>Alarm Status:</b>	Critical
<b>&lt;text&gt; Value:</b>	RDI
<b>Note:</b>	The sonetLineCurrentStatus field in the sonetLineCurrentTable will have a value sonetLineRDI (4).
<b>2. Condition:</b>	RDI condition is not present.
<b>Alarm Status:</b>	Cleared

#### 4.17.1.4 Component: System#0<n> and Board#0<n>

The source varbind text for all the alarms under this component depends on the device:

- 3000 Series: **System#0<n>**
- 2000 Series, Mediant 1000, and MediaPack: **Board#0<n>**

where *n* is the slot number in which the blade resides in the chassis. For Mediant 1000 and MediaPack, *n* always equals to 1.

**Table 4-23: acBoardFatalError Alarm Trap**

<b>Alarm:</b>	acBoardFatalError
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.1
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	equipmentAlarm
<b>Probable Cause:</b>	underlyingResourceUnavailable (56)
<b>Alarm Text:</b>	Board Fatal Error: <text>
<b>Status Changes:</b>	
<b>1. Condition:</b>	Any fatal error
<b>Alarm Status:</b>	Critical
<b>&lt;text&gt; Value:</b>	A run-time specific string describing the fatal error
<b>2. Condition:</b>	After fatal error
<b>Alarm Status:</b>	Status stays critical until reboot. A clear trap is not sent.
<b>Corrective Action:</b>	Capture the alarm information and the Syslog clause, if active. Contact your first-level support group. The support group will likely want to collect additional data from the device and perform a reset.

**Table 4-24: acBoardConfigurationError Alarm Trap**

<b>Alarm:</b>	acBoardConfigurationError
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.2
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	equipmentAlarm
<b>Probable Cause:</b>	underlyingResourceUnavailable (56)
<b>Alarm Text:</b>	Board Config Error: <text>
<b>Status Changes:</b>	
<b>1. Condition:</b>	A configuration error was detected
<b>Alarm Status:</b>	critical
<b>&lt;text&gt; Value:</b>	A run-time specific string describing the configuration error.
<b>2. Condition:</b>	After configuration error
<b>Alarm Status:</b>	Status stays critical until reboot. A clear trap is not sent.
<b>Corrective Action:</b>	Inspect the run-time specific string to determine the nature of the configuration error. Fix the configuration error using the appropriate tool: Web interface, EMS, or <i>ini</i> file. Save the configuration and if necessary reset the device.

**Table 4-25: acBoardTemperatureAlarm Alarm Trap (Applicable to 2000 and 3000 Series - Except Mediant 3000 HA)**

<b>Alarm:</b>	acBoardTemperatureAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.3
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	equipmentAlarm
<b>Probable Cause:</b>	temperatureUnacceptable (50)
<b>Alarm Text:</b>	Board temperature too high
<b>Status Changes:</b>	
<b>1. Condition:</b>	Temperature is above 60°C (140°F)
<b>Alarm Status:</b>	Critical
<b>2. Condition:</b>	After raise, temperature falls below 55°C (131°F)
<b>Alarm Status:</b>	Cleared
<b>Corrective Action:</b>	Inspect the system. Determine if all fans in the system are properly operating.

**Table 4-26: acBoardEvResettingBoard Alarm Trap**

<b>Alarm:</b>	acBoardEvResettingBoard
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.5
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	equipmentAlarm
<b>Probable Cause:</b>	outOfService (71)
<b>Alarm Text:</b>	User resetting board
<b>Status Changes:</b>	
<b>1. Condition:</b>	When a soft reset is triggered via the Web interface or SNMP.
<b>Alarm Status:</b>	Critical
<b>2. Condition:</b>	After raise
<b>Alarm Status:</b>	Status stays critical until reboot. A clear trap is not sent.
<b>Corrective Action:</b>	A network administrator has taken action to reset the device. No corrective action is required.

The following trap is applicable only to 2000 Series, Mediant 1000, and MediaPack devices. This trap relates to the Ethernet Link Module (the #0 numbering doesn't apply to the physical Ethernet link).

**Table 4-27: acBoardEthernetLinkAlarm Alarm Trap (Applicable only to 2000 Series, Mediant 1000, and MediaPack)**

<b>Alarm:</b>	acBoardEthernetLinkAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.10
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	equipmentAlarm
<b>Probable Cause:</b>	underlyingResourceUnavailable (56)
<b>Alarm Text:</b>	Ethernet link alarm: <text>
<b>Status Changes:</b>	
<b>1. Condition:</b>	Fault on single interface
<b>Alarm Status:</b>	Major
<b>&lt;text&gt; Value:</b>	Redundant link is down
<b>2. Condition:</b>	Fault on both interfaces
<b>Alarm Status:</b>	critical
<b>&lt;text&gt; Value:</b>	No Ethernet link
<b>3. Condition:</b>	Both interfaces are operational
<b>Alarm Status:</b>	cleared
<b>Corrective Action:</b>	Ensure that both Ethernet cables are plugged into the back of the system. Inspect the system's Ethernet link lights to determine which interface is failing. Reconnect the cable or fix the network problem

**Table 4-28: acBoardCallResourcesAlarm Alarm Trap**

<b>Alarm:</b>	acBoardCallResourcesAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.8
<b>Default Severity:</b>	Major
<b>Event Type:</b>	processingErrorAlarm
<b>Probable Cause:</b>	softwareError (46)
<b>Alarm Text:</b>	Call resources alarm
<b>Status Changes:</b>	
<b>1. Condition:</b>	Percentage of busy channels exceeds the predefined RAI high threshold.
<b>Alarm Status:</b>	Major
<b>Note:</b>	To enable this alarm the RAI mechanism must be activated (EnableRAI = 1).
<b>2. Condition:</b>	Percentage of busy channels falls below the predefined RAI low threshold.
<b>Alarm Status:</b>	Cleared



**Table 4-29: acBoardControllerFailureAlarm Alarm Trap**

<b>Alarm:</b>	acBoardControllerFailureAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.9
<b>Default Severity:</b>	Minor
<b>Event Type:</b>	processingErrorAlarm
<b>Probable Cause:</b>	softwareError (46)
<b>Alarm Text:</b>	Controller failure alarm
<b>Status Changes:</b>	
<b>1. Condition:</b>	Proxy has not been found
<b>Alarm Status:</b>	Major
<b>Additional Info:</b>	Proxy not found. Use internal routing or Proxy lost. looking for another Proxy
<b>2. Condition:</b>	Proxy is found. The clear message includes the IP address of this Proxy.
<b>Alarm Status:</b>	Cleared

**Table 4-30: acBoardOverloadAlarm Alarm Trap**

<b>Alarm:</b>	acBoardOverloadAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.11
<b>Default Severity:</b>	Major
<b>Event Type:</b>	processingErrorAlarm
<b>Probable Cause:</b>	softwareError (46)
<b>Alarm Text:</b>	Board overload alarm
<b>Status Changes:</b>	
<b>1. Condition:</b>	An overload condition exists in one or more of the system components.
<b>Alarm Status:</b>	Major
<b>2. Condition:</b>	The overload condition passed
<b>Alarm Status:</b>	Cleared

**Table 4-31: acFeatureKeyError Alarm Trap (Applicable only to Digital devices)**

<b>Alarm:</b>	acFeatureKeyError
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.6
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	processingErrorAlarm
<b>Probable Cause:</b>	configurationOrCustomizationError (7)
<b>Alarm Text:</b>	Feature key error
<b>Status Changes:</b>	
<b>Note:</b>	Support for this alarm is pending.

The following trap is applicable only to the 3000 Series devices. The source varbind text for the alarm under this component is Chassis#0/SA#<m>, where *m* is the shelf Alarm module's slot number.

**Table 4-32: acSAMissingAlarm Alarm Trap (Applicable only to the 3000 Series Devices)**

<b>Alarm:</b>	acSAMissingAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.32
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	equipmentAlarm
<b>Probable Cause:</b>	underlyingResourceUnavailable
<b>Alarm Text:</b>	SA Module Alarm. SA-Module from slot #n is missing.
<b>Status Changes:</b>	
<b>1. Condition:</b>	SA module removed or missing
<b>Alarm Status:</b>	Critical
<b>2. Condition:</b>	SA module is in slot 2 or 4 and working.
<b>Alarm Status:</b>	Cleared

#### 4.17.1.5 Component: System#0



**Note:** This subsection is applicable only to AudioCodes' 3000 Series devices.

**Table 4-33: acHitlessUpdateStatus Alarm Trap**

<b>Alarm:</b>	acHitlessUpdateStatus
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.48
<b>Default Severity:</b>	-
<b>Event Type:</b>	Other
<b>Probable Cause:</b>	Other
<b>Alarm Text:</b>	Hitless Update Event
<b>Status Changes:</b>	
<b>Condition:</b>	<p>A Notification trap that is sent out at the beginning and the end of a Hitless SW update. Failure during the process will also instigate the trap. May include the following information:</p> <p>Hitless: start SW upgrade.</p> <p>Hitless: Stream read error, aborting CMP file processing.</p> <p>Hitless: Invalid cmp file - missing Ver parameter.</p> <p>Hitless fail: Hitless SW upgrade is not supported under version 5.2.</p> <p>Hitless fail: SW ver stream name too long.</p> <p>Hitless fail: Invalid cmp file - missing UPG parameter.</p> <p>Hitless fail: Hitless SW upgrade not supported.</p> <p>Hitless fail: Communication with redundant module failed.</p> <p>Hitless: SW upgrade ended successfully.</p>
<b>Alarm Status:</b>	Indeterminate
<b>Corrective Action:</b>	

#### 4.17.1.6 Component: AlarmManager#0

The source varbind text for all the alarms under this component is `System#0<n>/AlarmManager#0`.

**Table 4-34: acActiveAlarmTableOverflow Alarm Trap**

<b>Alarm:</b>	acActiveAlarmTableOverflow
<b>OID:</b>	1.3.6.1.4.15003.9.10.1.21.2.0.12
<b>Default Severity:</b>	Major
<b>Event Type:</b>	processingErrorAlarm
<b>Probable Cause:</b>	resourceAtOrNearingCapacity (43)
<b>Alarm Text:</b>	Active alarm table overflow
<b>Status Changes:</b>	
<b>1. Condition:</b>	Too many alarms to fit in the active alarm table
<b>Alarm Status:</b>	Major
<b>2. Condition:</b>	After raise
<b>Alarm Status:</b>	Status remains Major until reboot. A Clear trap is not sent.
<b>Note:</b>	The status remains Major until reboot as it denotes a possible loss of information until the next reboot. If an alarm is raised when the table is full, it is possible that the alarm is active, but does not appear in the active alarm table.
<b>Corrective Action:</b>	Some alarm information may have been lost, but the ability of the device to perform its basic operations has not been impacted. A reboot is the only way to completely clear a problem with the active alarm table. Contact your first-level group.

#### 4.17.1.7 Component: AudioStaging#0



**Note:** This subsection is applicable only to AudioCodes' IPmedia Series and Mediant 1000 devices.

**Table 4-35: acAudioProvisioningAlarm Alarm Trap**

<b>Alarm:</b>	acAudioProvisioningAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.14
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	processingErrorAlarm
<b>Probable Cause:</b>	configurationOrCustomizationError (7)
<b>Alarm Text:</b>	Unable to provision audio
<b>Status Changes:</b>	
<b>1. Condition:</b>	Media server times out waiting for a successful audio distribution from the APS (Audio Provisioning Server)
<b>Alarm Status:</b>	Critical
<b>2. Condition:</b>	After raise, media server is successfully provisioned with audio from the APS
<b>Alarm Status:</b>	Cleared
<b>Corrective Action:</b>	From the APS (Audio Provisioning Server) GUI ensure that the device is properly configured with audio and that the device has been enabled. Ensure that the IP address for the APS has been properly specified on the device. Ensure that both the APS server and application are in-service. For more information regarding the problem, view the Syslogs from the device as well as the APS manager logs.

### 4.17.1.8 Component: AnalogPorts#0



**Note:** This subsection is applicable only to AudioCodes' MediaPack and Mediant 1000 (analog) devices.

The source varbind text for all the alarms under this component is System#0/analogports#<n>, where *n* is the port number.

**Table 4-36: acAnalogPortSPIOutOfService Alarm Trap**

<b>Alarm:</b>	acAnalogPortSPIOutOfService
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.46
<b>Default Severity:</b>	Major
<b>Event Type:</b>	physicalViolation
<b>Probable Cause:</b>	equipmentMalfunction
<b>Alarm Text:</b>	Analog Port SPI out of service
<b>Status Changes:</b>	
<b>1. Condition:</b>	Analog port has gone out of service
<b>Alarm Status:</b>	Major
<b>2. Condition:</b>	Analog port is back in service.
<b>Alarm Status:</b>	Cleared
<b>Corrective Action:</b>	None

**Table 4-37: acAnalogPortHighTemperature Alarm Trap**

<b>Alarm:</b>	acAnalogPortHighTemperature
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.47
<b>Default Severity:</b>	Major
<b>Event Type:</b>	physicalViolation
<b>Probable Cause:</b>	equipmentMalfunction
<b>Alarm Text:</b>	Analog Port High Temperature
<b>Status Changes:</b>	
<b>1. Condition:</b>	Analog device has reached critical temperature. Device is automatically disconnected.
<b>Alarm Status:</b>	Major
<b>2. Condition:</b>	Temperature is back to normal - analog port is back in service.
<b>Alarm Status:</b>	Cleared
<b>Corrective Action:</b>	None
<b>Note:</b>	Relevant to FXS only.

### 4.17.1.9 Component: SS7#0



**Note:** This subsection is applicable only to AudioCodes' 2000 and 3000 (except IPmedia 3000/IPM-8410) Series devices.

The source varbind text for all alarms under this component is System#0<n>/SS7#0/SS7Link#<m>, where *m* is the link number.

**Table 4-38: acSS7LinkStateChangeAlarm Trap**

<b>Alarm:</b>	acSS7LinkStateChangeAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.19
<b>Default Severity:</b>	Major
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	Other
<b>Alarm Text:</b>	*** SS7 *** Link %i is %s \$s
<b>Status Changes:</b>	
<b>1. Condition:</b>	Operational state of the SS7 link becomes 'BUSY'.
<b>Alarm Status:</b>	Major
<b>&lt;text&gt; Value:</b>	%i - <Link number> %s - <state name>: { "OFFLINE", "BUSY", "INSERVICE"} %s - If link has MTP3 layer, then this string equals: (SP %i linkset %i slc %i) Where: %i - <SP number> %i - <Link-Set number> %i - <SLC number> Otherwise there is NO additional text.
<b>Additional Info1 Varbind</b>	BUSY
<b>2. Condition:</b>	Operational state of the link becomes 'IN-SERVICE' or 'OFFLINE'.
<b>Alarm status:</b>	Cleared
<b>Corrective Action:</b>	For full details refer to the SS7 MTP2 and MTP3 relevant standards.

**Table 4-39: acSS7LinkCongestionStateChangeAlarm Trap**

<b>Alarm:</b>	acSS7LinkCongestionStateChangeAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.22
<b>Default Severity:</b>	Major
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	Other
<b>Alarm Text:</b>	*** SS7 *** Link %i is %s %s
<b>Status Changes:</b>	
<b>1. Condition:</b>	SS7 link becomes congested (local or remote).
<b>Alarm Status:</b>	Major
<b>&lt;text&gt; Value:</b>	%i - <Link number> %s - If link has MTP3 layer, then this string equals: (SP %i linkset %i slc %i) Where: %i - <SP number> %i - <Link-Set number> %i - <SLC number> Otherwise there is NO additional text. %s - <congestion state>: { "UNCONGESTED", "CONGESTED" }
<b>Additional Info1 Varbind</b>	CONGESTED
<b>2. Condition:</b>	Link becomes un-congested (local AND remote).
<b>Alarm Status:</b>	Cleared
<b>Corrective Action:</b>	Reduce SS7 traffic on the link.
<b>Note:</b>	This alarm is raised for any change in the remote or local congestion status.

**Table 4-40: acSS7LinkInhibitStateChangeAlarm Trap**

<b>Alarm:</b>	acSS7LinkInhibitStateChangeAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.20
<b>Default Severity:</b>	Major
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	Other
<b>Alarm Text:</b>	*** SS7 *** Link %i (SP %i linkset %i slc %i) is %s
<b>Status Changes:</b>	
<b>1. Condition:</b>	SS7 link becomes inhibited (local or remote).
<b>Alarm Status:</b>	Major
<b>&lt;text&gt; Value:</b>	%i - <Link number> %i - <SP number> %i - <Link-Set number> %i - <SLC number> %s - <congestion state>: { "UNINHIBITED", "INHIBITED" }
<b>Additional Info1 Varbind</b>	INHIBITED
<b>2. Condition:</b>	Link becomes uninhibited - local AND remote
<b>Alarm Status:</b>	Cleared
<b>Corrective Action:</b>	Make sure the link is uninhibited – on both local and remote sides
<b>Note:</b>	This alarm is raised for any change in the remote or local inhibition status.

**Table 4-41: acSS7LinkBlockStateChangeAlarm Trap**

<b>Alarm:</b>	acSS7LinkBlockStateChangeAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.21
<b>Default Severity:</b>	Major
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	Other
<b>Note:</b>	Support pending

**Table 4-42: acSS7LinkSetStateChangeAlarm Trap**

<b>Alarm:</b>	acSS7LinkSetStateChangeAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.23
<b>Default Severity:</b>	Major
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	Other
<b>Alarm Text:</b>	*** SS7 *** Linkset %i on SP %i is %s
<b>Status Changes:</b>	
<b>1. Condition:</b>	Operational state of the SS7 link-set becomes BUSY
<b>Alarm Status:</b>	Major
<b>&lt;text&gt; Value:</b>	%i - <Link-Set number> %i - <SP number> %s - <state name: { "OFFLINE", "BUSY", "INSERVICE" }
<b>Additional Info1 Varbind</b>	BUSY
<b>2. Condition:</b>	Operational state of the link-set becomes IN-SERVICE or OFFLINE
<b>Alarm Status:</b>	Cleared
<b>Corrective Action:</b>	For full details see the SS7 section
<b>Note:</b>	The source varbind text for all the alarms under this component is System#0/SS7#0/SS7LinkSet#<m>, where <i>m</i> is the link set number.

**Table 4-43: acSS7RouteSetStateChangeAlarm Trap**

<b>Alarm:</b>	acSS7RouteSetStateChangeAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.24
<b>Default Severity:</b>	Major
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	Other
<b>Alarm Text:</b>	*** SS7 *** Routeset %i on SP %i is %s
<b>Status Changes:</b>	
<b>1. Condition:</b>	Operational state of the SS7 link-set becomes BUSY
<b>Alarm Status:</b>	Major
<b>&lt;text&gt; Value:</b>	%i - <Route-Set number> %i - <SP number> %s - <state name: { "OFFLINE", "BUSY", "INSERVICE" }
<b>Additional Info1 Varbind</b>	BUSY
<b>2. Condition:</b>	Operational state of the link-set becomes IN-SERVICE or OFFLINE
<b>Alarm Status:</b>	Cleared
<b>Corrective Action:</b>	For full details see the SS7 section
<b>Note:</b>	The source varbind text for all the alarms under this component is System#0/SS7#0/SS7RouteSet#<m>, where <i>m</i> is the route set number.

**Table 4-44: acSS7SNSetStateChangeAlarm Trap**

<b>Alarm:</b>	acSS7SNSetStateChangeAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.25
<b>Default Severity:</b>	Major
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	Other
<b>Alarm Text:</b>	*** SS7 *** SP %i is %s
<b>Status Changes:</b>	
<b>1. Condition:</b>	Operational state of the SS7 node becomes BUSY
<b>Alarm Status:</b>	Major
<b>&lt;text&gt; Value:</b>	%i - <SP number> %s - <state name: { "OFFLINE", "BUSY", "INSERVICE" }
<b>Additional Info1 Varbind</b>	BUSY
<b>2. Condition:</b>	Cleared when the operational state of the node becomes IN-SERVICE or OFFLINE
<b>Alarm Status:</b>	Cleared
<b>Corrective Action:</b>	Signaling Node must complete its MTP3 restart procedure and become un-isolated. For full details see the SS7 section
<b>Note:</b>	The source varbind text for all the alarms under this component is System#0/SS7#0/SS7SN#<m>, where <i>m</i> is the (signaling node) number.



**Table 4-45: acSS7RedundancyAlarm Trap**

<b>Alarm:</b>	acSS7RedundancyAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.26
<b>Default Severity:</b>	Major
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	Other
<b>Note:</b>	Support pending.

#### 4.17.1.10 Component: System#0/Module#<m>



**Note:** The alarm traps discussed in this subsection applies only to the Mediant 3000 in High Availability mode.

The source varbind text for the alarms under the component below is System#0/Module#<m>, where *m* is the blade module's slot number.

**Table 4-46: acHASystemConfigMismatchAlarm Trap**

<b>Trap:</b>	acHASystemConfigMismatchAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.34
<b>Default Severity:</b>	major
<b>Event Type:</b>	processingErrorAlarm
<b>Probable Cause:</b>	configurationOrCustomizationError
<b>Trap Text:</b>	Configuration mismatch in the system.
<b>Status Changes:</b>	
<b>1. Condition:</b>	HA feature is active. The active module was unable to pass on to the redundant module the License Key.
<b>Trap Status:</b>	Major
<b>&lt;text&gt; Value:</b>	Fail to update the redundant with feature key
<b>2. Condition:</b>	Successful License Key update.
<b>Trap Status:</b>	Cleared
<b>&lt;text&gt; Value:</b>	The feature key was successfully updated in the redundant module

**Table 4-47: acHASystemFaultAlarm Trap**

<b>Trap:</b>	acHASystemFaultAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.33
<b>Default Severity:</b>	critical
<b>Event Type:</b>	qualityOfServiceAlarm
<b>Probable Cause:</b>	outOfService
<b>Trap Text:</b>	No HA! <text>
<b>Status Changes:</b>	
<b>1. Condition:</b>	HA feature is active but the system is not working in HA mode.
<b>Trap Status:</b>	Critical
<b>&lt;text&gt; Value:</b>	<p>There are many possible values for the text:</p> <p>Fatal exception error</p> <p>TCPIP exception error</p> <p>Network processor exception error</p> <p>SW WD exception error</p> <p>HW WD exception error</p> <p>SAT device is missing</p> <p>SAT device error</p> <p>DSP error</p> <p>BIT tests error</p> <p>PSTN stack error</p> <p>Keep Alive error</p> <p>Software upgrade</p> <p>Manual switch over</p> <p>Manual reset</p> <p>Board removal</p> <p>Can't read slot number</p> <p>TER misplaced</p> <p>HW fault. TER in slot 2 or 3 is missing</p> <p>HW fault. TER has old version or is not functional</p> <p>HW fault. invalid TER Type</p> <p>HW fault. invalid TER active/redundant state</p> <p>HW fault. Error reading GbE state</p> <p>Redundant module is missing</p> <p>Unable to sync SW versions</p> <p>Redundant is not connecting</p> <p>Redundant is not reconnecting after deliberate restart</p> <p>No Ethernet Link in redundant module</p> <p>SA module faulty or missing</p>
<b>2. Condition:</b>	HA feature is active and the redundant module is in start up mode and hasn't connected yet.
<b>Trap Status:</b>	Minor
<b>&lt;text&gt; Value:</b>	Waiting for redundant to connect
<b>3. Condition:</b>	HA system is active.
<b>Trap Status:</b>	Cleared

**Table 4-48: acHASystemSwitchOverAlarm Trap**

<b>Trap:</b>	acHASystemSwitchOverAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.35
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	qualityOfServiceAlarm
<b>Probable Cause:</b>	outOfService
<b>Trap Text:</b>	Switch-over: <text>
<b>Status Changes:</b>	
<b>1. Condition:</b>	Switch over has taken place.
<b>Trap Status:</b>	Critical
<b>&lt;text&gt; Value:</b>	See the acHASystemFaultAlarm table above.
<b>2. Condition:</b>	10 seconds have passed since the switch over.
<b>Trap Status:</b>	cleared

**Table 4-49: acBoardTemperatureAlarm Trap**

<b>Trap:</b>	acBoardTemperatureAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.3
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	equipmentAlarm
<b>Probable Cause:</b>	temperatureUnacceptable (50)
<b>Trap Text:</b>	Board temperature too high
<b>Status Changes:</b>	
<b>1. Condition:</b>	Temperature in the active module or redundant is above 67°C (140°F)
<b>Trap Status:</b>	Critical
<b>2. Condition:</b>	After raise, temperature falls below 55°C (131°F)
<b>Trap Status:</b>	Cleared
<b>Corrective Action:</b>	Inspect the system. Determine if all fans in the system are properly operating.

If the lost link is from the active module, the source varbind text for the alarm under this component is Chassis#0/Module#<m>/EthernetLink#0, where *m* is the blade's slot number.

**Table 4-50: acBoardEthernetLinkAlarm Trap**

<b>Trap:</b>	acBoardEthernetLinkAlarm
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.10
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	equipmentAlarm
<b>Probable Cause:</b>	underlyingResourceUnavailable (56)
<b>Trap Text:</b>	Ethernet link alarm: <text>
<b>Status Changes:</b>	
<b>1. Condition:</b>	Fault on single interface of the Active module.
<b>Trap Status:</b>	Major
<b>&lt;text&gt; Value:</b>	Redundant link (physical link n) is down
<b>2. Condition:</b>	Fault on both interfaces
<b>Trap Status:</b>	Critical
<b>&lt;text&gt; Value:</b>	No Ethernet link
<b>3. Condition:</b>	Fault on single interface of the Redundant module.
<b>Trap Status:</b>	Major
<b>&lt;text&gt; Value:</b>	Redundant link in the redundant module (physical link n) is down
<b>4. Condition:</b>	Both interfaces are operational
<b>Trap Status:</b>	Cleared
<b>Corrective Action:</b>	Ensure that both Ethernet cables are plugged into the back of the system. Inspect the system's Ethernet link lights to determine which interface is failing. Reconnect the cable or fix the network problem
<b>Note:</b>	The alarm behaves differently when coming from the redundant or the active modules of an HA system. The alarm from the redundant is raised when there is an operational HA configuration in the system. There is no critical severity for the redundant module losing both its Ethernet links as that is conveyed in the no HA alarm that follows such a case.

#### 4.17.1.11 Component: Interfaces#0/Trunk#<m>



**Note:** The alarm traps discussed in this subsection applies only to the Digital devices, except IPmedia 3000/IPM-8410.

The source varbind text for the alarms under the component below is Interfaces#0/Trunk#<m>, where *m* is the trunk IF number, 1 being the first trunk.

**Table 4-51: acTrunksAlarmNearEndLOS Alarm Trap**

<b>Alarm:</b>	acTrunksAlarmNearEndLOS
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.49
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	lossOfSignal

<b>Alarm Text:</b>	Trunk LOS Alarm.
<b>Status Changes:</b>	
<b>Condition:</b>	Near-end LOS
<b>Alarm Status:</b>	Critical
<b>Condition:</b>	End of LOS
<b>Alarm Status:</b>	Cleared
<b>Corrective Action:</b>	Ensure the trunk is properly connected.

**Table 4-52: acTrunksAlarmNearEndLOF Alarm Trap**

<b>Alarm:</b>	acTrunksAlarmNearEndLOF
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.50
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	lossOfFrame
<b>Alarm Text:</b>	Trunk LOF Alarm.
<b>Status Changes:</b>	
<b>Condition:</b>	Near end LOF
<b>Alarm Status:</b>	Critical
<b>Condition:</b>	End of LOF
<b>Alarm Status:</b>	Cleared
<b>Corrective Action:</b>	Ensure the trunk is connected to a proper follow up device. Ensure correct clocking setup.

**Table 4-53: acTrunksAlarmRcvAIS Alarm Trap**

<b>Alarm:</b>	acTrunksAlarmRcvAIS
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.51
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	receiveFailure
<b>Alarm Text:</b>	Trunk AIS Alarm
<b>Status Changes:</b>	
<b>Condition:</b>	Receive AIS
<b>Alarm Status:</b>	Critical
<b>Condition:</b>	End of AIS
<b>Alarm Status:</b>	Cleared
<b>Corrective Action:</b>	None

**Table 4-54: acTrunksAlarmFarEndLOF Alarm Trap**

<b>Alarm:</b>	acTrunksAlarmFarEndLOF
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.52
<b>Default Severity:</b>	Critical
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	transmitFailure
<b>Alarm Text:</b>	Trunk RAI Alarm.
<b>Status Changes:</b>	
<b>Condition:</b>	RAI
<b>Alarm Status:</b>	Critical
<b>Condition:</b>	End of RAI
<b>Alarm Status:</b>	Cleared
<b>Corrective Action:</b>	Ensure correct transmission.

## 4.17.2 Log Traps (Notifications)

This subsection details traps that are not alarms. These traps are sent with the severity varbind value of 'indeterminate'. These traps don't 'clear' and they don't appear in the alarm history or active tables. (The only log trap that does send clear is acPerformanceMonitoringThresholdCrossing.)

**Table 4-55: acKeepAlive Log Trap**

<b>Trap:</b>	acKeepAlive
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.16
<b>Default Severity:</b>	Indeterminate
<b>Event Type:</b>	other (0)
<b>Probable Cause:</b>	other (0)
<b>Trap Text:</b>	Keep alive trap
<b>Status Changes:</b>	
<b>Condition:</b>	The STUN client in is enabled and identified a NAT device or doesn't locate the STUN server. The <i>ini</i> file contains the following line: 'SendKeepAliveTrap=1'
<b>Trap Status:</b>	Trap is sent
<b>Note:</b>	Keep-alive is sent every 9/10 of the time defined in the parameter NatBindingDefaultTimeout.

**Table 4-56: acPerformanceMonitoringThresholdCrossing Log Trap**

<b>Trap:</b>	acPerformanceMonitoringThresholdCrossing
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.27
<b>Default Severity:</b>	Indeterminate
<b>Event Type:</b>	other (0)
<b>Probable Cause:</b>	other (0)
<b>Trap Text:</b>	"Performance: Threshold trap was set", with source = name of performance counter which caused the trap
<b>Status Changes:</b>	
<b>Condition:</b>	A performance counter has crossed the high threshold
<b>Trap Status:</b>	Indeterminate
<b>Condition:</b>	A performance counter has crossed the low threshold
<b>Trap Status:</b>	Cleared

**Table 4-57: acHTTPDownloadResult Log Trap**

<b>Trap:</b>	acHTTPDownloadResult
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.28
<b>Default Severity:</b>	Indeterminate
<b>Event Type:</b>	processingErrorAlarm (3) for failures and other (0) for success.
<b>Probable Cause:</b>	other (0)
<b>Status Changes:</b>	
<b>Condition:</b>	Successful HTTP download.
<b>Trap Text:</b>	HTTP Download successful
<b>Condition:</b>	Failed download.
<b>Trap Text:</b>	HTTP download failed, a network error occurred.
<b>Note:</b>	There are other possible textual messages describing NFS failures or success, FTP failure or success.

**Table 4-58: acDialPlanFileReplaced Log Trap (Applicable Only to Digital Devices, Except IPmedia 3000/IPM-8410)**

<b>Alarm:/8410</b>	acDialPlanFileReplaced
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.45
<b>Default Severity:</b>	Indeterminate
<b>Event Type:</b>	Other (0)
<b>Probable Cause:</b>	Other (0)
<b>Status Change:</b>	
<b>Condition:</b>	Successful dial plan file replacement
<b>Trap Text:</b>	Dial plan file replacement complete.

**Table 4-59: acHitlessUpdateStatus Log Trap (Applicable Only to 3000 Series Devices)**

<b>Alarm:</b>	acHitlessUpdateStatus
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.48
<b>Default Severity:</b>	Indeterminate
<b>Event Type:</b>	Other (0)
<b>Probable Cause:</b>	Other (0)
<b>Source:</b>	Automatic Update
<b>Status Changes:</b>	
<b>Condition:</b>	Successful SW upgrade
<b>Trap Text:</b>	Hitless: SW upgrade ended successfully
<b>Condition:</b>	Failed SW upgrade
<b>Trap Text:</b>	Hitless fail: Waiting for module in slot <n> to burn new SW and reboot Timed out. (n – slot number).

### 4.17.3 Other Traps

The following are provided as SNMP traps and are not alarms.

**Table 4-60: coldStart Trap**

<b>Trap Name:</b>	coldStart
<b>OID:</b>	1.3.6.1.6.3.1.1.5.1
<b>MIB:</b>	SNMPv2-MIB
<b>Note:</b>	This is a trap from the standard SNMP MIB.

**Table 4-61: authenticationFailure Trap**

<b>Trap Name:</b>	authenticationFailure
<b>OID:</b>	1.3.6.1.6.3.1.1.5.5
<b>MIB:</b>	SNMPv2-MIB

**Table 4-62: acBoardEvBoardStarted Trap**

<b>Trap Name:</b>	acBoardEvBoardStarted
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.4
<b>MIB:</b>	AcBoard
<b>Severity:</b>	cleared
<b>Event Type:</b>	equipmentAlarm
<b>Probable Cause:</b>	Other(0)
<b>Alarm Text:</b>	Initialization Ended
<b>Note:</b>	This is the AudioCodes Enterprise application cold start trap.



**Table 4-63: AcDChannelStatus Trap (Applicable Only to Digital Devices)**

<b>Trap Name:</b>	acDChannelStatus
<b>OID:</b>	1.3.6.1.4.1.5003.9.10.1.21.2.0.37
<b>MIB</b>	AcBoard
<b>Severity:</b>	Minor
<b>Event Type:</b>	communicationsAlarm
<b>Probable Cause:</b>	communicationsProtocolError
<b>Alarm Text:</b>	D-Channel Trap.
<b>Source:</b>	Trunk <m> where m is the trunk number (starts from 0).
<b>Status Changes:</b>	
<b>Condition:</b>	D-Channel un-established.
<b>Trap Status:</b>	Trap is sent with the severity of Minor.
<b>Condition:</b>	D-Channel established.
<b>Trap Status:</b>	Trap is sent with the severity of Cleared.

#### 4.17.4 Trap Varbinds

Each trap described above provides the following fields (known as *varbinds*). Refer to the AcBoard MIB for additional details on these varbinds.

- acBoardTrapGlobalsName
- acBoardTrapGlobalsTextualDescription
- acBoardTrapGlobalsSource
- acBoardTrapGlobalsSeverity
- acBoardTrapGlobalsUniqID
- acBoardTrapGlobalsType
- acBoardTrapGlobalsProbableCause
- acBoardTrapGlobalsDateAndTime
- acBoardTrapGlobalsAdditionalInfo1
- acBoardTrapGlobalsAdditionalInfo2
- acBoardTrapGlobalsAdditionalInfo3



**Note:** 'acBoardTrapGlobalsName' is actually a number. The value of this varbind is 'X' minus 1, where 'X' is the last number in the trap's OID. For example, the 'name' of 'acBoardEthernetLinkAlarm' is '9'. The OID for 'acBoardEthernetLinkAlarm' is 1.3.6.1.4.1.5003.9.10.1.21.2.0.10.

### 4.17.5 Customizing Trap's Enterprise OID

You can change the enterprise value in the device's SNMP Traps to a variable value using the *ini* parameter `SNMPTrapEnterpriseOid`. This parameter replaces the Traps' OID prefix from 'AcTrap' (1.3.6.1.4.1.5003.9.10.1.21) to user-defined root. All other OIDs remain the same.

For example, the current `acBoardEvBoardStarted` parameter's OID is '1.3.6.1.4.1.5003.9.10.1.21.2.0.4'. Its prefix ('1.3.6.1.4.1.5003.9.10.1.21') can be changed, and all other OIDs remain the same.

## 5 Automatic Device Configuration



**Note:** This section is applicable only to AudioCodes' Analog and 2000 Series devices, unless otherwise specified.

Large-scale deployment of devices calls for automated installation and setup capabilities. 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 devices may sometimes be pre-configured during the manufacturing process by AudioCodes (commonly known as *private labeling*). Typically, a two-stage configuration process is implemented such that initial configuration includes only the basic configurations, while the final configuration is performed when the device is deployed in a live network.

### 5.1 Automatic Configuration Options

This available options for performing fast, automatic device configuration includes the following:

- Local configuration using BootP/TFTP server (refer to "Local Configuration Server with BootP/TFTP" on page 95)
- Configuration using DHCP (refer to "DHCP-based Configuration Server" on page 96)
- Configuration using DHCP Option 67 (refer to "Configuration using DHCP Option 67" on page 96)
- Configuration using DHCP Option 66 (refer to "TFTP Configuration using DHCP Option 66" on page 97)
- Configuration using HTTP (refer to "HTTP-based Automatic Updates" on page 97)
- Configuration using FTP or NFS ("Configuration using FTP or NFS" on page 98)
- Configuration using AudioCodes Element Management System (refer to "Configuration using AudioCodes EMS" on page 98)

#### 5.1.1 Local Configuration Server with BootP/TFTP

Local configuration server with BootP/TFTP provides the most efficient and easiest method for automatic configuration, where configuration occurs at a staging warehouse, as described below:

1. A computer running BootP and TFTP software is located in a staging warehouse.
2. A standard *ini* configuration file is prepared and located in the TFTP directory.
3. BootP is configured with the MAC address of each device.
4. Each device is connected to the network and powered-up.
5. The BootP reply contains the *cmp* and *ini* file names entered in the 'Boot File' field. The device retrieves these files using BootP and stores them in its flash memory.
6. If auxiliary files are required (coefficients, call progress tones etc.), they may be specified in the *ini* file and downloaded from the same TFTP server.

7. When the devices' LEDs turn green (i.e., files successfully loaded to the devices), the devices may be disconnected and shipped to the customer.
8. Local IP addressing at the customer site would typically be provided by DHCP.

For additional information, refer to "Using BootP / DHCP" on page 19.

### 5.1.2 DHCP-based Configuration Server

This alternative is similar to the setup described in "Local Configuration Server with BootP/TFTP" on page 95, except that DHCP is used instead of BootP. The DHCP server may be specially configured to automatically provide AudioCodes devices with a temporary IP address so that individual MAC addresses are not required. In this method, configuration occurs at a staging warehouse. For additional information, refer to "Using BootP / DHCP" on page 19.

Below is a sample configuration file for 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 "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;
        filename "MP118 SIP 5.00A.001.cmp -fb;mp118.ini";
        option routers                10.31.0.1;
        option subnet-mask            255.255.0.0;
    }
}
```

### 5.1.3 Configuration using DHCP Option 67

This method is suitable for deployments where DHCP server configuration is feasible at the customer site. Most DHCP servers allow configuring individual DHCP option values for different devices on the network. The DHCP configuration should be modified so that the device receives a configuration URL in option 67, along with IP addressing and DNS server information. The DHCP response is processed by the device upon startup, and consequently the HTTP server specified by the configuration URL is contacted to complete the configuration. This method does not require additional servers at the customer premises and 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;
    }
}
```

### 5.1.4 TFTP Configuration using DHCP Option 66



**Note:** This subsection is applicable only to AudioCodes' Analog devices.

This method is suitable when the customer's network contains a provisioning TFTP server for all network equipment, without the possibility of distinction between AudioCodes and non-AudioCodes devices. Upon startup, the device searches for option 66 in the DHCP response. If option 66 contains a valid IP address, a TFTP download is attempted for a file named after the device's MAC address, e.g., "00908f0130aa.ini". This method requires a configuration 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 for 10 seconds while the Ethernet cable is not connected). Note that access to the core network using TFTP is not NAT-safe.

### 5.1.5 HTTP-based Automatic Updates

An HTTP (or HTTPS) server can be placed in the customer's core network where configuration and software updates are available for download. This alternative does not require additional servers at the customer premises and is NAT-safe. For example, assume the core network HTTP server is <https://www.corp.com>. A master configuration *ini* file should be placed on the HTTP server, e.g., <https://www.corp.com/audiocodes/master.ini>. This *ini* file could point to additional *ini* files, auxiliary files (voice prompts, call progress tones, coefficients etc.), and software upgrades *cmp* files, all on the HTTP server or other HTTP servers in the core network.

The main advantage of this method is that the HTTP configuration can be checked periodically when the device is deployed at the customer site; HTTP(S) is not sensitive to NAT devices, allowing configuration whenever needed without on-site intervention.

For additional security, the URL may contain a different port, and a user name and password.

The devices should only be configured with the URL of the initial *ini* file. There are several methods for performing this:

- Using methods described in "DHCP-based Configuration Server" on page 96 or above, via TFTP at a staging warehouse. The *ini* file parameter controlling the configuration URL is *IniFileURL*.
- Private labeling at AudioCodes.
- Using DHCP option 67 (see method described in "Configuration using DHCP Option 67" on page 96).
- Manually on-site, using the RS-232 port or Web interface.

When the device is deployed at the customer site, local DHCP provides IP addressing and DNS server information. The device can then contact the HTTP server at the core network and complete 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` becomes: `http://corp.com/config-00908f030012.ini`
- `http://corp.com/<IP>/config.ini` becomes: `http://corp.com/192.168.0.7/config.ini`

Software upgrades may be performed using the parameter *CmpFileURL*. Inclusion of this parameter in the master *ini* file causes the devices to download and store the specified software image.

For additional information, refer to "Automatic Update Mechanism" on page 24.

## 5.1.6 Configuration using FTP or NFS

Some networks block access to HTTP(S). The Automatic Update facility provides limited support for FTP/FTPS connectivity. However, periodic polling for updates is not possible (since these protocols don't support conditional fetching, i.e., updating files only if it is changed on the server).

The difference between this method and methods described in "HTTP-based Automatic Updates" on page 97 and "Configuration using DHCP Option 67" on page 96 is simply the protocol in the URL -- 'ftp' instead of 'http'. NFS v2/v3 is supported as well.



**Note:** Unlike FTP, NFS is not NAT-safe.

## 5.1.7 Configuration 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.

## 5.2 Loading Files Securely (Disabling TFTP)

The TFTP protocol is not considered secure; some network operators block it using a firewall. It is possible to disable TFTP completely, using the *ini* file parameter `EnableSecureStartup` (set to 1). This way, secure protocols such as HTTPS may be used to fetch the device configuration.

➤ **To download the ini file to the device using HTTPS instead of TFTP, take the following 4 steps:**

1. Prepare the device's configuration file on an HTTPS server, and obtain a URL to the file (e.g., `https://192.168.100.53/audiocodes.ini`).
2. Enable DHCP if necessary.
3. Enable SSH and connect to it (refer to "Starting a CLI Management Session" on page 27).
4. In the CLI, use the *ini* file parameters `IniFileURL` (for defining the URL of the configuration file) and `EnableSecureStartup` (for disabling TFTP), and then restart the device with the new configuration:

```
/conf/scp IniFileURL https://192.168.100.53/audiocodes.ini
/conf/scp EnableSecureStartup 1
/conf/sar bootp
```



**Note:** Once Secure Startup has been enabled, it can only be disabled by setting `EnableSecureStartup` to 0 using the CLI. Loading a new *ini* file using BootP/TFTP is not possible until `EnableSecureStartup` is disabled.

## Reader's Notes



## 6 Security

This section describes the security mechanisms and protocols implemented on the device. The following list specifies the available security protocols and their objectives:

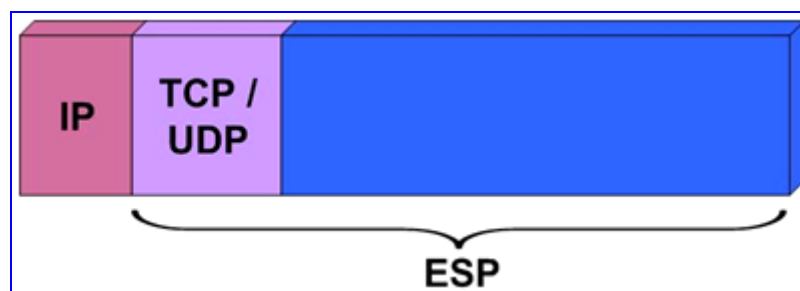
- IPsec and IKE protocols are part of the IETF standards for establishing a secured IP connection between two applications. IPsec and IKE are used in conjunction to provide security for control and management protocols but not for media (refer to "IPsec and IKE" on page 101).
- SSL (Secure Socket Layer) / TLS (Transport Layer Security). The SSL / TLS protocols are used to provide privacy and data integrity between two communicating applications over TCP/IP. They are used to secure the following applications: SIP Signaling (SIPS), Web access (HTTPS) and Telnet access (refer to "SSL/TLS" on page 107).
- Secured RTP (SRTP) according to RFC 3711 - used to encrypt RTP and RTCP transport (refer to "SRTP" on page 109).
- RADIUS (Remote Authentication Dial-In User Service) - RADIUS server is used to enable multiple-user management on a centralized platform (refer to "RADIUS Login Authentication" on page 110).
- Internal Firewall for filtering unwanted inbound traffic (refer to "Internal Firewall" on page 114).

### 6.1 IPsec and IKE

IPsec and Internet Key Exchange (IKE) protocols are part of the IETF standards for establishing a secured IP connection between two applications (also referred to as peers). Providing security services at the IP layer, IPsec and IKE are transparent to IP applications. IPsec and IKE are used in conjunction to provide security for control and management (e.g., SNMP and Web) protocols, but not for media (i.e., RTP, RTCP and T.38).

IPsec is responsible for securing the IP traffic. This is accomplished by using the Encapsulation Security Payload (ESP) protocol to encrypt the IP payload (illustrated in the following figure). The IKE protocol is responsible for obtaining the IPsec encryption keys and encryption profile (known as IPsec Security Association - SA).

Figure 6-1: IPsec Encryption



**Note:** IPsec doesn't function fully if the device's IP address is changed on-the-fly due to the fact that the crypto hardware can only be configured on reset. Therefore, reset the device after you change its IP address.

## 6.1.1 IKE

IKE is used to obtain the Security Associations (SA) between peers (the device and the application it's trying to contact). The SA contains the encryption keys and profile used by the IPSec to encrypt the IP stream. The IKE table lists the IKE peers with which the device performs the IKE negotiation (up to 20 peers are available).

The IKE negotiation is separated into two phases: main mode and quick mode. The main mode employs the Diffie-Hellman (DH) protocol to obtain an encryption key (without any prior keys), and uses a pre-shared key to authenticate the peers. The created channel secures the messages of the following phase (quick mode) in which the IPSec SA properties are negotiated.

The IKE negotiation is as follows:

- Main mode (the main mode creates a secured channel for the quick mode):
  - **SA negotiation:** The peers negotiate their capabilities using two proposals. Each proposal includes three parameters: Encryption method, Authentication protocol and the length of the key created by the DH protocol. The key's lifetime is also negotiated in this stage. For detailed information on configuring the main mode proposals, refer to 'IKE Configuration' in the device's *User's Manual*.
  - **Key exchange (DH):** The DH protocol is used to create a phase-1 key.
  - **Authentication:** The two peers authenticate one another using the pre-shared key (configured by the parameter IKEPolicySharedKey).
- Quick mode (quick mode negotiation is secured by the phase-1 SA):
  - **SA negotiation:** The peers negotiate their capabilities using a single proposal. The proposal includes two parameters: Encryption method and Authentication protocol. The lifetime is also negotiated in this stage. For detailed information on configuring the quick mode proposal, refer to the SPD table under 'IPSec Configuration' in the device's *User's Manual*.
  - **Key exchange:** a symmetrical key is created using the negotiated SA.

### IKE Specifications:

- Authentication mode: pre-shared key only
- Main mode is supported for IKE Phase 1
- Supported IKE SA encryption algorithms: Data Encryption Standard (DES), 3DES, and Advanced Encryption Standard (AES)
- Hash types for IKE SA: SHA1 and MD5

## 6.1.2 IPSec

IPSec is responsible for encrypting and decrypting the IP streams. The IPSec Security Policy Database (SPD) table defines up to 20 IP peers to which the IPSec security is applied. IPSec can be applied to all packets designated to a specific IP address or to a specific IP address, port (source or destination), and protocol type.

Each outgoing packet is analyzed and compared to the SPD table. The packet's destination IP address (and optionally, destination port, source port, and protocol type) are compared to each entry in the table. If a match is found, the device checks if a Security Association (SA) already exists for this entry. If it doesn't, the IKE protocol is invoked (refer to "IKE" on page 102) and an IPSec SA is established. The packet is encrypted and transmitted. If a match isn't found, the packet is transmitted un-encrypted.



**Note:** An incoming packet whose parameters match one of the entries in the SPD table and is received un-encrypted is dropped.

#### IPSec Specifications:

- Transport or Tunneling mode
- Encapsulation Security Payload (ESP) only
- Support for Cipher Block Chaining (CBC)
- Supported IPSec SA encryption algorithms - DES, 3DES, and AES
- Hash types for IPSec SA include SHA1 and MD5

### 6.1.3 IPSec and IKE Configuration Table's Confidentiality

Since the pre-shared key parameter of the IKE table must remain undisclosed, measures are taken by the *ini* file, Web interface and SNMP agent to maintain this parameter's confidentiality. In the Web interface, a list of asterisks is displayed instead of the pre-shared key. In SNMP, the pre-shared key parameter is a write-only parameter and cannot be read. In the *ini* file, the following measures to assure the secrecy of the IPSec and IKE tables are taken:

- **Hidden IPSec and IKE tables:** When uploading the *ini* file from the device, the IPSec and IKE tables are not displayed. Instead, the notifications shown in the following figure are displayed.

```
; *** TABLE IPSEC IKEDB TABLE ***
; This table contains hidden elements and will not be exposed.
; This table exists on blade and will be saved during restarts
;
; *** TABLE IPSEC SPD TABLE ***
; This table contains hidden elements and will not be exposed.
; This table exists on blade and will be saved during restarts
```

- **Preserving the values of the parameters in the IPSec and IKE tables from one *ini* file loading to the next:** The values configured for the parameters in the IPSec tables in the *ini* file are preserved from one loading to another. If a newly loaded *ini* file doesn't define IPSec tables, the previously loaded tables remain valid. To invalidate a previously loaded *ini* file IPSec table, load a new *ini* file with an empty IPSec table, as shown below:

```
[IPSec IKEDB Table]
[\IPSec_IKEDB_Table]

[IPSEC SPD TABLE]
[\IPSEC SPD TABLE]
```

## 6.1.4 Dead Peer Detection (RFC 3706)

When two peers communicate with IKE and IPSec, the situation may arise in which connectivity between the two goes down unexpectedly. In such cases, there is often no way for IKE and IPSec to identify the loss of peer connectivity. As such, the Security Associations (SA) can remain until their lifetimes naturally expire, resulting in a 'black hole' situation where the packets are lost.

The detection of such a scenario is achieved by performing message exchanges between the peers and when no reply is received, the sender assumes SA's are no longer valid on the remote peer and attempt to renegotiate.

The device can be configured to query the liveliness of its Internet Key Exchange (IKE) peer at regular intervals or on-demand, using the IPSec Dead Peer Detection (DPD) feature. DPD is auto negotiated.

To activate the DPD feature, the *ini* file parameter IPsecDPDMode must be set to one of the below values:

- [0] = Disabled (default).
- [1] = Periodic message exchanges at regular intervals.
- [2] = On-demand message exchanges as needed (i.e., before sending data to the peer). If the liveliness of the peer is questionable, the device sends a DPD message to query the status of the peer. If the device has no traffic to send, it never sends a DPD message.

## 6.1.5 Certificate Revocation Checking

Some Public-Key Infrastructures (PKI) can revoke a certificate after it has been issued. The device, which employs SSL/TLS and IPSec, may be configured to check whether a peer's certificate has been revoked, using the Online Certificate Status Protocol (OCSP).

To enable OCSP, the following *ini* file parameters must be configured:

- OCSPEnable
- OCSPServerIP
- OCSPServerPort
- OCSPDefaultResponse

For a description on these parameters, refer to the device's User's Manual.

When OCSP is enabled, the device queries the OCSP server for revocation information whenever a peer certificate is received (IPSec, TLS client mode, or TLS server mode with mutual authentication).



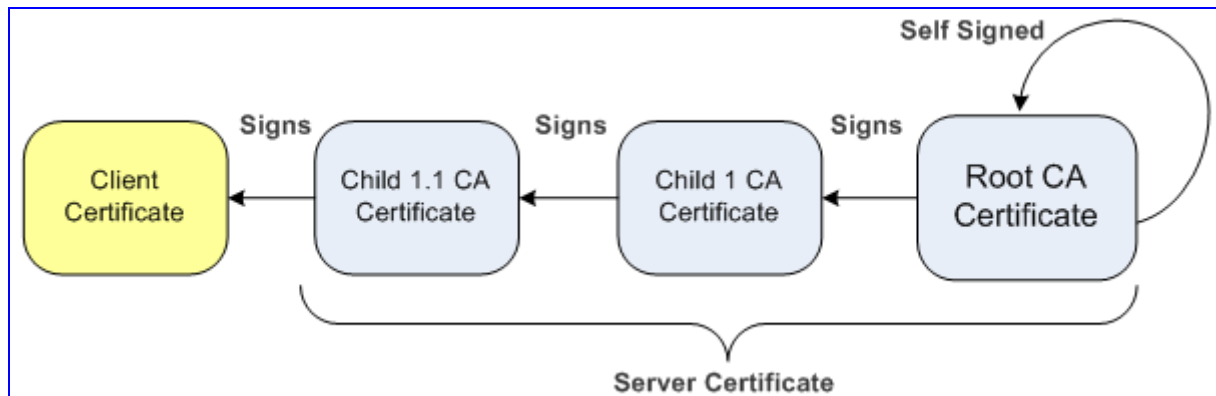
### Notes:

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

### 6.1.6 Certificate Chain

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 6-2: Certificate Chain Hierarchy**



**Note:** The chained certificate is limited to up to 9,000 characters (including the certificates headers).

## 6.2 Secure Shell

The device's command-line interface (CLI) may be accessed using Telnet. However, unless configured for TLS mode, 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.

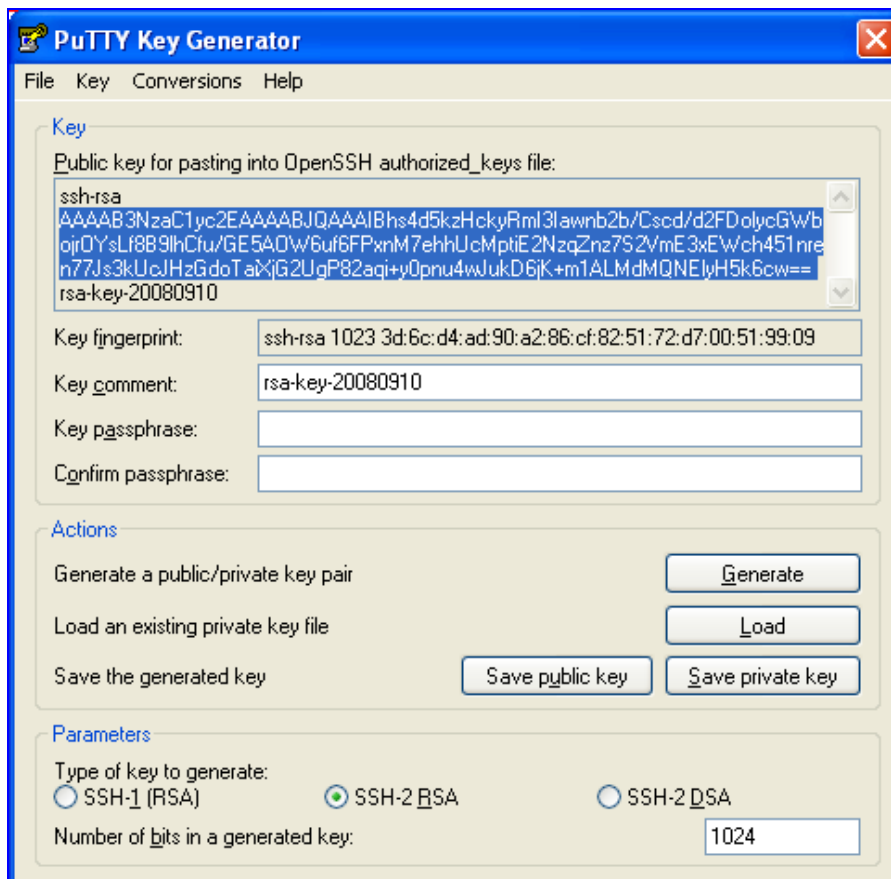
SSH requires appropriate client software for the management PC. Most Linux distributions have OpenSSH pre-installed; Windows-based PCs require an SSH client software such as PuTTY, which can be downloaded from <http://www.chiark.greenend.org.uk/~sgtatham/putty/>.

By default, SSH uses the same user name and password as the Telnet server and Web server. In addition, SSH supports 1024-bit RSA public keys, which provide carrier-grade security. Follow the instructions below to configure the device with an administrator RSA key as a means of strong authentication.

➤ **To configure RSA public keys for Windows (using PuTTY SSH software), take these steps:**

1. Run the file *puttygen.exe*; the PuTTY Key Generator program starts, displaying the main window.

2. Under the 'Parameters' group, perform the following:
  - a. Select the option 'SSH-2 RSA'.
  - b. In the field 'Number of bits in a generated key', enter "1024" bits.
3. Under the 'Actions' group, click **Generate**, and then follow the on-screen instructions.
4. Under the 'Actions' group, save the new private key to a file (\*.ppk) on your PC, by clicking **Save private key**.
5. Under the 'Key' group, select the displayed encoded text between "ssh-rsa" and "rsa-key-....", as shown in the example below:



6. Open the device's ini file, and then paste the public key (that you copied in Step 5) as the value for the parameter SSHAdminKey, as shown below:

```
SSHAdminKey = AAAAB3NzaC1yc2EAAAABJQ...
```

7. Load the ini file to the device.
8. Run the file *PuTTY.exe*; the PuTTY Configuration program starts.
9. In the 'Category' tree, drill down the tree by selecting **Connection**, then **SSH**, and then **Auth**; the 'Options controlling SSH authentication' pane appears.
10. Under the 'Authentication parameters' group, click **Browse** and then locate the private key file that you created and saved in Step 4.
11. Connect to the device with SSH, using the user name "Admin"; RSA key negotiation occurs automatically and no password is required.

➤ **To configure RSA public keys for Linux (using OpenSSH 4.3), take these 5 steps:**

1. Run the following command:  

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

A new key is created in the file *admin.key* and the public portion is saved to the file *admin.key.pub*.
2. Open the file *admin.key.pub*, and then copy the long encoded string from "ssh-rsa" up to the white-space.
3. Open the device's ini file, and then set the SSHAdminKey to the value copied in Step 2, e.g.:

```
SSHAdminKey = AAAAB3NzaC1yc2EAAAABJQ...
```

4. Load the ini file to the device.
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.

For additional security, you can set the *ini* file parameter *SSHRequirePublicKey* to 1. This ensures that SSH access is only possible by using the RSA key and not by using user name and password.

## 6.3 SSL / TLS

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

Specifications for the SSL/TLS implementation include the following:

- **Transports:** SSL 2.0, SSL 3.0, TLS 1.0.
- **Ciphers:** DES, RC4 compatible, Advanced Encryption Standard (AES).
- **Authentication:** X.509 certificates (CRLs are currently not supported). The device supports the receipt of wildcards ("\*") in X.509 Certificates when establishing TLS connections. These wildcards can be part of the CN attribute of the Subject field or the DNSName attribute of the SubjectAltName field.



**Tip:** A common security practice is to disable SSLv2/SSLv3 and use only TLSv1. This can be achieved by setting the *ini* file parameter *TLSVersion* to 1. If using Microsoft Internet Explorer, ensure you disable SSL 2.0 / SSL 3.0 and enable TLS 1.0 in Internet Explorer (**Tools > Internet Options > Advanced**).



### 6.3.1 SIP Over TLS (SIPS)

The device uses TLS over TCP to encrypt SIP transport and (optionally) to authenticate it. To enable TLS on the device, set the selected transport type to TLS (SIPTransportType = 2). In this mode, 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 EnableSIPS to 1. When a TLS connection with the device is initiated, the device also responds using TLS, regardless of the configured SIP transport type (in this case, the parameter EnableSIPS is also ignored).

TLS and SIPS use the Certificate Exchange process described in Server Certificate Replacement and Client Certificates in the *User's Manual*. To change the port number used for SIPS transport (by default, 5061), use the parameter TLSLocalSIPPort.

When SIPS is implemented, it is sometimes required to use two-way authentication. When acting as the TLS server (in a specific connection), it is possible to demand the authentication of the client's certificate. To enable two-way authentication on the device, set the *ini* file parameter SIPSRequireClientCertificate to 1. For information on installing a client certificate, refer to Client Certificates described in the *User's Manual*.

### 6.3.2 Secured HTTPS Web Interface Configuration

For additional security, you can configure the Web interface to accept only secured (HTTPS) connections by setting the parameter HTTPSONly to 1 (described in the device's *User's Manual*). You can also change the port number used for the secured Web server (by default, 443), by changing the *ini* file parameter, HTTPSPort (described in the device's *User's Manual*).

➤ **To use the secured Web interface, take these 3 steps:**

1. Access the device using the following URL: `https://[host name or IP address]`  
Depending on the browser's configuration, a security warning dialog may be displayed. The reason for the warning is that the device initial certificate is not trusted by your PC. The browser may allow you to install the certificate, thus skipping the warning dialog the next time you connect to the device.
2. If you are using Internet Explorer, click **View Certificate**, and then **Install Certificate**.
3. The browser also warns you if the host name used in the URL is not identical to the one listed in the certificate. To solve 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:

```
# This is a sample HOSTS file used by Microsoft TCP/IP for
Windows.
# Location: C:\WINDOWS\SYSTEM32\DRIVERS\ETC\hosts
#
127.0.0.1    localhost
10.31.4.47   ACL 280152
```

### 6.3.3 Secured Telnet

To enable the embedded Telnet server on the device, set the parameter TelnetServerEnable (described in 'Web and Telnet Parameters' in the device's *User's Manual*) to 1 (standard mode) or 2 (SSL mode); no information is transmitted in the clear when SSL mode is used.



If the Telnet server is set to SSL mode, 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, Kermit-95 for Windows, and AudioCodes' acSSLTelnet utility for Windows (that requires prior installation of the free OpenSSL toolkit). Contact AudioCodes to obtain the acSSLTelnet utility.

For security reasons, some organizations require the display of a proprietary notice upon starting a Telnet session. The following is an example of a configuration *ini* file for defining such a message:

```
[ WelcomeMessage ]
FORMAT WelcomeMessage Index = WelcomeMessage Text ;
WelcomeMessage 01 = "WARNING! This computer system and network is
PRIVATE and PROPRIETARY and may" ;
WelcomeMessage 02 = "only be accessed by authorized users.
Unauthorized use of this computer" ;
WelcomeMessage 03 = "system or network is strictly prohibited and
may be subject to criminal" ;
WelcomeMessage 04 = "prosecution, employee discipline up to and
including discharge, or the" ;
WelcomeMessage 05 = "termination of vendor/service contracts. The
owner, or its agents, may" ;
WelcomeMessage 06 = "monitor any activity or communication on the
computer system or network." ;
WelcomeMessage 07 = "The owner, or its agents, may retrieve any
information stored within the" ;
WelcomeMessage 08 = "computer system or network. By accessing and
using this computer system or" ;
WelcomeMessage 09 = "network, you are consenting to such
monitoring and information retrieval for" ;
WelcomeMessage 10 = "law enforcement and other purposes. Users
should have no expectation of" ;
WelcomeMessage 11 = "privacy as to any communication on or
information stored within the computer" ;
WelcomeMessage 12 = "system or network, including information
stored locally or remotely on a hard" ;
WelcomeMessage 13 = "drive or other media in use with this
computer system or network." ;
[ /WelcomeMessage ]
```

## 6.4 SRTP

The device supports Secured RTP (SRTP) according to RFC 3711. SRTP is used to encrypt RTP and RTCP transport as it is best suited for protecting VoIP traffic.

SRTP requires a Key Exchange mechanism that is performed according to RFC 4568 – “Session Description Protocol (SDP) Security Descriptions for Media Streams”. The Key Exchange is executed by adding a ‘Crypto’ attribute to the SDP. This attribute is used (by both sides) to declare the various supported cipher suites and to attach the encryption key to use. If negotiation of the encryption data is successful, the call is established.

SRTP implementation supports the following suites:

- AES\_CM\_128\_HMAC\_SHA1\_32
- AES\_CM\_128\_HMAC\_SHA1\_80

All other suites are ignored.

The only supported key parameter is Master Key Identifier (MKI) value. When the device is the offering side, it generates an MKI of a size defined by the *ini* file parameter `SRTPTxPacketMKISize`. The length of the MKI is limited to four bytes. If the remote side sends a longer MKI, then the key is ignored. The key lifetime field is not supported. However, if it is included in the key it is ignored and the call does not fail.

The device supports the following Session parameters (as defined in RFC 4568, SDP Security Descriptions for Media Streams):

- UNENCRYPTED\_S RTP
- UNENCRYPTED\_S RTPC
- UNAUTHENTICATED\_S RTP

Session parameters should be the same for the local side and remote side. When the device is the offering side, the session parameters are defined according to the following *ini* file parameters: `RTPEncryptionDisableTx`, `RTCPEncryptionDisableTx`, and `RTPAuthenticationDisableTx`. 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
```

Use the parameter `MediaSecurityBehaviour` (described in the device's *User's Manual*) to select the device's mode of operation that determines the behavior of the device if negotiation of the cipher suite fails:

- **Mandatory:** the call is terminated. Incoming calls that don't include encryption information are rejected.
- **Preferable:** an unencrypted call is established. Incoming calls that don't include encryption information are accepted.

To enable SRTP, set the parameter `EnableMediaSecurity` to 1 (described in the device's *User's Manual*).



#### Notes:

- When SRTP is used, the channel capacity is reduced (refer to the parameter `EnableMediaSecurity`).
- The device supports only the AES 128 in CM mode cipher suite.

## 6.5 RADIUS Login Authentication

Users can enhance the security and capabilities of logging to the device's Web and Telnet embedded servers by using a Remote Authentication Dial-In User Service (RADIUS) to store numerous user names, passwords and access level attributes (Web only), allowing multiple user management on a centralized platform. RADIUS (RFC 2865) is a standard authentication protocol that defines a method for contacting a predefined server and verifying a given name and password pair against a remote database in a secure manner.

When accessing the Web and Telnet servers, users must provide a valid user name and password. When RADIUS authentication isn't used, the user name and password are authenticated with the Web interface's user names and passwords of the primary or secondary accounts (refer to 'User Accounts' in the device's *User's Manual*) or with the Telnet server's user name and password stored internally in the device's memory. When RADIUS authentication is used, the device doesn't store the user name and password but simply forwards them to the pre-configured RADIUS server for authentication (acceptance or rejection). The internal Web / Telnet passwords can be used as a fallback mechanism in case the RADIUS server doesn't respond (configured by the parameter `BehaviorUponRadiusTimeout`). Note that when RADIUS authentication is performed, the Web / Telnet servers are blocked until a response is received (with a timeout of 5 seconds).

RADIUS authentication requires HTTP basic authentication, meaning the user name and password are transmitted in clear text over the network. Therefore, users are recommended to set the parameter `HttpsOnly` to 1, to force the use of HTTPS, since the transport is encrypted.

## 6.5.1 Setting Up a RADIUS Server

The following examples refer to FreeRADIUS, a free RADIUS server that can be downloaded from [www.freeradius.org](http://www.freeradius.org). Follow the directions on that site for information on installing and configuring the server. If you use a RADIUS server from a different vendor, refer to its appropriate documentation.

### ➤ To set up a RADIUS server, take these 5 steps:

1. Define the device as an authorized client of the RADIUS server, with a predefined 'shared secret' (a password used to secure communication) and a vendor ID. Below is an example of the file `clients.conf` (FreeRADIUS client configuration).

```
#
# clients.conf - client configuration directives
#
client 10.31.4.47 {
    secret          = FutureRADIUS
    shortname       = tp1610 master tpm
}
```

2. If access levels are required, set up a VSA dictionary for the RADIUS server and select an attribute ID that represents each user's access level. The following example shows a dictionary file for FreeRADIUS (FreeRADIUS Client Configuration) that defines the attribute 'ACL-Auth-Level' with ID=35.

```
#
# 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. In the RADIUS server, define the list of users authorized to use the device, using one of the password authentication methods supported by the server implementation. The following example 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

larry   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', vendor ID and VSA access level identifier (if access levels are used) used by the RADIUS server.
5. Configure the device's relevant parameters according to "Configuring RADIUS Support" on page 112.

## 6.5.2 Configuring RADIUS Support

The procedure below describes how to configure RADIUS for the device using the Web interface. For information on the RADIUS parameters, refer to the device's *User's Manual*.

➤ **To configure RADIUS support using the Web interface, take these 13 steps:**

1. Access the Web interface (refer to the device's *User's Manual*).
2. Open the 'General Security Settings' screen (**Advanced Configuration** menu > **Security Settings** > **General Security Settings** option); the 'General Security Settings' screen is displayed.
3. Under section 'General RADIUS Settings', in the field 'Enable RADIUS Access Control', select 'Enable'; the RADIUS application is enabled.
4. In the field 'Use RADIUS for Web / Telnet Login', select 'Enable'; RADIUS authentication is enabled for Web and Telnet login.
5. Enter the RADIUS server IP address, port number and shared secret in the relevant fields.

6. Under section 'RADIUS Authentication Settings', in the field 'Device Behavior Upon RADIUS Timeout', select the device's operation if a response isn't received from the RADIUS server after the 5 seconds timeout expires:
  - Deny Access: the device denies access to the Web and Telnet embedded servers.
  - Verify Access Locally: the device checks the local user name and password.
7. In the field 'Local RADIUS Password Cache Timeout', enter a time (in seconds); when this time expires, the user name and password verified by the RADIUS server becomes invalid and a user name and password must be re-validated with the RADIUS server.
8. In the field 'Local RADIUS Password Cache Mode', select the device's mode of operation regarding the above-mentioned 'Local RADIUS Password Cache Timer' option:
  - Reset Timer Upon Access: upon each access to a Web screen, the timer resets (reverts to the initial value configured in the previous step).
  - Absolute Expiry Timer: when you access a Web screen, the timer doesn't reset but rather continues decreasing.
9. In the field 'RADIUS VSA Vendor ID', enter the vendor ID you configured in the RADIUS server:
10. When using the Web access-level mechanism, perform one of the following options:
  - When RADIUS responses include the access level attribute:  
In the field 'RADIUS VSA Access Level Attribute', enter the code that indicates the access level attribute in the Vendor Specific Attributes (VSA) section of the received RADIUS packet.
  - When RADIUS responses don't include the access level attribute:  
In the field 'Default Access Level', enter the default access level that is applied to all users authenticated by the RADIUS server.
11. In the field 'Require Secured Web Connection (HTTPS)', select 'HTTPS only'.  
It is important you use HTTPS (secure Web server) when connecting to the device over an open network, since the password is transmitted in clear text. Similarly, for Telnet, use SSL TelnetServerEnable = 2 or SSH (refer to "Secured Telnet" on page 108).
12. Save the changes so they are available after a power fail.
13. Reset the device (refer to the device's *User's Manual*).

After reset, when accessing the Web or Telnet servers, use the user name and password you configured in the RADIUS database. The local system password is still active and can be used when the RADIUS server is down.

➤ **To configure RADIUS support on the device using the *ini* file, take these 3 steps:**

1. Add the following parameters to the *ini* file.
  - EnableRADIUS = 1
  - WebRADIUSLogin = 1
  - RADIUSAuthServerIP = IP address of RADIUS server
  - RADIUSAuthPort = port number of RADIUS server, usually 1812

- SharedSecret = your shared secret
  - HTTPSONly = 1
  - RadiusLocalCacheMode = 1
  - RadiusLocalCacheTimeout = 300
  - RadiusVSAVendorID = your vendor's ID
  - RadiusVSAAccessAttribute = code that indicates the access level attribute
  - DefaultAccessLevel = default access level (0 to 200)
2. Authenticating via RADIUS with credentials in the URL:
- The device is capable of authenticating via RADIUS server when the UserName/Password are in the URL, e.g.,:  
  
`http://10.4.4.112/Forms/RadiusAuthentication?WSBackUserName=Guyy&WSBackPassword=1234`
  - This method is applicable when using RADIUS server with HTTP basic authentication. Note that only one connection is possible at a time.
3. To set this feature, use RADIUS with Basic authentication settings:
- a. Default settings: You are prompted for your login every time you connect to the blade.
  - b. Enable RADIUS configuration as described above.
  - c. Enable Basic HTTP authentication settings.
  - d. Connect to the device using a URL as in the example.

This feature is restricted to five simultaneous users only.

## 6.6 Internal Firewall

The device accommodates an internal access list facility, allowing the security administrator to define network traffic filtering rules. The access list provides the following features:

- Block traffic from known malicious sources
- Only allow traffic from known friendly sources, and block all others
- Mix allowed and blocked network sources
- Limit traffic to a predefined rate (blocking the excess)
- Limit traffic to specific protocols, and specific port ranges on the device

The access list consists of a table with up to 50 ordered lines. For each packet received on the network interface, the table is scanned from the top until a matching rule is found (or the table end is reached). This rule can either block the packet or allow it; however it is important to note that subsequent rules aren't scanned. If the table end is reached without a match, the packet is accepted.

Each rule is composed of the following fields (described in the device's *User's Manual*):

- IP address (or DNS name) of source network
- IP network mask
- Destination UDP/TCP ports (on this device)
- Protocol type
- Maximum packet size, byte rate per second, and allowed data burst

■ Action upon match (allow or block)

The internal firewall can be configured using the *ini* file or the Web interface (refer to the device's *User's Manual*).

Below is an example of an access list definition via *ini* file:

```
[ACCESSLIST]
FORMAT AccessList Index = AccessList Source IP,
AccessList_Net_Mask, AccessList_Start_Port, AccessList_End_Port,
AccessList_Protocol, AccessList_Packet_Size, AccessList_Byte_Rate,
AccessList_Byte_Burst, AccessList_Allow_Type;
AccessList 10 = mgmt.customer.com, 255.255.255.255, 0, 80, tcp, 0,
0, 0, allow;
AccessList 15 = 192.0.0.0, 255.0.0.0, 0, 65535, any, 0, 40000,
50000, block;
AccessList 20 = 10.31.4.0, 255.255.255.0, 4000, 9000, any, 0, 0,
0, block;
AccessList 22 = 10.4.0.0, 255.255.0.0, 4000, 9000, any, 0, 0, 0,
block ;
[\\ACCESSLIST]
```

Explanation of the example access list:

- Rule #10: traffic from the host 'mgmt.customer.com' destined to TCP ports 0 to 80, is always allowed.
- Rule #15: traffic from the 192.xxx.yyy.zzz subnet, is limited to a rate of 40 Kbytes per second (with an allowed burst of 50 Kbytes). Note that the rate is specified in bytes, not bits, per second; a rate of 40000 bytes per second, nominally corresponds to 320 kbps.
- Rule #20: traffic from the subnet 10.31.4.xxx destined to ports 4000 to 9000 is always blocked, regardless of protocol.
- Rule #22: traffic from the subnet 10.4.xxx.yyy destined to ports 4000 to 9000 is always blocked, regardless of protocol.
- All other traffic is allowed.

More complex rules may be defined, relying on the 'single-match' process described above. Below is an advanced example of an access list definition via *ini* file:

```
[ACCESSLIST]
FORMAT AccessList Index = AccessList Source IP,
AccessList_Net_Mask, AccessList_Start_Port, AccessList_End_Port,
AccessList_Protocol, AccessList_Packet_Size, AccessList_Byte_Rate,
AccessList_Byte_Burst, AccessList_Allow_Type;
AccessList 10 = 10.0.0.0, 255.0.0.0, 0, 65535, any, 0, 40000,
50000, allow;
AccessList 15 = 10.31.4.0, 255.255.255.0, 4000, 9000, any, 0, 0,
0, allow;
AccessList 20 = 0.0.0.0, 0.0.0.0, 0, 65535, any, 0, 0, 0, block;
[\\ACCESSLIST]
```

This access list (in the example above) consists of three rules:

- Rule #10: traffic from the subnet 10.xxx.yyy.zzz is allowed if the traffic rate does not exceed 40 KB/s.
- Rule #15: if a packet didn't match rule #10, that is, the excess traffic is over 40 KB/s, and coming from the subnet 10.31.4.xxx to ports 4000 to 9000, then it is allowed.
- Rule #20: all other traffic (which didn't match the previous rules), is blocked.

## 6.7 Network Port Usage

The following table lists the default TCP/UDP network port numbers used by the device. Where relevant, the table lists the *ini* file parameters that control the port usage and provide source IP address filtering capabilities.

**Table 6-1: Default TCP/UDP Network Port Numbers**

Port Number	Peer Port	Application	Notes
2	2	Debugging interface	Always ignored
23	-	Telnet	Disabled by default (TelnetServerEnable). Configurable (TelnetServerPort), access controlled by WebAccessList
68	67	DHCP	Active only if DHCPEnable = 1
80	-	Web server (HTTP)	Configurable (HTTPPort), can be disabled (DisableWebTask or HTTPSONly). Access controlled by WebAccessList
161	-	SNMP GET/SET	Configurable (SNMPPort), can be disabled (DisableSNMP). Access controlled by SNMPTrustedMGR
443	-	Web server (HTTPS)	Configurable (HTTPSPort), can be disabled (DisableWebTask). Access controlled by WebAccessList
500	-	IPSec IKE	Can be disabled (EnableIPSec)
6000, 6010 and up	-	RTP traffic	Base port number configurable (BaseUDPPort), fixed increments of 10. The number of ports used depends on the channel capacity of the device.
6001, 6011 and up	-	RTCP traffic	Always adjacent to the RTP port number
6002, 6012 and up	-	T.38 traffic	Always adjacent to the RTCP port number
5060	5060	SIP	Configurable (LocalSIPPort [UDP], TCPLocalSIPPort [TCP]).
5061	5061	SIP over TLS (SIPS)	Configurable (TLSLocalSIPPort)
(random) > 32767	514	Syslog	Configurable (SyslogServerPort). Disabled by default (EnableSyslog).
(random) > 32767	-	Syslog ICMP	Disabled by default (EnableSyslog).
(random) > 32767	-	ARP listener	
(random) > 32767	162	SNMP Traps	Can be disabled (DisableSNMP)
(random) > 32767	-	DNS client	



## 6.8 Recommended Practices

To improve network security, the following guidelines are recommended when configuring the device:

- Define the password of the primary Web user account (refer to 'Configuring the Web User Accounts' in the device's *User's Manual*) to a unique, hard-to-hack string. Do not use the same password for several devices as a single compromise may lead to others. Keep this password safe at all times and change it frequently.
- If possible, use a RADIUS server for authentication. RADIUS allows you to set different passwords for different users of the device, with centralized management of the password database. Both Web and Telnet interfaces support RADIUS authentication (refer to "RADIUS Login Authentication" on page 110). (**Note:** RADIUS is not applicable to the 3000 Series.)
- If the number of users that access the Web and Telnet interfaces is limited, you can use the 'Web and Telnet Access List' to define up to ten IP addresses that are permitted to access these interfaces. Access from an undefined IP address is denied (refer to 'Configuring the Web and Telnet Access List' in the device's *User's Manual*).
- Use IPSec to secure traffic to all management and control hosts. Since IPSec encrypts all traffic, hackers cannot capture sensitive data transmitted on the network, and malicious intrusions are severely limited.
- Use HTTPS when accessing the Web interface. Set HTTPSEnabled to 1 to allow only HTTPS traffic (and block port 80). If you don't need the Web interface, disable the Web server (DisableWebTask).
- If you use Telnet, do not use the default port (23). Use SSL mode to protect Telnet traffic from network sniffing.
- If you use SNMP, do not leave the community strings at their default values as they can be easily guessed by hackers (refer to "SNMP Community Names" on page 53).
- Use a firewall to protect your VoIP network from external attacks. Network robustness may be compromised if the network is exposed to Denial of Service (DoS) attacks. DoS attacks are mitigated by Stateful firewalls. Do not allow unauthorized traffic to reach the device.

## 6.9 Legal Notice

By default, the device supports export-grade (40-bit and 56-bit) encryption due to US government restrictions on the export of security technologies. To enable 128-bit and 256-bit encryption on your device, contact your AudioCodes representative.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

This product includes cryptographic software written by Eric Young' (eay@cryptsoft.com).

## **Reader's Notes**

## 7 RTP Control Protocol Extended Reports (RTCP-XR)



**Note:** This section is applicable only to AudioCodes' 2000 Series and Mediant 1000 devices.

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 diagnosing problems. RTCP-XR (RFC 3611) extends the RTCP reports defined in RFC 3550 by providing additional VoIP metrics.

RTCP-XR information publishing is implemented in the device according to <draft-johnston-sipping-rtcp-summary-07>. This draft defines how a SIP User Agent (UA) publishes the detailed information to a defined collector.

RTCP-XR messages containing key call-quality-related metrics are exchanged periodically (user-defined) between the device and the SIP UA. This allows an analyzer to monitor these metrics midstream, or a device to retrieve them using SNMP. The device can send RTCP-XR reports to an Event State Compositor (ESC) server using PUBLISH messages. These reports can be sent at the end of each call (configured using RTCPXRReportMode) and according to a user-defined interval (RTCPInterval or DisableRTCPRandomize) between consecutive reports.

To enable RTCP-XR reporting, the VQMonEnable *ini* file parameter must be set to 1. For a detailed description of the RTCP-XR *ini* file parameters, refer to Channel Parameters in the device's *User's Manual*.

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 the metrics listed in the table below.

**Table 7-1: RTCP-XR Published VoIP Metrics**

	Metric Name
<b>General</b>	Start Timestamp
	Stop Timestamp
	Call-ID
	Local Address (IP, Port & SSRC)
	Remote Address (IP, Port & SSRC)
<b>Session Description</b>	Payload Type
	Payload Description
	Sample Rate
	Frame Duration
	Frame Octets
	Frames per Packets
	Packet Loss Concealment
	Silence Suppression State

	Metric Name
<b>Jitter Buffer</b>	Jitter Buffer Adaptive
	Jitter Buffer Rate
	Jitter Buffer Nominal
	Jitter Buffer Max
	Jitter Buffer Abs Max
<b>Packet Loss</b>	Network Packet Loss Rate
	Jitter Buffer Discard Rate
<b>Burst Gap Loss</b>	Burst Loss Density
	Burst Duration
	Gap Loss Density
	Gap Duration
	Minimum Gap Threshold
<b>Delay</b>	Round Trip Delay
	End System Delay
	One Way Delay
	Interarrival Jitter
	Min Absolute Jitter
	Signal
	Signal Level
	Noise Level
	Residual Echo Return Noise
<b>Quality Estimates</b>	Listening Quality R
	RLQ Est. Algorithm
	Conversational Quality R
	RCQ Est. Algorithm
	External R In
	Ext. R In Est. Algorithm
	External R Out
	Ext. R Out Est. Algorithm
	MOS-LQ
	MOS-LQ Est. Algorithm
	MOS-CQ
	MOS-CQ Est. Algorithm
	QoE Est. Algorithm

## 8 RTP / RTCP Payload Types and Port Allocation

RTP Payload Types are defined in RFC 3550 and RFC 3551. We have added new payload types to enable advanced use of other coder types. These types are reportedly not used by other applications.

### 8.1 Payload Types Defined in RFC 3551

Table 8-1: Packet Types Defined in RFC 3551

Payload Type	Description	Basic Packet Rate (msec)
0	G.711 $\mu$ -Law	10,20
2	G.726-32	10, 20
3	GSM-FR <b>Note:</b> Only applicable to 2000 Series and 3000 Series.	20
4	G.723 (6.3/5.3 kbps)	30
8	G.711 A-Law	10,20
9	G.722 <b>Note:</b> Only applicable to 3000 Series.	20
12	QCELP <b>Note:</b> Only applicable to 2000 Series.	20
18	G.729A/B	20
200	RTCP Sender Report	Randomly, approximately every 5 seconds (when packets are sent by channel)
201	RTCP Receiver Report	Randomly, approximately every 5 seconds (when channel is only receiving)
202	RTCP SDES packet	
203	RTCP BYE packet	
204	RTCP APP packet	

## 8.2 Defined Payload Types

The defined payload types are listed in the table below.



**Note:** Not all coders are supported on all devices. For a detailed description of supported coders, please refer to the specific device's *Release Notes*.

**Table 8-2: Defined Payload Types**

Payload Type	Description	Basic Packet Rate (msec)
3	MS-GSM	40
3	GSM-EFR	20
22	G.726-24	20
23	G.726-16	20
38	G.726-40	20
56	Transparent PCM	20
60	EVRC	20
64	AMR	20
64	AMR-WB	20
65	iLBC	20, 30
68	EVRC-B (4GV)	20
96	DTMF relay per RFC 2833	
102	Fax Bypass	20
103	Modem Bypass	20
104	RFC 2198 (Redundancy)	Same as channel's voice coder.
105	NSE Bypass	

## 8.3 Default RTP / RTCP / T.38 Port Allocation

The local default local User Datagram Protocol (UDP) ports for Audio, Video, and Fax media streams are calculated using the formula below:

$\text{BaseUDPPort} + \text{Channel ID} * 10 + \text{Port Offset}$

The BaseUDPPort is a configurable parameter, which by default is 4000. The port offsets are listed in the table below.

**Table 8-3: Local UDP Port Offsets**

Port Type	Port Offset
Audio RTP	0
Audio RTCP	1
Fax T.38	2
Video RTP	4
Video RTCP	5

For example, the T.38 local UDP port for channel 30 is calculated as follows (using the default value for BaseUDPPort, i.e., 4000):

$4000 + 30 * 10 + 2 = 4302.$



**Notes:**

- For a description of the *ini* file parameter BaseUDPPort, refer to the device's *User's Manual*.
- Video RTP and Video RTCP are applicable only to IPmedia 3000/IPM-8410.
- To configure the device to use the same port for both RTP and T.38 packets, set the parameter T38UseRTPPort to 1.
- For the 2000 Series, the number of channels depends on the configuration (i.e., device with one or two TP-1610 / IPM-1610 blades).

## Reader's Notes



## 9 CAS Protocol Table



**Note:** This section is applicable only to AudioCodes' Digital devices.

### 9.1 Constructing CAS Protocol Tables for CAS-Terminated Protocols

The protocol table file is a text file containing the protocol's state machine that defines the entire protocol process. It is constructed of States, predefined Actions/Events, and predefined functions. With this file, you have full control over CAS protocol and can define or modify any CAS protocol by writing the protocol state machine in a text file according to a few AudioCodes-defined rules.

➤ **To generate the protocol file, take these 5 steps:**

1. Learn the protocol text file rules from which the CAS state machine is built.
2. Refer to the supplied CAS files for an example.
3. Build the specific protocol/script text file (for example, xxx.txt) file and its related numerical value h file (for example, UserProt\_defines\_xxx.h). Note that the xxx.txt file must include the following 'C include' (for example, #include 'UserProt\_defines\_xxx.h').
4. Compile the xxx.txt with the 'TrunkPack Downloadable Conversion Utility' to produce the xxx.dat file. Note that the files xxx.txt, CASSetup.h, cpp.exe and UserProt\_defines\_xxx.h must be located in the same folder (you should choose Dynamic Format at the list).
5. Download the xxx.dat file to the board using the function acOpenBoard() in the initialization phase.

### 9.2 Protocol Table Elements

The *CASSetup.h* file includes all the predefined definitions necessary to build a new protocol text file or to modify an existing one.

The CAS protocol table file (xxx.txt) is composed of the following elements:

- INIT Variables
- Actions
- Functions
- States

#### 9.2.1 INIT Variables

INIT variables are numeric values defined by users in UserProt\_defines\_xxx.h. These values can be used in the file xxx.txt.

For example, INIT\_RC\_IDLE\_CAS defines the ABCD bits expected to be received in IDLE state. INIT\_DTMF\_DIAL defines the On-time and Off-time for the DTMF digits generated towards the PSTN. Refer to the detailed list in UserProt\_defines\_xxx.h and in the sample protocol text file (AudioCodes-supplied CAS files). Refer to the following ST\_INIT detailed explanation.

## 9.2.2 Actions

Actions (i.e., protocol table events) are protocol table events activated either by the DSP (e.g., EV\_CAS\_01) or by users (e.g., EV\_PLACE\_CALL, EV\_TIMER\_EXPIRED1). The full list of available predefined events is located in the file CASSetup.h.

## 9.2.3 Functions

Functions define a certain procedure that can be activated in any state or in the transition from one state to another. The available functions include, for example, SET\_TIMER (timer number, timeout in milliseconds), SEND\_CAS (AB value, CD value). A full list of the possible predefined functions can be found in the file CASSetup.h.

## 9.2.4 States

Each Protocol Table consists of several states that it switches between during the call setup and tear-down process. Every state definition begins with the prefix 'ST\_' followed by the state name and colon. The body of the state is composed of up to four unconditional performed functions and a list of actions that may trigger this state.

Below shows an example taken from an E&M wink start table protocol file:

**Table 9-1: ST\_DIAL: Table Elements**

Action	Function	Parameter		Next State
		#1	#2	
FUNCTION0	SET_TIMER	2	Extra Delay Before Dial	DO
EV_TIMER_EXPIRED2	SEND_DEST_NUM	ADDRESS	None	NO_STATE
EV_DIAL_ENDED	SET_TIMER	4	No Answer Time	ST_DIAL_ENDED

When the state machine reaches the dial state, it sets timer number 2 and then waits for one of two possible actions to be triggered: Either timer 2 expiration or end of dial event. When timer 2 expires, the protocol table executes function SEND\_DEST\_NUM and remains in the same state (NEXT\_STATE=NO\_STATE). When the dial event ends, the protocol table sets timer 4 and moves to ST\_DIAL\_ENDED written in the field NEXT\_STATE.

Although you can define your own states, there are two states defined in the file CASSetup.h that must appear in every protocol table created:

- **ST\_INIT:** When channels initialization is selected, the table goes into 'Init' state. This state contains functions that initialize the following global parameters:
  - **INIT\_RC\_IDLE\_CAS:** Defines the ABCD bits expected to be received in the IDLE state in the specific protocol. The third parameter used to enable detection of 4 bits` CAS value (see below).

- **INIT\_TX\_IDLE\_CAS:** Defines the ABCD bits transmitted in IDLE state in the specific protocol.
- **INIT\_DIAL\_PLAN:** A change regarding the issue of an incoming call dialed number. In version 4.2 and earlier, users were required to predefine the expected number of digits to receive an incoming call. If a lower number of digits than expected was received, the call setup would have failed.
- **ST\_IDLE:** When no active call is established or is in the process of being established, the table resides in Idle state, allowing it to start the process of incoming or outgoing calls. When the call is cleared, the state machine table returns to its Idle state.

In Versions 4.2 and later, process the incoming call detection event by declaring end of digit reception in the following ways (both for ADDRESS/destination number and ANI/source number):

- Receiving '#' digit (in MF or DTMF).
- The number of digits collected reaches its maximum value as defined in DIAL\_PLAN parameter #1 and #2 for destination and ANI numbers respectively.
- A predefined time-out value defined in DIAL\_PLAN parameter #3 elapses.
- In MFC-R2 reception of signal I-15 (depending on the variant).

Parameter	Description
<b>INIT_DTMF_DIAL</b>	Defines the On-time and Off-time for the DTMF digits generated towards the PSTN.
<b>INIT_COMMA_PAUSE_TIME</b>	Defines the delay between each digit when a comma is used as part of the dialed number string (refer to acPSTNPlaceCall for details).
<b>INIT_DTMF_DETECTION</b>	Defines the minimum/maximum On-time for DTMF digit dialing detection.
<b>INIT_PULSE_DIAL_TIME</b>	Not supported by the current stack version. Defines the Break and Make time for pulse dialing.
<b>INIT_PULSE_DIAL</b>	Not supported by the current stack version. Defines the Break and Make ABCD bits for pulse dialing.
<b>INIT_DEBOUNCE</b>	Defines the interval time of CAS to be considered (a stable one).
<b>INIT_COLLECT_ANI</b>	Enables or Disables reception of ANI in a specific protocol.
<b>INIT_DIGIT_TYPE</b>	<p>The #1 parameter defines the dialing method used (DTMF, MF). With MFC-R2 protocols, this parameter is not applicable (digits are assumed to be R2 digits).</p> <p>The #2 parameter enabled to usage of SS5 tones (not used).</p> <p>The #3 parameter used to enable digits detection at the OutGoing side of the call (which needed at some protocols).</p>
<b>INIT_NUM_OF_EVENT_IN_STATE</b>	Inserted for detection on TOTAL_NUMBER_OF_EVENTS_IN_STATE (CASSetup.h).

Parameter	Description
INIT_INIT_GLOBAL_TIMERS	Initiates specific timers; it is used with Parameter#1 for metering pulse timer duration.
INIT_PULSE_DIAL_ADDITIONAL_PARAMS	Not used.
INIT_RINGING_TO_ANALOGUE	When using analogue gateway option, it defines the CAS value of ringing (#1) CAS value of silence (#2) and CAS value of polarity reversal(#3).
INIT_DIGIT_TYPE_1	Defines the signaling system used to send operator service.
INIT_REJECT_COLLECT	Defines the method for reject collect calls: <i>disabled</i> , <i>using Line signaling</i> , or <i>using register signaling</i> .
INIT_VERSION	Defines the version number. The version number is relevant to the release version number and is a text information string (not related to the utility compilation version number).
INIT_SIZE_OF_TABLE_PARAM	Users must insert the definition of TOTAL_NUMBER_OF_EVENTS_IN_STATE from CASSetup.h.

## 9.3 Reserved Words

For reserved words such as DO, NO\_STATE, etc., refer to the detailed list in CASSetup.h.

## 9.4 State Line Structure

Each text line in the body of each state comprises 6 columns:

1. Action/event
2. Function
3. Parameter #1
4. Parameter #2
5. Additional parameters
6. Next state

## 9.5 Action / Event

Action / event is the name of the table's events that are the possible triggers for the entire protocol state machine. These can be selected from the list of events in file CASSetup.h (e.g., EV\_DISCONNECT\_INCOMING).

At the beginning of the state, there can be up to four unconditional actions / events called FUNCTION. These events are functions that are unconditionally performed when the table reaches the state. These actions are labeled FUNCTION0 to FUNCTION3.

The following subsections provide a list of available protocols table actions (events to the state machine).

## 9.5.1 User Command Oriented Action / Event

Table 9-2: User Command Orientated Action / Event

User Command Oriented Action/Event	Description
EV_PLACE_CALL	When acpstnplacecall() is used.
EV_SEIZE_LINE	Used by Megaco control protocol.
EV_SEND_SEIZE_ACK	Used by Megaco control protocol.
EV_ANSWER	When acpstnanswercall() is used.
EV_MAKE_DOUBLE_ANSWER_CAS	When the function acpstnanswercall is used and the INIT_REJECT_COLLECT parameter is set to Line Signaling.
EV_MAKE_DOUBLE_ANSWER_MF	When the function acpstnanswercall is used and the INIT_REJECT_COLLECT parameter is set to Register Signaling.
EV_DISCONNECT	When function acpstndisconnectcall() is used and the call is outgoing.
EV_DISCONNECT_INCOMING	When function acpstndisconnectcall() is used and the call is incoming.
EV_RELEASE_CALL	When acpstnreleasecall() is used.
EV_FORCED_RELEASE	When accasforcedrelease () is used.
EV_USER_BLOCK_COMND	When accasblockchannel() is used. This event is used to block or unblock the channel.
EV_MAKE_METERING_PULSE	When the function accasmeteringpulse is used, it triggers the start of the metering pulse while using function set_pulse_timer to start the timer to get the off event (refer to event ev_metering_timer_pulse_off).
EV_METERING_TIMER_PULSE_OFF	An event sent after the timer (invoked by function set_pulse_timer) expires. Refer to ev_make_metering_pulse.
EV_MAKE_FLASH_HOOK	When accasflashhook is used, a flash hook is triggered.

## 9.5.2 CAS Change Oriented Events

Table 9-3: CAS Change Orientated Events

Event	Description
EV_CAS_1_1	A new cas a, b bits received (a=1, b=1, was stable for the bouncing period).
EV_CAS_1_0	A new cas a, b bits received (a=1, b=0, was stable for the bouncing period).
EV_CAS_0_1	A new cas a, b bits received (a=0, b=1, was stable for the bouncing period).

Event	Description
EV_CAS_0_0	A new cas a, b bits received (a=0, b=0, was stable for the bouncing period).
EV_CAS_1_1_1_1	A new cas a, b bits received (a=1, b=1, c=1, d=1 was stable for the bouncing period). To receive such detection (that is different from EV_CAS_1_1) you must set YES at the #3 parameter of INIT_RC_IDLE_CAS.

### 9.5.3 Timer Oriented Events

**Table 9-4: Time-Orientated Events**

Event	Description
EV_TIMER_EXPIRED1	Timer 1 that was previously set by the table has expired.
EV_TIMER_EXPIRED2	Timer 2 that was previously set by the table has expired.
EV_TIMER_EXPIRED3	Timer 3 that was previously set by the table has expired.
EV_TIMER_EXPIRED4	Timer 4 that was previously set by the table has expired.
EV_TIMER_EXPIRED5	Timer 5 that was previously set by the table has expired.
EV_TIMER_EXPIRED6	Timer 6 that was previously set by the table has expired.
EV_TIMER_EXPIRED7	Timer 7 that was previously set by the table has expired.
EV_TIMER_EXPIRED8	Timer 8 that was previously set by the table has expired.

### 9.5.4 Counter Oriented Events

**Table 9-5: Counter Orientated Events**

Event	Description
EV_COUNTER1_EXPIRED	The value of counter 1 reached 0.
EV_COUNTER2_EXPIRED	The value of counter 2 reached 0.

### 9.5.5 IBS Oriented Events

**Table 9-6: IBS Orientated Events**

Event	Explanation
EV_RB_TONE_STARTED	Ringback tone as defined in the Call Progress Tone <i>ini</i> file (type and index) is detected.
EV_RB_TONE_STOPPED	Ringback tone as defined in the Call Progress Tone <i>ini</i> file (type and index) is stopped after it was previously detected.
EV_BUSY_TONE	Not used.

Event	Explanation
EV_BUSY_TONE_STOPPED	Not used.
EV_FAST_BUSY_TONE	Not used.
EV_FAST_BUSY_TONE_STOPPED	Not used.
EV_ANI_REQ_TONE_DETECTED	R1.5 ANI-request tone as defined in the Call Progress Tone <i>ini</i> file (type and index) is detected.
EV_R15_ANI_DETECTED	R1.5 ANI digit-string was detected.
EV_DIAL_TONE_DETECTED	Dial tone as defined in the Call Progress Tone <i>ini</i> file (type and index) is detected.
EV_DIAL_TONE_STOPPED	Dial tone as defined in the Call Progress Tone <i>ini</i> file (type and index) is stopped after it was previously detected.

### 9.5.6 DTMF/MF Oriented Events

Table 9-7: DTMF / MF Orientated Events

Event	Explanation
EV_MFRn_0	MF digit 0 is detected (only DTMF & MFr1).
EV_MFRn_1	MF digit 1 is detected.
EV_MFRn_2	MF digit 2 is detected.
EV_MFRn_3	MF digit 3 is detected.
EV_MFRn_4	MF digit 4 is detected.
EV_MFRn_5	MF digit 5 is detected.
EV_MFRn_6	MF digit 6 is detected.
EV_MFRn_7	MF digit 7 is detected.
EV_MFRn_8	MF digit 8 is detected.
EV_MFRn_9	MF digit 9 is detected.
EV_MFRn_10	MF digit 10 is detected.
EV_MFRn_11	MF digit 11 is detected.
EV_MFRn_12	MF digit 12 is detected.
EV_MFRn_13	MF digit 13 is detected.
EV_MFRn_14	MF digit 14 is detected.
EV_MFRn_15	MF digit 15 is detected.
EV_MFRn_1_STOPPED	MF digit 1 previously detected is now stopped.
EV_MFRn_2_STOPPED	MF digit 2 previously detected is now stopped.
EV_MFRn_3_STOPPED	MF digit 3 previously detected is now stopped.
EV_MFRn_4_STOPPED	MF digit 4 previously detected is now stopped.
EV_MFRn_5_STOPPED	MF digit 5 previously detected is now stopped.

Event	Explanation
EV_MFRn_6_STOPPED	MF digit 6 previously detected is now stopped.
EV_MFRn_7_STOPPED	MF digit 7 previously detected is now stopped.
EV_MFRn_8_STOPPED	MF digit 8 previously detected is now stopped.
EV_MFRn_9_STOPPED	MF digit 9 previously detected is now stopped.
EV_MFRn_10_STOPPED	MF digit 10 previously detected is now stopped.
EV_MFRn_11_STOPPED	MF digit 11 previously detected is now stopped.
EV_MFRn_12_STOPPED	MF digit 12 previously detected is now stopped.
EV_MFRn_13_STOPPED	MF digit 13 previously detected is now stopped.
EV_MFRn_14_STOPPED	MF digit 14 previously detected is now stopped.
EV_MFRn_15_STOPPED	MF digit 15, previously detected is now stopped.
EV_END_OF_MF_DIGIT	When DialMF() is used and no more dialed number digits are available (they already were sent). For example, the far side requests the next ANI digit but all digits already have been sent. This event usually appears in MFC-R2 tables.
EV_FIRST_DIGIT	The first digit of the DNI / ANI number is detected.
EV_DIGIT_IN	An incoming digit (MFR1 or DTMF) is detected.
EV_WRONG_MF_LENGTH	An incoming digit was detected, but its duration (ON-TIME) is too long or too short.
EV_DIALED_NUM_DETECTED	The whole destination number is detected.
EV_ANI_NUM_DETECTED	The whole source number is detected.
EV_DIAL_ENDED	The dialing process finished and all digits dialed.
EV_NO_ANI	When DialMF() is used and no ANI is specified by the outgoing user in function acPSTNPlaceCall(). MFC



**Note:** MF digit includes MF R1, R2-FWD, or R2-BWD, according to the context, protocol type, and call direction.

The following actions / events cause the MFC-R2 table to send the correct MF tone to the backward direction:

**Table 9-8: Actions / Events Causing MFC-R2 Table to Send Correct MF Tone to Backward Direction**

Actions/Events	Explanation
EV_ACCEPT	When acCASAacceptCall is used (only in MFC-R2) with CALLED_IDLE as its reason parameter (for example, this sends MF backward B-6).
EV_ACCEPT_SPARE_MF1	When acCASAacceptCall is used with SPARE_MF1 as its reason parameter.
EV_ACCEPT_SPARE_MF9	When acCASAacceptCall is used with SPARE_MF9 as its reason parameter.



Actions/Events	Explanation
<b>EV_ACCEPT_SPARE_MF10</b>	When acCASAcceptCall is used with SPARE_MF10 as its reason parameter.
<b>EV_ACCEPT_SPARE_MF11</b>	When acCASAcceptCall is used with SPARE_MF11 as its reason parameter.
<b>EV_ACCEPT_SPARE_MF12</b>	When acCASAcceptCall is used with SPARE_MF12 as its reason parameter.
<b>EV_ACCEPT_SPARE_MF13</b>	When acCASAcceptCall is used with SPARE_MF13 as its reason parameter.
<b>EV_ACCEPT_SPARE_MF14</b>	When acCASAcceptCall is used with SPARE_MF14 as its reason parameter.
<b>EV_ACCEPT_SPARE_MF15</b>	When acCASAcceptCall is used with SPARE_MF 15 as its reason parameter.
<b>EV_REJECT_BUSY</b>	When acCASAcceptCall is used with CALLED_BUSY as its reason parameter.
<b>EV_REJECT_CONGESTION</b>	When acCASAcceptCall is used with CALLED_CONGESTION as its reason parameter.
<b>EV_REJECT_UNALLOCATED</b>	When acCASAcceptCall is used with CALLED_UNALLOCATED as its reason parameter.
<b>EV_REJECT_SIT</b>	When acCASAcceptCall is used with SIT as its reason parameter.
<b>EV_REJECT_RESERVE1</b>	When acCASAcceptCall is used with CALLED_RESERVE1 as its reason parameter.
<b>EV_REJECT_RESERVE2</b>	When acCASAcceptCall is used with CALLED_RESERVE2 as its reason parameter.

### 9.5.7 Operator Service Events (up to GR-506)

Table 9-9: Operator Service Events (Up to GR-506)

Event	Explanation
<b>EV_SEND_LINE_OPERATOR_SERVICE1</b>	Send operator service 1 (=Operator Released) using line signaling.
<b>EV_SEND_LINE_OPERATOR_SERVICE2</b>	Send operator service 2 (=Operator Attached) using line signaling.
<b>EV_SEND_LINE_OPERATOR_SERVICE3</b>	Send operator service 3 (=Coin Collect) using line signaling.
<b>EV_SEND_LINE_OPERATOR_SERVICE4</b>	Send operator service 4 (=Coin Return) using line signaling.
<b>EV_SEND_LINE_OPERATOR_SERVICE5</b>	Send operator service 5 (=Ring-back) using line signaling.
<b>EV_SEND_REGISTER_OPERATOR_SERVICE1</b>	Send operator service 1 (=Operator Released) using register signaling.

Event	Explanation
EV_SEND_REGISTER_OPERATOR_SERVICE2	Send operator service 2 (=Operator Attached) using register signaling.
EV_SEND_REGISTER_OPERATOR_SERVICE3	Send operator service 3 (=Coin Collect) using register signaling.
EV_SEND_REGISTER_OPERATOR_SERVICE4	Send operator service 4 (=Coin Return) using register signaling.
EV_SEND_REGISTER_OPERATOR_SERVICE5	Send operator service 5 (=Ring-back) using register signaling.
EV_SEND_REGISTER_OPERATOR_SERVICE6	Send operator service 6 (=Coin Collect/Operator Released) using register signaling.



**Note:** The following actions/events are for internal use only:

- EV\_INIT\_CHANNEL
- EV\_TO\_USER
- EV\_CLOSE\_CHANNEL
- EV\_OPEN\_CHANNEL
- EV\_FAIL\_DIAL
- EV\_FAIL\_SEND\_CAS
- EV\_ALARM

## 9.6 Function

The function's column holds the name of the function to be activated when the action specified in the action / events field occurs. Select the functions from the list of eight functions defined in CasSetup.h (e.g., START\_COLLECT). When NONE is specified in this column, no function is executed.



**Note:** Do not define the same timer number (by SET\_TIMER) twice before the first one expires or is deleted.

## 9.7 Parameters

The following columns are used as the function's parameters:

- **Parameter #1**
- **Parameter #2**

The list of global parameters can be found in CasSetup.h. If a parameter is not essential, it can also be written as NONE.



**Note:** In previous versions, you must include three parameters per function. From Release 5.2 and on, to enable the dynamic format of the CAS file and reduce memory usage, you can only include the used parameters.

Table 9-10: Available User Functions and Corresponding Parameters

User Function	User Function Parameters and Descriptions
<b>SET_TIMER</b>	(Timer number, timeout). Sets the timers managed per B-channel. Their expiration triggers the state machine table. Each protocol table/state machine can use up to 8 timers per B-channel/call (timeout in msec) when the timers have 25 msec resolution.
<b>SEND_CAS</b>	(AB value, CD value). ABCD bits are sent as line signaling for the specific channel when the call is setup.
<b>GENERATE_CAS_EV</b>	Check the ABCD bits value, and send a proper event to the state machine.
<b>SEND_EVENT</b>	(Event type, cause). The specific event type is sent to the host/user and retrieved by applying the function acGetEvent().
<b>SEND_DEST_NUM</b>	En-bloc dialing: refers to the digits string located in function acPSTNPlaceCall. Three types are available: (1) DestPhoneNum (2) InterExchangePrefixNum (3) SourcePhoneNum.
<b>DEL_TIMER</b>	(Timer number). Deletes a specific timer or all the timers (0 represents all the timers) for the B-channel.
<b>START_COLLECT</b>	Initiates the collection of address information, i.e., the dialed (destination) number for incoming calls where appropriate, according to the protocol. In the time between START_COLLECT and STOP_COLLECT, no digit is reported to users (EV_DIGIT is blocked) and the destination number is reported in event EV_INCOMING_CALL_DETECTED.
<b>STOP_COLLECT</b>	Refer to START_COLLECT.
<b>SET_COUNTER</b>	(Counter number, counter value or NONE). Sets counters managed per B-channel. Their expiration triggers the state machine. The counter initialization value should be a non-negative number. To delete all timers, invoke this function with 0 in the counter number field.
<b>DEC_COUNTER</b>	(Counter number). Decreases the counter value by 1. When the counter value reaches 0, EV_COUNTERx_EXPIRES is sent to the table (where x represents the counter number).
<b>RESTRICT_ANI</b>	Indicate the incoming side to hide the ANI from the Far-end user.
<b>SEND_MF</b>	(MF type, MF digit or index or NONE, MF sending time). This function is used only with MFC-R2 protocols.

The Channel Parameter structure contains three parameters associated with sending digits:

Table 9-11: Parameters Associated with Sending Digits

Parameter	Description
<b>AddressVector and ANIDigitVector</b>	<p>These parameters are initialized when function PlaceCall is used. When the code reaches the dialing section, it sends the MF digit according to the MF type specified in the MF type cell (the types are defined in file CASSetup.h):</p> <ul style="list-style-type: none"> <li>▪ <b>ADDRESS:</b> Sends the digit from the address vector (destination number) according to the index requested. Refer to the Index definition.</li> <li>▪ <b>ANI:</b> Sends the digit from the ANI vector (source number) according to the requested index.</li> <li>▪ <b>SPECIFIC:</b> Sends the MF digit specified in the cell Parameter #2.</li> <li>▪ <b>SOURCE_CATEGORY:</b> Sends the predefined source category MF digit.</li> </ul>

Parameter	Description
	<p>The source category digit is set as the parameter SourceNumberingType when function PlaceCall is used. The second and third parameters are ignored when this type is used.</p> <ul style="list-style-type: none"> <li>▪ <b>TRANSFER_CAPABILITY:</b> Sends the predefined line category MF digit. The line category digit is set as the parameter TransferCapability when function PlaceCall is used. The second and third parameters are ignored when this type is used.</li> </ul>
<b>Index</b>	<p>Specifies the Offset of the next digit to be sent from the vector (ADDRESS or ANI types, described above):</p> <ul style="list-style-type: none"> <li>▪ <b>Index 1:</b> Sends the next digit in the vector.</li> <li>▪ <b>Index -n:</b> Sends the last n digit. Underflow can occur if n is greater than the number of digits sent so far.</li> <li>▪ <b>Index 0:</b> Sends the last sent digit.</li> <li>▪ <b>Index SEND_FIRST_DIGIT:</b> Starts sending the digits vector from the beginning (refer to CASSetup.h).</li> </ul>
<b>MF Send Time</b>	<p>This send time parameter specifies the maximum transmission time of the MF.</p> <ul style="list-style-type: none"> <li>▪ <b>STOP_SEND_MF:</b> Stops sending the current MF.</li> <li>▪ <b>SEND_PROG_TON:</b> Operation, Tone or NONE.</li> </ul>

Two operations are available:

- Sends the Call Progress Tone specified in the cell Parameter #2 (The second parameter can be taken from CASsetup.h)
- Stops sending the last parameter

Parameter	Description
<b>CHANGE_COLLECT_TYPE</b>	<p>(Collect Type). Used by the incoming user to indicate that waiting for receipt of the digit of the requested type. The type can be one of the following:</p> <ul style="list-style-type: none"> <li>▪ <b>ADDRESS:</b> The user waits for receipt of address digits.</li> <li>▪ <b>ANI:</b> The user waits for receipt of ANI digits.</li> <li>▪ <b>SOURCE_CATEGORY:</b> The user waits for receipt of the source category.</li> <li>▪ <b>TRANSFER_CAPABILITY:</b> The user waits for receipt of the source transfer capability (line category).</li> </ul>

## 9.8 Next State

The Next State column contains the next state the table moves to after executing the function for that action/event line. When you select to stay in the same state, insert NO\_STATE or use the current state.

Note the difference between NO\_STATE and the current state name in this field. If you select to stay in the same current state, the unconditional actions (FUNCTION0) at the beginning of the state are performed. In contrast, NO\_STATE skips these functions and waits for another action to arrive.

Reserved word 'DO' must be written in the next state field if the unconditional actions (FUNCTION0) at the beginning of the state are used.

## 9.9 Changing the Script File

- CAS bouncing is filtered globally for each received CAS for each channel. Define the time for the filtering criteria in the protocol table file (refer to INIT\_DEBOUNCE) and this exceeds the bouncing in the DSP detection of 30 msec.
- ANI/CLI is enabled using parameter ST\_INIT ANI with 'YES'. ANI/CLI is supported using EV\_ANI\_NUM\_DETECTED as the table action for collecting the ANI number in an incoming call. For outgoing calls, the table's function SEND\_DEST\_NUM with ANI parameter I initiates ANI dialing. The ANI number is provided by you in the Source phone number parameter of acPSTNPlaceCall().
- You can use ANSI C pre-compile flags such as #ifdef, #ifndef, #else and #endif in the CAS script file. For example, you can decide whether or not to play dial tone according to fulfillment of #ifdef statement. The definition itself must be in CASSetup.h.

### 9.9.1 MFC-R2 Protocol

- Use the SEND\_MF script function to generate the outgoing call destination number. In this case, the first parameter should be ADDRESS (or ANI for source phone number) and the second parameter -3 to 1 (+1), indicating which digit is sent out of the number that the string conveyed by you in acPSTNPlaceCall().
  - 1 (+1) implies sending of the next digit
  - 0 implies a repeat of the last digit
  - -1 implies the penultimate digitThis parameter actually changes the pointer to the phone number string of digits. Thus, a one-to-one mapping with the MF backward signals of the R2 protocol exists.
- Using parameter SEND\_FIRST\_DIGIT initiates resending the string from the beginning, (change the pointer back to first digit and then proceed as above). This parameter is defined in CASSetup.h.
- When MFC-R2 protocol is used, the two detectors (opened by default) are the Call Progress Tones and MFC-R2 Forward MF. When you invoke an outgoing call via acPSTNPlaceCall(), MFC-R2 Forward MF detector is replaced with MFC-R2 Backward MF detector, since only two detectors per DSP channel are permitted to operate simultaneously.
- The correct MF is automatically generated according to the call direction: Forward for outgoing calls and Backward for incoming calls.
- MFC-R2 protocol fault can cause a channel block. In this case, the script file provided by AudioCodes releases the call to enable the user to free the call resources and be notified as to being in blocking state.
- START\_COLLECT and STOP\_COLLECT must be used in the script file for MF collecting both in outgoing and incoming calls.



**Warning:** If this script function isn't used, the script gets stuck and forward\backward MF are not detected.

- The Ringback Call Progress Tone is translated to a unique event `acEV_PSTN_ALERTING`, since the Ringback tone is actually used in all AudioCodes protocols' state machines. All other Call Progress Tones are conveyed via `acEV_TONE_DETECTED` and retrieved by the user according to their type and index (note that the Ringback tone should be defined in the Call Progress Tones table with the relevant type in order to get this event).
- When the tone detection event is received, users can perform any action. For example, if the event is received with BUSY tone indication, users can invoke `acPSTNDisconnectCall()` to end the call.
- The MFC-R2 destination number is collected using parameter `EXPECTED_NUM_OF_DIGITS_MINUS_1` for `SET_COUNTER` that the user defines with `UserProt_defines_R2_MF.h`. The counter function is used to trigger the script file for the penultimate received. After receiving the last digit, the script file (acting as the outgoing register) initiates the A6/A3 FWD MF. Normally, variant supports end of digit information (MF15 or MF12) or silence at the end of the dialing (when MF15 is not used). A short pulse of MF3 (A3) is sent to indicate that the entire string of digits (according to Q442, 476) is received.
- Sending Group B digit by an incoming register requires invoking `acCASAacceptCall()` with a certain reason parameter. Six reason parameters are available:

Reason Parameter	Description
<b>CALLED_IDLE</b>	Subscriber's line is free. Continue the call sequence. Should usually be followed by accept or reject.
<b>CALLED_BUSY</b>	Subscriber line is busy. Perform disconnect procedures.
<b>CALLED_CONGESTION</b>	Congestion encountered. Perform disconnect procedures.
<b>CALLED_UNALLOCATED</b>	Dial number was not allocated. Perform disconnect procedures.
<b>CALLED_RESERVE1</b>	Reserved for additional group B (user additional requirements).
<b>CALLED_RESERVE2</b>	Reserved for additional group B (user additional requirements).

Each reason generates a specific action, defined by the user, who modifies the script file. The action is then used to generate/respond with a group B MF (free, busy, etc.).

- **Transfer Capability:** This parameter under function `acPSTNPlaceCall()` is used by the outgoing register to generate the service nature of the originating equipment. In most variants (countries), this is the same as the Calling Subscriber Categories, but in some countries it is different, such as in R2 China protocol where it is referred to as the KD (Group II) digit.



**Note:** This parameter only receives MF values from the enumerator `acTISDNTransferCapability`. Choose the MF digit according to the service type that should be sent.

- **Source Category:** This parameter under function `acPSTNPlaceCall()` determines the calling subscriber category. For example, a subscriber with priority, a subscriber without priority, etc. The parameter is usually sent as part of the Group II forward digits (except for R2 China where it is sent as the KA digit using Group I forward digits).



**Note:** This parameter is only applicable only to MFC-R2 protocol type.



## 10 SS7 Tunneling

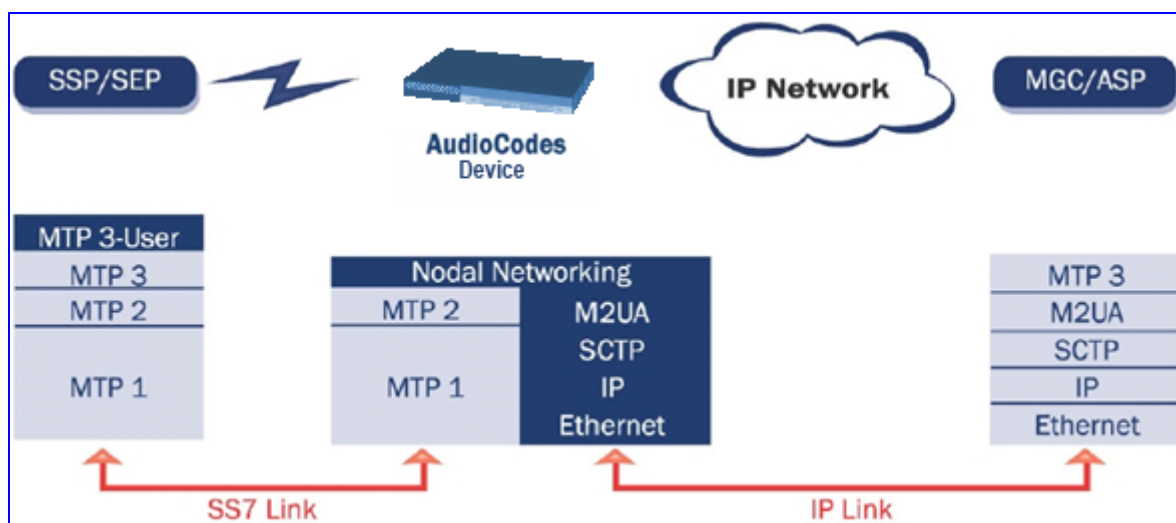


**Note:** This section is applicable only to AudioCodes' 3000 Series (except IPmedia 3000/IPM-8410) and 2000 Series devices.

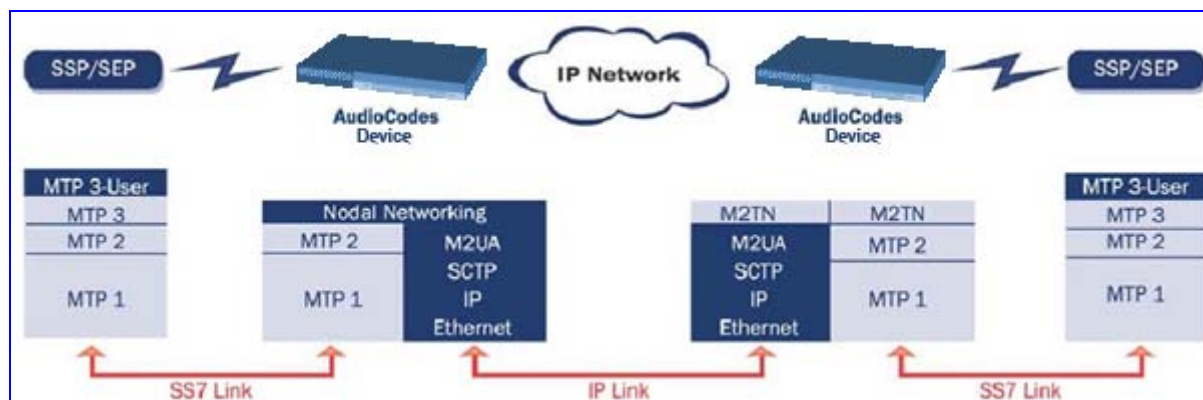
The Signaling System 7 (SS7) tunneling feature facilitates peer-to-peer transport of SS7 links between devices that support AudioCodes' unique MTP2 (Message Transfer Part) Tunneling application (M2TN) for transferring SS7 MTP2 link data over IP. In this scenario, both sides of the link are pure TDM switches and are unaware of the IP tandem that is utilized between them. Using M2TN, the network operator can support SS7 connections over IP, carrying MTP level 3, as well as higher level SS7 layers (e.g., user parts and application protocols, such as TUP (Telephone User Part), Integrated ISUP (Services User Part), SCCP (Signaling Connection Control Part), TCAP (Transaction Capabilities Application Part)).

M2TN uses standard protocols, such as SIGTRAN (RFC 2719 Architectural Framework for Signaling Transport), SCTP (RFC 2960, Stream Control Transmission Protocol), M2UA (RFC 3331, MTP2 User Adaptation Layer), the latter being used for transporting SS7-MTP2 signaling information over IP. M2UA and M2TN architectures are shown in the following figures respectively:

**Figure 10-1: M2UA Architecture**



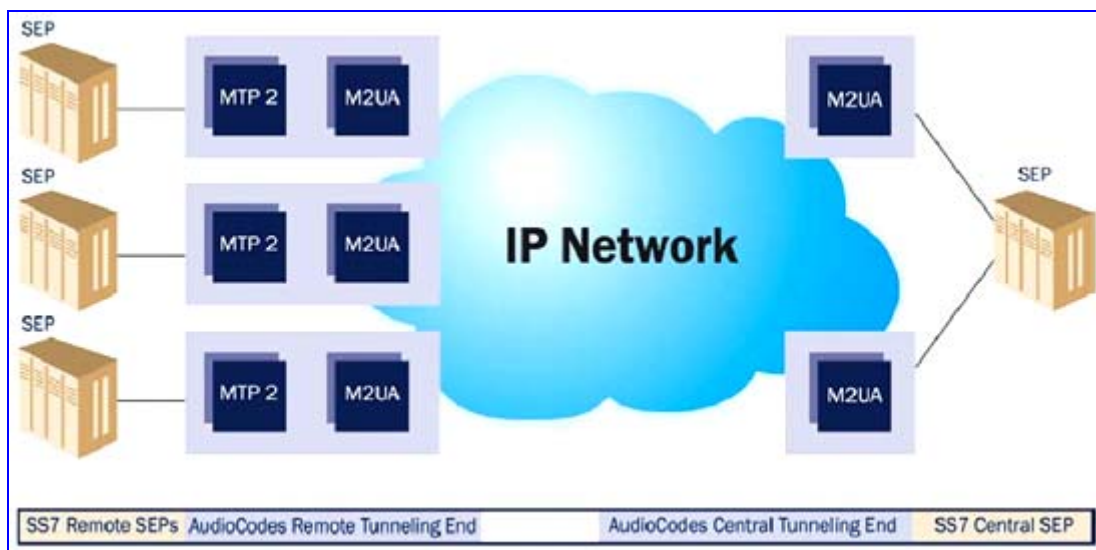
**Figure 10-2: M2TN Architecture**



## 10.1 MTP2 Tunneling Technology

The SS7 tunneling technology is based on a pairing of remote and central devices, as shown in the following figure. The remote devices are configured to backhaul MTP layer 2 signaling over the IP network using standard M2UA protocol (over SCTP protocol). The function of the M2TN entity is to transmit traffic and handle all management events between MTP2 on the TDM side and M2UA's MGC (Media Gateway Controller) entity on the IP side. Only the actual SS7 MSU (Message Signaling Unit) data is sent. Management of the SS7 link is performed using M2UA without transporting the MTP2 LSSU (Link Status Signaling Unit) and FISU (Fill in Signaling Unit) messages over IP. These messages, in addition to MTP2 timing, are terminated and supported, respectively, by the remote and central sides. Therefore, the MTP2 connections are not affected by the fact that they are transported over IP.

**Figure 10-3: Protocol Architecture for MTP2 Tunneling**



## 10.2 SS7 Characteristics

The SS7 characteristics are listed below:

- Only standard protocols are used on external interfaces (MTP2 on PSTN side, and M2UA over SCTP on IP side) - the M2TN application resides internally in the device.
- No extra signaling point codes are required; both endpoints are unaware that the SS7 connection is via IP.
- Several links from multiple SS7 nodes can be concentrated into a single blade on the 'Central' side (using several SCTP associations per device).
- The devices can handle SS7 MTP2 tunneling and voice concurrently (does not require additional device or other server).
- Voice and signaling can be transferred on the same E1/T1 trunk (F-Links).
- IP traffic can be monitored via standard sniffing tools (e.g. protocol analyzers).



**Note:** Channels that are used for SS7 Tunneling mustn't be defined in the Trunk Group table.



## 10.3 SS7 Parameters

Table 10-1: SS7 Parameters

<i>ini</i> File Name	Valid Range and Description
<b>SS7 MTP2 Parameter Table</b>	
<b>SS7Mtp2Parms</b>	<p>This <i>ini</i> file table parameter configures the SS7 MTP2 table parameters. The format of this parameter is as follows:</p> <pre>[SS7Mtp2Parms] FORMAT SS7Mtp2Parms_Index = SS7Mtp2Parms_LinkRate, SS7Mtp2Parms_ErrorCorrectionMethod, SS7Mtp2Parms_lacCp, SS7Mtp2Parms_SuermT, SS7Mtp2Parms_AermTin, SS7Mtp2Parms_AermTie, SS7Mtp2Parms_SuermSuD, SS7Mtp2Parms_OctetCounting, SS7Mtp2Parms_LssuLength, SS7Mtp2Parms_PcrN2, SS7Mtp2Parms_T1, SS7Mtp2Parms_T2, SS7Mtp2Parms_T3, SS7Mtp2Parms_T4n, SS7Mtp2Parms_T4e, SS7Mtp2Parms_T5, SS7Mtp2Parms_T6, SS7Mtp2Parms_T7; [SS7Mtp2Parms]</pre> <p>For example:</p> <pre>[SS7Mtp2Parms] SS7Mtp2Parms 0 = D, P, 0, 0, 0, 0, 0, 0, 1, 200, 13000, 11800, 11800, 2300, 600, 100, 3000, 1000; SS7Mtp2Parms 1 = A, B, 5, 64, 4, 1, 256, 16, 1, 200, 50000, 150000, 2000, 8200, 500, 120, 6000, 2000; [SS7Mtp2Parms]</pre> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>To configure the MTP2 table using the device's Web interface, refer to "Configuring MTP2 Attributes" on page 149.</li> <li>For a description on using <i>ini</i> file table parameters, refer to the device's <i>User's Manual</i>.</li> </ul>
<b>SIGTRAN Interface Groups Table</b>	
<b>SS7_SIG_IF_GROUP_TABLE</b>	<p>This <i>ini</i> file table parameter configures the Sigtran Interface Group table. The format of this parameter is as follows:</p> <pre>[SS7_SIG_IF_GROUP_TABLE] FORMAT SS7_SIG_IF_GR_INDEX = SS7_IF_GR_ID,SS7_SIG_SG_MGC, SS7_SIG_LAYER, SS7_SIG_TRAF_MODE, SS7_SIG_T_REC, SS7_SIG_T_ACK, SS7_SIG_T_HB, SS7_SIG_MIN_ASP, SS7_SIG_BEHAVIOUR, SS7_LOCAL_SCTP_PORT, SS7_SIG_NETWORK, SS7_DEST_SCTP_PORT, SS7_DEST_IP, SS7_MGC_MX_IN_STREAM, SS7_MGC_NUM_OUT_STREAM; [SS7_SIG_IF_GROUP_TABLE]</pre> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>To configure the MTP2 table using the device's Web</li> </ul>

<i>ini</i> File Name	Valid Range and Description
	<p>interface, refer to "Configuring Sigtran Group IDs" on page 163.</p> <ul style="list-style-type: none"> <li>For a description on using <i>ini</i> file table parameters, refer to the device's <i>User's Manual</i>.</li> </ul>
<b>RdcyBoardNum</b>	For a description of this parameter, refer to "Configuring Sigtran Group IDs" on page 163.
<b>SIGTRAN Interface IDs Table</b>	
<b>SS7_SIG_INT_ID_TABLE</b>	<p>This <i>ini</i> file table parameter configures the Sigtran Interface IDs table. The format of this parameter is as follows:  [SS7_SIG_INT_ID_TABLE]  FORMAT SS7_SIG_IF_ID_INDEX =  SS7_SIG_IF_ID_VALUE, SS7_SIG_IF_ID_NAME,  SS7_SIG_IF_ID_OWNER_GROUP,  SS7_SIG_IF_ID_LAYER, SS7_SIG_IF_ID_NAI,  SS7_SIG_M3UA_SPC;  [SS7_SIG_INT_ID_TABLE]</p> <p>For example:  [SS7_SIG_INT_ID_TABLE]  SS7_SIG_INT_ID_TABLE 0 = 1, INT_ID, 0, 1, 3, 0;  SS7_SIG_INT_ID_TABLE 1 = 0, INT_ID, 0, 1, 2, 0;  [SS7_SIG_INT_ID_TABLE]</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>To configure the Sigtran Interface IDs table using the device's Web interface, refer to "Configuring Sigtran Interface IDs" on page 165.</li> <li>For a description on using <i>ini</i> file table parameters, refer to the device's <i>User's Manual</i>.</li> </ul>
<b>SS7 Signaling Link Table</b>	
<b>SS7_LINK_TABLE</b> (SS7 Link table)	<p>This <i>ini</i> file table parameter configures the SS7 Links table. The format of this parameter is as follows:  [SS7_LINK_TABLE]  FORMAT SS7_LINK_INDEX = SS7_LINK_NAME,  SS7_LINK_TRACE_LEVEL,  SS7_LINK_ADMINISTRATIVE_STATE,  SS7_LINK_TRUNK_NUMBER,  SS7_LINK_TIMESLOT_NUMBER,  SS7_LINK_LAYER2_VARIANT, SS7_LINK_L2_TYPE,  SS7_LINK_L3_TYPE, SS7_LINK_MTP2_ATTRIBUTES,  SS7_CONGESTION_LOW_MARK,  SS7_CONGESTION_HIGH_MARK,  SS7_LINK_M2UA_IF_ID, SS7_LINK_GROUP_ID;  [SS7_LINK_TABLE]</p> <p>For example:  SS7_LINK_TABLE 0 = link_0_SP_A, 0, 2, 0, 16, 2, 1,2,0, 15, 80;  SS7_LINK_TABLE 1 = link_1_SP_B, 0, 2, 1, 16, 2, 1,2,0, 15, 80;  [SS7_LINK_TABLE]</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>To configure the SS7 Links table using the device's Web interface, refer to "Configuring Links" on page 157.</li> </ul>

<i>ini</i> File Name	Valid Range and Description
	<ul style="list-style-type: none"> <li>For a description on using <i>ini</i> file table parameters, refer to the device's <i>User's Manual</i>.</li> </ul>
<b>SS7_LINK_ACTION</b>	<p>Determines the management field for actions.</p> <ul style="list-style-type: none"> <li>[0] = acSS7LINK_PS_ACTION_NONE (default)</li> <li>[1] = acSS7LINK_PS_ACTION_OFFLINE</li> <li>[2] = acSS7LINK_PS_ACTION_INSERTSERVICE</li> <li>[3] = acSS7LINK_PS_ACTION_ACTIVATE</li> <li>[4] = acSS7LINK_PS_ACTION_DEACTIVATE</li> <li>[5] = acSS7LINK_PS_ACTION_INHIBIT</li> <li>[6] = acSS7LINK_PS_ACTION_UNINHIBIT</li> </ul>
<b>SS7_LINK_ACTION_RESULT</b>	<p>Determines the management field for actions result. The valid range is acPARAMSET_ACTION_RESULT_SUCCEEDED to acPARAMSET_ACTION_RESULT_FAILED. The default value is acPARAMSET_ACTION_RESULT_SUCCEEDED.</p>
<b>SS7_LINK_OPERATIONAL_STATE</b>	<p>For a description of this parameter, refer to "Configuring Links" on page 157.</p>
<b>SS7_LINK_ADMINISTRATIVE_STATE</b>	<p>For a description of this parameter, refer to Configuring Links.</p>
<b>SS7_LINK_MTC_BUSY</b>	<p>For a description of this parameter, refer to "Configuring Links" on page 157.</p>
<b>SS7_LINK_TNL_MGC_LINK_NUMBER</b>	<p>Determines the MTP2 Tunneling: MGC link number (MTP2 \other side\ of signaling link. The valid range is 0 to 83. The default value is 0.</p>
<b>SS7_LINK_TNL_ALIGNMENT_MODE</b>	<p>Determines the MTP2 Tunneling: Alignment mode of signaling links in tunnel.</p> <ul style="list-style-type: none"> <li>[0] = M3B_ALIGNMENT_NORMAL</li> <li>[1] = M3B_ALIGNMENT_EMERGENCY (default)</li> </ul>
<b>SS7_LINK_TNL_CONGESTION_MODE</b>	<p>Determines the MTP2 Tunneling: Congestion mode of signaling links in tunnel.</p> <ul style="list-style-type: none"> <li>[0] = M3B_CONGESTION_ACCEPT (default)</li> <li>[1] = M3B_CONGESTION_DISCARD</li> </ul>
<b>SS7_LINK_TNL_WAIT_START_COMPLETE_TIMER</b>	<p>Determines the MTP2 Tunneling Timer: wait start complete. The valid range is 500 to 0xFFFFFFFF. The default value is 30000.</p>
<b>SS7_LINK_TNL_OOS_START_DELAY_TIMER</b>	<p>Determines the MTP2 Tunneling Timer: OOS start delay. The valid range is 500 to 0xFFFFFFFF. The default value is 5000.</p>
<b>SS7_LINK_TNL_WAIT_OTHER_SIDE_INSERTSERVICE_TIMER</b>	<p>Determines the MTP2 Tunneling Timer: wait other side insertservice. The valid range is 500 to 0xFFFFFFFF. The default value is 30000.</p>
<b>SS7 Link-Set Parameters</b>	
<b>SS7_LINKSET_TABLE</b>	<p>This <i>ini</i> file table parameter configures the SS7 Link-Set table. The format of this parameter is as follows:</p>

<i>ini</i> File Name	Valid Range and Description
	<p>[ SS7_LINKSET_TABLE ]            FORMAT SS7_LINKSET <b>SN_INDEX</b>,            SS7_LINKSET <b>LINKSET_INDEX</b> = SS7_LINKSET <b>NAME</b>,            SS7_LINKSET <b>ADMINISTRATIVE_STATE</b>,            SS7_LINKSET <b>DPC</b>, SS7_LINKSET <b>TIMERS_INDEX</b>;            [ \SS7_LINKSET_TABLE ]</p> <p>Where,</p> <ul style="list-style-type: none"> <li>▪ <b>SS7_LINKSET_SN_INDEX</b> = First index field for line. The valid range is 0 to 1. The default value is 0.</li> <li>▪ <b>SS7_LINKSET_LINKSET_INDEX</b> = Second index field for line. The valid range is 0 to 83. The default value is 0.</li> </ul> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>▪ To configure the SS7 Link-Set table using the device's Web interface, refer to "Configuring Link-Set Timers" on page <a href="#">154</a>.</li> <li>▪ For a description on using <i>ini</i> file table parameters, refer to the device's <i>User's Manual</i>.</li> </ul>
<b>SS7_LINKSET_ROWSTATUS</b>	<p>Determines the RowStatusField for line.            The valid range is            acPARAMSET_ROWSTATUS_DOESNOTEXIST to            acPARAMSET_ROWSTATUS_DESTROY. The default            value is acPARAMSET_ROWSTATUS_DOESNOTEXIST.</p>
<b>SS7_LINKSET_ACTION</b>	<p>Determines the management field for actions.</p> <ul style="list-style-type: none"> <li>▪ <b>[0]</b> = acSS7LINKSET_PS_ACTION_NONE (default)</li> <li>▪ <b>[1]</b> = acSS7LINKSET_PS_ACTION_OFFLINE</li> <li>▪ <b>[2]</b> = acSS7LINKSET_PS_ACTION_INSERTSERVICE</li> <li>▪ <b>[3]</b> = acSS7LINKSET_PS_ACTION_ACTIVATE</li> <li>▪ <b>[4]</b> = acSS7LINKSET_PS_ACTION_DEACTIVATE</li> </ul>
<b>SS7_SN_ACTION_RESULT</b>	<p>Determines the management field for actions result.            The valid range is            acPARAMSET_ACTION_RESULT_SUCCEEDED to            acPARAMSET_ACTION_RESULT_FAILED. The default            value is acPARAMSET_ACTION_RESULT_SUCCEEDED.</p>
<b>SS7 Signaling Node Timers Parameter Table</b>	
<b>SS7_SN_TIMERS_TABLE</b>	<p>This <i>ini</i> file table parameter configures the SS7 Signaling Node Timers table. The format of this parameter is as follows:</p> <p>[SS7_SN_TIMERS_TABLE]            FORMAT SS7_SNTIMERS <b>INDEX</b> =            SS7_SNTIMERS <b>NAME</b>, SS7_SNTIMERS <b>T6</b>,            SS7_SNTIMERS <b>T8</b>, SS7_SNTIMERS <b>T10</b>,            SS7_SNTIMERS <b>T11</b>, SS7_SNTIMERS <b>T15</b>,            SS7_SNTIMERS <b>T16</b>, SS7_SNTIMERS <b>T22_ANSI</b>,            SS7_SNTIMERS <b>T23_ANSI</b>, SS7_SNTIMERS <b>T24_ANSI</b>,            SS7_SNTIMERS <b>T25_ANSI</b>, SS7_SNTIMERS <b>T26_ANSI</b>,            SS7_SNTIMERS <b>T28_ANSI</b>, SS7_SNTIMERS <b>T29_ANSI</b>,            SS7_SNTIMERS <b>T30_ANSI</b>, SS7_SNTIMERS <b>T18_ITU</b>,            SS7_SNTIMERS <b>T19_ITU</b>, SS7_SNTIMERS <b>T20_ITU</b>,</p>

<i>ini</i> File Name	Valid Range and Description
	<p>SS7_SNTIMERS_T21_ITU, SS7_SNTIMERS_T24_ITU; [SS7_SN_TIMERS_TABLE]</p> <p>For example: [SS7_SN_TIMERS_TABLE] SS7_SN_TIMERS_TABLE 1 = BABILON_0, 800, 1000, 30000, 30000, 2000, 1400, 180000, 180000, 5000, 30000, 12000, 3000, 60000, 30000; [SS7_SN_TIMERS_TABLE]</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>▪ To configure the SS7 Signaling Node Timers table using the device's Web interface, refer to "Configuring SS7 Signaling Node Timers" on page 152.</li> <li>▪ For a description on using <i>ini</i> file table parameters, refer to the device's <i>User's Manual</i>.</li> </ul>
<b>SS7 Link-Set Timers Parameter Table</b>	
<p><b>SS7_LINKSET_TIMERS_TABLE</b> (SS7 Link Set Timers table)</p>	<p>This <i>ini</i> file table parameter configures the SS7 Link Set Timers table. The format of this parameter is as follows: [SS7_LINKSET_TIMERS_TABLE] FORMAT SS7_LKSETTIMERS_INDEX = SS7_LKSETTIMERS_NAME, SS7_LKSETTIMERS_T1SLT, SS7_LKSETTIMERS_T2SLT, SS7_LKSETTIMERS_T1, SS7_LKSETTIMERS_T2, SS7_LKSETTIMERS_T3, SS7_LKSETTIMERS_T4, SS7_LKSETTIMERS_T5, SS7_LKSETTIMERS_T7, SS7_LKSETTIMERS_T12, SS7_LKSETTIMERS_T13, SS7_LKSETTIMERS_T14, SS7_LKSETTIMERS_T17, SS7_LKSETTIMERS_T20_ANSI, SS7_LKSETTIMERS_T21_ANSI, SS7_LKSETTIMERS_T22_ITU, SS7_LKSETTIMERS_T23_ITU; [SS7_LINKSET_TIMERS_TABLE]</p> <p>For example: [SS7_LINKSET_TIMERS_TABLE] SS7_LINKSET_TIMERS_TABLE 1 = DUBLIN, 8000, 30000, 800, 1400, 800, 800, 800, \$\$, 1000, 1500, 2000, 1500, 90000, 90000, \$\$, \$\$; [SS7_LINKSET_TIMERS_TABLE]</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>▪ To configure the SS7 Link Set Timers table using the device's Web interface, refer to "Configuring Link-Set Timers" on page 154.</li> <li>▪ For a description on using <i>ini</i> file table parameters, refer to the device's <i>User's Manual</i>.</li> </ul>

<i>ini</i> File Name	Valid Range and Description
<b>SS7 Static Routing Context Parameter Table</b>	
<b>[SS7_ROUTING_CONTEXT_TABLE ]</b>	<p>This <i>ini</i> file table parameter configures the SS7 Static Routing Context table. The format of this parameter is as follows:</p> <pre>[ SS7_ROUTING_CONTEXT_TABLE ] FORMAT SS7_RC_INDEX, SS7_RC_INNER_INDEX = SS7_RC_SN_INDEX, SS7_RC_OPC1, SS7_RC_OPC2, SS7_RC_OPC3, SS7_RC_OPC4, SS7_RC_SI1, SS7_RC_SI2, SS7_RC_SI3, SS7_RC_SI4; [ \SS7_ROUTING_CONTEXT_TABLE ]</pre> <p>For example:</p> <pre>[ SS7_ROUTING_CONTEXT_TABLE ] FORMAT SS7_RC_INDEX, SS7_RC_INNER_INDEX = SS7_RC_SN_INDEX, SS7_RC_OPC1, SS7_RC_OPC2, SS7_RC_OPC3, SS7_RC_OPC4, SS7_RC_SI1, SS7_RC_SI2, SS7_RC_SI3, SS7_RC_SI4;  SS7_ROUTING_CONTEXT_TABLE 0, 0 = 0, -1, -1, -1, -1, -1, -1, -1;  [ \SS7_ROUTING_CONTEXT_TABLE ]</pre>

## 10.4 SS7 MTP2 Tunneling ini File Example

For the SS7 MTP2 tunneling *ini* file example, note the following:

- The first *ini* file acts as an MTP2 tunneling central side (M2UA MGC links).
  - There are eight SS7 links - four links of type: MTP2 MGC, and four links of type MTP2. Each pair of links (one MTP2 MGC and one MTP2) defines an MTP2 tunnel.
  - There is one interface that is used for the M2UA MGC <=> M2UA SG (Signaling Gateway) connection.
  - There are four interface IDs defined: one per link (M2UA MGC side).
  - This file is intended for ITU link variant (E1 trunks).
- **To load the example SS7 MTP2 tunneling *ini* files to the devices, take these 4 steps:**
1. Load the *ini* file that is shown below (**SS7 MTP2 Tunneling ini File Example - MGC**) to a tunnel central gateway (MTP2 MGC).
  2. Load the *ini* file that is shown below (**SS7 MTP2 Tunneling ini File Example - SG**) to a tunnel remote gateway (MTP2 SG); the MGC gateway connects (over IP) to the SG gateway. For information on loading an *ini* file to the device, refer to 'Modifying an ini File' in the device's *User's Manual*.
  3. In the MGC gateway, change the parameter 'SS7\_DEST\_IP' to the actual IP address of the M2UA SG gateway.
  4. Change the value of the 'SyslogServerIP' parameter in the MGC and SG gateways to your Syslog server IP address.

**SS7 MTP2 Tunneling ini File Example - MGC:**

```

[TDM BUS configuration]
; 1=aLaw 3=ulaw
PCMLawSelect= 1
;1 - internal, 3 - mvip, 4 - Network, 8 - h110a, 9 - h110b, 10 -
Netref
TDMBusClockSource= 1
[Trunk Configuration]
;e1 euro isdn=1 t1 isdn=2 ;e1 cas r2=8 (8 for fcd); e1 trans 62=5
ProtocolType = 5
TraceLevel = 0
; acCLOCK MASTER ON =1
CLOCKMASTER= 1
;acUSER TERMINATION SIDE = 0
TerminationSide = 1
;acEXTENDED SUPER FRAME=0
FramingMethod = 0
;acB8ZS = 0 2 for E1 CAS - FCD
LineCode = 0
[SS7]
SS7 MTP2 PARAM TIMER T1 0=50000
SS7 MTP2 PARAM TIMER T2 0=150000
SS7 MTP2 PARAM TIMER T3 0=1000
SS7 MTP2 PARAM TIMER T4E 0=500
SS7 MTP2 PARAM TIMER T4N 0=8200
SS7 MTP2 PARAM TIMER T5 0=100
SS7 MTP2 PARAM TIMER T6 0=3000
SS7 MTP2 PARAM TIMER T7 0=2000
[syslog]
SYSLOGSERVERIP = 168.100.0.1
ENABLESYSLOG = 1
WATCHDOGSTATUS = 0
[ SS7 LINK TABLE ]
FORMAT SS7 LINK INDEX = SS7 LINK NAME, SS7 LINK TRACE LEVEL,
SS7 LINK ADMINISTRATIVE STATE,SS7 LINK L2_TYPE, SS7 LINK L3_TYPE,
SS7 LINK GROUP ID, SS7 LINK M2UA IF ID;
SS7 LINK TABLE 1 = new link 1, 0, 2, 2, 3, 4, 50;
SS7 LINK TABLE 3 = new link 3, 0, 2, 2, 3, 4, 12;
SS7 LINK TABLE 5 = new link 5, 0, 2, 2, 3, 4, 18;
SS7 LINK TABLE 7 = new link 7, 0, 2, 2, 3, 4, 1;
[ \SS7 LINK TABLE ]
[ SS7 LINK TABLE ]
FORMAT SS7 LINK INDEX = SS7 LINK NAME, SS7 LINK TRACE LEVEL,
SS7 LINK ADMINISTRATIVE STATE,SS7 LINK L2_TYPE, SS7 LINK L3_TYPE,
SS7 LINK TRUNK NUMBER,SS7 LINK TIMESLOT NUMBER,
SS7 LINK LAYER2 VARIANT,SS7 LINK MTP2 ATTRIBUTES,SS7 CONGESTION LO
W MARK, SS7 CONGESTION HIGH MARK, SS7 LINK TNL MGC LINK NUMBER,
SS7 LINK TNL ALIGNMENT MODE, SS7 LINK TNL CONGESTION MODE,
SS7 LINK TNL WAIT START COMPLETE TIMER,
SS7 LINK TNL OOS START DELAY TIMER,
SS7 LINK TNL WAIT OTHER SIDE INSV TIMER;
SS7 LINK TABLE 0 = new link 0, 0, 2, 1, 3, 0, 15, 1, 0, 5, 50, 1,
1, 0, 30000, 5000, 30000;
SS7 LINK TABLE 2 = new link 2, 0, 2, 1, 3, 3, 12, 1, 0, 5, 50, 3,
1, 0, 30000, 5000, 30000;
SS7 LINK TABLE 4 = new link 4, 0, 2, 1, 3, 6, 7, 1, 0, 5, 50, 5,
1, 0, 30000, 5000, 30000;
SS7 LINK TABLE 6 = new link 6, 0, 2, 1, 3, 7, 3, 1, 0, 5, 50, 7,
1, 0, 30000, 5000, 30000;
[ \SS7 LINK TABLE ]
[ SS7 SIG IF GROUP TABLE ]
FORMAT SS7 SIG IF GR INDEX = SS7 IF GR ID,SS7 SIG SG MGC,
SS7 SIG LAYER, SS7 SIG TRAF MODE, SS7 SIG T REC, SS7 SIG T ACK,
SS7 SIG T HB, SS7 SIG MIN ASP, SS7 SIG BEHAVIOUR,
SS7_LOCAL SCTP_PORT, SS7_SIG NETWORK, SS7_DEST SCTP_PORT,
SS7_DEST IP, SS7_MGC MX IN_STREAM, SS7_MGC_NUM OUT_STREAM;

```

```

SS7 SIG IF GROUP TABLE 4 = 4, 77, 4, 1, 2000, 2000, 30000, 1, 0,
2904, 1,2904,168.100.0.2,3,3;
[ \SS7_SIG_IF_GROUP_TABLE ]
[ SS7 ROUTING CONTEXT TABLE ]
FORMAT SS7 RC INDEX, SS7 RC INNER INDEX = SS7 RC SN INDEX,
SS7_RC_OPC1, SS7_RC_OPC2, SS7_RC_OPC3, SS7_RC_OPC4, SS7_RC_SI1,
SS7_RC_SI2, SS7_RC_SI3, SS7_RC_SI4;
SS7 ROUTING CONTEXT TABLE 0, 0 = 0, -1, -1, -1, -1, -1, -1, -
1;
[ \SS7 ROUTING CONTEXT TABLE ]
[ SS7 SIG INT ID TABLE ]FORMAT SS7 SIG IF ID INDEX =
SS7 SIG IF ID VALUE, SS7 SIG IF ID NAME,
SS7 SIG IF ID OWNER GROUP, SS7 SIG IF ID LAYER, SS7 SIG IF ID NAI,
SS7 SIG M3UA SPC;
SS7 SIG INT ID TABLE 7 = 50, BELFAST12, 4, 4, 1, 0;
SS7 SIG INT ID TABLE 8 = 12, AMSTERDAM, 4, 4, 3, 0;
SS7 SIG INT ID TABLE 9 = 18, ROTTERDAM , 4, 4, 5, 0;
SS7 SIG INT ID TABLE 10 = 1, GAUDA , 4, 4, 7, 0;
[ \SS7_SIG_INT_ID_TABLE ]

```

### SS7 MTP2 Tunneling ini File Example - SG

```

[TDM BUS configuration]
; 1=aLaw 3=ulaw
PCMLawSelect= 1
;1 - internal, 3 - mvip, 4 - Network, 8 - h110a, 9 - h110b, 10 -
Netref
TDMBusClockSource= 1
[Trunk Configuration]
;e1 euro isdn=1 t1 isdn=2 ;e1 cas r2=8 (8 for fcd); e1 trans 62=5
ProtocolType = 5
TraceLevel = 0
; acCLOCK MASTER ON =1
ClockMaster= 1
TerminationSide = 1
;acEXTENDED SUPER FRAME=0
FramingMethod = 0
;acB8ZS = 0 2 for E1 CAS - FCD
LineCode = 0
WATCHDOGSTATUS = 0
[ SS7 LINK TABLE ]
FORMAT SS7 LINK INDEX = SS7 LINK NAME, SS7 LINK TRACE LEVEL,
SS7_LINK_ADMINISTRATIVE_STATE,SS7_LINK_L2_TYPE, SS7_LINK_L3_TYPE,
SS7_LINK_TRUNK_NUMBER,SS7_LINK_TIMESLOT_NUMBER,SS7_LINK_M2UA_IF_ID
;
SS7_LINK_TABLE 0 = new_link_0, 0, 2, 1,1, 1, 15,50;
SS7_LINK_TABLE 1 = new link 1, 0, 2, 1,1, 2, 12, 12;
SS7_LINK_TABLE 2 = new link 2, 0, 2, 1, 1, 4, 7,18;
SS7_LINK_TABLE 3 = new_link_3, 0, 2, 1, 1, 5, 3,1;
[ \SS7 LINK TABLE ]
[ SS7 SIG IF GROUP TABLE ]
FORMAT SS7_SIG_IF_GR_INDEX = SS7_IF_GR_ID,SS7_SIG_SG_MGC,
SS7_SIG_LAYER, SS7_SIG_TRAF_MODE, SS7_SIG_T_REC, SS7_SIG_T_ACK,
SS7_SIG_T_HB, SS7_SIG_MIN_ASP, SS7_SIG_BEHAVIOUR,
SS7_LOCAL_SCTP_PORT, SS7_SIG_NETWORK;
SS7_SIG_IF_GROUP_TABLE 4 = 4,83, 2, 1, 2000, 2000, 30000, 1, 0,
2904, 1;
[ \SS7_SIG_IF_GROUP_TABLE ]
[ SS7 SIG INT ID TABLE ]
FORMAT SS7 SIG IF ID INDEX = SS7 SIG IF ID VALUE,
SS7_SIG_IF_ID_NAME, SS7_SIG_IF_ID_OWNER_GROUP,
SS7_SIG_IF_ID_LAYER, SS7_SIG_IF_ID_NAI, SS7_SIG_M3UA_SPC;
SS7_SIG_INT_ID_TABLE 7 = 50, BELFAST12, 4, 4, 0, 0;
SS7_SIG_INT_ID_TABLE 8 = 12, AMSTERDAM, 4, 4, 1, 0;
SS7_SIG_INT_ID_TABLE 9 = 18, ROTTERDAM , 4, 4, 2, 0;
SS7_SIG_INT_ID_TABLE 10 = 1, GAUDA , 4, 4, 3, 0;
[ \SS7_SIG_INT_ID_TABLE ]

```



## 10.5 Configuring SS7 Tunneling

You can configure SS7 in the Web interface using the **SS7 Configuration** menu:

- Configure M2P2 Attributes (refer to "Configuring M2P2 Attributes" on page [149](#))
- Configure SS7 Signaling Node Timers (refer to "Configuring SS7 Signaling Node Timers" on page [152](#))
- Configure SS7 Link-Set Timers (refer to "Configuring Link-Set Timers" on page [154](#))
- Configure Links (refer to "Configuring Links" on page [157](#))
- Configure SS7 Signaling Nodes (refer to "Configuring SS7 Signaling Nodes" on page [159](#))
- Configure SS7 MTP3 redundancy (refer to "Configuring MTP3 Redundancy" on page [161](#))
- Configure SS7 Static Routing Context (refer to "Configuring Static Routing Context" on page [162](#))
- Configure Sigtran Group IDs (refer to "Configuring Sigtran Group IDs" on page [163](#))
- Configure Sigtran Interface IDs (refer to "Configuring Sigtran Interface IDs" on page [165](#))

### 10.5.1 Configuring MTP2 Attributes

The 'MTP2 Attributes' page allows you to configure Message Transfer Part level 2 (MTP2) parameters. These parameters can also be configured using the *ini* file parameter table SS7Mtp2Parms (refer to "SS7 Parameters" on page [141](#)). For a detailed description of MTP2, refer to "MTP2 Tunneling Technology" on page [140](#).

➤ **To configure the MTP2 attributes parameters, take these 4 steps:**

1. Open the 'MTP2 Attributes' page (**Configuration** tab > **SS7 Configuration** menu > **MTP2 Attributes** page item).

**Figure 10-4: MTP2 Attributes Page**

▼	
Profile Number	0 ▼
▼	
Link Rate	A ▼
Error Correction Method	B ▼
IAC CP	5
SUERM T	64
AERM TIN	4
AERM TIE	1
SUERM SU D	256
Octet Counting	16
LSSU Length	1
PCR N2	200
▼ MTP2 Timers	
T1	50000
T2	150000
T3	2000
T4N	8200
T4E	500
T5	120
T6	6000
T7	2000

2. Configure the parameters according to the table below.
3. Click **Submit**.
4. To save the changes to flash memory, refer to 'Saving Configuration' in the device's *User's Manual*.

**Table 10-2: MTP2 Parameters**

Parameter	Description
Profile Number	Profile number for this link.
Link Rate [SS7Mtp2Parms_LinkRate]	Defines the SS7 SLI Link Rate. Choose either: <ul style="list-style-type: none"> <li>▪ <b>[0]</b> 0 = 64 kbps (default)</li> <li>▪ <b>[A]</b> A = 64 kbps</li> </ul>

Parameter	Description
	<ul style="list-style-type: none"> <li>▪ <b>[D]</b> D = 56 kbps</li> </ul>
Error Correction Method <b>[SS7Mtp2Parms_ErrorCorrectionMethod]</b>	<p>Defines the SLI error correction method.</p> <ul style="list-style-type: none"> <li>▪ <b>[0]</b> 0 = Basic (default)</li> <li>▪ <b>[B]</b> B = Basic</li> <li>▪ <b>[P]</b> P = PCR (Preventive Cyclic Retransmission)</li> </ul>
IAC CP <b>[SS7Mtp2Parms_lacCp]</b>	<p>Defines the number of aborted proving attempts before sending an out-of-service to MTP-3. The valid range is 0 to 10. The default value is 5.</p>
SUERM T <b>[SS7Mtp2Parms_SuermT]</b>	<p>Defines the SS7 SUERM (Signal Unit Error Rate Monitor) T threshold. The valid range is 0 to 256. The default value is 64.</p>
AERM TIN <b>[SS7Mtp2Parms_AermTin]</b>	<p>Defines the SS7 alignment normal error rate threshold. The valid range is 0 to 20. The default value is 4.</p>
AERM TIE <b>[SS7Mtp2Parms_AermTie]</b>	<p>Defines the SS7 alignment emergency error rate threshold. The valid range is 0 to 10. The default value is 1.</p>
SUERM SU D <b>[SS7Mtp2Parms_SuermSuD]</b>	<p>Defines the SS7 Signal Unit error rate monitor D threshold. The valid range is 0 to 256. The default value is 256.</p>
Octet Counting <b>[SS7Mtp2Parms_OctetCounting]</b>	<p>Defines the SS7 MTP2 Octet received while the OCTET is in counting mode (# of Octets received - N Octets - while in Octet counting mode). The valid range is 0 to 256. The default value is 16.</p>
LSSU Length <b>[SS7Mtp2Parms_LSSULength]</b>	<p>Defines the SS7 MTP2 LSSU length as 1 or 2 (bytes). The valid range is 1 to 2. The default value is 1.</p>
PCR N2 <b>[SS7Mtp2Parms_PcrN2]</b>	<p>Number of message signal unit octets available for retransmission. The valid range is 0 to 512. The default value is 200.</p>
<b>MTP2 Timers</b>	
T1 <b>[SS7Mtp2Parms_T1]</b>	<p>Defines the SS7 MTP2 T1 alignment ready timer (in msec). The valid range is 0 to 100000. The default value is 50000.</p>
T2 <b>[SS7Mtp2Parms_T2]</b>	<p>Defines the SS7 MTP2 T2 unaligned timer (in msec). The valid range is 0 to 200000. The default value is 150000.</p>
T3 <b>[SS7Mtp2Parms_T3]</b>	<p>Defines the SS7 MTP2 T3 timer aligned. The valid range is 0 to 20000. The default value is 2000.</p>
T4N <b>[SS7Mtp2Parms_T4N]</b>	<p>Defines the SS7 MTP2 T4n Nominal proving period timer. The valid range is 0 to 15000. The default value is 8200.</p>
T4E <b>[SS7Mtp2Parms_T4E]</b>	<p>Defines the SS7 MTP2 T4e Emergency proving period timer (msec). The valid range is 0 to 5000. The default value is 500.</p>
T5 <b>[SS7Mtp2Parms_T5]</b>	<p>Defines the SS7 MTP2 Sending SIB timer. The valid range is 0 to 2400. The default value is 120.</p>
T6 <b>[SS7Mtp2Parms_T6]</b>	<p>Defines the SS7 MTP2 Remote Congestion timer (in msec). The valid range is 0 to 10000. The default value is 6000.</p>

Parameter	Description
T7 [SS7Mtp2Parms_T7]	Defines the SS7 MTP2 excessive delay of the ack timer (in msec). The valid range is 0 to 5000. The default value is 2000.

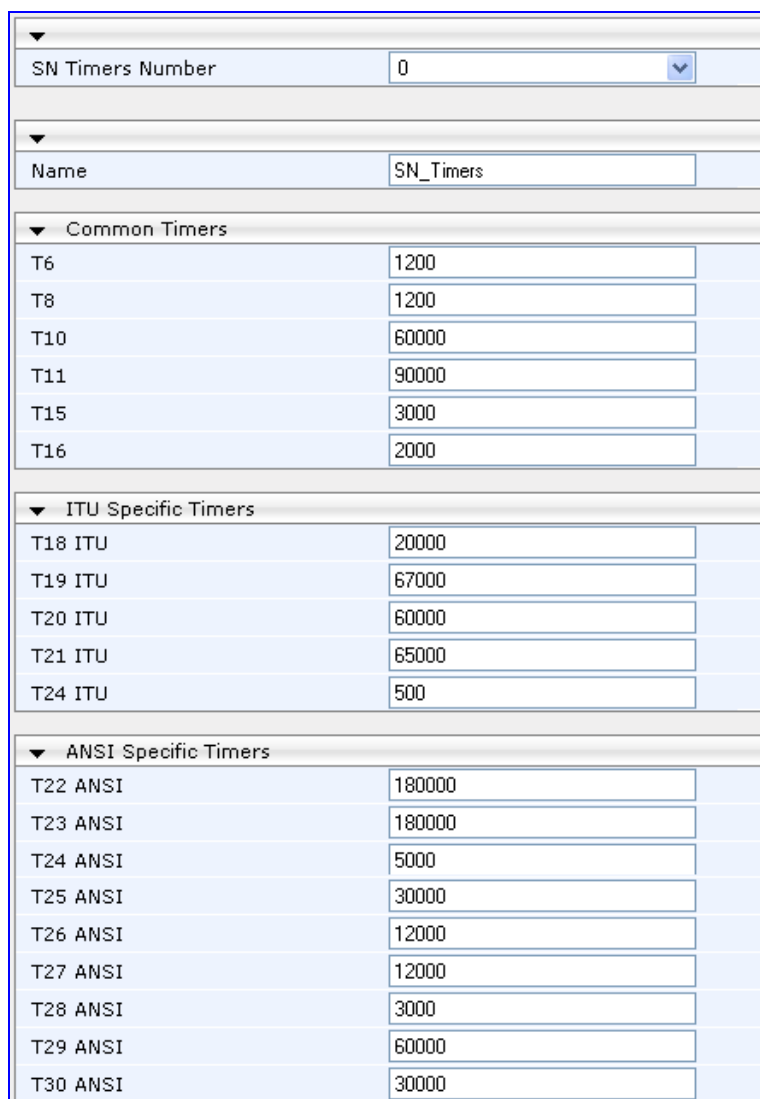
## 10.5.2 Configuring SS7 Signaling Node Timers

The 'SS7 Signaling Node Timers' page allows you to configure the SS7 Signaling Node Timers. These parameters can also be configured using the *ini* file parameter table SS7\_SN\_TIMERS\_TABLE (refer to "SS7 Parameters" on page 141).

➤ **To configure the Signaling Node Timers parameters, take these 4 steps:**

1. Open the 'SS7 Signaling Node Timers' page (**Configuration** tab > **SS7 Configuration** menu > **SN Timers** page item).

**Figure 10-5: SS7 Signaling Node Timers Page**



Common Timers	
T6	1200
T8	1200
T10	60000
T11	90000
T15	3000
T16	2000

ITU Specific Timers	
T18 ITU	20000
T19 ITU	67000
T20 ITU	60000
T21 ITU	65000
T24 ITU	500

ANSI Specific Timers	
T22 ANSI	180000
T23 ANSI	180000
T24 ANSI	5000
T25 ANSI	30000
T26 ANSI	12000
T27 ANSI	12000
T28 ANSI	3000
T29 ANSI	60000
T30 ANSI	30000

2. Configure the parameters according to the table below.

3. Click **Submit**.
4. To save the changes to flash memory, refer to 'Saving Configuration' in the device's *User's Manual*.

**Table 10-3: SS7 Signaling Node Timers Parameters**

Parameter	Description
SN Timers Number [SS7_SNTIMERS_INDEX]	Index field for the table row entry. The range is 0 to MTP3_SN_TIMER_SETS-1. The default is 0.
Name [SS7_SNTIMERS_NAME]	String name for SN timer-set. The default is 'SN_Timers'.
<b>Common Timers</b>	
T6 [SS7_SNTIMERS_T6]	Delay to avoid message mis-sequencing on controlled rerouting. The range is 500 to 4294967295. The default is 1200.
T8 [SS7_SNTIMERS_T8]	Transfer prohibited inhibition timer (transient solution). The range is 500 to 4294967295. The default is 1200.
T10 [SS7_SNTIMERS_T10]	Waiting to repeat signaling route set test message. The range is 500 to 4294967295. The default is 60000.
T11 [SS7_SNTIMERS_T11]	Transfer restricted timer. The range is 500 to 4294967295. The default is 90000.
T15 [SS7_SNTIMERS_T15]	Waiting to start signaling route set congestion test. The range is 500 to 4294967295. The default is 3000.
T16 [SS7_SNTIMERS_T16]	Waiting for route set congestion status update. The range is 500 to 4294967295. The default is 2000.
<b>ITU Specific Timers</b>	
T18 ITU [SS7_SNTIMERS_T18_ITU]	Timer within a signaling point whose MTP restarts for supervising link and link set activation as well as the receipt of routing information. The range is 500 to 4294967295. The default is 20000.
T19 ITU [SS7_SNTIMERS_T19_ITU]	Supervision timer during MTP restart to avoid possible ping-pong of TFP, TFR and TRA messages. The range is 500 to 4294967295. The default is 67000.
T20 ITU [SS7_SNTIMERS_T20_ITU]	Overall MTP restart timer at the signaling point whose MTP restarts. The range is 500 to 4294967295. The default is 60000.
T21 ITU [SS7_SNTIMERS_T21_ITU]	Overall MTP restart timer at a signaling point adjacent to one whose MTP restarts. The range is 500 to 4294967295. The default is 65000.
T24 ITU [SS7_SNTIMERS_T24_ITU]	Stabilizing timer after removal of local processor outage, used in LPO latching to RPO (national option). The range is 500 to 4294967295. The default is 500.
<b>ANSI Specific Timers</b>	
T22 ANSI [SS7_SNTIMERS_T22_ANSI]	Timer at restarting SP waiting for signaling links to become available. The range is 500 to 4294967295. The default is 180000.

Parameter	Description
T23 ANSI [SS7_SNTIMERS_T23_ANSI]	Timer at restarting SP, started after T22, waiting to receive all traffic restart allowed messages. The range is 500 to 4294967295. The default is 180000.
T24 ANSI [SS7_SNTIMERS_T24_ANSI]	Timer at restarting SP with transfer function, started after T23, waiting to broadcast all traffic restart allowed messages. The range is 500 to 4294967295. The default is 5000.
T25 ANSI [SS7_SNTIMERS_T25_ANSI]	Timer at SP adjacent to restarting SP waiting for traffic restart allowed message. The range is 500 to 4294967295. The default is 30000.
T26 ANSI [SS7_SNTIMERS_T26_ANSI]	Timer at restarting SP waiting to repeat traffic restart waiting message. The range is 500 to 4294967295. The default is 12000.
T28 ANSI [SS7_SNTIMERS_T28_ANSI]	Timer at SP adjacent to restarting SP waiting for traffic restart waiting message. The range is 500 to 4294967295. The default is 3000.
T29 ANSI [SS7_SNTIMERS_T29_ANSI]	Timer started when TRA sent in response to unexpected TRA or TRW. The range is 500 to 4294967295. The default is 60000.
T30 ANSI [SS7_SNTIMERS_T30_ANSI]	Timer to limit sending of TFPs and TFRs in response to unexpected TRA or TRW. The range is 500 to 4294967295. The default is 30000.

### 10.5.3 Configuring Link-Set Timers

The 'SS7 Link-set Timers' page allows you to configure SS7 Link-set Timers.

➤ **To configure the SS7 Link-set Timers parameters, take these 4 steps:**

1. Open the 'SS7 Link-set Timers' page (**Configuration** tab > **SS7 Configuration** menu > **Link Set Timers** page item).

**Figure 10-6: SS7 Link-set Timers Page**

▼	
Link-set Timers Number	0 ▼
▼	
Name	LINKSET_Timers
▼ Common Timers	
T1SLT	8000
T2SLT	30000
T1	1000
T2	2000
T3	1200
T4	1200
T5	1200
T7	2000
T12	1200
T13	1300
T14	3000
T17	1500
▼ ITU Specific Timers	
T22 ITU	180000
T23 ITU	180000
▼ ANSI Specific Timers	
T20 ANSI	90000
T21 ANSI	90000

2. Configure the parameters according to the table below.
3. Click **Submit**.
4. To save the changes to flash memory, refer to 'Saving Configuration' in the device's *User's Manual*.

**Table 10-4: SS7 Link-Set Timers Parameters**

Parameter	Description
Link-set Timers Number [SS7_LKSETTIMERS_INDEX]	Index field for table entry. The range is 0 to (MTP3_LKSET_TIMER_SETS-1). The default is 0.
Name [SS7_LKSETTIMERS_NAME]	String name for SN timer-set. The default name is 'LINKSET_Timers'.
<b>Common Timers</b>	
T1SLT [SS7_LKSETTIMERS_T1SLT]	Supervision timer for signaling link test acknowledgement message. The range is 500 to 4294967295. The default is 8000.
T2SLT [SS7_LKSETTIMERS_T2SLT]	Interval timer for sending signaling link test messages. The range is 500 to 4294967295. The default is 30000.
T1 [SS7_LKSETTIMERS_T1]	Delay to avoid message mis-sequencing on changeover. The range is 500 to 4294967295. The default is 1000.
T2 [SS7_LKSETTIMERS_T2]	Waiting for changeover acknowledgement. The range is 500 to 4294967295. The default is 2000.
T3 [SS7_LKSETTIMERS_T3]	Time controlled diversion-delay to avoid mis-sequencing on changeback. The range is 500 to 4294967295. The default is 1200.
T4 [SS7_LKSETTIMERS_T4]	Waiting for changeback acknowledgement (first attempt). The range is 500 to 4294967295. The default is 1200.
T5 [SS7_LKSETTIMERS_T5]	Waiting for changeback acknowledgement (second attempt). The range is 500 to 4294967295. The default is 1200.
T7 [SS7_LKSETTIMERS_T7]	Waiting for signaling data link connection acknowledgement. The range is 500 to 4294967295. The default is 2000.
T12 [SS7_LKSETTIMERS_T12]	Waiting for uninhibit acknowledgement. The range is 500 to 4294967295. The default is 1200.
T13 [SS7_LKSETTIMERS_T13]	Waiting for force uninhibit. The range is 500 to 4294967295. The default is 1300.
T14 [SS7_LKSETTIMERS_T14]	Waiting for inhibition acknowledgement. The range is 500 to 4294967295. The default is 3000.
T17 [SS7_LKSETTIMERS_T17]	Delay to avoid oscillation of initial alignment failure and link restart. The range is 500 to 4294967295. The default is 1500.
<b>ITU Specific Timers</b>	
T22 ITU [SS7_LKSETTIMERS_T22_ITU]	Local inhibit ITU test timer. The range is 500 to 4294967295. The default is 180000.
T23 ITU [SS7_LKSETTIMERS_T23_ITU]	Remote inhibit ITU test timer. The range is 500 to 4294967295. The default is 180000.
<b>ANSI Specific Timers</b>	
T20 ANSI [SS7_LKSETTIMERS_T20_ANSI]	Local inhibit ANSI test timer. The range is 500 to 4294967295. The default is 90000.
T21 ANSI [SS7_LKSETTIMERS_T21_ANSI]	Remote inhibit ANSI test timer. The range is 500 to 4294967295. The default is 90000.



## 10.5.4 Configuring Links

The 'Links' page allows you to configure SS7 links. These parameters can also be configured using the *ini* file parameter table SS7\_LINK\_TABLE (refer to "SS7 Parameters" on page 141).

➤ **To configure the Links parameters, take these 5 steps:**

1. Open the 'Links' page (**Configuration** tab > **SS7 Configuration** menu > **Links** page item).

**Figure 10-7: Links Page**

2. Select an SS7 link icon that you want to configure.
3. Configure or modify the parameters according to the table below.
4. Click **Create**.
5. To save the changes to flash memory, refer to 'Saving Configuration' in the device's *User's Manual*.

**Table 10-5: SS7 Links Parameters**

Parameter	Description
Link Number [SS7_LINK_INDEX]	Determines the index field for a line. The valid range is 0 to max. signaling links. The default value is 0.
Name [SS7_LINK_NAME]	String name for link parameters The default string is 'LINK'.
Trace [SS7_LINK_TRACE_LEVEL]	Determines the trace level of a signaling link (level 2). The valid range is 0 to 1. The default value is 0.
Variant [SS7_LINK_LAYER2_VARIANT]	Determines the variant (layer 2) of signaling link (TDM). <ul style="list-style-type: none"> <li>▪ [0] = NET_VARIANT_OTHER</li> </ul>

Parameter	Description
	<ul style="list-style-type: none"> <li>▪ <b>[1]</b> ITU-T = NET_VARIANT_ITU (default)</li> <li>▪ <b>[2]</b> ANSI = NET_VARIANT_ANSI</li> <li>▪ <b>[3]</b> CHINA = NET_VARIANT_CHINA</li> </ul>
Operative State <b>[SS7_LINK_OPERATIONAL_STATE]</b>	Determines the operational state of a signaling link. <ul style="list-style-type: none"> <li>▪ <b>[0]</b> Offline = L3_OFFLINE (default)</li> <li>▪ <b>[1]</b> Busy = L3_BUSY</li> <li>▪ <b>[2]</b> In service = L3_INSERTSERVICE</li> </ul>
Layer 2 Type <b>[SS7_LINK_L2_TYPE]</b>	Determines the link layer type - defines level 2 media of signaling link. <ul style="list-style-type: none"> <li>▪ <b>[0]</b> None = SS7_SUBLINK_L2_TYPE_NONE (default)</li> <li>▪ <b>[1]</b> MTP2 = SS7_SUBLINK_L2_TYPE_MTP2</li> <li>▪ <b>[2]</b> M2UA MGC = SS7_SUBLINK_L2_TYPE_M2UA_MGC</li> <li>▪ <b>[3]</b> SAAL = SS7_SUBLINK_L2_TYPE_SAAL</li> </ul>
Layer 3 Type <b>[SS7_LINK_L3_TYPE]</b>	Determines the link high layer type - defines level 3 or L2 high layer of signaling link. <ul style="list-style-type: none"> <li>▪ <b>[0]</b> None = SS7_SUBLINK_L3_TYPE_NONE (default)</li> <li>▪ <b>[1]</b> M2UA SG = SS7_SUBLINK_L3_TYPE_M2UA_SG</li> <li>▪ <b>[2]</b> MTP3 = SS7_SUBLINK_L3_TYPE_MTP3</li> <li>▪ <b>[3]</b> MTP2 Tunneling = SS7_SUBLINK_L3_TYPE_MTP2_TUNNELING</li> </ul>
<b>MTP2 Layer 2</b>	
Trunk Number <b>[SS7_LINK_TRUNK_NUMBER]</b>	Determines the trunk number of a signaling link (TDM). The valid range is 0 to maximum trunk capacity. The default value is 0.
Timeslot Number <b>[SS7_LINK_TIMESLOT_NUMBER]</b>	Determines the time-slot number of a signaling link (TDM). The valid range is 0 to 31. The default value is 16.
MTP2 Attributes Index <b>[SS7_LINK_MTP2_ATTRIBUTES]</b>	Determines the MTP2 attributes of signaling link (TDM). The valid range is 0 to MAX_C7_MTP2_PARAMS_INDEX. The default value is 3.
Congestion Low Watermark <b>[SS7_CONGESTION_LOW_MARK]</b>	Determines the link congestion low mark of signaling link (TDM). The valid range is 0 to 255. The default value is 5.
Congestion High Watermark <b>[SS7_CONGESTION_HIGH_MARK]</b>	Determines the link congestion high mark of signaling link (TDM). The valid range is 0 to 255. The default value is 20.
<b>M2UA MGC Layer 2</b>	
Group ID <b>[SS7_LINK_GROUP_ID]</b>	Determines the group ID (M3UA) of signaling link. The valid range is 0 to 0xFFFF. The default value is 0.
Interface ID <b>[SS7_LINK_M2UA_IF_ID]</b>	Determines the interface ID (M2UA) of signaling link. The valid range is 0 to 4294967295. The default value is 0.
Local Busy <b>[SS7_LINK_MTC_BUSY]</b>	Determines the link local busy indicator – if set, indicates link is busy due to local mtc action. The valid range is 0 to 1. The default value is 0.

### 10.5.5 Configuring SS7 Signaling Nodes

The 'SS7 Signaling Nodes' page allows you to configure SS7 Signaling Nodes.

➤ **To configure the SS7 Signaling Nodes parameters, take these 4 steps:**

1. Open the 'SS7 Signaling Nodes' page (**Configuration** tab > **SS7 Configuration** menu > **SNs** page item).

**Figure 10-8: SS7 Signaling Nodes Page**

2. Configure the parameters according to the table below.
3. Click **Create**.
4. To save the changes to flash memory, refer to 'Saving Configuration' in the device's *User's Manual*.

**Table 10-6: SS7 Signaling Nodes Parameters**

Parameter	Description
Name [SS7_SN_NAME]	String name for SN. The default name is 'SN'.
Variant [SS7_SN_VARIANT]	Variant of signaling node: <ul style="list-style-type: none"> <li>▪ [1] ITU-T (default)</li> <li>▪ [2] ANSI</li> <li>▪ [3] CHINA</li> </ul>

Parameter	Description
Trace [SS7_SN_TRACE_LEVEL]	Trace level of signaling node (level 3). <ul style="list-style-type: none"> <li>▪ <b>[0]</b> 0 (default)</li> <li>▪ <b>[1]</b> 1</li> </ul>
Point Code [SS7_SN_OPC]	Origination (local) point-code of signaling node. The range is 0 to 4294967295. The default is 0.
Network Indicator [SS7_SN_NI]	Network Indicator of signaling node. <ul style="list-style-type: none"> <li>▪ <b>[0]</b> International (default)</li> <li>▪ <b>[1]</b> International(Spare)</li> <li>▪ <b>[2]</b> National</li> <li>▪ <b>[3]</b> National(Spare)</li> </ul>
STP Function [SS7_SN_SP_STP]	Routing function of signaling node. <ul style="list-style-type: none"> <li>▪ <b>[0]</b> SP (default)</li> <li>▪ <b>[1]</b> STP</li> </ul>
SN Timers Index [SS7_SN_TIMERS_INDEX]	Index of SNTimers tables used for this signaling node. The range is 0 to (MTP3_SN_TIMER_SETS-1). The default is 0.
<b>Layer 4 Applications</b>	
ISUP [SS7_SN_ISUP_APP]	Level 4 application that handles ISUP traffic for this signaling node. <ul style="list-style-type: none"> <li>▪ <b>[0]</b> None = NIL (default)</li> <li>▪ <b>[4]</b> UAL = UAL</li> </ul>
SCCP [SS7_SN_SCCP_APP]	Level 4 application that handles SCCP traffic for this signaling node. <ul style="list-style-type: none"> <li>▪ <b>[0]</b> None = NIL (default)</li> <li>▪ <b>[4]</b> UAL = UAL</li> </ul>
BISUP [SS7_SN_BISUP_APP]	Level 4 application that handles BISUP traffic for this signaling node. <ul style="list-style-type: none"> <li>▪ <b>[0]</b> None = NIL (default)</li> <li>▪ <b>[4]</b> UAL = UAL</li> </ul>
TUP [SS7_SN_TUP_APP]	Level 4 application that handles TUP traffic for this signaling node. <ul style="list-style-type: none"> <li>▪ <b>[0]</b> None = NIL (default)</li> <li>▪ <b>[4]</b> UAL = UAL</li> </ul>
BICC	Name of the Level 4 application that handles BICC traffic for this signaling node. <ul style="list-style-type: none"> <li>▪ <b>[0]</b> None = NIL (default)</li> <li>▪ <b>[4]</b> UAL = UAL</li> </ul>
ALCAP [SS7_SN_ALCAP_APP]	Level 4 application that handles ALCAP traffic for this signaling node. <ul style="list-style-type: none"> <li>▪ <b>[0]</b> None = NIL (default)</li> <li>▪ <b>[4]</b> UAL = UAL</li> <li>▪ <b>[5]</b> ALCAP</li> </ul>

## 10.5.6 Configuring MTP3 Redundancy

SS7 MTP3 is the network layer of SS7. It defines and manages the behavior of signaling nodes (point-codes). Each signaling node may use several signaling links to communicate with the rest of the SS7 network. AudioCodes has a working MTP3 layer in a single CPU. However, it is also important to manage a point code that is distributed over several CPUs for the following reasons:

- Eliminating a 'single point of failure' at the network layer. Since MTP3 runs on a single CPU, failure of a device causes isolation of higher layer applications such as a softswitch.
- Increasing the number of DPCs that can be connected directly to one single point code, since devices that are located physically in different locations have the same point code number

There are two operating modes for MTP3:

- Regular mode: all links are handled by a single CPU.
- Redundancy mode: two devices (i.e. CPUs) may participate in a distributed point-code

### ➤ To configure MTP3 redundancy parameters, take these 4 steps:

1. Open the 'MTP3 Redundancy Configuration' page (**Configuration** tab > **SS7 Configuration** menu > **MTP3 Redundancy Configuration** page item).

**Figure 10-9: MTP3 Redundancy Configuration Page**

⚡ Redundancy Mode	0
⚡ Board Number	0
⚡ Keep-Alive Window	2
⚡ Keep-Alive Interval [Sec]	1

2. Configure the parameters according to the table below.
3. Click **Submit**.
4. To save the changes to flash memory, refer to 'Saving Configuration' in the device's *User's Manual*.

**Table 10-7: MTP3 Redundancy Parameters**

Parameter	Description
Redundancy Mode [SS7MTP3RdcyMode]	Determines whether or not configured for SS7 MTP3 redundancy mode. <ul style="list-style-type: none"> <li>▪ 0 = Disabled, i.e., no redundancy (default)</li> <li>▪ 1 = Enabled</li> </ul>
Board Number [SS7MTP3RdcyBoardNum]	Device number for SS7 MTP3 redundancy mode. Each device is assigned a unique number, since they all share a common redundancy table. The default is 0.

Parameter	Description
Keep-Alive Window [SS7MTP3RdcyKeepAliveWindow]	Defines the redundancy X-link keep-alive tolerance window. This is the X-link between devices in SS7 MTP3-User Adaptation Layer redundancy mode. The range is 1 to 15. the default is 2.
Keep-Alive Interval [Sec] [SS7MTP3RdcyKeepAliveInterval]	Defines the redundancy X-link keep-alive interval (in seconds). This is the X-link between devices in SS7 MTP3-User Adaptation Layer redundancy mode. The range is 0 to 100, where 0 denotes that no keep-alive mechanism is activated. The default is 1.

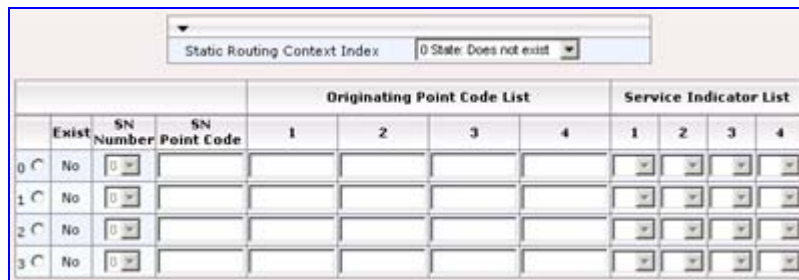
## 10.5.7 Configuring Static Routing Context

The 'Static Routing Context Table' page allows you to configure SS7 Signaling Nodes.

➤ **To configure the Static Routing Context parameters, take these 4 steps:**

1. Open the 'Static Routing Context Table' page (**Configuration** tab > **SS7 Configuration** menu > **Static Routing Context** page item).

**Figure 10-10: Static Routing Context Table Page**



2. Configure the parameters according to the table below.
3. Click **Create**; the 'Exist' field displays "Yes".
4. To save the changes to flash memory, refer to 'Saving Configuration' in the device's *User's Manual*.

**Table 10-8: SS7 Static Routing Context Parameters**

Parameter	Description
[SS7_RC_INDEX]	Routing Context index line of the table. The range is 0 to 15. The default is 0.
SN Number [SS7_RC_INNER_INDEX]	Second Index Field for line. The range is 0 to 3. The default is 0.
SN Point Code [SS7_RC_SN_INDEX]	Specifies the M3UA Routing Context DPC SN-Index. The range is 0 to 1. The default is 0.
Originating Point Code List [SS7_RC_OPC1], [SS7_RC_OPC2], [SS7_RC_OPC3], [SS7_RC_OPC4]	Specifies the first, second, third, and fourth elements in M3UA Routing Context OPC List. The range is -1, 0 to 0xFFFFFFFF. The default is -1.

Parameter	Description
Service Indicator List [SS7_RC_SI1], [SS7_RC_SI1], [SS7_RC_SI1], [SS7_RC_SI1]	Specifies the first, second, third, and fourth elements in M3UA Routing Context SI List. The range is -1, 0 to 15. The default is -1.

### 10.5.8 Configuring Sigtran Group IDs

The 'Sigtran Group IDs' page allows you to configure Signaling Transport (Sigtran) Group IDs. These parameters can also be configured using the *ini* file parameter table SS7\_SIG\_IF\_GROUP\_TABLE (refer to "SS7 Parameters" on page 141).

➤ **To configure the Sigtran Group IDs parameters, take these 4 steps:**

1. Open the 'Sigtran Group IDs' page (**Configuration** tab > **Sigtran Configuration** menu > **Sigtran Group IDs** page item).

**Figure 10-11: Sigtran Group IDs Page**

Group Number: 0 State: Does not exist

ASP Status: Invalid ASP Status

Sigtran Group does not exist

Group ID	0
Rdcy Board Number	0
UAL Group Function	SG NAT
Group Layer	M2UA
Group Traffic Mode	Override
Group Minimal ASP Number	1
Group Behavior Field	0
Group Local SCTP Port	0
Group Network Variant	ITU
Inbound Streams Number	2
Outbound Streams Number	2
Group Destination SCTP IP	0.0.0.0
Group Destination SCTP Port	65534

Interface Group Timers

Tr - Group Recovery Timer	2000
Tr - Group Acknowledge Timer	2000
Tr - Group Heartbeat Timer	30000

2. Configure the parameters according to the table below.

3. Click **Create**.
4. To save the changes to flash memory, refer to 'Saving Configuration' in the device's *User's Manual*.

**Table 10-9: Sigtran Group IDs Parameters**

Parameter	Description
Group Number [SS7_SIG_IF_GR_INDEX]	Indicates the interface group index for a line. The valid range is 0 to 7.
ASP Status	Sigtran group Application Server Process (ASP) status (read-only).
Group ID [SS7_IF_GR_ID]	Determines the SS7 SIGTRAN interface group index, for a line. The valid range is 0 to 65535. The default value is 65535.
Rdcy Board Number [RdcyBoardNum]	Specifies the Sigtran group redundancy board number.
UAL Group Number [SS7_SIG_SG_MGC]	Determines the SS7 SIGTRAN interface group Signaling Gateway (SG) and Media Gateway Controller (MGC) option. The valid range is 77 (MGC) and 83 (SG). The default value is 83.
Group Layer [SS7_SIG_LAYER]	Determines the SIGTRAN group layer (IUA/M2UA/M3UA). Choose either: <ul style="list-style-type: none"> <li>▪ [0] = no_layer (default)</li> <li>▪ [1] IUA = iua</li> <li>▪ [2] M2UA = m2ua</li> <li>▪ [3] M3UA = m3ua</li> <li>▪ [4] M2Tunnel = m2tunnel</li> <li>▪ [6] DUA = V5ua</li> </ul>
Group Traffic Mode [SS7_SIG_TRAF_MODE]	Determines the SS7 SIGTRAN interface group traffic mode. The valid range is 1 to 3. The default value is 1.
Group Minimal ASP Number [SS7_SIG_MIN_ASP]	Determines the SIGTRAN group minimal Application Server Process (ASP) number (minimum = 1). The valid range is 1 to 10. The default value is 1.
Group Behavior Field [SS7_SIG_BEHAVIOUR]	Determines the SIGTRAN group behavior bit. The valid range is 0 to 4294967294. The default value is 0.
Group Local SCTP Port [SS7_LOCAL_SCTP_PORT]	Determines the SIGTRAN group SCTP port. The valid range is 0 to 65534. The default value is 65534.
Group Network Variant [SS7_SIG_NETWORK]	Determines the SIGTRAN group Network (ITU, ANSI, CHINA). The valid range is 1 to 3. The default value is 1.
Inbound Streams Number [SS7_MGC_MX_IN_STREAM]	Determines the SIGTRAN group maximum inbound stream. The valid range is 2 to 65534. The default value is 2.
Outbound Streams Number [SS7_MGC_NUM_OUT_STREAM]	Determines the SIGTRAN group's number of outbound streams. The valid range is 2 to 65534. The default value is 2.
Group Destination SCTP Port [SS7_DEST_SCTP_PORT]	Determines the SIGTRAN group destination SCTP port. The valid range is 0 to 65534. The default value is 65534.
Group Destination SCTP IP [SS7_DEST_IP]	Determines the SIGTRAN group destination IP address The valid range is 0 to 4294967294. The default value is 0.



Parameter	Description
<b>Interface Group Timers</b>	
Tr - Group Recovery Timer <b>[SS7_SIG_T_REC]</b>	Determines the SIGTRAN group T recovery. The valid range is 0 to 10000000. The default value is 2000.
Ta - Group Acknowledge Timer <b>[SS7_SIG_T_ACK]</b>	Determines the SIGTRAN group T Ack (in msec). The valid range is 0 to 10000000. The default value is 2000.
Th - Group Heartbeat Timer <b>[SS7_SIG_T_HB]</b>	Determines the SIGTRAN group T Hb (in msec). The valid range is 0 to 10000000. The default value is 2000.

### 10.5.9 Configuring Sigtran Interface IDs

The 'Sigtran Interface IDs' page allows you to configure the Sigtran interface IDs. These parameters can also be configured using the *ini* file parameter table SS7\_SIG\_INT\_ID\_TABLE (refer to "SS7 Parameters" on page 141).

➤ **To configure the Sigtran Interface IDs parameters, take these 4 steps:**

1. Open the 'Sigtran Interface IDs' page (**Configuration** tab > **Sigtran Configuration** menu > **Sigtran Interface IDs** page item).

**Figure 10-12: Sigtran Interface IDs Page**

Interface Number: 0 State: Does not exist

Sigtran Interface does not exist

Interface ID	0
Interface ID Name	
Owner Group	0
Sigtran Layer Type	M2UA
IF ID NAI	0

2. Configure the parameters according to the table below.
3. Click **Create**.
4. To save the changes to flash memory, refer to 'Saving Configuration' in the device's *User's Manual*.

**Table 10-10: Sigtran Interface IDs Parameters**

Parameter	Description
Interface Number <b>[SS7_SIG_IF_ID_INDEX]</b>	Determines the SS7 interface ID index, for a line. The valid range is 0 to 15. The default value is 1.
Interface ID <b>[SS7_SIG_IF_ID_VALUE]</b>	Determines the SIGTRAN interface ID value. The valid range is 0 to 4294967294. The default value is 0.
Interface ID Name <b>[SS7_SIG_IF_ID_NAME]</b>	Determines the SIGTRAN interface ID (text string). The default string is 'INT_ID'.
Owner Group <b>[SS7_SIG_IF_ID_OWNER_GROUP]</b>	Determines the SIGTRAN interface ID owner group. The valid range 0 to 65534. The default value is 0.
Sigtran Layer Type <b>[SS7_SIG_IF_ID_LAYER]</b>	Determines the SIGTRAN group layer (IUA/M2UA/M3UA). <ul style="list-style-type: none"> <li>▪ <b>[0]</b> None = no layer (default)</li> <li>▪ <b>[1]</b> IUA</li> <li>▪ <b>[2]</b> M2UA</li> <li>▪ <b>[4]</b> MTP2 Tunnel</li> <li>▪ <b>[6]</b> DUA</li> </ul>
IF ID NAI <b>[SS7_SIG_IF_ID_NAI]</b>	Determines the SIGTRAN interface ID NAI. The valid range 0 to 65534. The default value is 65534.

# 11 Accessory Programs and Tools

The accessory programs and tools shipped with your AudioCodes device provide you with user-friendly interfaces that enhance device usability and facilitates your transition to the new VoIP infrastructure. The following proprietary applications are available:

- **AudioCodes BootP / TFTP Server** configuration utility (refer to "BootP/TFTP Configuration Utility" on page 167)
- **AudioCodes TrunkPack Downloadable Conversion Utility** (DConvert) (refer to "TrunkPack Downloadable Conversion Utility" on page 180)
- **AudioCodes Call Progress Tones Wizard** (applicable only to Analog devices) (refer to "Call Progress Tones Wizard" on page 190)

## 11.1 BootP/TFTP Server Configuration Utility

The proprietary BootP/TFTP Server utility enables you to easily configure and provision AudioCodes devices. Similar to third-party BootP/TFTP utilities (which are also supported) the BootP/TFTP Server utility can be installed on Windows™ 98 or Windows™ NT/2000/XP/Vista. The BootP/TFTP utility enables remote reset of the device to trigger the initialization procedure (BootP and TFTP). It contains BootP and TFTP utilities with specific adaptations to our requirements.

### 11.1.1 When to Use the BootP/TFTP

The BootP/TFTP utility can be used as an alternative means for initializing the device. Initialization provides the device with an IP address, subnet mask, and default Gateway IP address. The tool also loads default software files, *ini* file, and other configuration files. BootP/TFTP Tool can also be used to restore a device to its initial configuration such as in the following instances:

- The IP address of the device is unknown.
- The Web browser has been inadvertently turned off.
- The Web browser password has been lost.
- The device has encountered a fault that cannot be recovered using the Web browser.
- The device has encountered a fatal error that cannot be recovered using the Web browser or the hardware reset button.



**Note:** The BootP/TFTP utility is typically used to configure the device's initial parameters. Once this information has been provided, the BootP/TFTP utility is no longer needed. All parameters are stored in non-volatile memory and used when the BootP/TFTP is not accessible.

### 11.1.2 An Overview of BootP

Bootstrap Protocol (BootP) is a protocol defined in RFC 951 and RFC 1542 that enables an Internet device to obtain its own IP address and the IP address of a BootP on the network. In addition, it's also used to obtain the files required for operating the device.

When a device uses BootP and powers up, the device broadcasts a BootRequest message on the network. A BootP on the network receives this message and generates a BootReply. The BootReply indicates the IP address that must be used by the device and specifies an IP address from which the device may load configuration files using Trivial File Transfer Protocol (TFTP) described in RFC 906 and RFC 1350.

### 11.1.3 Key Features

The BootP/TFTP utility offers the following key features:

- Internal BootP supporting hundreds of entities
- Internal TFTP
- Contains all required data for AudioCodes' products in pre-defined format
- Provides a TFTP address, enabling network separation of TFTP and BootP utilities
- Tools to backup and restore the local database
- Templates
- User-defined names for each entity
- Option for changing MAC address
- Protection against entering faulty information
- Remote reset
- Unicast BootP response
- User-initiated BootP respond for remote provisioning over WAN
- Filtered display of BootP requests
- Location of other BootP utilities that contain the same MAC entity
- Common log window for both BootP and TFTP sessions
- Runs on Windows™ 98/NT/2000/XP/Vista

### 11.1.4 Specifications

The BootP/TFTP utility provides the following specifications:

- BootP standards: RFC 951 and RFC 1542
- TFTP standards: RFC 1350 and RFC 906
- Operating Systems: Windows™ 98, Windows™ NT, Windows™ 2000 and Windows™ XP
- Maximum number of MAC entries: 200

### 11.1.5 Installation

The following procedures describe how to install and run the AudioCodes BootP / TFTP Server application.

➤ **To install the BootP/TFTP on your computer, take these 2 steps:**

1. Locate the 'BootP & TFTP Configuration utility' folder on the supplied CD-ROM, and then click the *Setup.exe* file.
2. Follow the prompts from the installation wizard to complete the installation.

➤ **To start BootP/TFTP, take these 2 steps:**

1. From the **Start** menu on your computer, point to **Programs**, point to **BootP**, and then click **bootp**.
2. The first time you run the BootP/TFTP utility, the program prompts you to set user preferences. Refer to "Setting the Preferences" on page 172 for information on setting the preferences.

### 11.1.6 Loading the cmp File - Booting the Device

Once the BootP is running and the preferences are defined (refer to "Setting the Preferences" on page 172), you need to configure the network configuration information and initialization file names. Each device is identified by a MAC address. For information on how to configure (add, delete and edit) devices, refer to "Configuring the BootP Clients" on page 174.

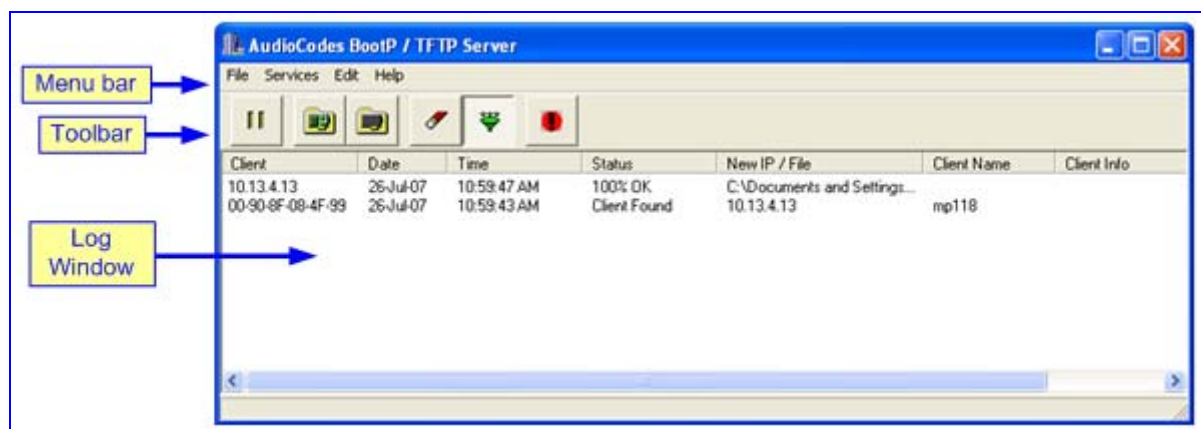
➤ **To load the software and configuration files, take these 4 steps:**

1. Create a folder on your computer that contains all the software and configuration files that are needed as part of the TFTP process.
2. Define the BootP and TFTP preferences (refer to "Setting the Preferences" on page 172).
3. Add a new client for the device that you want to initialize by BootP (refer to "Adding Clients" on page 177).
4. Reset the device, either physically or remotely. This causes the device to use BootP to access the network and configuration information.

### 11.1.7 BootP/TFTP Application User Interface







The figure below shows the main window of the BootP/TFTP utility.

**Figure 11-1: Main Screen**

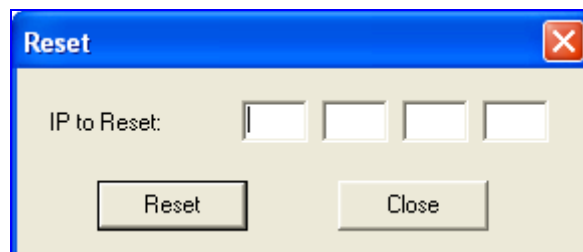


## 11.1.8 Toolbar Buttons in the Main Screen

The buttons on the toolbar are described in the table below:

Button	Name	Description
	<b>Pause</b>	Pauses the BootP / TFTP utility so that no replies are sent to BootP requests. Click the button again to restart the BootP utility so that it responds to all BootP requests. The <b>Pause</b> button provides a depressed graphic when the feature is active.
	<b>Edit Clients</b>	Opens the Client Configuration window that enables you to enter configuration information for each supported device. Details on the Client Configuration window are provided in "Configuring the BootP Clients" on page 174.
	<b>Edit Templates</b>	Opens the Templates window that enables you to create or edit standard templates. These templates can be used when configuring new clients that share most of the settings. Details on the Templates window are provided in "Managing Client Templates" on page 179.
	<b>Clear Log</b>	Clears all entries from the Log window portion of the main window. Details on the Log window are provided in "Log Window" on page 170.
	<b>Filter Unknown Clients</b>	Prevents the BootP / TFTP utility from logging BootP requests received from disabled clients or from clients which do not have entries in the Clients table.
	<b>Reset</b>	Opens the Reset window where you enter an IP address requests for a device that you want to reset. Refer to the figure below.

**Figure 11-2: Reset Screen**



When a device resets, it first sends a BootRequest. Therefore, the Reset button can be used to force a BootP session with a device without needing to power cycle the device. As with any BootP session, the computer running the BootP tool must be located in the same subnet as the controlled device.

## 11.1.9 Log Window

The Log window (refer to "BootP/TFTP Application User Interface" on page 169) records all BootP request and BootP reply transactions, as well as TFTP transactions. For each transaction, the Log window displays the following information:

- **Client:** address of the device, which is the MAC address of the client for BootP transactions or the IP address of the client for TFTP transactions.
- **Date:** date of the transaction, based on the internal calendar of the computer.
- **Time:** time of day of the transaction, based on the internal clock of the computer.

- **Status:** status of the transaction:
  - *Client Not Found:* A BootRequest was received but there is no matching client entry in the BootP / TFTP utility.
  - *Client Found:* A BootRequest was received and there is a matching client entry in the BootP / TFTP utility. A BootReply is sent.
  - *Client's MAC Changed:* There is a client entered for this IP address but with a different MAC address.
  - *Client Disabled:* A BootRequest was received and there is a matching client entry in the BootP / TFTP utility, but this entry is disabled.
  - *Listed At:* Another BootP utility is listed as supporting a particular client when the **Test Selected Client** button is clicked (for details on Testing a client, refer to "Testing the Client" on page 178).
  - *Download Status:* Progress of a TFTP load to a client, shown in %.
- **New IP / File:** IP address applied to the client as a result of the BootP transaction as well as the file name and path of a file transfer for a TFTP transaction.
- **Client Name:** client name as configured for that client in the Client Configuration window.

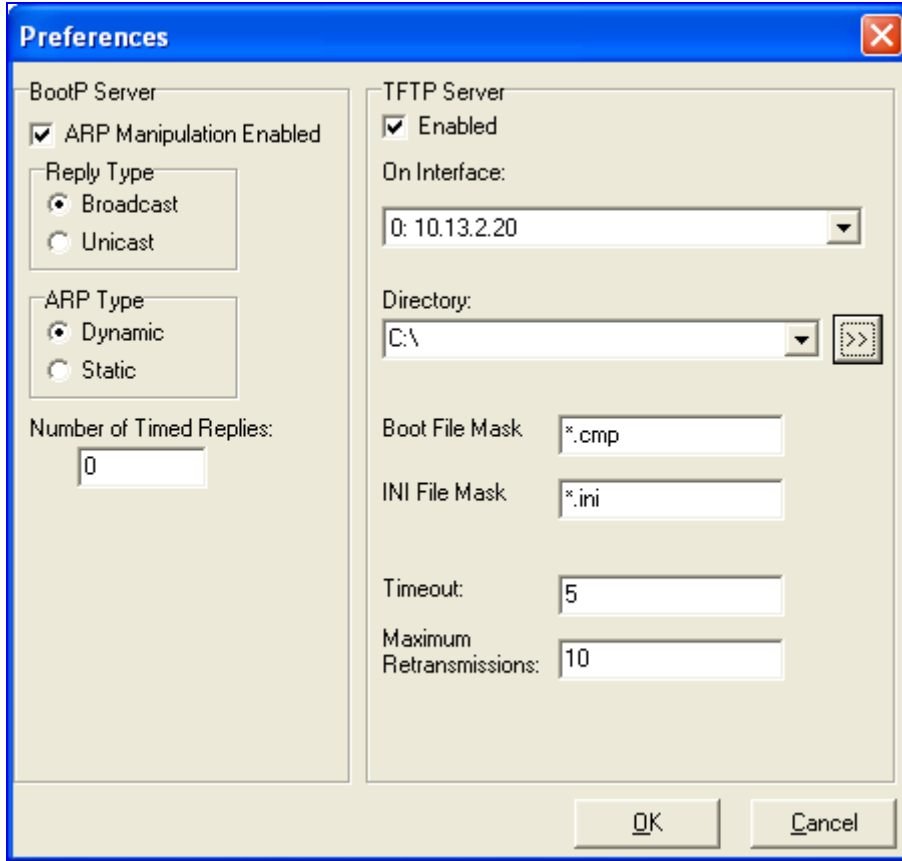
Right-click a row in the Log window to open a pop-up window with the following options:

- **Reset:** Selecting this option results in a reset command being sent to the client device. The program searches its database for the MAC address indicated in the line. If the client is found in that database, the program adds the client MAC address to the Address Resolution Protocol (ARP) table of the computer. The program then sends a reset command to the client. This enables a reset to be sent without knowing the current IP address of the client as long as the computer sending the reset is on the same subnet.  
**Note:** To use reset, you must have administrator privileges on the computer. Attempting to perform this type of reset without administrator privileges on the computer results in an error message. **ARP Manipulation Enable** must also be turned on in the Preferences window.
- **View Client:** Selecting this option, or double clicking on the line in the log window, opens the Client Configuration window. If the MAC address indicated on the line exists in the client database, it is highlighted. If the address is not in the client database, a new client is added with the MAC address filled out. You can enter data in the remaining fields to create a new client entry for that client.

### 11.1.10 Setting the Preferences

The Preferences window (**Edit** menu > **Preferences**), as shown below is used to configure the BootP / TFTP parameters.

**Figure 11-3: Preferences Screen**



The Preferences window is divided into two main sections: BootP Server and TFTP Server.

**BootP Server:**

- ☒ ARP Manipulation Enabled
- Reply Type:
  - ☒ Broadcast
  - ☐ Unicast
- ARP Type:
  - ☒ Dynamic
  - ☐ Static
- Number of Timed Replies:

**TFTP Server:**

- ☒ Enabled
- On Interface:
- Directory:
- Boot File Mask:
- INI File Mask:
- Timeout:
- Maximum Retransmissions:

Buttons:

#### 11.1.10.1 BootP Preferences

Address Resolution Protocol (ARP) is the method used by all Internet devices to determine the link layer address such as the Ethernet MAC address to route Datagrams to devices that are on the same subnet. When ARP Manipulation is enabled, the BootP/TFTP utility creates an ARP cache entry on your computer when it receives a BootP BootRequest from the device. Your computer uses this information to send messages to the device without using ARP again. This is particularly useful when the device does not yet have an IP address and, therefore, cannot respond to an ARP. Because this feature creates an entry in the computer ARP cache, administrator privileges are required. If the computer is not set to allow administrator privileges, ARP Manipulation cannot be enabled.

- **ARP Manipulation Enabled:** Enable ARP Manipulation to remotely reset a device that does not yet have a valid IP address.

If ARP Manipulation is enabled, the following two option groups are available:

- **Reply Type:** Reply to a BootRequest can be either **Broadcast** or **Unicast**. The default is **Broadcast** and for the reply to be set to **Unicast**, **ARP Manipulation** must first be enabled. This then enables the BootP / TFTP utility to find the MAC address for the client in the ARP cache so that it can send a message directly to the requesting device. Typically, this setting can be left at **Broadcast**.



- **ARP Type:** The type of entry (**Dynamic** or **Static**) made in the ARP cache on the computer once **ARP Manipulation** is enabled. **Dynamic** entries (default) expire after a period of time, keeping the cache clean so that old entries do not consume computer resources. Static entries do not expire.
- **Number of Timed Replies:** This is useful for communicating to devices that are located behind a firewall that would block their BootRequest messages from getting through to the computer that is running BootP / TFTP. You can set this value to any whole digit. Once set, BootP / TFTP can send that number of BootReply messages to the destination immediately after you send a remote reset to a device at a valid IP address. This enables the replies to pass through to the device even if the BootRequest is blocked by the firewall. To turn off this feature, set the **Number of Timed Replies** to 0.

### 11.1.10.2 TFTP Preferences

The Preferences window allows you to define the following TFTP preferences:

- **Enabled:** Select this check box to enable the TFTP functionality of the BootP/TFTP utility. If you want to use another TFTP application other than the one included with the BootP/TFTP utility, clear this check box.
- **On Interface:** From the drop-down list, select the network interface available on your PC that you want to use for the TFTP server. (Typically, only one interface is listed.)
- **Directory:** This option is enabled only when TFTP is enabled. Specify the folder that contains the files for the TFTP utility to manage (*cmp*, *ini*, Call Progress Tones, etc.).
- **Boot File Mask:** Specify the file extension used by the TFTP utility for the boot file that is included in the BootReply message. This is the file that contains the device's software and typically appears as *cmp*.
- **INI File Mask:** Specify the file extension used by the TFTP utility for the configuration file that is included in the BootReply message. This is the file that contains device's configuration parameters and typically appears as *ini*.
- **Timeout:** Specifies the number of seconds that the TFTP utility waits before retransmitting TFTP messages. The default value is 30, however, it is recommended to set it to 50 (the more congested your network, the higher you should set this value).
- **Maximum Retransmissions:** Specifies the number of times that the TFTP utility tries to resend messages after timeout. This can be left at the default value of 10 (the more congested your network, the higher you should set this value).



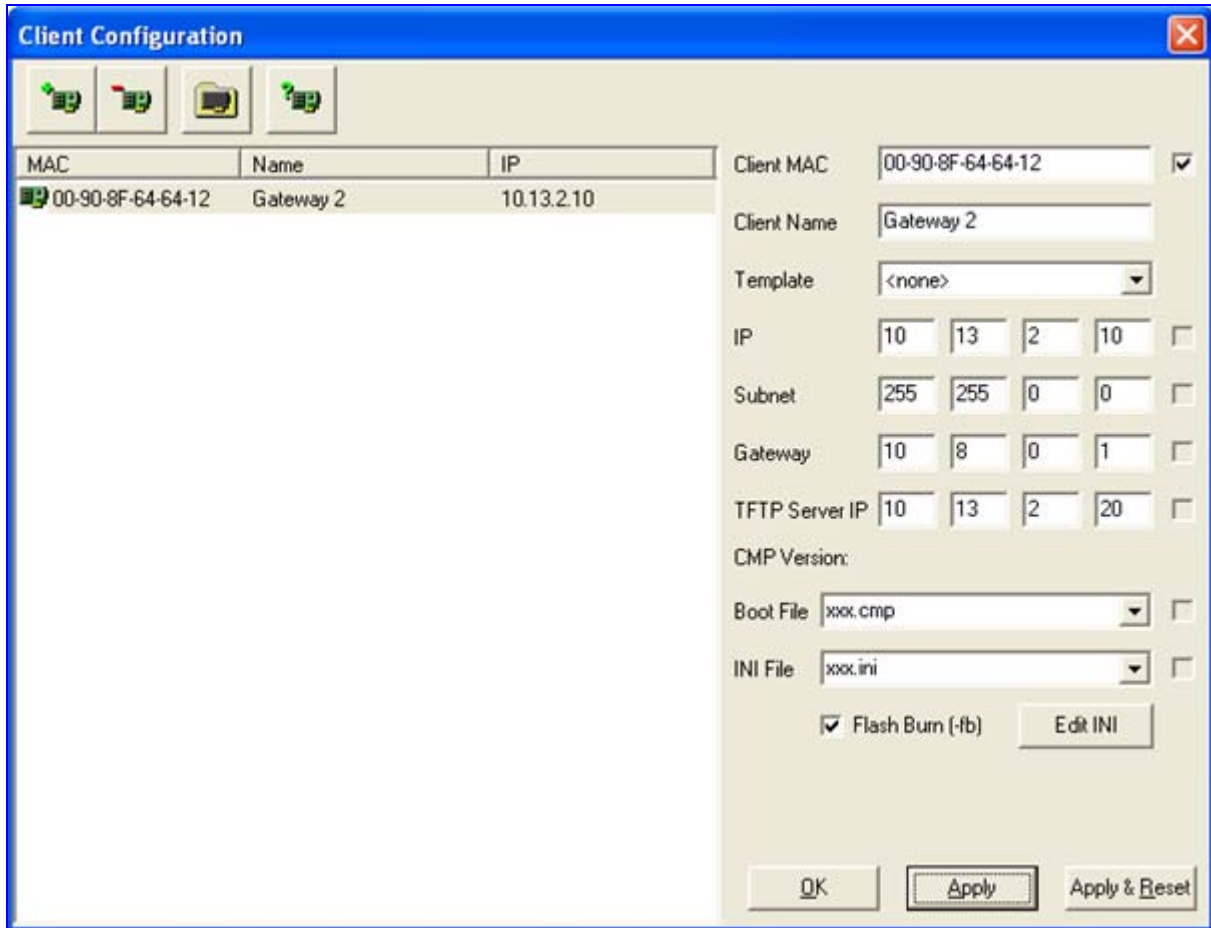
#### Notes:

- The 'On Interface' field is automatically set to the IP address of the PC on which the BootP/TFTP Server program is running.
- When upgrading the device between major software releases (e.g., from 5.2 to 5.4), it is recommended to set the maximum retransmissions to 20.

### 11.1.11 Configuring the BootP Clients

The Client Configuration window (**Services** menu > **Client**), as shown below is used to define the parameters for each specific device.

**Figure 11-4: BootP Client Configuration Screen**



The screenshot shows the 'Client Configuration' window. On the left, there is a table with columns 'MAC', 'Name', and 'IP'. The first row shows '00-90-8F-64-64-12', 'Gateway 2', and '10.13.2.10'. On the right, there are configuration fields for the selected client. The 'Client MAC' field is '00-90-8F-64-64-12' with a checked checkbox. The 'Client Name' field is 'Gateway 2'. The 'Template' dropdown is set to '<none>'. The 'IP' field is '10.13.2.10' with an unchecked checkbox. The 'Subnet' field is '255.255.0.0' with an unchecked checkbox. The 'Gateway' field is '10.8.0.1' with an unchecked checkbox. The 'TFTP Server IP' field is '10.13.2.20' with an unchecked checkbox. The 'CMP Version' dropdown is empty. The 'Boot File' dropdown is 'xxx.cmp' with an unchecked checkbox. The 'INI File' dropdown is 'xxx.ini' with an unchecked checkbox. There is a checked checkbox for 'Flash Burn (-fb)' and an 'Edit INI' button. At the bottom, there are 'OK', 'Apply', and 'Apply & Reset' buttons.

MAC	Name	IP
00-90-8F-64-64-12	Gateway 2	10.13.2.10

Client MAC: 00-90-8F-64-64-12 ☒

Client Name: Gateway 2

Template: <none>

IP: 10.13.2.10 ☐

Subnet: 255.255.0.0 ☐

Gateway: 10.8.0.1 ☐

TFTP Server IP: 10.13.2.20 ☐

CMP Version:

Boot File: xxx.cmp ☐

INI File: xxx.ini ☐

☒ Flash Burn (-fb)

#### 11.1.11.1 Client Parameters

Client parameters are listed on the right side of the Client Configuration window.

- **Client MAC:** Used by BootP to identify the device. The MAC address of the device is printed on a label located on the device hardware. Enter the Ethernet MAC address of the device in this field. Select the check box to the right of this field to enable this particular client in the BootP tool (if the client is disabled, no replies are sent to BootP requests).  
**Note:** When the MAC address of an existing client is edited, a new client is added with the same parameters as the previous client.
- **Client Name:** Enter a descriptive name for the client so that it is easy to identify to which device the record refers. For example, this name could refer to the location of the device.
- **Template:** From the drop-down list, select one of the templates that you configured (refer to "Managing Client Templates" on page 179). This applies the parameters from that template to the remaining fields. Parameter values that are applied by the template are indicated by a check mark in the check box to the right of that parameter. Clear this check box if you want to enter a different value. If templates are not used, the check box is colored gray and is not selectable.

- **IP:** Enter the IP address you want to assign to the device. Use the normal dotted-decimal notation.
- **Subnet:** Enter the subnet mask you want to apply to the device. Use the normal dotted-decimal notation. Ensure that the subnet mask is correct. If the address is incorrect, the device may not function until the entry is corrected and a BootP reset is applied.
- **Gateway:** Enter the IP address for the data network gateway used on this subnet that you want to apply to the device. The data network gateway is a device such as a router that is used in the data network to interface this subnet to the rest of the enterprise network.
- **TFTP Server IP:** IP address of the TFTP utility that is used for file transfer of software and initialization files to the device. When creating a new client, this field is populated with the IP address used by the BootP/TFTP utility. If a different TFTP utility is used, change the IP address in this field to the IP address used by the other utility.
- **Boot File:** Specify the file name for the software file (*cmp*) that is loaded by the TFTP utility to the device after the device receives the BootReply message. The software file is located in the TFTP utility directory that is specified in the Preferences window. The software file can be followed by command line switches. For information on available command line switches, refer to "Using Command Line Switches" on page 175.

**Notes:**

- Once the software file loads to the device, the device begins operating from that software. To save this software to non-volatile memory (only the *cmp* file, i.e., the compressed firmware file can be burned to the device's flash memory), the -fb flag must be added to the end of the file name or the **Flash Burn** check box must be selected. If the file is not saved, the device reverts to the previous software version after the next reset.
- The **Boot File** field can contain up to two file names: *cmp* file name for loading the application image and the *ini* file name for device provisioning. To use both file names, use the ';' separator (without blank spaces) between the xxx.cmp and the yyy.ini files (e.g., *ram.cmp;SIPgw.ini*).

- **INI File:** Specify the configuration *ini* file that the device uses to configure its various settings. Enter the name of the file, which is loaded by the TFTP utility to the device after it receives the BootReply message. The *ini* file is located in the TFTP utility directory that is specified in the Preferences window.

### 11.1.11.2 Using Command Line Switches

You can add command line switches in the field 'Boot File'.

➤ **To use a Command Line Switch, take these 4 steps:**

1. In the field 'Boot File', leave as is the defined file name (e.g., *ramxxx.cmp*).
2. Place your cursor after *cmp*.
3. Press the space bar.
4. Type in the switch you require.

Examples:

- 'ramxxx.cmp -fb' to burn flash memory.
- 'ramxxx.cmp -fb -em 4' to burn flash memory and for Ethernet Mode 4 (auto-negotiate).

The table below lists and describes the switches that are available:

**Table 11-1: Command Line Switch Descriptions**


Switch	Description	
<b>-fb</b>	Burns the <i>cmp</i> file (software file) to the flash memory. <b>Note:</b> Instead of using this switch, you can simply select the <b>Flash Burn</b> check box.	
<b>-em #</b>	Defines the Ethernet mode: <ul style="list-style-type: none"> <li>■ 0 = 10Base-T half-duplex (Not applicable to 3000 Series)</li> <li>■ 1 = 10Base-T full-duplex</li> <li>■ 2 = 100Base-TX half-duplex (Not applicable to 3000 Series)</li> <li>■ 3 = 100Base-TX full-duplex</li> <li>■ 4 = auto-negotiate (default)</li> </ul> For detailed information on Ethernet interface configuration, refer to the device's <i>User's Manual</i> .	
<b>-br</b>	This parameter is used to perform the following:	
	Defines the number of BootP requests the device sends during startup. The device stops sending BootP requests when either BootP reply is received or number of retries is reached. <ul style="list-style-type: none"> <li>■ 1 = 1 BootP retry, 1 second</li> <li>■ 2 = 2 BootP retries, 3 seconds</li> <li>■ 3 = 3 BootP retries, 6 seconds</li> <li>■ 4 = 10 BootP retries, 30 seconds</li> <li>■ 5 = 20 BootP retries, 60 seconds</li> <li>■ 6 = 40 BootP retries, 120 seconds</li> <li>■ 7 = 100 BootP retries, 300 seconds</li> <li>■ 15 = BootP retries indefinitely</li> </ul>	Defines the number of DHCP packets the device sends. After all packets are sent, if there's still no reply, the device loads from flash. <ul style="list-style-type: none"> <li>■ 1 = 4 DHCP packets</li> <li>■ 2 = 5 DHCP packets</li> <li>■ 3 = 6 DHCP packets (default)</li> <li>■ 4 = 7 DHCP packets</li> <li>■ 5 = 8 DHCP packets</li> <li>■ 6 = 9 DHCP packets</li> <li>■ 7 = 10 DHCP packets</li> <li>■ 15 = 18 DHCP packets</li> </ul>
	<b>Note:</b> This switch takes effect only from the next device reset.	
<b>-bd</b>	Defines the interval between the device's startup and the first BootP/DHCP request that is issued by the device (BootP delays). The switch only takes effect from the next reset of the device. <ul style="list-style-type: none"> <li>■ 1 = 1 second delay (default)</li> <li>■ 2 = 10 second delay</li> <li>■ 3 = 30 second delay</li> <li>■ 4 = 60 second delay</li> <li>■ 5 = 120 second delay</li> </ul>	

Switch	Description
<b>-bs</b>	<ul style="list-style-type: none"> <li>▪ <b>-bs 1</b>: enables the Selective BootP mechanism</li> <li>▪ <b>-bs 0</b>: disables the Selective BootP mechanism</li> </ul> <p>The Selective BootP mechanism enables the device's integral BootP client to filter unsolicited BootP/DHCP replies (accepts only BootP replies that contain the text 'AUDC' in the Vendor Specific Information field). This option is useful in environments where enterprise BootP/DHCP servers provide undesired responses to the device's BootP requests.</p>
<b>-be</b>	<p>Use <b>-be 1</b> for the device to send device-related initial startup information (such as device type, current IP address, software version) in the Vendor Specific Information field (in the BootP request). This information can be viewed in the main window of the BootP/TFTP utility under column 'Client Info' (refer to "BootP/TFTP Application User Interface" on page 169). For a full list of the Vendor Specific Information fields, refer to "Vendor Specific Information Field" on page 22.</p> <p><b>Note:</b> This option is not available on DHCP servers.</p>

### 11.1.11.3 Adding Clients


Adding a client creates an entry in the BootP/TFTP utility for a specific device.

➤ **To add a client without using a template, take these 3 steps:**

1. In the Client Configuration window, click the **Add New Client** icon ; a client with blank parameters is displayed.
2. Enter values in the fields on the right side of the window, using the guidelines for the fields in "Setting Client Parameters" on page 174.
3. Click **Apply** to save this entry to the list of clients, or click **Apply & Reset** to save this entry to the list of clients and send a reset message to the device to immediately implement the settings.

An easy way to create several clients that use similar settings is to create a template. For information on how to create a template, refer to "Managing Client Templates" on page 179.

➤ **To add a client using a template, take these 5 steps:**

1. In the Client Configuration window, click the **Add New Client** icon ; a client with blank parameters is displayed.
2. From the **Template** drop-down list, select the template that you want to use.
3. The values provided by the template are automatically entered into the parameter fields. To use the template parameters, leave the check boxes corresponding to each parameter selected. The parameter values appear in gray text.
4. To change a parameter to a different value, clear the check box corresponding to the parameter and enter another value. Clicking the check box again restores the template settings.

5. Click **Apply** to save this entry to the list of clients or click **Apply & Reset** to save this entry to the list of clients and send a reset message to the device to immediately implement the settings.



**Note:** To use **Apply & Reset**, you must enable **ARP Manipulation** in the Preferences window. In addition, you must have administrator privileges for the computer you are using.

#### 11.1.11.4 Editing Client Parameters

The procedure below describes how to edit a BootP client.

➤ **To edit the parameters of an existing client, take these 3 steps:**

1. In the left pane of the Client Configuration window, select the client that you wish to edit; parameters for that client are displayed on the right of the window.
2. Make the changes as required for each parameter.
3. Click **Apply** to save the changes, or click **Apply & Reset** to save the changes and send a reset message to the device to immediately implement the settings.




**Note:** To use **Apply & Reset**, you must enable **ARP Manipulation** in the Preferences window. In addition, you must have administrator privileges for the computer you are using.

#### 11.1.11.5 Deleting Clients

The procedure below describes how to delete a BootP client.


➤ **To delete a client from the BootP/TFTP utility, take these 3 steps:**

1. In the left pane of the Client Configuration window, select the client that you wish to delete.
2. Click the **Delete Current Client** button ; a warning message box appears.
3. To delete the client, click **Yes**.

#### 11.1.11.6 Testing the Client

There must only be one BootP utility supporting any particular client MAC active on the network at any given time.

➤ **To test if other BootP utilities support this client, take these 4 steps:**

1. In the left pane of the Client Configuration window, select the client that you wish to test.
2. Click the **Test Selected Client** button .

3. In the Log area of the main window, check that there is no other BootP utility supporting this client MAC (indicated in the Status column as Listed At together with the IP address of that utility).
4. If there is another utility responding to this client, you must remove that client from either this utility or the other one.


### 11.1.12 Managing Client Templates

The Templates window (**Services** menu > **Templates**) can be used to add templates to simplify configuration of clients when most of the parameters are identical.

Figure 11-5: Templates Screen

The screenshot shows the 'Templates' window. It features a title bar with the text 'Templates' and a close button. Below the title bar, there are two icons: a green plus sign and a red minus sign. A 'Template Name' label is followed by a text input field. To the right of the input field, there are several parameter fields: 'IP' (four boxes with '0'), 'Subnet' (four boxes with '255', '255', '0', '0'), 'Gateway' (four boxes with '0'), 'Server IP' (four boxes with '10', '13', '2', '20'), 'Boot File' (a dropdown menu), and 'INI File' (a dropdown menu). At the bottom right are 'OK' and 'Apply' buttons.

➤ **To add a new template, take these 5 steps:**

1. From the Services menu, choose **Templates**; the Templates window appears.
2. Click the **Add New Template** button .
3. Fill in the required parameter values in the parameter fields.
4. Click **Apply** to save each template.
5. Click **OK** when you are finished adding all your templates.

➤ **To edit a template, take these 4 steps:**

1. In the Template Name list, select the template.
2. Make changes to the parameters, as required.
3. Click **Apply** to save this new template.
4. Click **OK** when you are finished editing templates.

➤ **To delete a template, take these 3 steps:**

1. In the Template Name list, select the template.
2. Click the **Delete Current Template**  button; a warning message appears.
3. To delete the template, click **Yes**.

Note that



**Note:** A template cannot be deleted if it is currently in use.

## 11.2 TrunkPack Downloadable Conversion Utility

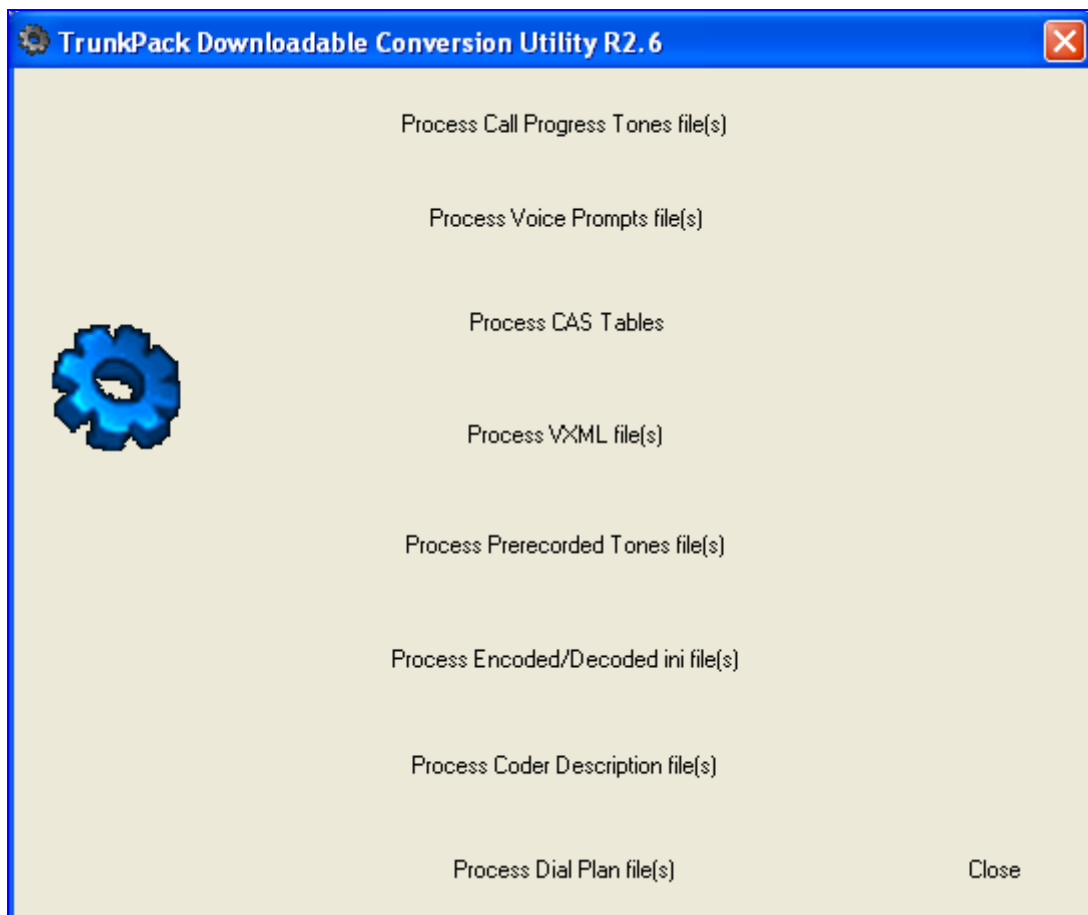
The AudioCodes TrunkPack Downloadable Conversion Utility is used to perform the following:

- Create a loadable Call Progress Tones (CPT) file (refer to "Converting a CPT ini File to a Binary dat File" on page [181](#))
- Create a loadable Voice Prompts (VP) file from prerecorded voice messages (Only applicable to 3000 Series and 2000 Series devices) (refer to "Creating a Loadable Voice Prompts File" on page [183](#))
- Create a loadable CAS protocol table file (Only applicable to Digital devices) (refer to "Creating a loadable CAS Protocol Table" on page [184](#))
- Create Dial Plan file(s) (Only applicable to Digital devices)
- Encode / decode an *ini* file (refer to "Encoding / Decoding an ini File" on page [187](#))
- Create a loadable Prerecorded Tones file (refer to "Creating a Loadable Prerecorded Tones File" on page [188](#))



The TrunkPack Downloadable Conversion Utility is run by clicking the file *DConvert.exe*, supplied with your software package.

**Figure 11-6: TrunkPack Downloadable Conversion Utility Main Screen**



**Note:** The 'Process VXML file(s)' and 'Process Coder Description files(s)' options are not applicable to SIP devices.

## 11.2.1 Converting a CPT ini File to a Binary dat File

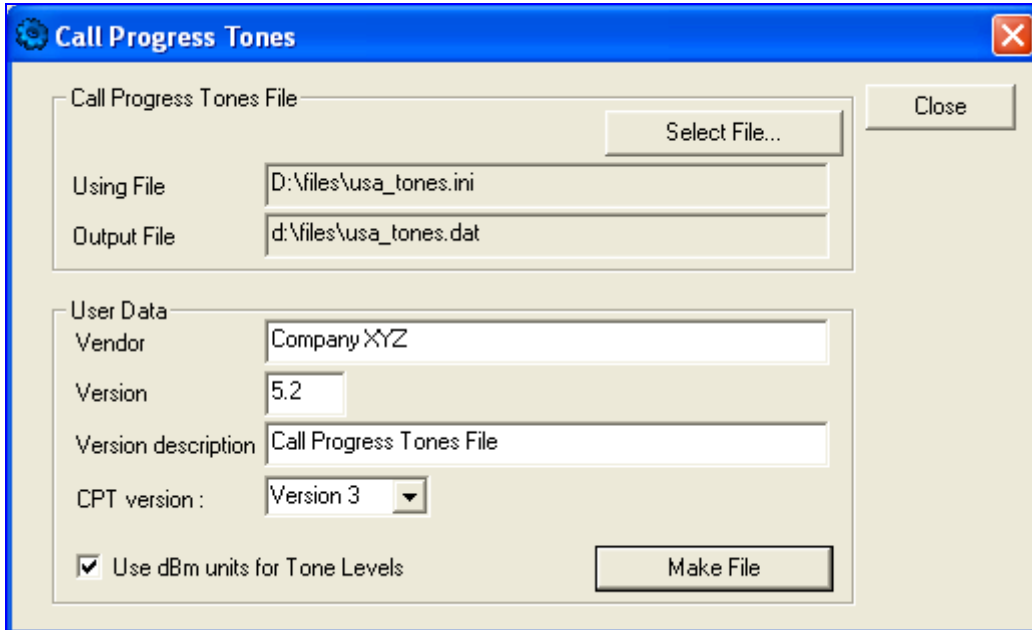
The procedure below describes how to convert a Call Progress Tones (CPT) *ini* file to a binary \*.dat file, using the TrunkPack Downloadable Conversion Utility. For detailed information on creating a CPT *ini* file, refer to Configuring the Call Progress Tones and Distinctive Ringing File in the device's *User's Manual*.

➤ **To convert a CPT *ini* file to a binary *dat* file, take these 10 steps:**

1. Start the TrunkPack Downloadable Conversion Utility; the main window opens (shown in "TrunkPack Downloadable Conversion Utility" on page 180).

2. Click the **Process Call Progress Tones File(s)** button; the 'Call Progress Tones' dialog box opens, shown in the figure below.

**Figure 11-7: Call Progress Tones Screen**



3. Under the 'Call Progress Tones File' group, click the **Select File** button.
4. Navigate to the folder that contains the CPT *ini* file that you want to convert.
5. Select the *ini* file, and then click the **Open** button; the name and path of both the *ini* file and the (output) *dat* file appears in the fields below the **Select File** button.
6. Under the 'User Data' group, enter the perform the following:
  - a. In the 'Vendor' field, enter the vendor's name (maximum length is 256 characters).
  - b. In the 'Version' field, enter the version number. The format is composed of two integers separated by a period '.' (e.g., 1.2, 23.4, 5.22)/
  - c. In the 'Version Description' field, enter a brief description of this file. The maximum length is 256 characters.
7. The default value of the 'CPT Version' drop-down list is Version 3. Perform one of the following:
  - If the software version you are using is prior to version 4.4, select Version 1 (to maintain backward compatibility).
  - If the software version you are using is 4.4, select Version 2.
  - Otherwise, leave the value at its default.
8. Select the 'Use dBm units for Tone Levels' check box. Note that the levels of the call progress tones (in the CPT file) must be in -dBm units.
9. Click the **Make File** button; the file is created and a message box is displayed when successfully complete.
10. Close the application.

## 11.2.2 Creating a Loadable Voice Prompts File

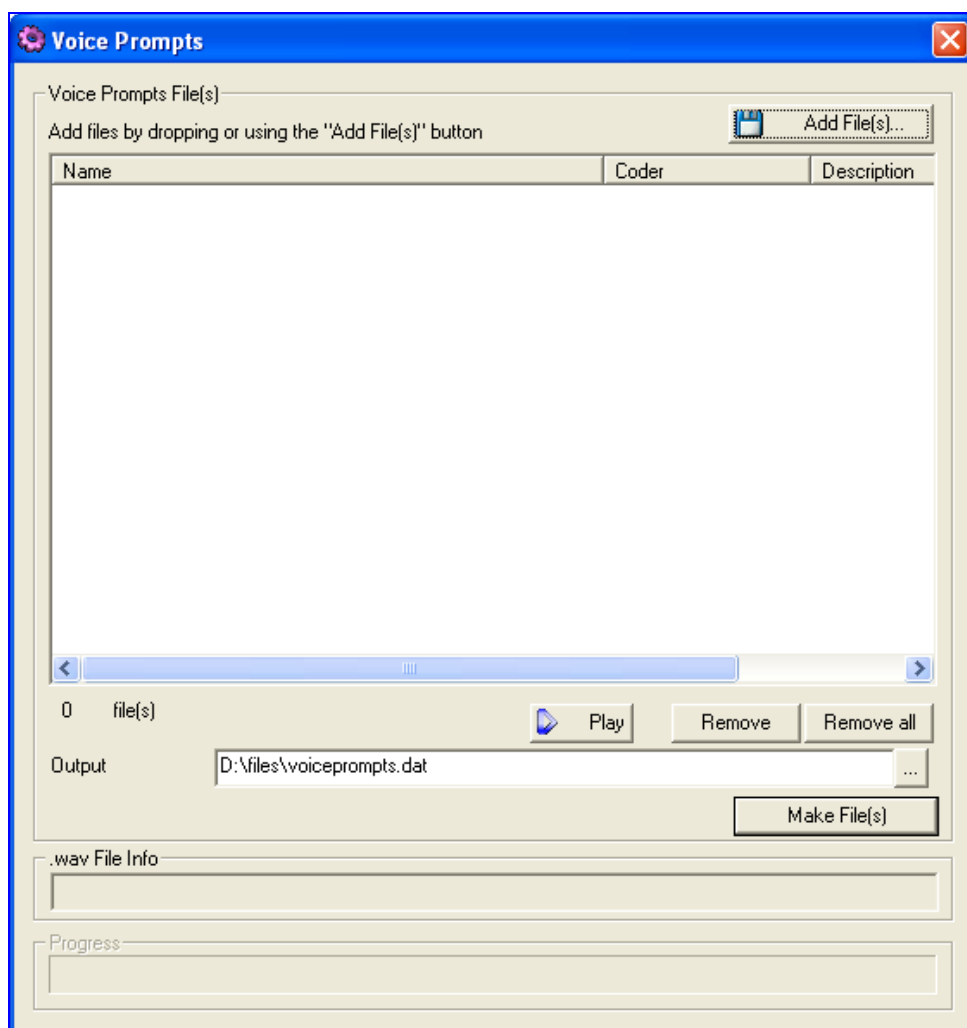
The procedure below describes how to create a loadable Voice Prompts file, using the TrunkPack Downloadable Conversion Utility. For detailed information on the Voice Prompts file, refer to Voice Prompts File in the device's *User's Manual*.



**Note:** This subsection is only applicable to AudioCodes 3000 Series and 2000 Series devices.

- **To create a loadable Voice Prompts *dat* file from your voice recording files, take these 7 steps:**
1. Start the TrunkPack Downloadable Conversion Utility; the main window appears (shown in "TrunkPack Downloadable Conversion Utility" on page 180).
  2. Click the **Process Voice Prompts File(s)** button; the 'Voice Prompts' dialog box opens.

**Figure 11-8: Voice Prompts Screen**



3. To add the prerecorded voice files to the 'Voice Prompts' screen, perform one of the following:
  - Select the files and drag them into the 'Voice Prompts' screen.
  - Click the **Add File(s)** button; the 'Select Files' screen opens. Select the required Voice Prompt files, and then click the **Add** button. Close the 'Select Files' screen.
4. Arrange the files according to your requirements by dragging and dropping them from one location in the list to another. Note that the order of the files determines their assigned Voice Prompt ID.

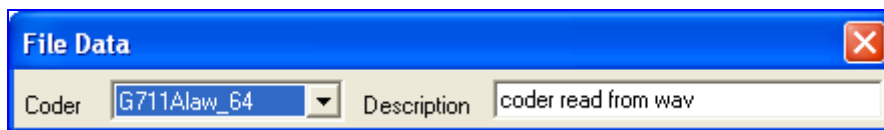

**Tips:**

- Use the **Play** button to listen to the wav files.
- Use the **Remove** and **Remove all** buttons to delete files from the list.

1. For each of the raw files, select a coder that corresponds to the coder in which it was originally recorded, by completing the following steps:
  - a. Double-click or right-click the required file(s); the 'File Data' window (shown in the figure below) appears.
  - b. From the 'Coder' drop-down list, select the required coder type.
  - c. In the 'Description' field, enter additional identifying information.
  - d. Close the 'File Data' window.

**Note:** For wav files, a coder is automatically selected from the wav file's header.

**Figure 11-9: File Data Window**



2. In the 'Output' field, specify the directory to which the Voice Prompts file is generated, followed by the name of the Voice Prompts file (the default name is voiceprompts.dat).
3. Click the **Make File(s)** button; the Voice Prompts loadable file is produced.

### 11.2.3 Creating a Loadable CAS Protocol Table File

The procedure below describes how to create a loadable CAS Protocol Table file, using the TrunkPack Downloadable Conversion Utility.



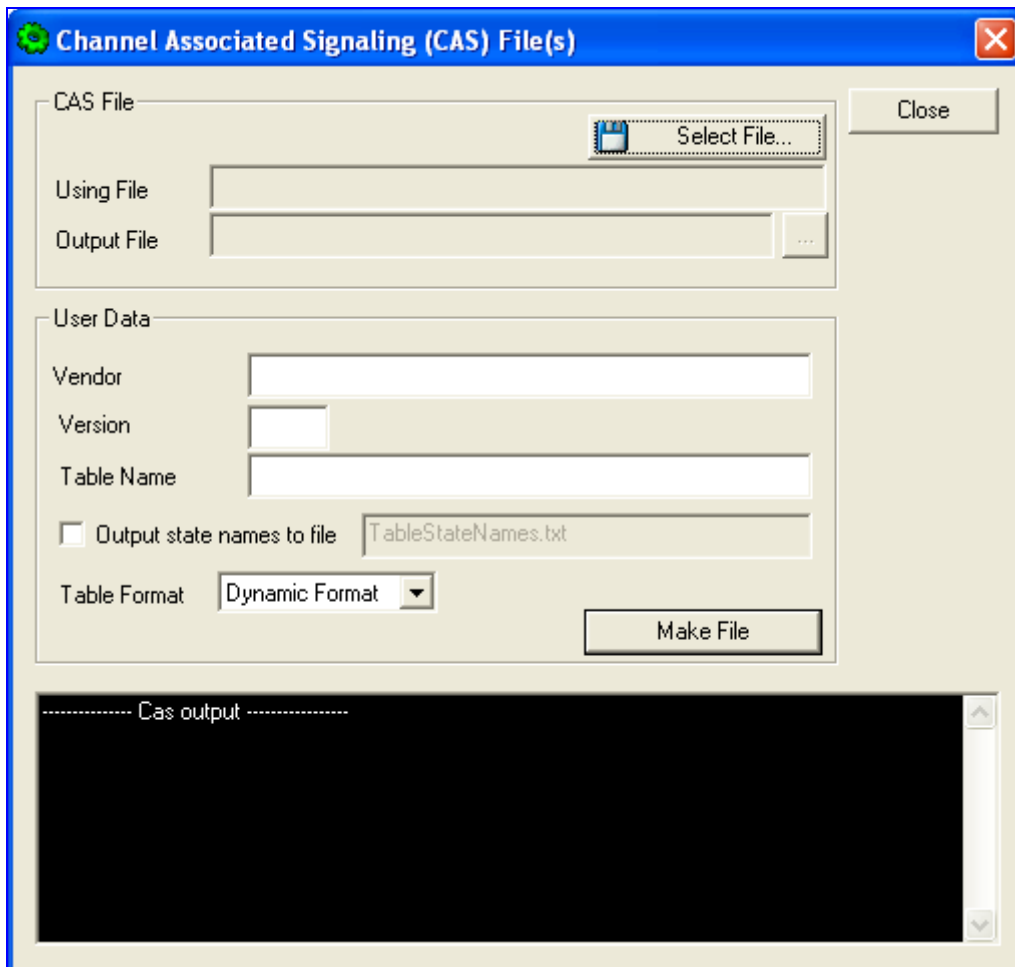
**Note:** This subsection is only applicable to AudioCodes Digital devices.

➤ **To create a loadable CAS protocol table file, take these 10 steps:**

1. Create the CAS protocol files (*xxx.txt* and *UserProt\_defines\_xxx.h*).
2. Copy the files generated in the previous step to the same directory in which the TrunkPack Downloadable Conversion Utility is located. Ensure that the files *CASSetup.h* and *cpp.exe* are also located in the same directory.

3. Start the TrunkPack Downloadable Conversion utility; the main window opens (shown in "TrunkPack Downloadable Conversion Utility" on page 180).
4. Click **Process CAS Tables**; the Channel Associated Signaling (CAS) screen opens, shown in the figure below.

**Figure 11-10: Call Associated Signaling (CAS) Screen**



5. Under the 'CAS File' group, click **Select File**, navigate to the folder in which the file is located, and then select the *txt* file you want converted; the 'Output File' field displays the file name and path, but with a *dat* extension. The table's name is also automatically designated.
6. Under the 'User Data' group, perform the following:
  - a. In the 'Vendor' field, enter the vendor's name (maximum of 32 characters).
  - b. In the 'Version' field, enter the version number. The value must be in the following format: [number] [single period '.'] [number] (e.g., 1.2, 23.4, 5.22)
7. In the 'Table Name' field, modify the name according to your requirements.
8. To create a file (for troubleshooting purposes) that contains the name of the States and their actual values, select the 'Output state names to file' check box; the default file name *TableStateNames.txt* appears in the adjacent field (you can modify the name of the file). The generated file is to be located in the same directory as the TrunkPack Downloadable Conversion utility.

9. From the 'Table Format' drop-down list, select the format you want to use:
  - Old Format: supported by all versions. Many CAS features are not supported in this format.
  - New Format: supported from 4.2 and later. From 5.2 and later a few new features are not supported by this format.
  - Dynamic Format: supported from 5.2 and later. Some 5.2 features are only supported by this format. The size of the file with dynamic format is significantly lower than other formats.
10. Click **Make File**; the *.dat* file is generated and saved in the directory specified in the 'Output File' field. A message box informing you that the operation was successful indicates that the process is completed. In the pane at the bottom of the Call Assisted Signaling (CAS) Files(s) screen, the CAS output log box displays the log generated by the process. It can be copied as needed. The information in it isn't retained after the screen is closed.

## 11.2.4 Creating a Dial Plan File

The procedure below describes how to create a Dial Plan file, using the TrunkPack Downloadable Conversion Utility.

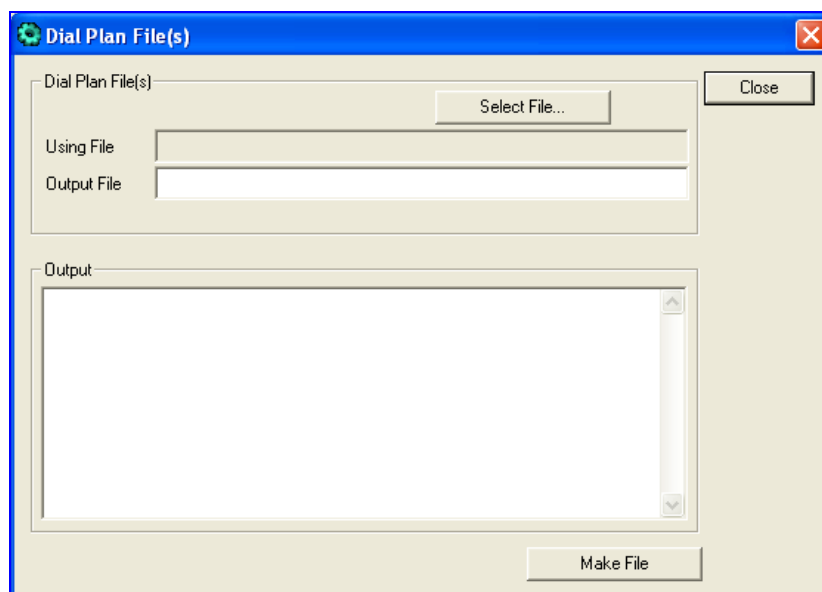


**Note:** This subsection is only applicable to AudioCodes Digital devices.

### ➤ To create a Dial Plan file, take these 6 steps:

1. Construct a Dial Plan text file according to the instructions in Dial Plan File in the device's *User's Manual*.
2. Start the TrunkPack Downloadable Conversion Utility; the main window appears.
3. Click the **Process Dial Plan File(s)** button; the 'Dial Plan File(s)' window appears.

**Figure 11-11: Dial Plan Screen**



4. Click the **Select File** button, navigate to the desired folder, and then select the file to be converted; the selected file name (but with the *.dat* extension) and path is displayed in the 'Output File' field. The output file name may be altered.
5. Click the **Make File** button. The *.dat* file is generated and saved in the same directory as shown in the 'Output File' field. A message box informing you that the operation was successful indicates that the process has been completed.
6. On the bottom of the 'Coders' window, the 'Output' log box displays the log generated by the process. It may be copied as needed. This information is not retained after the window is closed.



**Note:** The process verifies the input file for validity. Invalid data causes an error and aborts the process. In such a case, the log box contains further information.

## 11.2.5 Encoding / Decoding an ini File

The procedure below describes how to encode and decode an *ini* file, using the TrunkPack Downloadable Conversion Utility. For detailed information on secured *ini* file, refer to Secured ini File in the device's *User's Manual*.

➤ **To encode an *ini* file, take these 6 steps:**

1. Start the TrunkPack Downloadable Conversion Utility; the main window opens (shown in in "TrunkPack Downloadable Conversion Utility" on page 180).
2. Click the **Process Encoded/Decoded ini file(s)** button; the 'Encode/Decode ini File(s)' screen, shown below, opens.

**Figure 11-12: Encode / Decode ini File(s) Screen**

3. Under the 'Encode *ini* File(s)' group, click the **Select File** button.
4. Navigate to the folder that contains the *ini* file you want to encode.
5. Select the *ini* file, and then click the **Open** button; the name and path of both the *ini* file and the output encoded file appear in the fields under the **Select File** button. Note that the name and extension of the output file can be modified.
6. Click the **Encode File(s)** button; an encoded *ini* file with the name and extension you specified is created.

➤ **To decode an encoded *ini* file, take these 4 steps:**

1. Under the 'Decode *ini* File(s)' group, click the **Select File** button.
2. Navigate to the folder that contains the file you want to decode.
3. Click the file and click the **Open** button; the name and path of both the encode *ini* file and the output decoded file appear in the fields under the **Select File** button. Note that the name of the output file can be modified.
4. Click the **Decode File(s)** button; a decoded *ini* file with the name you specified is created.



**Note:** The decoding process verifies the input file for validity. Any change made to the encoded file causes an error and the decoding process is aborted.

## 11.2.6 Creating a Loadable Prerecorded Tones File

The procedure below describes how to create a loadable Prerecorded Tones (PRT) file, using the TrunkPack Downloadable Conversion Utility. For detailed information on PRT files, refer to Prerecorded Tones (PRT) File in the device's *User's Manual*.



**Note:** It is highly recommended to avoid using the Linear PCM coder.

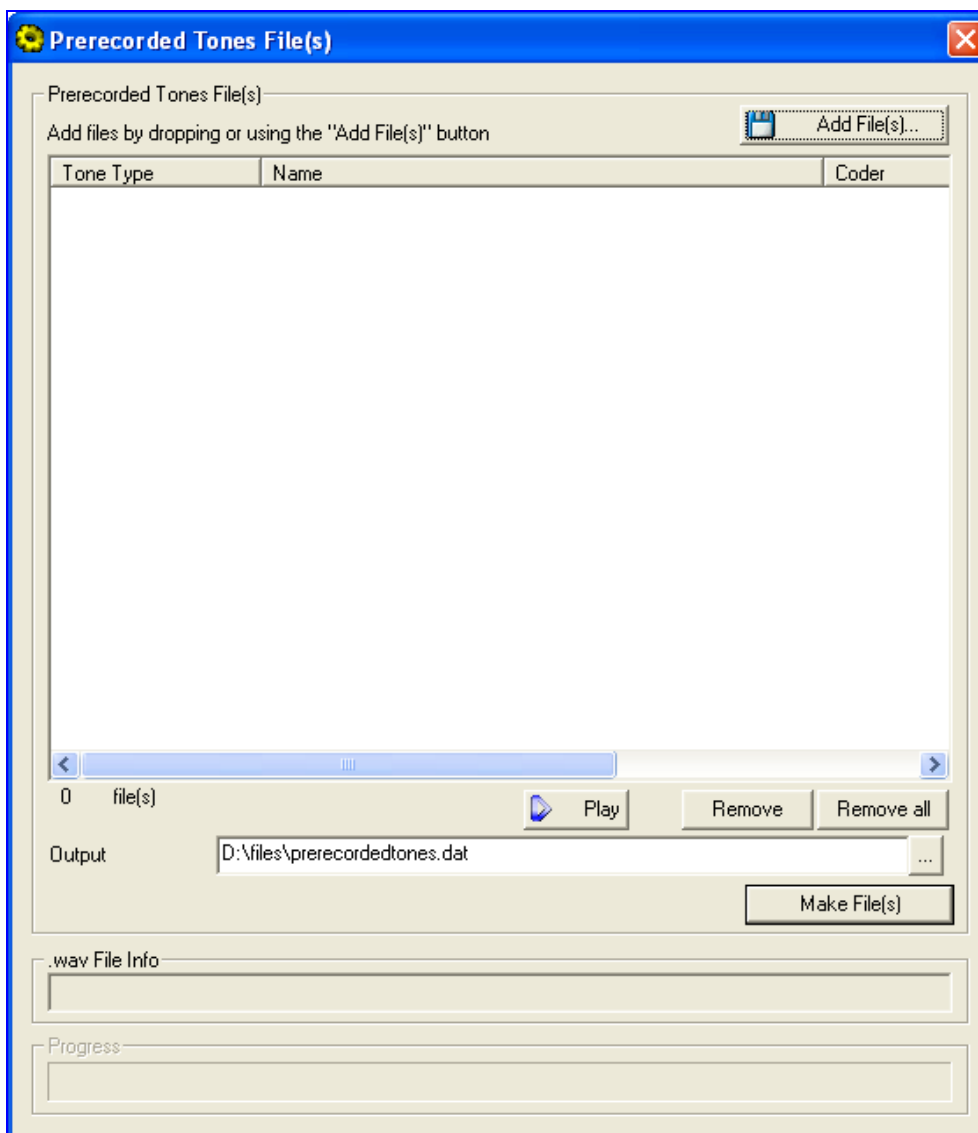
➤ **To create a loadable PRT *dat* file from your raw data files, take these 7 steps:**

1. Prepare the PRT files that you want to combine into a single *dat* file using standard recording utilities.
2. Start the TrunkPack Downloadable Conversion utility; the main window opens (shown in "TrunkPack Downloadable Conversion Utility" on page 180).



3. Click the **Process Prerecorded Tones File(s)** button; the Prerecorded Tones File(s) screen opens.

Figure 11-13: Prerecorded Tones Screen



4. To add the PRT files (that you created in Step 1), perform one of the following:
  - Select the files and drag them into the 'Prerecorded Tones File(s)' screen.
  - Click the **Add File(s)** button; the 'Select Files' screen opens. Select the required PRT files, and then click the **Add** button. Close the 'Select Files' screen.
5. For each raw data file, define a tone type, a coder, and the default duration, by completing the following steps:
  - a. Double-click or right-click the required file; the 'File Data' window (shown in the figure below) appears.
  - b. From the 'Type' drop-down list, select the tone type with which this raw data file is associated.
  - c. From the 'Coder' drop-down list, select the coder that corresponds to the coder with which this raw data file was originally recorded.
  - d. In the 'Description' field, enter brief identifying information (optional).

- e. In the 'Default' field, enter the default duration this raw data file is repeatedly played.
- f. Close the 'File Data' window (press the **Esc** key to cancel your changes); you are returned to the 'Prerecorded Tones File(s)' screen.

**Figure 11-14: File Data Window**



6. In the 'Output' field, specify the output directory in which the PRT file is generated, followed by the name of the PRT file (the default name is *prerecordedtones.dat*). Alternatively, use the **Browse** button to select a different output file, navigate to the desired file, and then select it; the selected file name and its path appear in the 'Output' field.
7. Click the **Make File(s)** button; the progress bar at the bottom of the window is activated. The *dat* file is generated and saved in the directory specified in the 'Output' field. A message box informing you that the operation was successful indicates that the process is completed.

## 11.3 Call Progress Tones Wizard



**Note:** This subsection is only applicable to AudioCodes Analog devices.

The Call Progress Tones Wizard (CPTWizard) is an application designed to facilitate the provisioning of an FXO device by recording and analyzing Call Progress Tones (CPT) generated by any PBX or telephone network. The CPTWizard creates a basic CPT *ini* file and *dat* files, providing a good starting point when configuring an FXO device. The *ini* file contains definitions for all relevant CPT; the *dat* file (which can also be created using the TrunkPack Downloadable Conversion utility -- "Converting a CPT ini File to a Binary dat File" on page 181) is in a format that is suitable for downloading to the device.

To use this wizard, an FXO device connected to your PBX with two physical phone lines is required. This device must be configured with factory-default settings and mustn't be used for phone calls during the operation of the wizard.



**Note:** You must use the CPTWizard version that corresponds to the device's software version.

### 11.3.1 Installation

The CPTWizard can be installed on any PC running Windows 2000 or Windows XP. Windows-compliant networking and audio peripherals are required for full functionality. To install the CPTWizard, copy the files from the supplied installation kit to any folder on your PC. No further setup is required (approximately 5 MB of hard disk space is required).

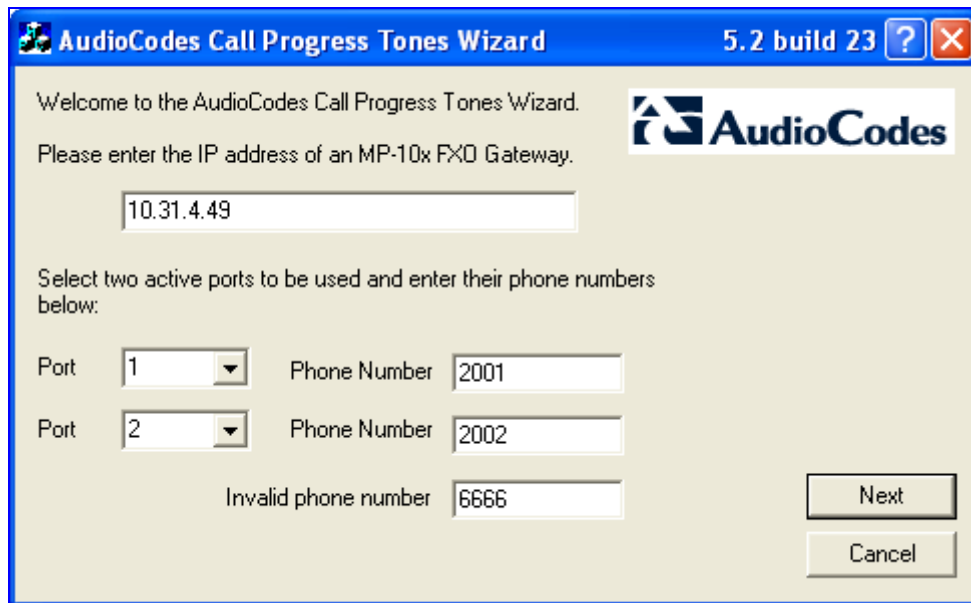
### 11.3.2 Initial Settings

The procedure below describes how to start the CPTWizard.

➤ **To start the CPTWizard, take these 5 steps:**

1. Run the *CPTWizard.exe* file; the wizard's initial settings screen is displayed:

**Figure 11-15: Initial Settings Screen**



The screenshot shows the 'AudioCodes Call Progress Tones Wizard' window, version 5.2 build 23. The window has a blue title bar and a yellow background. It contains the following fields and controls:

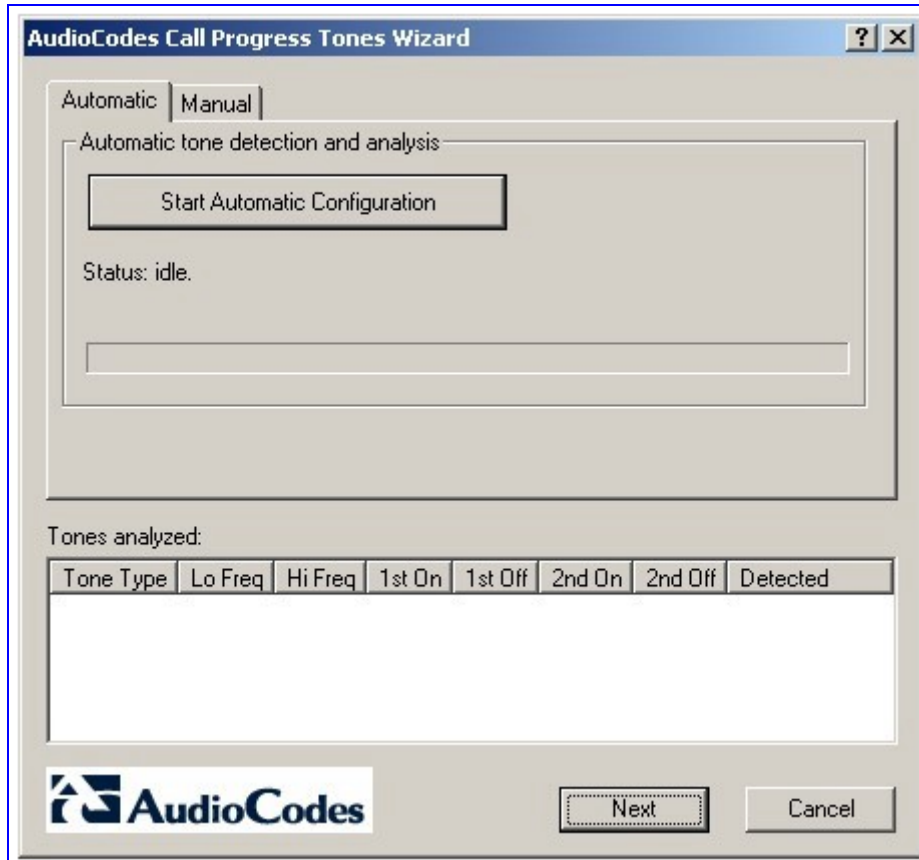
- Text: 'Welcome to the AudioCodes Call Progress Tones Wizard.'
- Text: 'Please enter the IP address of an MP-10x FXO Gateway.'
- Text input field: '10.31.4.49'
- Text: 'Select two active ports to be used and enter their phone numbers below:'
- Port 1: Dropdown menu showing '1', Phone Number input field showing '2001'
- Port 2: Dropdown menu showing '2', Phone Number input field showing '2002'
- Invalid phone number: Input field showing '6666'
- Buttons: 'Next' and 'Cancel'
- AudioCodes logo in the top right corner.

2. Enter the IP address of the FXO device.
3. Select the device's ports that are connected to your PBX, and specify the phone number of each extension.
4. In the 'Invalid phone number' field, enter a number that generates a 'fast busy' tone when dialed. Usually any incorrect phone number should cause a 'fast busy' tone.
5. Click **Next**.

### 11.3.3 Recording Screen - Automatic Mode

Once the connection between the CPTWizard and the FXO device is established, the recording screen is displayed:

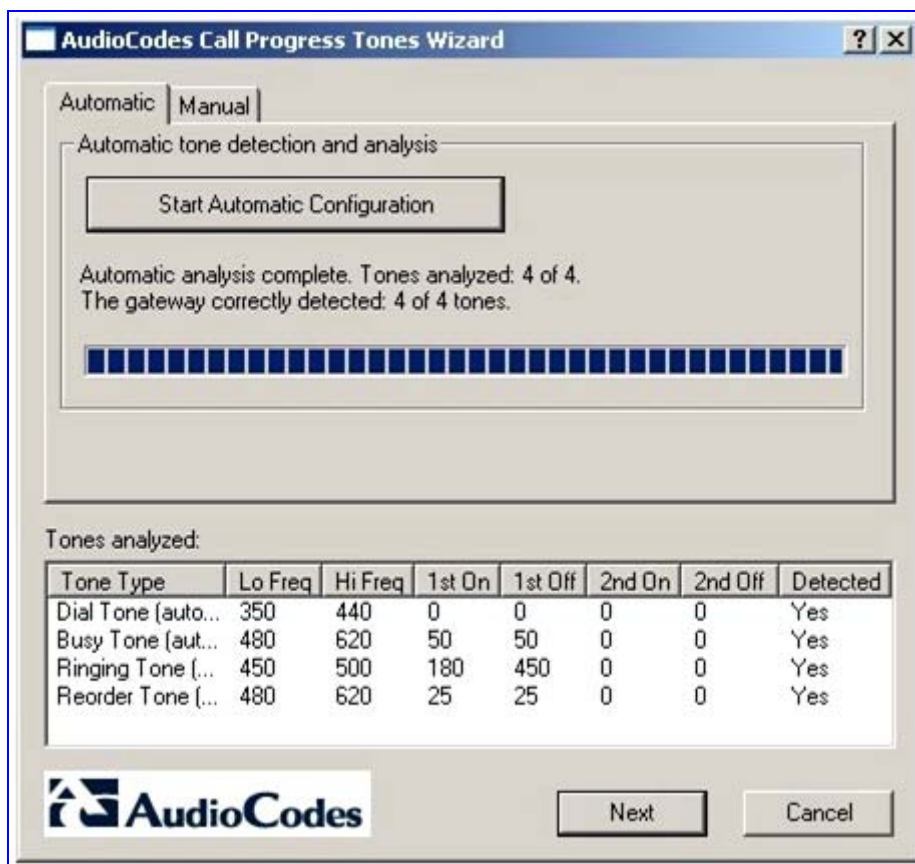
**Figure 11-16: Recording Screen - Automatic Mode**



➤ **To start recording in automatic mode, take these 4 steps:**

1. Click the **Start Automatic Configuration** button; the wizard starts the following Call Progress Tones detection sequence (the operation takes approximately 60 seconds to complete):
  - a. Sets port 1 to offhook, and then listens to the dial tone.
  - b. Sets port 1 and port 2 to offhook, dials the number of port 2, and then listens to the busy tone.
  - c. Sets port 1 to offhook, dials the number of port 2, and then listens to the Ringback tone.
  - d. Sets port 1 to offhook, dials an invalid number, and then listens to the reorder tone.
2. The wizard then analyzes the recorded Call Progress Tones and displays a message specifying the tones that were detected (by the device) and analyzed (by the wizard) correctly. At the end of a successful detection operation, the detected Call Progress Tones are displayed in the Tones Analyzed pane, as shown in the figure below:

Figure 11-17: Recording Screen after Automatic Detection



3. All four Call Progress Tones are saved (as standard A-law PCM at 8000 bits per sample) in the same directory as the *CPTWizard.exe* file is located, with the following names:
  - *cpt\_recorded\_dialtone.pcm*
  - *cpt\_recorded\_busytone.pcm*
  - *cpt\_recorded\_rington.pcm*
  - *cpt\_recorded\_invalidtone.pcm*
4. At this stage, you can either click **Next** to generate a Call Progress Tones *ini* and *dat* files and terminate the wizard, or continue to manual recording mode.

**Notes:**

- If the device is configured correctly (with a Call Progress Tones *dat* file loaded to the device), all four Call Progress Tones are detected by the device. By noting whether the device detects the tones or not, you can determine how well the Call Progress Tones *dat* file matches your PBX. During the first run of the CPTWizard, it is likely that the device does not detect any tones.
- Some tones cannot be detected by the FXO device (such as 3-frequency tones and complex cadences). CPTWizard is therefore, limited to detecting only those tones that can be detected on the FXO device.

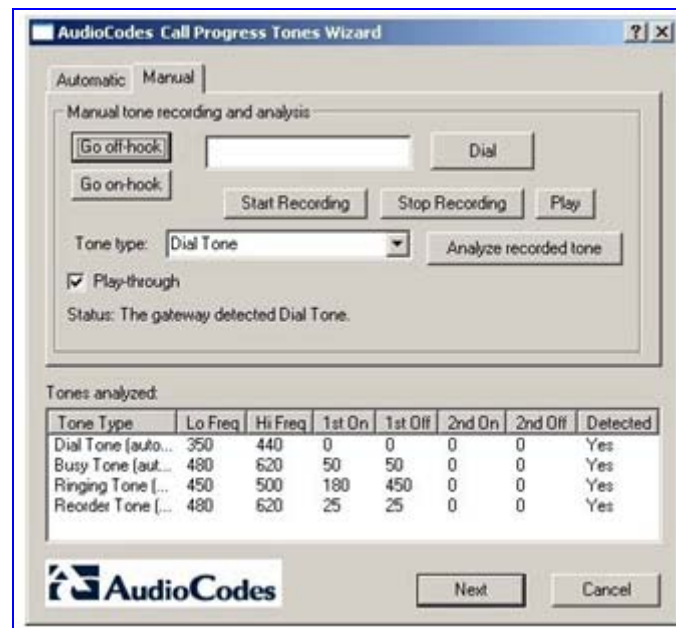
### 11.3.4 Recording Screen - Manual Mode

In manual mode, you can record and analyze tones included in the Call Progress Tones *ini* and *dat* files in addition to the tones analyzed when in automatic mode.

➤ **To start recording in manual mode, take these 8 steps:**

1. In the recording screen, click the **Manual** tab; the 'Manual Tone Recording' pane is displayed.

**Figure 11-18: Recording Screen - Manual Mode**



2. Select the **Play-through** check box to hear the tones through your PC speakers.
3. Click the **Go off-hook** button, enter a number to dial in the 'Dial' field, and then click the **Dial** button.
4. When you're ready to record, click the **Start Recording** button.
5. When the desired tone is complete, click **Stop Recording**. (The recorded tone is saved as 'cpt\_manual\_tone.pcm'.)



**Note:** Due to some PC audio hardware limitations, you may hear 'clicks' in play-through mode. You can ignore these clicks.

6. From the 'Tone type' drop-down list, select the tone type, and then click **Analyze recorded tone**; the analyzed tone is added to the 'Tones analyzed' list at the bottom of the screen. It is possible to record and analyze several different tones for the same tone type (e.g., different types of 'busy' signal).
7. Repeat the process for more tones, as necessary.
8. When you're finished adding tones to the list, click **Next** to generate a Call Progress Tones *ini* and *dat* files and terminate the wizard.

### 11.3.5 Call Progress Tones ini and dat Files

Once the wizard completes the Call Progress Tone detection, a text file named *call\_progress\_tones.ini* and a binary file named *call\_progress\_tones.dat* are created in the same directory in which the *CPTWizard.exe* file is located. The latter is ready for download to the device and it contains the same output which the DConvert utility would produce when processing the *ini* file.

The *ini* file contains the following information:

- Information on each tone that was recorded and analyzed by the wizard. This information includes frequencies and cadence (on/off) times, which is required when converting the *ini* file to *dat*.

Below shows an example of an *ini* file with Call Progress Tone properties:

```
[CALL PROGRESS TONE #1]
Tone Type=1
Low Freq [Hz]=350
High Freq [Hz]=440
Low Freq Level [-dBm]=0
High Freq Level [-dBm]=0
First Signal On Time [10msec]=0
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=0
Second Signal Off Time [10msec]=0
```

- Information relating to possible matches of *each* tone with the CPTWizard's internal database of common tones. This information is specified as comments in the file and is ignored when converting the *ini* file to a *dat* file.

Below shows an example of a file with Call Progress Tone database matches:

```
# Recorded tone: Busy Tone (automatic configuration)
## Matches: PBX name=ITU Anguilla, Tone name=Busy tone
## Matches: PBX name=ITU Antigua and Barbuda, Tone name=Busy tone
## Matches: PBX name=ITU Barbados, Tone name=Busy tone
## Matches: PBX name=ITU Bermuda, Tone name=Busy tone
## Matches: PBX name=ITU British Virgin Islan, Tone name=Busy tone
## Matches: PBX name=ITU Canada, Tone name=Busy tone
## Matches: PBX name=ITU Dominica (Commonweal, Tone name=Busy tone
## Matches: PBX name=ITU Hongkong, China, Tone name=Busy tone
## Matches: PBX name=ITU Jamaica, Tone name=Busy tone
## Matches: PBX name=ITU Korea (Republic of), Tone name=Busy tone
## Matches: PBX name=ITU Montserrat, Tone name=Busy tone
```

- Information relating to matches of *all* tones recorded with the CPTWizard's internal database. The database is scanned to find one or more PBX definitions that match all recorded tones (i.e., dial tone, busy tone, ringing tone, reorder tone and any other manually-recorded tone - all match the definitions of the PBX). If a match is found, the entire PBX definition is reported (as comments) in the *ini* file using the same format.

Below shows an example of a file with full PBX/Country Database match:

```
## Some tones matched PBX/country Audc US
## Additional database tones guessed below (remove #'s to use).
#
# # Audc US, US Ringback tone
# [CALL PROGRESS TONE #5]
# Tone Type=2
# Low Freq [Hz]=450
# High Freq [Hz]=500
# Low Freq Level [-dBm]=0
# High Freq Level [-dBm]=0
# First Signal On Time [10msec]=180
# First Signal Off Time [10msec]=450
# Second Signal On Time [10msec]=0
# Second Signal Off Time [10msec]=0
```




**Notes:**

- If a match is found in the database, consider using the database's definitions instead of the recorded definitions, as they might be more accurate.
- For full operability of the FXO device, it may be necessary to edit this file and add more Call Progress Tone definitions. Sample Call Progress Tones *ini* files are available in the release package.
- When the call progress tones *ini* is complete, the corresponding dat file is ready for download. After loading this file to the device, repeat the automatic detection phase discussed above, and verify that the device detects all four call progress tones correctly.
- Manually changing the *ini* file causes the dat file to be outdated and it therefore, needs to be re-generated according to the new *ini* file. A dat file may be regenerated by clicking the **Regenerate** button at the final dialog or by using the DConvert utility.

### 11.3.6 Adding a Reorder Tone to the CPT File

The following procedure describes how to add a Reorder tone that a PBX generates to indicate a disconnected call, to the CPT file.

➤ **To add a Reorder tone to the CPT file, take these 11 steps:**

1. Make a call (using G.711) between the device FXO, which is connected to the PBX, and a remote entity in the IP network.
2. Capture the call using a network sniffer such as Whiteshark.
3. Disconnect the call from the PBX side, and then wait approximately 30 seconds before stopping the Whiteshark recording.
4. In the network trace, locate the RTP stream sent from the FXO.
5. Save the RTP payload on your PC as a \*.pcm file by clicking **Save Payload (Statistics menu > RTP > Stream Analysis)**. (Note: ensure that you select the 'forward' option.)
6. Open the \*.pcm file in a voice recording utility such as CoolEdit.
7. Locate the tone that the PBX played to indicate the disconnected call (if such a tone exists).
8. Locate the attributes of the tone -- its frequency and interval (on / off time).
9. In the Call Progress Tones file, add a new Reorder Tone with the attributes you found in the previous step. Ensure that you update the numbers of the successive tones and the total number of tones in the beginning of the file.
10. Create a Call Progress Tones.dat file using the DConvert Utility (refer to "TrunkPack Downloadable Conversion Utility" on page 180).
11. Load the new file to the device, and then reset the device.



## 12 Diagnostics

Several diagnostic tools are provided to enable you to identify correct functioning of the device or an error condition with a probable cause and a solution or workaround.

The diagnostic tools include the following:

- Front-panel LEDs on the device (refer to the device's *Fast Track Guide*).
- Self-Testing on hardware initialization (refer to Section "Self-Testing" on page 197).
- FXS Line testing (refer to Section "FXS Line Testing" on page 198). (Applicable only to Analog devices.)
- Error / notification messages via the following interfaces:
  - **Syslog:** Log messages can be viewed using an external Syslog server (refer to "Syslog Support" on page 199) or in the 'Message Log' page of the Web interface (refer to Activating the Internal Syslog Viewer). Note that the 'Message Log' page is not recommended for prolong debugging.
  - RS-232 terminal (for establishing a serial communications link with the device, refer to device's *Fast Track Guide*). (Applicable only to Analog devices.)
- Debug Recording using CLI (refer to "Debug Recording (DR)" on page 201)

### 12.1 Self-Testing

The device features the following self-testing modes to identify faulty hardware components:

- **Startup Tests:** These tests have minor impact in real-time. While the Startup tests are executed, the regular operation of the device is disabled. If an error is detected, an error message is sent to the Syslog. The following hardware components are tested:
  - **CPU speed** - applicable only to 3000 Series devices
  - **TSA** (Time Slot Assigner) - applicable only to Digital devices
  - **CPU Version** - applicable only to 2000 Series devices
  - **PSTN framers** (when used) - applicable only to Digital devices except IPmedia 3000/IPM-8410
  - **Missing DSP's** - applicable only to Digital devices
  - **Lattice TPM and TER** - applicable only to 3000 Series devices
  - **GB Ethernet** - applicable only to 3000/6310 Series devices
  - **Voice path** - applicable only to Digital devices
- **Periodic Test:** (Applicable only to 2000 and 3000 Series devices.) Monitors the device during run-time. This test is performed after startup, even when there is full traffic on the device (quality is not degraded). This is a short test phase in which the only error detected and reported is failure in initializing hardware components or malfunction in running hardware components. If an error is detected, an error message is sent to the Syslog. The following hardware components are tested:
  - **Time Slot Assigner (TSA)** - applicable only to Digital devices
  - **PSTN framers** (when used) - applicable only to Digital devices except IPmedia 3000/IPM-8410
  - **Missing DSP's** - applicable only to Digital devices

- **Lattice TPM and TER** - applicable only to 3000 Series devices
- **Gb Ethernet ports** - applicable only to 3000/6310 Series devices
- **Voice path** - applicable only to 3000 Series and 2000 Series devices for redundant blade
- **User-Initiated (Detailed) Test:** initiated by the user when the device is offline (not used for regular service). This test is used in addition to the Startup tests. The test is performed on startup when initialization of the device completes and if the parameter EnableDiagnostics is set to 1 or 2 (refer to the device's *User's Manual* for a description of his parameter). For Analog devices, the **Ready** and **Fail** LEDs are lit while the Detailed test is running. The following hardware components are tested:
  - **RAM** (when EnableDiagnostics = 1 or 2) - applicable only to Digital devices
  - **Flash memory** (when EnableDiagnostics = 1 or 2)
  - **DSPs** (when EnableDiagnostics = 1 or 2)
  - **Physical Ethernet ports** (when EnableDiagnostics = 1 or 2)
  - **Analog interfaces** (when EnableDiagnostics = 1 or 2) - applicable only to Analog devices
  - **UTOPIA Bridge** (when EnableDiagnostics = 2) - applicable only to 3000 Series devices


**Notes:**

- To continue regular operation, disable the Detailed test. Set the parameter EnableDiagnostics to 0, and then reset the device.
- When the Detailed test is enabled, ignore errors sent to the Syslog server.

## 12.2 Analog Line Testing



**Note:** This subsection is applicable only to Analog devices.

Analog Line testing is executed using SNMP only:

- For FXO interfaces: acAnalogFxoLineTestTable SNMP table
- For FXS interfaces: acAnalogFxsLineTestTable SNMP table

The device features a mechanism that performs tests on the telephone lines connected to FXS and FXO ports. These tests provide various line measurements. In addition to these tests (detailed below), a keep-alive test is also performed every 100 msec on each of the analog ports to detect communication problems with the analog device and overheating (in FXS ports).

The following line tests are available on FXS interfaces:

- Hardware revision number
- Temperature (above or below limit, only if a thermometer is installed)
- Hook state

- Coefficients checksum
- Message waiting indication status
- Ring state
- Reversal polarity state
- Line current (only on port 0)
- Line voltage between TIP and RING (only on port 0)
- 3.3 V reading (only on port 0)
- Ring voltage (only on port 0)
- Long line current (only on port 0)

The following line tests are available on FXO interfaces:

- Line Current (mA)
- Line Voltage (V)
- Hook (0 = on-hook; 1 = off-hook)
- Ring (0 - Off; 1 - On)
- Line Connected (0 = Disconnected; 1 = Connected)
- Polarity state (0 = Normal; 1 = Reversed, 2 = N/A)
- Line polarity (0 = Positive; 1 = Negative)
- Message Waiting Indication (0 = Off; 1 = On)



**Note:** The line testing mechanism must be used only for monitoring and never when there are calls in progress.

## 12.3 Syslog Support

Syslog protocol is an event notification protocol that enables a machine to send event notification messages across IP networks to event message collectors, also known as Syslog servers. The Syslog protocol is defined in the IETF RFC 3164 standard.

Since each process, application, and operating system was written independently, there is little uniformity in Syslog messages. For this reason, no assumption is made on the contents of the messages other than the minimum requirements of its priority.

Syslog uses UDP as its underlying transport layer mechanism. By default, UDP port 514 is assigned to Syslog, but this can be changed (using the SyslogServerPort parameter).

The Syslog message is transmitted as an ASCII (American Standard Code for Information Interchange) message. The message starts with a leading 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 and Severity as described below. The Priority number consists of one, two, or three decimal integers.

For example:

```
<37> Oct 11 16:00:15 mymachine su: 'su root' failed for lonvick on /dev/pts/8
```



**Notes:** For 2000 and 3000 Series devices.

- When NTP is enabled, a timestamp string [hour:minutes:seconds] is added to all Syslog messages (for information on NTP, refer to the device's *User's Manual*).
- For Mediant 3000: All High Availability main operations and events are sent to the Syslog with the prefix 'M3K\_HA'. All Syslog messages and events of the redundant TP-6310 blade are sent to the Syslog by the active TP-6310 blade with the appropriate message prefix.

### 12.3.1 Syslog Servers

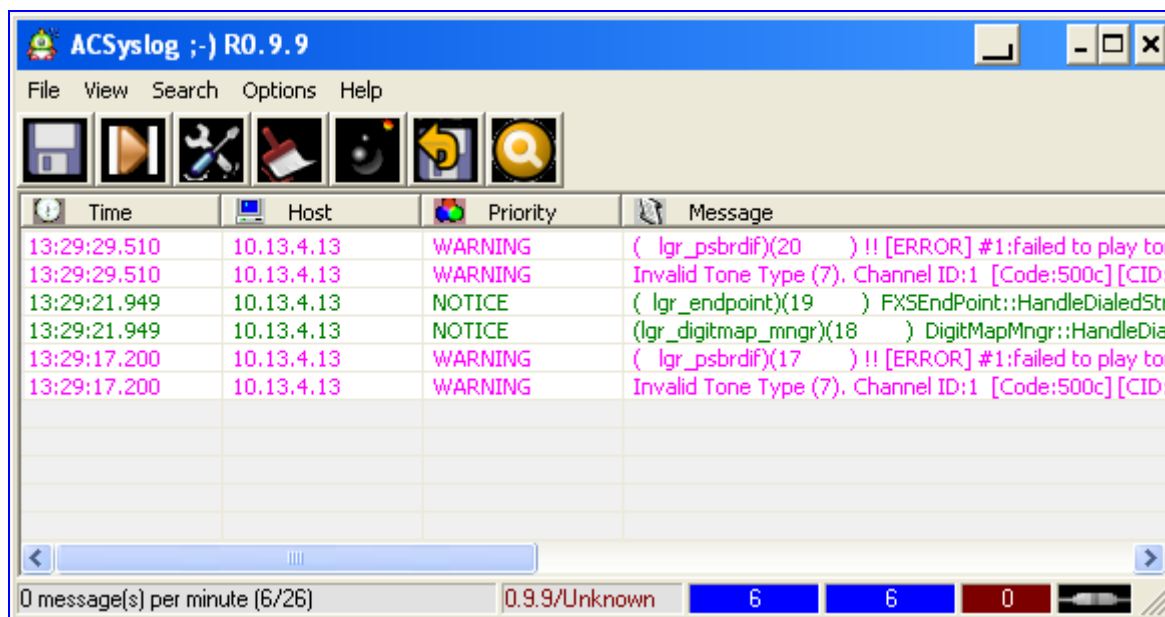
You can use the supplied AudioCodes proprietary Syslog server **ACSyslog** or any other third-party Syslog server for receiving Syslog messages. A typical Syslog server application enables filtering of messages according to priority, IP sender address, time, date, etc.

Below is a list of third-party Syslog servers available as shareware on the Internet:

- Kiwi Enterprises: <http://www.kiwisyslog.com/>
- The US CMS Server: [http://uscms.fnal.gov/hanlon/uscms\\_server/](http://uscms.fnal.gov/hanlon/uscms_server/)
- TriAction Software: <http://www.triaction.nl/Products/SyslogDaemon.asp>

- Netal SL4NT 2.1 Syslog Daemon: <http://www.netal.com/>

**Figure 12-1: AudioCodes' Proprietary Syslog Server**



### 12.3.2 Enabling the Syslog Server

The Syslog client, embedded in the device sends error reports/events generated by the device to a Syslog server using IP/UDP protocol. The Syslog client can be configured using either the *ini* file or the Web interface. The procedure below provides the *ini* file parameters; for the corresponding Web interface parameters, refer to the device's *User's Manual* (unless otherwise mentioned).

➤ **To activate the Syslog client on the device, take these 5 steps:**

1. Enable the Syslog feature (set the *ini* file parameter EnableSyslog to 1).
2. Define the IP address of the Syslog server (*ini* file parameter SyslogServerIP).
3. Define the port number of the Syslog server (*ini* file parameter SyslogServerPort).
4. Define the Syslog logging level (*ini* file parameter GWDebugLevel).
5. To enable the device to send log messages that report certain types of Web actions according to a predefined filter, use the *ini* file parameter ActivityListToLog.

## 12.4 Debug Recording (DR)

The debug recording (DR) tool can be used to capture media streams, networking and signaling traffic, and other internal blade information.

## 12.4.1 Collecting DR Messages

The client that is used to capture the DR packets is the open source Wireshark program (which can be downloaded from [www.wireshark.org](http://www.wireshark.org)). An AudioCodes proprietary plugin `acdr.dll` file (supplied in the software kit) must be placed in the 'plugin' folder of the installed Wireshark version (typically, `C:\Program Files\Wireshark\plugins\<Wireshark version>\`).

The default DR port is 925. This can be changed in Wireshark (**Edit** menu > **Preferences** > **Protocols** > **ACDR**). When loaded, the Wireshark plugin dissects all packets on port 925 as DR packets.



### Notes:

- The plugins for DR are per major AudioCodes release. The plugins that are released with version 5.6 are applicable to Wireshark version 99.08. The plugins are backward compatible.
- From Wireshark version 99.08, the `tpncp.dat` file must be placed in the folder `C:\Program Files\Wireshark\tpncp`.

## 12.4.2 Activating DR

Debug Recording activation is performed using the CLI interface under the `DebugRecording` directory. This subsection describes the basic procedures for quickly activating the DR and collecting the call traces. For a more detailed description of all the DR commands, refer to "Commands" on page 28.

### ➤ To activate the DR, take these 7 steps:

1. Start a CLI management session (refer to the "Starting a CLI Management Session" on page 27).
2. At the prompt, type **DR** to access the `DebugRecording` directory.
3. At the prompt, type **STOP** to terminate all active recordings, if any.
4. At the prompt, type **RTR ALL** to remove all previous recording rules.
5. At the prompt, type **RT ALL** to remove all DR targets (i.e., client IP addresses) from the list.
6. At the prompt, type **AIT <IP address of the target>** to define the IP address of the PC (running Wireshark) to which the device sends its debug packets.
7. Continue with the procedures described below for capturing PSTN and/or DSP traces.

### ➤ To capture DSP traces (internal DSP packets, RTP, RTCP, T38, events), take these 4 steps:

1. Setup the DR, as described at the beginning of this section.
2. At the prompt, type **ANCT ALL** (for MediaPack) or **ANCT ALL-WITH-PCM 1** (for Digital devices); the next call on the device is recorded.
3. At the prompt, type **START**.
4. Start Wireshark, and then filter according to the UDP port (default is 925) to where debug packets are sent.

For Digital devices, you can capture PSTN traces as described in the procedure below:

➤ **To capture PSTN (SS7, CAS, ISDN) traces, take these 7 steps:**

1. Setup the DR, as described at the beginning of this section.
2. At the prompt, type **pstn**.
3. At the prompt, type **PstnCommon**.
4. At the prompt, type **PstnSetTraceLevel <TrunkId> <BChannel> <TraceLevel>**.  
For example, to enable PSTN traces on the first Trunk, type **PstnSetTraceLevel 0 -1 1**.
5. At the prompt, type **APST<packet type -- ISDN, CAS, or SS7>**.
6. At the prompt, type **START**.
7. Start Wireshark, and then filter according to the UDP port (default is 925) to where debug packets are sent.



**Notes:**

- For Digital devices: PSTN and DSP recording can be performed simultaneously.
- All DR rules are deleted after the device is reset.

### 12.4.3 DR Command Reference

The below tables describe all the DR commands. You can also view the description of a DR command in the CLI interface, by simply typing the command name without any arguments.

**Table 12-1: Client Setup Commands**

Command	Parameters	Description
<b>AddIpTarget</b>	IPAddr [UDPPort]	Adds a Wireshark DR IP client to the list. UDPPort (optional): port on which to send the recorded packets (default is 925).
<b>RemoveTarget</b>	Index	Removes a DR client from the list. Index: index for the removed target (as displayed via ListTargets).
<b>ListTargets</b>		Displays the client list.
<b>SetDefaultTarget</b>	Index	Changes the default target. The default target is the first target added (AddTarget). Index: index for the default target (as displayed via ListTargets).

**Table 12-2: Trace Rules**

Command	Parameters	Description
<b>AddIPTrafficTrace</b>	TracePoint PDUType SourcePort DestPort [SourceIP] [DestIP] [DebugTarget]	<p>Record IP traffic.</p> <ul style="list-style-type: none"> <li>Trace Point: Net2Host = Inbound non-media traffic. Host2Net = outbound non-media traffic.</li> <li>PDUType: UDP = UDP traffic. TCP = TCP traffic. ICMP = ICMP traffic. IPType = Any other IP type (as defined by <a href="http://www.iana.com">http://www.iana.com</a>). A = All traffic types.</li> <li>SourcePort: datagram's source port number (ALL for IP wildcard).</li> <li>DestPort: datagram's destination port number (ALL for IP wildcard).</li> <li>SourceIP (optional): datagram's source IP address (ALL for IP wildcard).</li> <li>DestIP (optional): datagram's source IP address (ALL for IP wildcard).</li> <li>DebugTarget (optional): debug target list index; if not specified, the default target is used.</li> </ul>
<b>AddIPControlTrace</b>	TracePoint ControlType [DebugTarget]	<p>Records an IP control.</p> <ul style="list-style-type: none"> <li>Trace Point: Net2Host = Inbound/Outbound non-media traffic. ControlType: SIP = SIP traffic.</li> </ul> <p>DebugTarget (optional): debug target list index; if not specified, the default target is used.</p>
<b>AddPstnSignalingTrace</b>	PacketType [DebugTarget]	<p>Records PSTN signaling. Packet Type:</p> <ul style="list-style-type: none"> <li>CAS = CAS signaling.</li> <li>ISDN = ISDN signaling.</li> <li>SS7 = SS7 signaling.</li> </ul> <p>DebugTarget (optional): debug target list index; if not specified, the default target is used.</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>Applicable only to Digital devices.</li> <li>To record PSTN signaling, 'PSTN Trace Level' (TraceLevel ini file) must be set to 1.</li> </ul>



Command	Parameters	Description
<b>AddNextCallTrace</b>	PacketType NumOfCalls [TraceType] [DebugTarget]	<p>Records the next media calls.</p> <ul style="list-style-type: none"> <li>Packet Type:           <ul style="list-style-type: none"> <li>ALL = all media related (internal DSP packets, RTP, RTCP, T38, events) of a certain call.</li> <li>ALL-WITH-PCM = all media-related and PCM traffic of a certain call.</li> </ul> </li> <li>NumOfCalls: amount of next media calls to record. (<b>Note:</b> Currently, only 1 call can be recorded.)</li> </ul> <p>Trace Type (optional):</p> <ul style="list-style-type: none"> <li>New (default) = the next new NumOfCalls calls to record. When these calls end, new calls are not recorded.</li> <li>Dynamic = the next new NumOfCalls calls to record. When these calls end, new calls are recorded until this trace is deleted.</li> </ul> <p>DebugTarget (optional): debug target list index; if not specified, the default target is used.</p>
<b>AddTrunkBchannelTrace</b>	PacketType TRUNK [TO_TRUNK] [BCHANNEL] [TO_BCHANNEL][DebugTarget]	<p>Records media calls according to trunk and B-channel.</p> <ul style="list-style-type: none"> <li>Packet Type:           <ul style="list-style-type: none"> <li>ALL = all media related (internal DSP packets, RTP, RTCP, T38, events) of a certain call.</li> <li>ALL-WITH-PCM = all media-related and PCM traffic of a certain call.</li> </ul> </li> <li>Trunk: start of range trunk number for recording. (<b>Note:</b> Currently, only 1 channel can be recorded.)</li> </ul> <p>To_Trunk (optional): end of range trunk number.          BChannel (optional): start of range B-Channel number for recording.          To_BChannel (optional): end of range B-Channel number for recording.          DebugTarget (optional): debug target list index; if not specified, the default target is used.</p> <p><b>Note:</b> Applicable only to Digital devices.</p>

Command	Parameters	Description
<b>AddChannelIdTrace</b>	PacketType Channel-Id [To Channel-Id][DebugTarget]	<p>Records media calls according to CID.</p> <ul style="list-style-type: none"> <li>Packet Type: ALL = all media related (internal DSP packets, RTP, RTCP, T38, events) of a certain call. ALL-WITH-PCM = all media-related and PCM traffic of a certain call.</li> <li>Channel-Id: start of range channel ID number for recording. (<b>Note:</b> Currently, only 1 channel can be recorded for digital devices.) To Channel-Id (optional) = end of range channel ID number for recording. DebugTarget (optional): debug target list index; if not specified, the default target is used.</li> </ul>
<b>RemoveTraceRule</b>	Index	<p>Removes TraceRule from list. Index: rule index (as displayed via ListTraceRules). ALL for rule wildcard.</p>
<b>ListTraceRules</b>	--	Displays added TraceRules.

**Table 12-3: DR Activation**

Command	Parameters	Description
<b>STARTRecording</b>	--	Enables recording.
<b>STOPRecording</b>	--	Disables recording.

## 13 Glossary

**Table 13-1: Glossary of Terms**

Term	Meaning
ADPCM	Adaptive Differential PCM - voice compression
AIS	Alarm Indication Signal
A-law	Standard companding algorithm, used in European digital communications systems to optimize the dynamic range of an analog signal for digitizing.
AOR	Address of Record
AWG	American Wire Gauge
bps	Bits per second
BootP	AudioCodes Proprietary Bootstrap Loader Utility
CAS	Channel Associated Signaling
CoS	Class of Service
CMP	Compressed File (device Firmware)
cPCI	Compact PCI (Industry Standard)
CPT	Call Progress Tones
dB	Decibels
DHCP	Dynamic Host Control Protocol
DID	Direct Inward Dial
DiffServ	Differentiated Services
DNS	Domain Name System (or Server)
DR	Debug Recording
DS1	1.544 Mbps USA Digital Transmission System (see E1 and T1)
DS3	44.736 Mbps USA Digital Transmission System, Encapsulates 28 T1 streams, Also called T3
DSP	Digital Signal Processor (or Processing)
DTMF	Dual Tone Multiple Frequency (Touch Tone)
E1	2.048 Mbps European Digital Transmission System (see T1)
ETSI	European Telecommunications Standards Institute
FQDN	Fully Qualified Domain Name
GRUU	Globally Routable User Agent URIs
ICMP	Internet Control Message Protocol
IE	Information Element (ISDN layer 3 protocol, basic building block)
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange (for IPSec)
IP	Internet Protocol

Term	Meaning
IPSec	IP Security
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
ITU	International Telecommunications Union
ITU-T	Telecommunications section of the ITU
IVR	Interactive Voice Response
Jitter	Variation of interpacket timing interval
kbps	Kilobit per second. 1,000 bits per second
KLM	Numbering: K=TUG3, L=TUG2, M=TU number
LAPD	Line Access Protocol for the D-channel
LFA	Loss of Frame Alignment
LOF	Loss of Frame
Mbps	Megabit per second. Million bits per second
MIB	Management Information Base
MLPP	Multilevel Precedence and Preemption
ms or msec	Millisecond; a thousandth part of a second
MSCML	Media Server Control Markup Language
NT	Network Termination (ISDN)
MWI	Message Waiting Indicator
NAPTR	Naming Authority Pointer
NAT	Network Address Translation
NFAS	Non-Facility Associated Signalling (ISDN PRI)
NFS	Network File System
NPI	Numbering Plan Indicator
NTP	Network Time Protocol
OAMP	Operations, Administration, Maintenance and Provisioning
OSI	Open Systems Interconnection (Industry Standard)
PBX	Private Branch Exchange
PCI	Personal Computer Interface (Industry Standard)
PCM	Pulse-Code Modulation
PI	Progress Indicator
PKI	Public-Key Infrastructures
POTS	Plain Old Telephone System or Service
PRT	Prerecorded Tones (File)
PRI	Primary Rate Interface (ISDN)
PSTN	Public Switched Telephone Network

Term	Meaning
PVID	Port VLAN ID (VLAN ID assignment to Ethernet packet by switch)
QoS	Quality of Service
RAI	Remote Alarm Indication
RFC	Request for Comment issued by IETF
RTCP	Real-Time Transport (RTP) Control Protocol
RTP	Real-Time Transport Protocol
SA	Security Associations (contains encryption keys and profile used by IPSec to encrypt the IP stream)
SDH	Synchronous Digital Hierarchy
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SME	Small and Medium-sized Enterprise
SNMP	Simple Network Management Protocol
SONET	Synchronous Optical Network
SRTP	Secure Real-Time Transport Protocol
SRV	Service Record
SSH	Secure Shell
SSL	Secure Socket Layer (also known as Transport Layer Security (TLS))
STUN	Simple Traversal of UDP through NATs
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol
TE	Terminal Equipment (ISDN)
TDM	Time-Division Multiplexing
TFTP	Trivial File Transfer Protocol
TLS	Transport Layer Security
TON	Type of Numbering
UA	SIP User Agent
UDP	User Datagram Protocol
URI (SIP URIs)	SIP Uniform Resource Indicators
VBD	Voice-band data
VLAN	Virtual Local Area Network
VoIP	Voice over Internet Protocol
VoP	Voice over Packet(s)
VP	Voice Prompts (File)
VPN	Virtual Private Network
VT1.5	Virtual Tributary

# **Product Reference Manual**

**Version 5.6**