

Reference Manual for SIP Gateways and Media Servers

Version 5.2



Notice

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

- Media Gateways Systems: MediaPack series, Mediant 1000, Mediant 2000, Mediant 3000
- Media Servers Systems: IPmedia 2000, IPmedia 3000
- cPCI Blades: TP-1610, IPM-1610, TP-6310, IPM-6310
- PCI Boards: TP-260, IPM-260

Information contained in this document is believed to be accurate and reliable at the time of printing. However, due to ongoing product improvements and revisions, AudioCodes cannot guarantee accuracy of printed material after the Date Published nor can it accept responsibility for errors or omissions. Updates to this document and other documents can be viewed by registered Technical Support customers at <http://www.audiocodes.com/> under Support / Product Documentation.

© Copyright 2007 AudioCodes Ltd. All rights reserved.

This document is subject to change without notice.

Date Published: Aug-16-2007

Date Printed: Aug-19-2007



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, CTI² 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.

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-897xx (where xx represents the document version)	Mediant 3000 & TP-6310 SIP User's Manual
LTRT-898xx	IPmedia 3000 & IPM-6310 SIP User's Manual
LTRT-688xx	Mediant 2000 & TP-1610 & TP-260-UNI SIP User's Manual
LTRT-588xx	IPmedia 2000 & IPM-1610 & IPM-260-UNI SIP User's Manual
LTRT-833xx	Mediant 1000 SIP User's Manual
LTRT-654xx	MP-11x & MP-124 SIP User's Manual
LTRT-665xx	CPE SIP Configuration Guide for IP Voice Mail
LTRT-690xx	Mediant 3000 & Mediant 2000 & TP Series SIP Release Notes
LTRT-590xx	IPmedia Series & IPM Series SIP Release Notes
LTRT-831xx	Mediant 1000 SIP Release Notes
LTRT-656xx	MP-11x & MP-124 SIP Release Notes
LTRT-701xx	Mediant 2000 MGCP-MEGACO-SIP Fast Track Guide
LTRT-835xx	Mediant 1000 MEGACO-SIP Fast Track Guide
LTRT-598xx	MP-11x & MP-124 SIP Fast Track Guide



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

- **Device:** refers to all the AudioCodes' products listed in the Note above.
- **3000 Series:** refers to Mediant 3000, TP-6310, IPmedia 3000, and IPM-3000.
- **2000 Series:** refers to Mediant 2000, TP-1610, IPmedia 2000, IPM-1610, TP-260, and IPM-260.
- **IPmedia Series:** refers to IPmedia 3000, IPM-6310, IPmedia 2000, IPM-1610, and IPM-260.
- **Analog:** refers to the MediaPack series and the Mediant 1000 analog interface.
- **Digital:** refers to all products except MediaPack and the Mediant 1000 analog interface.
- **MediaPack or MediaPack Series:** 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

Table of Contents

1	Introduction	15
2	Using BootP / DHCP.....	17
2.1	BootP/DHCP Server Parameters	17
2.2	Using DHCP	17
2.3	Using BootP	19
2.3.1	Upgrading the Device	19
2.3.2	Vendor Specific Information Field.....	19
3	Command-Line Interface Based Management.....	21
3.1	Starting a CLI Management Session	21
3.2	CLI Navigation Concepts	22
3.3	Commands.....	22
3.3.1	General Commands.....	23
3.3.2	Call Detail Record (CDR) Commands	27
3.3.3	Configuration Commands.....	28
3.3.4	Management Commands	30
3.3.5	PSTN Commands.....	31
3.4	Debug Recording (DR).....	37
3.4.1	Collecting DR Messages	37
3.4.2	Activating DR	37
3.4.3	DR Command Reference	38
4	SNMP-Based Management	43
4.1	SNMP Standards and Objects	43
4.1.1	SNMP Message Standard	43
4.1.2	SNMP MIB Objects.....	44
4.1.3	SNMP Extensibility Feature	45
4.2	Carrier-Grade Alarm System.....	45
4.2.1	Active Alarm Table.....	45
4.2.2	Alarm History	46
4.3	Topology MIB - Objects.....	46
4.3.1	Physical Entity - RFC 2737	46
4.3.2	IF-MIB - RFC 2863	46
4.4	Cold Start Trap.....	50
4.5	Performance Measurements.....	50
4.5.1	Total Counters	51
4.6	TrunkPack-VoP Series Supported MIBs	52
4.7	Traps.....	57
4.8	SNMP Interface Details.....	60
4.8.1	SNMP Community Names.....	60
4.8.1.1	Configuring Community Strings via the Web	60
4.8.1.2	Configuring Community Strings via the ini File	60
4.8.1.3	Configuring Community Strings via SNMP	60
4.8.2	SNMPv3 USM Users	62
4.8.2.1	Configuring SNMPv3 Users via the ini File	63
4.8.2.2	Configuring SNMPv3 Users via SNMP	64

4.8.3	Trusted Managers.....	65
4.8.3.1	Configuring Trusted Managers via ini File.....	65
4.8.3.2	Configuring Trusted Managers via SNMP.....	65
4.8.4	SNMP Ports.....	66
4.8.5	Multiple SNMP Trap Destinations.....	67
4.8.5.1	Configuring Trap Managers via Host Name.....	67
4.8.5.2	Configuring Trap Managers via the ini File.....	67
4.8.5.3	Configuring Trap Managers via SNMP.....	69
4.8.5.4	SNMP Manager Backward Compatibility.....	70
4.9	Dual Module Interface.....	70
4.10	SNMP NAT Traversal.....	70
4.11	Media Server Configuration.....	71
4.12	Systems.....	72
4.13	High Availability Systems.....	72
4.14	SNMP Administrative State Control.....	73
4.14.1	Node Maintenance.....	73
4.14.2	Graceful Shutdown.....	73
4.15	AudioCodes' Element Management System.....	74
4.16	SNMP Traps.....	75
4.16.1	Alarm Traps.....	75
4.16.1.1	Component: Chassis#0.....	76
4.16.1.2	Component: Interfaces#0/Sonet#<m>.....	78
4.16.1.3	Component: System#0<n> and Board#0<n>.....	80
4.16.1.4	Component: AlarmManager#0.....	85
4.16.1.5	Component: AudioStaging#0.....	86
4.16.1.6	Component: SS7#0.....	87
4.16.1.7	Component: System#0/Module#<m>.....	92
4.16.2	Log Traps (Notifications).....	96
4.16.3	Other Traps.....	97
4.16.4	Trap Varbinds.....	98
4.16.5	Customizing Trap's Enterprise OID.....	99
5	Configuration Files.....	101
5.1	Configuring the Call Progress Tones File.....	101
5.2	Configuring the Distinctive Ringing Section of the ini File.....	104
5.2.1	Examples of Ringing Signals.....	105
5.3	Prerecorded Tones (PRT) File.....	106
5.4	Voice Prompts File.....	107
5.5	CAS Protocol Configuration Files.....	108
5.6	Coefficient Configuration File.....	108
5.7	Dial Plan File.....	109
5.8	User Information File.....	111
6	Automatic Configuration Options.....	113
6.1	Local Configuration Server with BootP/TFTP.....	113
6.2	DHCP-based Configuration Server.....	114
6.3	HTTP-based Automatic Updates.....	114
6.4	Configuration using DHCP Option 67.....	115
6.5	Configuration using FTP or NFS.....	116

6.6	TFTP Configuration using DHCP Option 66	117
6.7	Configuration using AudioCodes EMS	117
7	Security	119
7.1	IPSec and IKE	119
7.1.1	IKE	120
7.1.2	IPSec	120
7.1.3	IPSec and IKE Configuration Table's Confidentiality	121
7.2	SSL / TLS	122
7.2.1	SIP Over TLS (SIPS)	122
7.2.2	Secured HTTPS Embedded Web Server Configuration	122
7.2.3	Secured Telnet	124
7.3	SRTP	124
7.4	RADIUS Login Authentication	125
7.4.1	Setting Up a RADIUS Server	126
7.4.2	Configuring RADIUS Support	127
7.5	Internal Firewall	129
7.6	Network Port Usage	131
7.7	Recommended Practices	132
7.8	Legal Notice	132
8	RTP Control Protocol Extended Reports (RTCP-XR)	133
9	RTP / RTCP Payload Types and Port Allocation	137
9.1	Payload Types Defined in RFC 3551	137
9.2	Defined Payload Types	138
9.3	Default RTP / RTCP / T.38 Port Allocation	140
10	CAS Protocol Table	141
10.1	Constructing CAS Protocol Tables for CAS-Terminated Protocols	141
10.2	Protocol Table Elements	141
10.2.1	INIT Variables	142
10.2.2	Actions	142
10.2.3	Functions	142
10.2.4	States	142
10.3	Reserved Words	144
10.4	State Line Structure	144
10.5	Action / Event	145
10.5.1	User Command Oriented Action / Event	145
10.5.2	CAS Change Oriented Events	146
10.5.3	Timer Oriented Events	146
10.5.4	Counter Oriented Events	146
10.5.5	IBS Oriented Events	147
10.5.6	DTMF/MF Oriented Events	147
10.5.7	Operator Service Events (up to GR-506)	150
10.6	Function	151
10.7	Parameters	151

10.8	Next State	153
10.9	Changing the Script File.....	154
10.9.1	MFC-R2 Protocol	154
11	SS7 Tunneling	157
11.1	MTP2 Tunneling Technology	158
11.2	SS7 Characteristics.....	159
11.3	SS7 Parameter Tables.....	159
11.4	SS7 Parameters.....	160
11.5	SS7 MTP2 Tunneling ini File Example.....	168
11.6	Configuring SS7 Tunneling	171
11.6.1	Configuring MTP2 Attributes	172
11.6.2	Configuring SS7 Signaling Node Timers.....	174
11.6.3	Configuring Link-Set Timers	178
11.6.4	Configuring Links	180
11.6.5	Configuring SS7 Signaling Nodes	182
11.6.6	Configuring Sigtran Group IDs	184
11.6.7	Configuring Sigtran Interface IDs	186
12	Accessory Programs and Tools	189
12.1	BootP/TFTP Server Configuration Utility.....	189
12.1.1	When to Use the BootP/TFTP	189
12.1.2	An Overview of BootP.....	190
12.1.3	Key Features	190
12.1.4	Specifications.....	190
12.1.5	Installation.....	191
12.1.6	Loading the cmp File, Booting the Device.....	191
12.1.7	BootP/TFTP Application User Interface.....	192
12.1.8	Function Buttons on the Main Screen	192
12.1.9	Log Window	193
12.1.10	Setting the Preferences	194
12.1.10.1	BootP Preferences	195
12.1.10.2	TFTP Preferences	195
12.1.11	Configuring the BootP Clients	196
12.1.11.1	Client Parameters.....	196
12.1.11.2	Using Command Line Switches	198
12.1.11.3	Adding Clients	199
12.1.11.4	Editing Client Parameters	200
12.1.11.5	Deleting Clients	200
12.1.11.6	Testing the Client	201
12.1.12	Managing Client Templates.....	202
12.2	TrunkPack Downloadable Conversion Utility	204
12.2.1	Converting a CPT ini File to a Binary dat File.....	205
12.2.2	Creating a Loadable Voice Prompts File.....	206
12.2.3	Creating a Loadable CAS Protocol Table File.....	208
12.2.4	Creating a Dial Plan File.....	210
12.2.5	Encoding / Decoding an ini File	212
12.2.6	Creating a Loadable Prerecorded Tones File	214
12.3	Call Progress Tones Wizard	216
12.3.1	Installation.....	216
12.3.2	Initial Settings	217
12.3.3	Recording Screen - Automatic Mode.....	218
12.3.4	Recording Screen - Manual Mode.....	220

12.3.5	Call Progress Tones ini and dat Files.....	221
12.3.6	Adding a Reorder Tone to the CPT File	223
13	Installing and Configuring Apache HTTP Server.....	225
13.1	Windows 2000/XP Operation Systems	225
13.2	Linux Operation Systems.....	227
14	Diagnostics.....	229
14.1	Self-Testing.....	229
14.2	FXS Line Testing.....	230
14.3	Syslog Support.....	233
14.3.1	Syslog Servers.....	233
14.3.2	Operation	234

List of Figures

Figure 5-1: Example of a User Information File.....	112
Figure 7-1: IPsec Encryption	119
Figure 11-1: M2UA Architecture.....	157
Figure 11-2: M2TN Architecture	158
Figure 11-3: Protocol Architecture for MTP2 Tunneling.....	158
Figure 11-4: MTP2 Attributes Screen.....	172
Figure 11-5: SS7 Signaling Node Timers.....	175
Figure 11-6: SS7 Link-set Timers Screen	178
Figure 11-7: Links Screen	180
Figure 11-8: SS7 Signaling Nodes Screen.....	182
Figure 11-9: SS7 Sigtran Group IDs Screen.....	184
Figure 11-10: SS7 Sigtran Interface IDs	186
Figure 12-1: Main Screen.....	192
Figure 12-2: Reset Screen	193
Figure 12-3: Preferences Screen	194
Figure 12-4: Client Configuration Screen.....	196
Figure 12-5: Templates Screen.....	202
Figure 12-6: TrunkPack Downloadable Conversion Utility Main Screen	204
Figure 12-7: Call Progress Tones Screen	205
Figure 12-8: Voice Prompts Screen	207
Figure 12-9: File Data Window.....	208
Figure 12-10: Call Associated Signaling (CAS) Screen.....	209
Figure 12-11: Dial Plan Screen	211
Figure 12-12: Encode / Decode ini File(s) Screen	212
Figure 12-13: Prerecorded Tones Screen.....	214
Figure 12-14: File Data Window.....	215
Figure 12-15: Initial Settings Screen	217
Figure 12-16: Recording Screen - Automatic Mode.....	218
Figure 12-17: Recording Screen after Automatic Detection.....	219
Figure 12-18: Recording Screen - Manual Mode	220
Figure 14-1: Analog Line Testing Confirmation Box	232
Figure 14-2: FXS Line Testing for Channel 1.....	232

List of Tables

Table 2-1: Vendor Specific Information Field	20
Table 2-2: Structure of the Vendor Specific Information Field	20
Table 3-1: CLI Commands and Corresponding Options	22
Table 3-2: General Commands	23
Table 3-3: Sub-commands of Call Detail Record (CDR) Command	27
Table 3-4: Configuration Commands	28
Table 3-5: Management commands.....	30
Table 3-6: PSTN Commands	31
Table 3-7: Client Setup Commands	38
Table 3-8: Trace Rules	39
Table 3-9: DR Activation.....	41
Table 4-1: DS1 Digital Interfaces	47
Table 4-2: Ethernet (Gigabit for 3000 Series) Interface	47
Table 4-3: SONET /SDH Interfaces (3000 Series Only)	49
Table 4-4: DS3 Interfaces (3000 Series Only)	49
Table 4-5: Default TCP/UDP Network Port Numbers	57
Table 4-6: SNMP Predefined Groups.....	60
Table 4-7: SNMPv3 Security Levels.....	62
Table 4-8: SNMPv3 Predefined Groups.....	62
Table 4-9: SNMPv3 Table Columns Description.....	63
Table 4-10: acFanTrayAlarm Alarm Trap.....	76
Table 4-11: acPowerSupplyAlarm Alarm Trap	76
Table 4-12: acUserInputAlarm Alarm Trap.....	77
Table 4-13: acPEMAlarm Alarm Trap (Applicable only to 3000 Series)	77
Table 4-14: AcSonetSectionLOFAlarm Alarm Trap	78
Table 4-15: AcSonetSectionLOSAAlarm Alarm Trap	79
Table 4-16: AcSonetLineAISAlarm Alarm Trap.....	79
Table 4-17: AcSonetLineRDIAAlarm Alarm Trap	80
Table 4-18: acBoardFatalError Alarm Trap	80
Table 4-19: acBoardConfigurationError Alarm Trap	81
Table 4-20: acBoardTemperatureAlarm Alarm Trap	81
Table 4-21: acBoardEvResettingBoard Alarm Trap	82
Table 4-22: acBoardEthernetLinkAlarm Alarm Trap (Applicable only to 2000 Series, Mediant 1000, and MediaPack).....	82
Table 4-23: acBoardCallResourcesAlarm Alarm Trap (Applicable only to Mediant 1000)	83
Table 4-24: acBoardControllerFailureAlarm Alarm Trap (Applicable only to Mediant 1000)	83
Table 4-25: acBoardOverloadAlarm Alarm Trap (Applicable only to Mediant 1000)	84
Table 4-26: acFeatureKeyError Alarm Trap (Applicable only to Digital devices).....	84
Table 4-27: acSAMissingAlarm Alarm Trap (Applicable only to the 3000 Series devices).....	84
Table 4-28: acActiveAlarmTableOverflow Alarm Trap	85
Table 4-29: acAudioProvisioningAlarm Alarm Trap	86
Table 4-30: acSS7LinkStateChangeAlarm Trap	87
Table 4-31: acSS7LinkCongestionStateChangeAlarm Trap.....	88
Table 4-32: acSS7LinkInhibitStateChangeAlarm Trap	89
Table 4-33: acSS7LinkBlockStateChangeAlarm Trap	89
Table 4-34: acSS7LinkSetStateChangeAlarm Trap.....	90
Table 4-35: acSS7RouteSetStateChangeAlarm Trap.....	90
Table 4-36: acSS7SNSetStateChangeAlarm Trap	91
Table 4-37: acSS7RedundancyAlarm Trap.....	91
Table 4-38: acHASystemFaultAlarm Trap.....	92
Table 4-39: acHASystemConfigMismatchAlarm Trap.....	93
Table 4-40: acHASystemSwitchOverAlarm Trap	93
Table 4-41: acBoardTemperatureAlarm Trap	94
Table 4-42: acBoardEthernetLinkAlarm Trap.....	95

Table 4-43: acKeepAlive Log Trap	96
Table 4-44: acPerformanceMonitoringThresholdCrossing Log Trap	96
Table 4-45: acHTTPDownloadResult Log Trap	97
Table 4-46: coldStart Trap	97
Table 4-47: authenticationFailure Trap	97
Table 4-48: acBoardEvBoardStarted Trap	97
Table 4-49: AcDChannelStatus Trap (Applicable only to 3000 Series and 2000 Series devices).....	98
Table 5-1: User Information Items	112
Table 7-1: Default TCP/UDP Network Port Numbers	131
Table 8-1: RTCP-XR Published VoIP Metrics	133
Table 9-1: Packet Types Defined in RFC 3551	137
Table 9-2: Defined Payload Types	138
Table 9-3: Default RTP/RTCP/T.38 Port Allocation	140
Table 10-1: ST_DIAL: Table Elements.....	142
Table 10-2: User Command Orientated Action / Event.....	145
Table 10-3: CAS Change Orientated Events	146
Table 10-4: Time-Orientated Events	146
Table 10-5: Counter Orientated Events.....	146
Table 10-6: IBS Orientated Events.....	147
Table 10-7: DTMF / MF Orientated Events	147
Table 10-8: Actions / Events Causing MFC-R2 Table to Send Correct MF Tone to Backward Direction.....	149
Table 10-9: Operator Service Events (Up to GR-506)	150
Table 10-10: Available User Functions and Corresponding Parameters.....	151
Table 10-11: Parameters Associated with Sending Digits	152
Table 11-1: SS7 Parameters.....	160
Table 11-2: MTP2 Parameters	173
Table 11-3: SS7 Signaling Node Timers Parameters	176
Table 11-4: SS7 Link-Set Timers Parameters	179
Table 11-5: SS7 Links Parameters	181
Table 11-6: SS7 Signaling Nodes Parameters	183
Table 11-7: Sigtran Group IDs Parameters.....	185
Table 11-8: Sigtran Interface IDs Parameters.....	187
Table 12-1: Command Line Switch Descriptions	198

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 in the device's *User's Manual* and includes, for example, detailed descriptions on various supported features, AudioCodes proprietary applications, advanced configuration methods, and diagnostic methods.

This reference manual relates to the following devices:

- Mediant 3000 gateway hosting a single TP-6310 blade
- Mediant 3000 High-Availability (1+1) gateway hosting dual TP-6310 blades
- TP-6310 cPCI blade
- IPmedia 3000 media server hosting a single IPM-6310 blade
- IPmedia 3000 High Availability (1+1) media server hosting dual IPM-6310 blades
- IPM-6310 cPCI blade
- Mediant 2000 gateway
- TP-1610 cPCI blade
- TP-260/UNI PCI board
- IPmedia 2000 media server
- IPM-1610 cPCI blade
- IPM-260/UNI PCI board
- Mediant 1000 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 configure the device, please refer to the device's *User's Manual*.

Reader's Notes

2 Using BootP / DHCP

The device uses the Bootstrap Protocol (BootP) and the Dynamic Host Configuration Protocol (DHCP) to obtain its networking parameters and configuration automatically 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.

2.1 BootP/DHCP Server Parameters

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

- **IP address, subnet mask:** These **mandatory** parameters are sent to the device every time a BootP/DHCP process occurs.
- **Default Gateway IP address:** An optional parameter that is sent to the device only if configured in the BootP/DHCP server.
- **TFTP server IP address:** An optional parameter that contains the address of the TFTP server from which the firmware (*cmp*) and *ini* files are loaded.
- **DNS server IP address (primary and secondary):** Optional parameters that contain the IP addresses of the primary and secondary DNS servers. These parameters are available only in DHCP and from Boot version 1.92.
- **Syslog server IP address:** An optional parameter that is sent to the device only if configured. This parameter is available only in DHCP.
- **SIP server IP address:** Two optional parameters that are sent to the device only if configured. These parameters are available only in DHCP.
- **Firmware file name:** An optional parameter that contains the name of the firmware file to be loaded to the device via TFTP.
- **ini file name:** An optional parameter that contains the name of the *ini* file to be loaded to the device via TFTP.

2.2 Using DHCP

When the device is configured to use DHCP (Embedded Web Server -- 'Configuring the IP Settings' in the *User's Manual* or *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.


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 IP address of the device 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 Embedded Web Server/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.

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

- Starting with Boot version 1.92, 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 from Hex to decimal). 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 Application User Interface' on page 192).
- Use a serial communication software (refer to 'Assigning an IP Address Using the CLI' in the device's *User's Manual*).
- Contact your System Administrator.

2.3 Using BootP

2.3.1 Upgrading the Device

When upgrading the device (loading new software onto 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 Software Upgrade wizard, which is available through the Embedded Web Server (refer to 'Software Upgrade Wizard' in the device's *User's Manual*).



Note: To save the *cmp* file to 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 198.

2.3.2 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 configuration utility displays this information in the 'Client Info' column (refer to 'BootP/TFTP Application User Interface' on page 192).



Note: This option is not available on DHCP servers.

The Vendor Specific Information field is disabled by default. To enable / disable this feature, configure the *ini* file parameter ExtBootPReqEnable (refer to 'System Parameters' in the device's *User's Manual*) or use the **-be** command line switch (refer to 'Using Command Line Switches' on page 198).

The following table details the vendor specific information field according to device types:

Table 2-1: Vendor Specific Information Field

Tag #	Description	Value	Length
220	Device Type	<ul style="list-style-type: none"> ▪ #02 = IPM-6310; TP-6310; TP-1610; IPM-1610 ▪ #05 = TP-260; IPM-260 ▪ #09 = IPmedia 3000; Mediant 3000; Mediant 1000; ▪ #13 = MP-124 ▪ #14 = MP-118 ▪ #15 = MP-114 ▪ #16 = MP-112 	1
221	Current IP Address	XXX.XXX.XXX.XXX	4
222	Burned Boot Software Version	X.XX	4
223	Burned <i>cmp</i> Software Version	XXXXXXXXXXXX	12
224	Geographical Address	0 - 31 Note: Applicable only to IPM-260 and TP-260.	1
225	Chassis Geographical Address	0 - 31 Note: Applicable only to IPM-260 and TP-260.	1
228	Indoor / Outdoor	<ul style="list-style-type: none"> ▪ #0 = Indoor ▪ #1 = Outdoor Notes: <ul style="list-style-type: none"> ▪ Applicable only to Mediant 1000 analog modules and MediaPack gateways. ▪ Indoor is applicable only for FXS interfaces; Outdoor is applicable only for FXO interfaces. 	1
230	Analog Channels	<ul style="list-style-type: none"> ▪ Mediant 1000: 4; 8; 12; 16; 20; 24 ▪ MediaPack: 2; 4; 8; 24 Note: Applicable only to Mediant 1000 analog modules and MediaPack gateways.	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

3 Command-Line Interface Based Management

The command-line interface (CLI) is available through a Telnet or an SSH session with the device's management interface. It provides a predefined set of commands with a choice of options that comprehensively cover the maintenance tasks required for the device, including the following:

- Displaying status and configuration
- Modifying configuration
- Debugging



Note: The CLI is not applicable to the TP-6310, IPM-6310, TP-1610, and IPM-1610 blades.

3.1 Starting a CLI Management Session

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

1. Enable CLI (Telnet or SSH) using either the *ini* file, Embedded Web Server, or SNMP.
 - **ini file:** Configure the following *ini* file parameters as shown below:
 - ◆ TelnetServerEnable = 1
 - ◆ SSHServerEnable = 1
 - **Embedded Web Server:** Set the parameter 'Embedded Telnet Server' (under **Advanced Configuration > Network Settings > Application Settings**) to 'Enable (Unsecured)' or 'Enable Secured (SSL)' (refer to 'Accessing the CLI' in the device's *User's Manual*).
 - **SNMP:** set the objects acSysTelnetSSHServerEnable and acSysTelnetServerEnable to 'enable' (1).
2. Establish a Telnet or SSH session with the device's OAM IP address, using the system's username and password (as shown below).
 - Username: 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.


Notes:

- By default, all CLI access is disabled for security.
- The username and password are case-sensitive.
- The CLI username and password can be changed by the device administrator. Multiple users can be defined.

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
/>
```

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 (**SHow** and **PING**). 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 **PERFORMANCE** 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 in upper case (i.e., capital letters). For example, the **CHangePassWord** command can be entered by typing **chpw**.

3.3 Commands

The following table summarizes the CLI commands and their options.

Table 3-1: CLI Commands and Corresponding Options

Purpose	Commands	Description
Help	h	Displays the help for a specific command, action, or parameter.
Navigation	cd	Enters another directory.
	cd root	Navigates to the root directory (/).
	..	Goes up one level.
	exit	Terminates the CLI session.
Status	show	Displays the MG / MS operational status.

Purpose	Commands	Description
	ping	Sends Internet Control Message Protocol (ICMP) echo request packets from the device to a defined IP address.
Configuration	/conf/scp	Sets a value for the specific parameter.
	/conf/gcp	Queries a configuration parameter value.
	/conf/rfs	Restores factory defaults.
	/conf/sar	Restarts the device.

3.3.1 General Commands

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

Table 3-2: General Commands

Command	Short Format	Arguments	Description
SHow	sh	info mgcp tdm dsp ip log	Displays operational data. The individual subcommands are documented below.
SHow INFO	sh info	-	Displays device hardware information, versions, uptime, temperature reading, and the last reset reason.
SHow TDM	sh tdm	status perf summary	Displays the alarm status and performance statistics for E1/T1 trunks.
SHow DSP	sh dsp	status perf	Displays status and version for each DSP device, along with overall performance statistics.
SHow IP	sh ip	conf perf route	Displays IP interface status and configuration, along with performance statistics.
SHow LOG	sh log	[stop]	Displays (or stops displaying) Syslog messages inside the CLI session.
SHow VOICEPROMPT	sh voiceprompt	numofentries entries	Displays information about Voice-Prompt table.
SHow TONES	sh tones	tone type (cpt udt rngt prt) tone index	Displays properties of a tone according to its type and index in the tone database. If <tone index> is absent, general tones information of the tones is displayed. If <tone index> is negative, the next tone is displayed.
/ControlProtocol/Call DetailReport	/cp/cdr	start show send stop	Generates CDR records when a voice call ends. Refer to the 'Call Detail Reports' subsection for additional details.

Command	Short Format	Arguments	Description
PING	ping	[-n count] [-l size] [-w timeout] [-p cos] ip-address	Sends ICMP echo request packets to a specified IP address. <ul style="list-style-type: none"> ▪ count: number of packets to send. ▪ size: payload size in each packet. ▪ timeout: time (in seconds) to wait for a reply to each packet. ▪ cos: class-of-service (as per 802.1p) to use

Example:

```

/>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
  SHow TONES          Displays information about special tones
  SHow VOICEPROMPT    Displays information about Voice Prompt
table
/>sh info
Board type: device 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: 491096AE3 version 10713
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
DSP device 19: Inactive
DSP device 20: Inactive
DSP device 21: Inactive
/>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 tones cpt
Call Progress Tone - General information:
-----
Num of Tones: 20, (20 loaded to dsp)
Num of Frequencies: 0
High Energy Threshold=0
Low Energy Threshold=35
Max Frequency Deviation=10
Total Energy Threshold=44
Twist=10
SNR=15
/>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
/>show voiceprompt numofentries
First Used VoicePrompt Index: 0  First Free VoicePrompt Index: 18
/>show voiceprompt entries 11
VP-00011 Coder: 36 ,Length 7245
VP-00012 Coder: 48 ,Length 12930
VP-00013 Coder: 50 ,Length 5488
VP-00014 Coder: 53 ,Length 7486
VP-00015 Coder: 57 ,Length 15939
VP-00016 Coder: 21 ,Length 9207
VP-00017 Coder: 43 ,Length 28320
/>show voiceprompt entries 9 4
VP-00009 Coder: 32 ,Length 3812
VP-00010 Coder: 34 ,Length 5324
VP-00011 Coder: 36 ,Length 7245
VP-00012 Coder: 48 ,Length 12930
    
```

3.3.2 Call Detail Record (CDR) Commands

The command `/cp/cdr` can be used to generate Call Detail Records (CDR) when a voice call terminates. The following sub-commands are available:

Table 3-3: Sub-commands of Call Detail Record (CDR) Command

Subcommand	Description
start [syslog file both]	Starts generating CDR records. If 'syslog' is specified, the records are sent to the Syslog. If 'file' is specified, the records are collected in a file that can be viewed in the CLI or transferred to an NFS host using the '/cp/cdr send' command. If 'both' is specified, the records are sent to both the Syslog and the file.
show	Displays the current CDR file (history of last calls). Note: In a high-load system, the file is overwritten relatively quickly as it can hold approximately 1,000 CDRs (possibly less than a minute of activity). Using the '/cp/cdr show' command can yield unpredictable results.
send <nfs_location>	Sends the CDR file to an NFS host. The remote NFS file system must be pre-defined and mounted. The argument to this command must be a URI (Uniform Resource Identifier) in the form: file://server-ip-address/path/filename Note: The URI is case-sensitive.
stop	Stops generation of CDR records and clears the CDR file.

3.3.3 Configuration Commands

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

Table 3-4: Configuration Commands

Command	Short Format	Arguments	Description
CHangePassWord	/conf/chpw	old-pw new-pw	Changes the system password.
GetConfigParam	/conf/gcp	parameter ip	Queries the value of an <i>ini</i> file parameter. Refer to 'ini File Configuration' in the device's <i>User's Manual</i> for a list of the supported configuration parameters. 'gcp ip' displays the current IP configuration of the device.
SetConfigParam	/conf/scp	parameter + value	Sets a configuration parameter to a specified value. Refer to 'ini File Configuration' in the device's <i>User's Manual</i> for a list of supported configuration parameters.
SetConfigParam IP	/conf/scp ip	ip-addr subnet def-gw	Sets the IP address, subnet mask, and default gateway address of the device, on-the-fly. Note: This command may cause disruption of service. The CLI session may disconnect since the device changes its IP address.
GetParamDescription	/conf/gpd	parameter	Displays a short description of an <i>ini</i> file parameter.
RestoreFactorySettings	/conf/rfs		Restores all factory settings.
SaveAndRestart	/conf/sar		Saves all current configuration into non-volatile memory, and restarts the device.
ConfigFile	/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 within the CLI session. Note: The sub-command 'view' displays the file page-by-page. The sub-command 'get' displays the file without breaks.
AutoUPDate	/conf/aupd		Checks for a new <i>ini</i> file, or pending software upgrade configured via the IniFileUrl and CmpFileUrl parameters. Refer to 'Automatic Update Mechanism' in the device's <i>User's Manual</i> for additional information.

Example:

```
/>conf
SaveAndReset RestoreFactorySettings SetConfigParam
GetParameterDescription GetConfigParam CHangePassWord ConfigFile
AutoUPDate
/CONFIguration>gpd SyslogServerIP
SYSLOGSERVERIP = Defines the Syslog server IP address in dotted
format notation.
e.g., 192.10.1.255
SaveAndReset RestoreFactorySettings SetConfigParam
GetParameterDescription GetConfigParam CHangePassWord ConfigFile
AutoUPDate|
/CONFIguration>gcp syslogserverip
Result: SYSLOGSERVERIP = 10.31.4.51
SaveAndReset RestoreFactorySettings SetConfigParam
GetParameterDescription GetConfigParam CHangePassWord ConfigFile
AutoUPDate
/CONFIguration>scp syslogserverip 10.31.2.10
Old value: SYSLOGSERVERIP = 10.31.4.51
New value: SYSLOGSERVERIP = 10.31.2.10
SaveAndReset RestoreFactorySettings SetConfigParam
GetParameterDescription GetConfigParam CHangePassWord ConfigFile
AutoUPDate
/CONFIguration>cf set
Enter data below. Type a period (.) on an empty line to finish.
EnableSyslog = 1
SyslogServerIP = 10.31.2.10
.
INI File replaced.
SaveAndReset RestoreFactorySettings SetConfigParam
GetParameterDescription GetConfigParam CHangePassWord ConfigFile
AutoUPDate
/CONFIguration>..
MGmt/ CONFIguration/ IPNetworking/ TPApp/ BSP/
SHow PING
/>
```

3.3.4 Management Commands

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

Table 3-5: Management commands

Command	Short Format	Arguments	Description
/MGmt/FAult/ListActive	/mg/fa/lac		Displays the list of active alarms.
/MGmt/FAult/ListHistory	/mg/fa/lh		Displays the alarm history table.
/MGmt/PERformance	/mg/perf	basic control dsp net ds1 ss7 reset	Displays performance statistics. '/mg/perf reset' clears all statistics to zero.

Example:

```

/>mg
FAult/
PERformance
/MGmt>fa
ListHistory ListActive
/MGmt/FAult>lac
  1. Board#1                1 major      Board Config
Error: PSTN Trunk Validation Check Warning - TDMBusClockSource is
set to Netw
  2. Board#1/EthernetLink#0  9 major      Ethernet link
alarm. Redundant Link (Physical port #2) is down.
ListHistory ListActive
/MGmt/FAult>lh
  1. Board#1                1 major      Board Config
Error: PSTN Trunk Validation Check Warning - TDMBusClockSource is
set to Netw
  2. Board#1/EthernetLink#0  9 major      Ethernet link
alarm. Redundant Link (Physical port #2) is down.
ListHistory ListActive
/MGmt/FAult>
  
```

3.3.5 PSTN Commands

The commands under the 'PSTN' directory display the current status of the interface, and allows you to perform various PSTN actions.

Table 3-6: PSTN Commands

Command	Short Format	Arguments	Description
/PStn/PHsical/ PstnQueryTrunkStat us	/ps/ph/pqts	TrunkId	<p>Returns the current alarm status, configuration status, and the loop type of a specific trunk.</p> <p>Return values:</p> <ul style="list-style-type: none"> ▪ LOS (Loss of Signal) ▪ LOF (Loss of Frame alignment) ▪ RAI (Remote Alarm Indication) ▪ AIS (Alarm Indication Signal) ▪ RAI_CRC (Reception of RAI and continuous CRC error report) <p>For all these alarms: 0 = no alarm; 1 = alarm on</p> <p>TrunkStatus: 0 = TrunkStatusEquipped 1 = TrunkStatusStopped 2 = TrunkStatusDeleted</p> <p>LoopBackStatus: 0 = NO_LOOPS 1 = REMOTE_LOOP 2 = LINE_PAYLOAD_LOOP 3 = LOCAL_ALL_CHANNELS_LOOP 4 = LOCAL_SINGLE_CHANNEL_LOOP\ 10 = PRBS_START 11 = PRBS_STOP</p> <p>TrkMtc.AlarmBitMap: The trunk alarm bitmap represents the current alarms on a trunk. Note that multiple alarms may coexist on a specific trunk simultaneously. Thus, the use of a bitmap rather than singular values. The alarms that are represented by the bitmap are as follows:</p> <ul style="list-style-type: none"> ▪ Rx RAI - 0x02 (bit #1 is on starting from bit#0 as LSB) ▪ Rx AIS: 0x08 (bit #3) ▪ Rx LOF: 0x20 (bit #5) ▪ Rx LOS: 0x40 (bit #6) <p>NoMultiframeAlignment: Checks the No Multiframe Alignment Found bit in the framer. This bit is valid only in E1 and only in case FramingMethod is</p>

Command	Short Format	Arguments	Description
			acE1_FRAMING_MFF_CRC4_EXT. 0 = No framing indicated (LOS/LOF/AIS alarms), or the trunk is working with CRC (multi-frame). 1 = The framer detects a framing, but not a multi-frame, meaning that the trunk is working without CRC (double-frame).
/PStn/PHsical/PstnStarTPerformanceMonitoring	/ps/ph/pstpm	TrunkId	Initiates performance monitoring data accumulation for a specific trunk. It's used to monitor an E1/T1 and process the results.
/PStn/PHsical/PstnStoPPerformanceMonitoring	/ps/ph/psppm	TrunkId	Stops performance monitoring data accumulation for a specific trunk.
/PStn/PHsical/PstnGetPerformanceMonitoring	/ps/ph/pgpm	TrunkId Interval	Retrieves a specific performance monitoring data report. Interval: 0-96: SPECIFIC_15_MINUTES_INTERVAL_RESULTS 97 = CURRENT_DAY_RESULTS 98 = PREVIOUS_DAY_TOTAL_RESULTS Return values: <ul style="list-style-type: none"> ▪ TrunkId: the trunk number (counting from 0) ▪ Interval: the interval that is related to the performance monitoring data report. ▪ AlarmIndicationSignal: Blue alarm. ▪ LossOfSignal: Red alarm ▪ LossOfFrame: <ul style="list-style-type: none"> - Occurrence of a particular density of framing error events T1: more than 2 FERs in 3 msec E1: 3 consecutive frame alignment signals (FASs) have been received with an error. ▪ FramingErrorReceived: Incorrect FAS - bits are received. ▪ RemoteAlarmReceived: Yellow Alarm. ▪ LostCRC4multiframeSync: Loss of CRC4 multiframe synchronization. ▪ CRCErrrorReceived: CRC error in the remote side. ▪ EBitErrorDetected: If CRCmultiframe mode is enabled, this field functions as submultiframe error indication counter (16 bits), which counts zeros in Si-bit position of frame 13 and 15 of every received CRC multiframe. ▪ BitError: 16-bit counter incremented with every received PRBS bit error in the PRBS synchronous state. ▪ LineCodeViolation: <ul style="list-style-type: none"> Too many successive zeros. Bipolar Violation (BPV) or Excessive Zeros (EXZ).

Command	Short Format	Arguments	Description
			<p>BPV: T1 AMI occurrence of a pulse of the same polarity as the previous pulse. B8ZS or HDB3 occurrence of a pulse of the same polarity as the previous pulse without being a part of the zero substitution. EXZ: T1 AMI occurrence of more than 15 contiguous zeros. B8ZS occurrence of more than 7 contiguous zeros.</p> <ul style="list-style-type: none"> ▪ ControlledSlip: T1 replication or deletion of the payload bits of a frame (does not cause an OOF). ▪ ErroredSeconds: (ESF and E1-CRC) A second with one or more PCV or one or more LFA or one or more CS or a detected AIS defect. ▪ ControlledSlipSeconds: A second containing one or more CS. ▪ SeverelyErroredFramingSeconds: A second with one or more LFA or a detected AIS defect. ▪ SeverelyErroredSeconds: A second with 320 or more PCV, or one or more LFA, or a detected AIS defect. ▪ BurstyErroredSeconds: A second with less than 320 and more than one PCV, no Severely Errored Frame defects and no detected incoming AIS defect. CS are not included in this parameter. ▪ UnavailableSeconds: <ul style="list-style-type: none"> - UAS are calculated by counting the number of seconds that the interface is unavailable. The interface is unavailable if 10 contiguous SESs or the onset of the condition leading to a failure. - Failure states includes these alarms: Far End Alarm(Yellow alarm), RAI, AIS, LOF, LOS. - Once unavailable, the DS1 interface becomes available at the onset of 10 contiguous seconds with no SESs if no failure is present, or if a failure is present and its clearing time is less than or equal to 10 seconds. While the interface is deemed unavailable, the only count that is incremented is UASs. ▪ PathCodingViolation: <ul style="list-style-type: none"> - ESF and E1-CRC format: CRC or FER. - D4 and E1 - no CRC format: FER. ▪ LineErroredSeconds: A second in which one or more LCVs were detected. ▪ DegradedMinutes: currently not supported. ▪ AssessedSeconds: Counts the seconds in intervals, where Performance Monitoring is enabled.

Command	Short Format	Arguments	Description
/PStn/PHsical/IsdnGetDChannelStatus	/ps/ph/igdcs	TrunkID	Retrieves the synchronization status of the PRI D-channel. Return values: TrunkId: the trunk number (counting from 0) DChannelStatus: 0 = D_CHANNEL_ESTABLISHED 1 = D_CHANNEL_NOT_ESTABLISHED
PstnLoopCommands	PS/PH/PLC	<TrunkId> <LoopCode> > <BChannel> >	Activates a loopback on a specific trunk and BChannel. For loop on all the trunk, set BChannel=(-1). LoopCode: 0 = NO_LOOPS 1 = REMOTE_LOOP (whole trunk only) 2 = LINE_PAYLOAD_LOOP (whole trunk only) 3 = LOCAL_ALL_CHANNELS_LOOP (whole trunk only) 4 = LOCAL_SINGLE_CHANNEL_LOOP 10 = PRBS_START (whole trunk only) 11 = PRBS_STOP (whole trunk only)
PstnSendAlarm	PS/PH/PSA	<TrunkId> <AlarmSendCode>	Sends an alarm on a trunk. AlarmSendCode: 0 = NO_ALARMS 1 = AIS_ALARM
/PStn/PstnCommon/PstnSetTraceLevel	/ps/pco/pstl	TrunkId BChannel TraceLevel	Sets the trace level for the specific Trunk (b-channel field is for future use). A trace for all channels at the trunk are always received. The trace levels: 0 = NO_TRE 1 = FULL_TRE 2 = LAYER3_ISDN_TRE 3 = ONLY_ISDN_Q931_MSGS_TRE 4 = LAYER3_ISDN_TRE_NO_DUPLICATION 5 = FULL_ISDN_TRE_WITH_DUPLICATION
/PStn/PstnCommon/PstnRestartRequest	/ps/pco/prr	TrunkId BChannel	Sends a restart request for a specific B-Channel (for ISDN, it is the Q.931 restart message; for CAS it's channel reset and re-initialization). You can set the B-channel field to (-1) to restart all the trunks.
/PStn/PstnCommon/PstnQueryCallState	/ps/pco/pqcs	TrunkId BChannel ConnId	Retrieves information regarding a state of a call by supplying either the TrunkId and B-Channel (and set ConnId to (-1)) or by supplying the TrunkId and ConnId (and set B-Channel to (-1)).
/PStn/CAS/CasBlockChannel	/ps/cas/cbc	TrunkId BChannel Block	Sends local-block command to the b-channel. Depends on CAS file support.
/PStn/CAS/GenerateCasFlashHook	/ps/cas/gcfh	TrunkId ConnId BChannel	Sends CAS flash-hook to the far-end user by supplying either the TrunkId and B-Channel (and set ConnId to (-1)) or by supplying the TrunkId and ConnId (and set B-Channel to (-1)). The execution of this command depends on CAS file support to such signal.

Example: MGmt/ PStn/ DebugRecording/ ControlProtocol/ CONFIguration/ IPNetwOrking/ TPApP/ BSP/

```

PING SHow
/>ps
CAS/ PHysical/ PstnCOmmon/
/PStn>ph
IsdnGetDChannelStatus PstnQueryTrunkStatus PstnSendAlarm
PstnLoopCommands PstnGetPerformanceMonitoring
PstnStoPPerformanceMonitoring PstnStarTPerformanceMonitoring
/PStn/PHysical>pqts 1
TrunkId 1      LOS 0   LOF 0   RAI 1   AIS 0   RAI CRC 0
TrunkStatus 0   LoopBackStatus 0
TrunkIndexAlarmRedundancyDB 3
TrkMtc.Alarm                2
TrkMtc.AlarmBitMap          0x00000001
NoMultiframeAlignment 0
IsdnGetDChannelStatus PstnQueryTrunkStatus PstnSendAlarm
PstnLoopCommands PstnGetPerformanceMonitoring
PstnStoPPerformanceMonitoring PstnStarTPerformanceMonitoring
/PStn/PHysical>plc 22 1 10
Command sent to board. Use PstnQueryTrunkStatus to check the trunk
status.
IsdnGetDChannelStatus PstnQueryTrunkStatus PstnSendAlarm
PstnLoopCommands PstnGetPerformanceMonitoring
PstnStoPPerformanceMonitoring PstnStarTPerformanceMonitoring
/PStn/PHysical>psa 22 1
Command sent to board. Use PstnQueryTrunkStatus to check the trunk
status.
IsdnGetDChannelStatus PstnQueryTrunkStatus PstnSendAlarm
PstnLoopCommands PstnGetPerformanceMonitoring
PstnStoPPerformanceMonitoring PstnStarTPerformanceMonitoring
/PStn/PHysical>pstpm 1
Command sent to board. Use PstnGetPerformanceMonitoring to check
the trunk status.
IsdnGetDChannelStatus PstnQueryTrunkStatus PstnSendAlarm
PstnLoopCommands PstnGetPerformanceMonitoring
PstnStoPPerformanceMonitoring PstnStarTPerformanceMonitoring
/PStn/PHysical>psppm 1
Command sent to board
IsdnGetDChannelStatus PstnQueryTrunkStatus PstnSendAlarm
PstnLoopCommands PstnGetPerformanceMonitoring
PstnStoPPerformanceMonitoring PstnStarTPerformanceMonitoring
/PStn/PHysical>pgpm 0 0
TrunkId = 0
Interval = 0
AlarmIndicationSignal = 0
LossOfSignal = 0
LossOfFrame = 0
FramingErrorReceived = 0
RemoteAlarmReceived = 0
LostCRC4multiframeSync = 0
CRCErrorReceived = 0
EBitErrorDetected = 0
BitError = 0
LineCodeViolation = 0
ControlledSlip = 0
ErroredSeconds = 0
ControlledSlipSeconds = 0
SeverelyErroredFramingSeconds = 0
SeverelyErroredSeconds = 0
BurstyErroredSeconds = 0
UnAvailableSeconds = 0
PathCodingViolation = 0

```

```
LineErroredSeconds = 0
DegradedMinutes = 0
AssessedSeconds = 331
IsdnGetDChannelStatus PstnQueryTrunkStatus PstnSendAlarm
PstnLoopCommands PstnGetPerformanceMonitoring
PstnStoPPerformanceMonitoring PstnStarTPerformanceMonitoring
/PStn/PHysical> igdcs 0
TrunkId 0 DChannelStatus 0
IsdnGetDChannelStatus PstnQueryTrunkStatus PstnSendAlarm
PstnLoopCommands PstnGetPerformanceMonitoring
PstnStoPPerformanceMonitoring PstnStarTPerformanceMonitoring
/PStn/PHysical>..
CAS/ PHysical/ PstnCOmmon/
/PStn>pco
PstnQueryCallState PstnSetTraceLevel PstnRestartRequest
/PStn/PstnCOmmon>pstl 1 2 1
Command sent to board.
PstnQueryCallState PstnSetTraceLevel PstnRestartRequest
/PStn/PstnCOmmon>prr 1 2
Command sent to board.
PstnQueryCallState PstnSetTraceLevel PstnRestartRequest
/PStn/PstnCOmmon>..
CAS/ PHysical/ PstnCOmmon/
/PStn>cas
GenerateCasFlashHook CasBlockChannel
/PStn/CAS>cbc 1 2 1
Command sent to board.
GenerateCasFlashHook CasBlockChannel
/PStn/CAS>gcfh 1 0 2
```

3.4 Debug Recording (DR)

The debug recording (DR) tool can be used to capture media streams, networking and signaling traffic, and other internal blade information.

3.4.1 Collecting DR Messages

The client that is used to capture the DR packets is the open source Wireshark program (which can be downloaded from <http://www.wireshark.org>). An Audiocodes proprietary plugin (supplied in the software kit) must be placed in the 'plugin' folder of the installed Wireshark version (typically, C:\Program Files\WireShark\plugins\xxx\, where xxx is the installed 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.



Note: Wireshark plugins are not backward compatible. Loading incompatible plugins can crash the application.

3.4.2 Activating DR

Debug Recording activation is performed using the CLI interface under the DebugRecording directory. This section 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 'DR Command Reference' on page 38.

➤ **To activate the DR, take these 7 steps:**

1. Start a CLI management session (refer to 'Starting a CLI Management Session' on page 21).
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 and syslog), take these 4 steps:**

1. Setup the DR, as described at the beginning of this section.
2. At the prompt, type **ANCT ALL-WITH-PCM 1** Dynamic; 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 (i.e., not MediaPack gateways and Mediant 1000 analog modules) you can capture PSTN traces described in the procedure below:

➤ **To capture PSTN (SS7, CAS, ISDN) traces, take these 5 steps:**

1. Setup the DR, as described at the beginning of this section.
2. Set the *ini* file parameter TraceLevel to 1.
3. At the prompt, type **APST**<packet type -- ISDN, CAS, or SS7>.
4. At the prompt, type **START**.
5. Start Wireshark, and then filter according to the UDP port (default is 925) to where debug packets are sent.



Notes:

- PSTN and DSP recording can be performed simultaneously (applicable only to digital devices (i.e., not MediaPack gateways and Mediant 1000 analog modules)).
- All DR rules are deleted after the device is reset.

3.4.3 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 3-7: Client Setup Commands

Command	Parameters	Description
AddIpTarget	IPAddr [UDPPort]	Adds a Wireshark DR IP client to the list. UDPPort (optional): port on which to send the recorded packets (default is 925).
RemoveTarget	Index	Removes a DR client from the list. Index: index for the removed target (as displayed via ListTargets).

Command	Parameters	Description
ListTargets		Displays the client list.
SetDefaultTarget	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 3-8: Trace Rules

Command	Parameters	Description
AddIPTrafficTrace	TracePoint PDUType SourcePort DestPort [SourceIP] [DestIP] [DebugTarget]	Record IP traffic. <ul style="list-style-type: none"> Trace Point: Net2Host = Inbound non-media traffic. Host2Net = outbound non-media traffic. PDUType: UDP = UDP traffic. TCP = TCP traffic. ICMP = ICMP traffic. IPType = Any other IP type (as defined by http://www.iana.com). A = All traffic types. SourcePort: datagram's source port number (ALL for IP wildcard). DestPort: datagram's destination port number (ALL for IP wildcard). SourceIP (optional): datagram's source IP address (ALL for IP wildcard). DestIP (optional): datagram's source IP address (ALL for IP wildcard). DebugTarget (optional): debug target list index; if not specified, the default target is used.
AddIPControlTrace	TracePoint ControlType [DebugTarget]	Records an IP control. <ul style="list-style-type: none"> Trace Point: Net2Host = Inbound/Outbound non-media traffic. ControlType: SIP = SIP traffic. <p>DebugTarget (optional): debug target list index; if not specified, the default target is used.</p>

Command	Parameters	Description
AddPstnSignalingTrace	PacketType [DebugTarget]	<p>Records PSTN signaling.</p> <p>Packet Type:</p> <ul style="list-style-type: none"> CAS = CAS signaling. ISDN = ISDN signaling. SS7 = SS7 signaling. <p>DebugTarget (optional): debug target list index; if not specified, the default target is used.</p> <p>Notes:</p> <ul style="list-style-type: none"> Applicable only to digital devices (i.e., not MediaPack gateways and Mediant 1000 analog modules). To record PSTN signaling, 'PSTN Trace Level' (TraceLevel ini file) must be set to 1.
AddNextCallTrace	PacketType NumOfCalls [TraceType] [DebugTarget]	<p>Records the next media calls.</p> <ul style="list-style-type: none"> Packet Type: <ul style="list-style-type: none"> ALL = all media related (internal DSP packets, RTP, RTCP, T38, events, and syslog) of a certain call. ALL-WITH-PCM = all media-related and PCM traffic of a certain call (Note: applicable only to digital devices (i.e., not MediaPack gateways and Mediant 1000 analog modules)). NumOfCalls: amount of next media calls to record. (Note: Currently, only 1 call can be recorded.) <p>Trace Type (optional):</p> <ul style="list-style-type: none"> New (default) = the next new NumOfCalls calls to record. When these calls end, new calls are not recorded. Dynamic = the next new NumOfCalls calls to record. When these calls end, new calls are recorded until this trace is deleted. <p>DebugTarget (optional): debug target list index; if not specified, the default target is used.</p>
AddTrunkBchannelTrace	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"> Packet Type: <ul style="list-style-type: none"> ALL = all media related (internal DSP packets, RTP, RTCP, T38, events and Syslog) of a certain call. ALL-WITH-PCM = all media-related and PCM traffic of a certain call. Trunk: start of range trunk number for recording. (Note: Currently, only 1 channel can be recorded.) <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.</p> <p>DebugTarget (optional): debug target list index; if not specified, the default target is used.</p> <p>Note: Applicable only to digital devices (i.e., not MediaPack gateways and Mediant 1000 analog modules).</p>

Command	Parameters	Description
AddChannelIdTrace	PacketType Channel-Id [To Channel- Id][DebugTarget]	Records media calls according to CID. <ul style="list-style-type: none"> Packet Type: ALL = all media related (internal DSP packets, RTP, RTCP, T38, events and syslog) of a certain call. ALL-WITH-PCM = all media-related and PCM traffic of a certain call (Note: applicable only to digital devices (i.e., not MediaPack gateways and Mediant 1000 analog modules). Channel-Id: start of range channel ID number for recording. (Note: 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.
RemoveTraceRule	Index	Removes TraceRule from list. Index: rule index (as displayed via ListTraceRules). ALL for rule wildcard.
ListTraceRules	--	Displays added TraceRules.

Table 3-9: DR Activation

Command	Parameters	Description
STARTrecording	--	Enables recording.
STOPrecording	--	Disables recording.

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 and Maintenance (OAM).

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`
- `nlmLogTable` and `nlmLogVariableTable` in the standard `NOTIFICATION-LOG-MIB`

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

4.3 Topology MIB - Objects



Note: This section is applicable only to AudioCodes' 3000 Series and Mediant 1000 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: 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.

ifTable & ifXTable	Value
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, dot3StatsLateCollisions, dot3StatsExcessiveCollisions, dot3StatsInternalMacTransmitErrors and dot3StatsCarrierSenseErrors.
ifName	Ethernet (Gigabit for 3000 Series) port #1 or# 2
ifInMulticastPkts	As defined in IfMIB.
ifInBroadcastPkts	As defined in IfMIB.
ifOutMulticastPkts	As defined in IfMIB.
ifOutBroadcastPkts	As defined in IfMIB.
ifHCInOctets ifHCOctets	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.
ifHCInUcastPkts ifHCInMulticastPkts ifHCInBroadcastPkts ifHCOctetsUcastPkts ifHCOctetsMulticastPkts ifHCOctetsBroadcastPkts	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.
ifLinkUpDownTrapEnable	Refer to [RFC 2863]. Default is 'enabled'
ifHighSpeed	3000 Series: 1000 Mediant 1000: 10 or 100 according to acSysEthernetFirstPortSpeed
ifPromiscuousMode	Constant False. [R/O]
ifConnectorPresent	Constant True.

ifTable & ifXTable	Value
ifAlias	An 'alias' name for the interface as specified by a network manager (NVM)
ifCounterDiscontinuityTime	As defined in IfMIB.

Table 4-3: SONET /SDH Interfaces (3000 Series Only)

ifTable & ifXTable	Value
ifDescr	SONET/SDH interface. Module #n Port #n
ifType	sonet(39).
ifMtu	Constant zero.
ifSpeed	155520000
ifPhysAddress	The value of the Circuit Identifier. If no Circuit Identifier has been assigned this object should have an octet string with zero length.
ifAdminStatus	Read-only access -- Always UP.
ifOperStatus	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.
ifLastChange	The value of sysUpTime at the time the interface entered its current operational state.
ifName	SONET /SDH port no. n
ifLinkUpDownTrapEnable	Set to enabled(1)
ifHighSpeed	Speed of line in Megabits per second: 155
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 4-4: DS3 Interfaces (3000 Series Only)

ifTable & ifXTable	Value
ifDescr	DS3 interface, Module no.#d, Port no.#d
ifType	Ds3(30).
ifMtu	Constant zero.
ifSpeed	44736000
ifPhysAddress	The value of the Circuit Identifier. If no Circuit Identifier has been assigned this object should have an octet string with zero length.
ifAdminStatus	Read-only access -- Always UP.

ifTable & ifXTable	Value
ifOperStatus	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).
ifLastChange	The value of sysUpTime at the time the interface entered its current operational state.
ifName	DS3 port no. n
ifLinkUpDownTrapEnable	Set to enabled(1)
ifHighSpeed	Speed of line in Megabits per second: 45
ifConnectorPresent	Set to true(1)
ifCounterDiscontinuityTime	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 sub tree):

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

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

The log trap, acPerformanceMonitoringThresholdCrossing (non-alarm) is sent out 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 blade'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:** acPMAnalogConfigurationResetTotalCounters (Applicable only to Analog devices)
- **PM-Control:** acPMControlConfigurationResetTotalCounters
- **PM-Media:** acPMMediaConfigurationResetTotalCounters
- **PM-PSTN:** acPMPSTNConfigurationResetTotalCounters (Applicable only to Digital devices)
- **PM-System:** acPMSystemConfigurationResetTotalCounters

4.6 TrunkPack-VoP Series Supported MIBs

The device contains an embedded SNMP agent supporting the listed MIBs below.



Note: An HTML format description 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.



Note: For the Mediant 3000 / TP-6310 and Mediant 2000 / TP-1610 / TP-260: 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) OAM, (2) Media, (3) Control. (When there is only one interface then the only index is OAM (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`.
- **RTP MIB:** The RTP MIB is supported according to RFC 2959. It contains objects relevant to the RTP streams generated and terminated by the blade and to the RTCP information related to these streams.



Note: The inverse tables are not supported.

- **Notification Log MIB:** This standard MIB (RFC 3014 - `iso.org.dod.internet.mgmt.mib-2`) is supported as part of AudioCodes' implementation of Carrier Grade Alarms.
- **Alarm MIB:** This IETF MIB (RFC 3877) is 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:** This MIB (RFC 2273) allows for configuration of trap destinations and trusted managers.
- **SNMP MIB:** This MIB (RFC 3418) allows support of the `coldStart` and `authenticationFailure` traps.
- **SNMP Framework MIB:** (RFC 3411).
- **SNMP Usm MIB:** This MIB (RFC 3414) implements the user-based Security Model.
- **SNMP Vacm MIB:** This MIB (RFC 3415) implements the view-based Access Control Model.

- **SNMP Community MIB:** This MIB (RFC 3584) implements community string management.
- **ipForward MIB:** (RFC 2096) - fully supported
- **RTCP-XR:** This MIB (RFC) implements the following partial support:
 - The `rtcpXrCallQualityTable` is fully supported.
 - In the `rtcpXrHistoryTable`, support of the RCQ objects is provided only with no more than 3 intervals, 15 minutes long each.
 - Supports the `rtcpXrVoipThresholdViolation` trap.
- **ds1 MIB** supports for the following (Applicable only to Digital devices):
 - `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):
 - `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`

There are some proprietary MIB objects that are connected to the SONET/SDH configuration:
- **In the acSystem MIB** (Applicable only to the 3000 Series):
 - ◆ `acSysTransmissionType`:to set the transmission type to optical or DS3 (T3)
- **SONET MIB:** this 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
 - ◆ sonetSectionCurrentSEsSs
 - ◆ sonetSectionCurrentSEFSs
 - ◆ sonetSectionCurrentCVs
- In the SonetLineCurrentTable, the following objects are supported:
 - ◆ sonetLineCurrentStatus
 - ◆ sonetLineCurrentESs
 - ◆ sonetLineCurrentSEsSs
 - ◆ 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
- acSonetIlfHwFailureAlarm

(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:to set 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 blade and channels as well as to run-time information. Through this MIB, users can set up the blade configuration parameters, reset the blade, monitor the blade'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 has the following Groups:

- **boardConfiguration**
- **boardInformation**
- **channelConfiguration**
- **channelStatus**
- **reset**
- **acTrap**

Each AudioCodes proprietary MIBs contain 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**
- **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 blade when applied as a SIP device only. This MIB complements the other AudioCodes proprietary MIBs.

The acGateway MIB has 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 blades).

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 can be any value between 50 and 100 (default is 500) for Digital devices, and 10 and 100 (default is 100) for MediaPack devices.



Note: The following are special notes pertaining to MIBs:

- 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 blade functionality immediately or may require that the blade be soft reset for the change to take effect. This depends on the parameter type.



Note: The current (updated) blade configuration parameters are configured on the blade provided the user doesn't load an *ini* file to the blade 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 75.



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

The following proprietary traps are supported by the device:

Table 4-5: Default TCP/UDP Network Port Numbers

Trap	Description
acBoardFatalError	Sent whenever a fatal device error occurs.
acBoardConfigurationError	Sent when a device's settings are illegal. The trap contains a message stating/detailing/explaining the illegality of the setting.
acBoardTemperatureAlarm	Sent when a device exceeds its temperature limits.
acBoardEvResettingBoard	Sent after a device is reset.
acBoardEvBoardstarted	Sent after a device is successfully restored and initialized following reset.
acFeatureKeyError	Relays Feature Key errors etc.
acgwAdminStateChange	Sent when Graceful Shutdown commences and ends.
acBoardEthernetLinkAlarm	Ethernet link(s) is down.
acActiveAlarmTableOverflow	An active alarm could not be placed in the active alarm table because the table is full.
acAudioProvisioningAlarm	Raised if the device is unable to provision its audio.
acOperationalStateChange	Raised if the operational state of the node goes to disabled. Cleared when the operational state of the node goes to enabled.
acKeepAlive	Part of the NAT traversal mechanism. If the STUN application in the device detects a NAT then this trap is sent out on a regular time laps - 9/10 of the acSysSTUNBindingLifeTime object. The AdditionalInfo1 varbind has the MAC address of the device.
acBoardCallResourcesAlarm	Indicates that no free channels are available.
acBoardControllerFailureAlarm	The gatekeeper / Proxy is not found or registration failed. Internal routing table may be used for routing.
acBoardOverloadAlarm	Overload in one or some of the system's components.

Trap	Description
acNATTraversalAlarm	When the NAT is placed in front of a device, it is identified as a symmetric NAT - this alarm is raised. It is cleared when a non-symmetric NAT or no NAT replace the symmetric one.
acEnhancedBITStatus	This trap is used to 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.
acPerformanceMonitoringThresholdCrossing	This log trap is sent out for 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.
acHTTPDownloadResult	Log trap for the success or failures of the HTTP Download action.
acSS7LinkStateChangeAlarm	This alarm is raised if the operational state of the SS7 link becomes BUSY. The alarm is cleared when the operational state of the link becomes SERVICE or OFFLINE. Note: Applicable only to the 3000 Series and the 2000 Series devices.
acSS7LinkInhibitStateChangeAlarm	This alarm is raised if 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. Note: Applicable only to the 3000 Series and the 2000 Series devices.
acSS7LinkBlockStateChangeAlarm	This alarm is raised if 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. Note: Applicable only to the 3000 Series and the 2000 Series devices.
acSS7LinkCongestionStateChangeAlarm	This alarm is raised 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. Note: Applicable only to the 3000 Series and the 2000 Series devices.
acSS7LinkSetStateChangeAlarm	This alarm is raised if the operational state of the SS7 linkset becomes BUSY. The alarm is cleared when the operational state of the linkset becomes SERVICE or OFFLINE. Note: Applicable only to the 3000 Series and the 2000 Series devices.
acSS7RouteSetStateChangeAlarm	This alarm is raised if the operational state of the SS7 routeset becomes BUSY. The alarm is cleared when the operational state of the routeset becomes SERVICE or OFFLINE. Note: Applicable only to the 3000 Series and the 2000 Series devices.
acSS7SNSetStateChangeAlarm	This alarm is raised if 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. Note: Applicable only to the 3000 Series and the 2000 Series devices.
acFanTrayAlarm	Fault in the fan tray or fan tray missing. Note: Applicable only to the 3000 Series devices.
acPowerSupplyAlarm	Fault in one of the power supply modules or a PS module is missing. Note: Applicable only to the 3000 Series devices.

Trap	Description
acPEMAlarm	Fault in one of the PEM modules or a PEM module is missing. Note: Applicable only to the 3000 Series devices.
acSAMissingAlarm	SA module is missing or non operational. Note: Applicable only to the 3000 Series devices.
acUserInputAlarm	Alarm is raised when the input dry contact is short circuited, and cleared when the circuit is reopened. Note: Applicable only to the 3000 Series devices.
acHASystemFaultAlarm	The High Availability (HA) system is faulty and therefore there is no HA. Note: Applicable only to the HA Mediant 3000 and IPmedia 3000 devices.
acHASystemConfigMismatchAlarm	Configuration of the modules in the HA system is not identical, causing instability. Note: Applicable only to the HA Mediant 3000 and IPmedia 3000 devices.
acHASystemSwitchOverAlarm	A switchover from the active to the redundant module has occurred. Note: Applicable only to the High Availability Mediant 3000 and IPmedia 3000 devices.
acSonetSectionLOFAlarm	SONET section Loss of Frame alarm. Note: Applicable only to the 3000 Series devices.
acSonetSectionLOSAlarm	SONET section Loss of Signal alarm. Note: Applicable only to the 3000 Series devices.
acSonetLineAISAlarm	SONET Line AIS alarm. Note: Applicable only to the 3000 Series devices.
acSonetLineRDIAAlarm	SONET Line RDI alarm. Note: Applicable only to the 3000 Series devices.
acDChannelStatus	Non alarm trap sent at the establishment, re-establishment or release of LAPD link with its peer connection occurs. The trap is sent out with one of the following textual descriptions: <ul style="list-style-type: none"> ■ D-channel synchronized ■ D-channel not-synchronized Note: Applicable only to the Digital devices

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 section 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 119)
- Various combinations of the above

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

4.8.1 SNMP Community Names

By default, the blade uses a single, read-only community string of "public" and a single read-write community string of "private". Up to 5 read-only community strings and up to 5 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 4-6: 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 Embedded Web Server, refer to 'Configuring the SNMP Community Strings' in 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 EM must use the standard `snmpCommunityMIB`. To configure the trap community string, the EM 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 4-7: 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 4-8: 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 table to add, modify, and delete SNMPv3 users. The SNMPUsers INI table is a hidden parameter. Therefore, when you perform a “Get ini File” operation in the Embedded Web Server, the table is not included in the generated file.

The table columns are described below.

Table 4-9: SNMPv3 Table Columns Description

Parameter	Description	Default
Row number	This is the 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. Here 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 will be 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 will be 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 will be associated with ReadWriteGroup3.

4.8.2.2 Configuring SNMPv3 Users via SNMP

To configure SNMPv3 users, the EM 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 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 via the use of Trusted Managers. A Trusted Manager is an IP address from which the SNMP agent accepts and process get and set requests. An EM can be used to configure up to 5 Trusted Managers.



Note: If Trusted Managers are defined, then all community strings works from all Trusted Managers. That is, there is no way to associate a community string with particular trusted managers.

The concept of trusted managers is considered to be a weak form of security and is therefore, not a required part of SNMPv3 security, which uses authentication and privacy. However, the blade's SNMP agent applies the trusted manager concept as follows:

- There is no way to configure trusted managers for only a SNMPv3 user. An SNMPv2c community string must be defined.
- If specific IPs are configured as trusted managers (via the community table), then only SNMPv3 users on those trusted managers are given access to the agent's MIB objects.

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 EM 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 blade. 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 blade. 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 ports can be changed by setting parameters in the *ini* file. The parameter name is:

SNMPPort = <port_number>

Valid UDP port number; default = 161.

This parameter specifies the port number for SNMP requests and responses. Usually it should not be specified. Use the default.

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:

- Embedded Web server (refer to 'Configuring the Management Settings' in the device's *User's Manual*)
- *ini* file (refer to 'Configuring Trap Managers via the ini File' on page 67)
- SNMP (refer to 'Configuring Trap Managers via SNMP' on page 69)

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 blade 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 device *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 device ini file regarding SNMP. The device can be configured to send to multiple trap destinations. The lines in the file below are commented out with the “;” at the beginning of the line. All of the lines below are commented out since the first line character is a semi-colon.

```

; SNMP trap destinations
; The blade 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".

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 for now 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 section 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 Embedded Web Server interface.

A default *ini* (or initialization) template has been defined, which sets the configuration parameters to settings that users normally would not need to modify.

Configuration parameters in the `acBoardMIB` specific to services on the device are:

- **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 section 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 board 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 75):

- **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 section is applicable only to AudioCodes' 3000 Series devices.

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 us uneven causing instability.
- **acHASystemSwitchOverAlarm:** a switch over from the active to the redundant module has occurred.

4.14 SNMP Administrative State Control

4.14.1 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 73). These parameters are in the acBoardMIB as the following:

- **acgwAdminState:** used either to request (set) a shutdown (0), undo shutdown (2), or to view (get) the device condition (0 = locked, 1 = shutting down, 2 = unlocked).
- **acgwAdminStateLockControl:** used to set 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 first followed by the acgwAdminState.

4.14.2 Graceful Shutdown

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

The possible values received on a get request are:

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

On a set request, the manager supplies the desired administrative state: either locked(0) or unlocked(2).

When the blade changes to either shuttingDown or locked state, an adminStateChange alarm is raised. When the blade 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 are:

- 1 = Perform a graceful lock. Calls are allowed to complete. No new calls are allowed to be originated on this 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.

4.15 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.

4.16 SNMP Traps

This subsection provides information on proprietary SNMP traps currently supported by the media server. There is a separation between traps that are alarms and traps that are not (logs). Currently all have the same structure made up of the same 11 varbinds (Variable Binding) (1.3.6.1.4.1.5003.9.10.1.21.1).

The source varbind is composed of a string that details the 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.

4.16.1 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'.

4.16.1.1 Component: Chassis#0



Note: This section is only applicable to AudioCodes' 3000 Series and Mediant 1000 devices.

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

Table 4-10: acFanTrayAlarm Alarm Trap

Alarm:	acFanTrayAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.29
Default Severity:	Critical
Event Type:	equipmentAlarm
Probable Cause:	heatingVentCoolingSystemProblem
Alarm Text:	Fan-Tray Alarm
Status Changes:	
Condition:	Fan-Tray is missing
Alarm status:	Critical
<text> value:	Fan-Tray Alarm. Fan-Tray is missing
Condition:	One or more fans in the Fan-Tray are faulty.
Alarm status:	Major
Corrective Action:	Fan is faulty
Condition:	Fan tray is in place and fans are working.
Alarm status:	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 4-11: acPowerSupplyAlarm Alarm Trap

Alarm:	acPowerSupplyAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.30
Default Severity:	Critical
Event Type:	equipmentAlarm
Probable Cause:	powerProblem
Alarm Text:	Power-Supply Alarm. Power-Supply is missing.
Status Changes:	
Condition:	The HA (High Availability) feature is active and one of the power supply units is faulty or missing.
Alarm status:	Major
Condition:	PS unit is placed and working.
Alarm status:	Cleared

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

Table 4-12: acUserInputAlarm Alarm Trap

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

The following trap is only applicable 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 slot number.

Table 4-13: acPEMAlarm Alarm Trap (Applicable only to 3000 Series)

Alarm:	acPEMAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.31
Default Severity:	Critical
Event Type:	equipmentAlarm
Probable Cause:	underlyingResourceUnavailable
Alarm Text:	PEM Module Alarm.
Status Changes:	
Condition:	The HA (High Availability) feature is active and one of the PEM units is missing (PEM – Power Entry Module)
Alarm status:	Critical
<text> value:	PEM card is missing.
Condition:	PEM card is placed and both DC wires are in.
Alarm status:	Cleared

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



Note: This section is only applicable to AudioCodes' 3000 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 4-14: AcSonetSectionLOFAlarm Alarm Trap

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

Table 4-15: AcSonetSectionLOSAAlarm Alarm Trap

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

Table 4-16: AcSonetLineAISAlarm Alarm Trap

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

Table 4-17: AcSonetLineRDIAAlarm Alarm Trap

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

4.16.1.3 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 4-18: acBoardFatalError Alarm Trap

Alarm:	acBoardFatalError
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.1
Default Severity:	Critical
Event Type:	equipmentAlarm
Probable Cause:	underlyingResourceUnavailable (56)
Alarm Text:	Board Fatal Error: <text>
Status Changes:	
Condition:	Any fatal error
Alarm status:	Critical
<text> value:	A run-time specific string describing the fatal error
Condition:	After fatal error
Alarm status:	Status stays critical until reboot. A clear trap is not sent.
Corrective Action:	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 4-19: acBoardConfigurationError Alarm Trap

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



Note: The acBoardTemperatureAlarm alarm trap below does not apply to the MediaPack Series nor the 3000 Series High Availability Mode.

Table 4-20: acBoardTemperatureAlarm Alarm Trap

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

Table 4-21: acBoardEvResettingBoard Alarm Trap

Alarm:	acBoardEvResettingBoard
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.5
Default Severity:	Critical
Event Type:	equipmentAlarm
Probable Cause:	outOfService (71)
Alarm Text:	User resetting board
Status Changes:	
Condition:	When a soft reset is triggered via the Embedded Web Server or SNMP.
Alarm status:	Critical
Condition:	After raise
Alarm status:	Status stays critical until reboot. A clear trap is not sent.
Corrective Action:	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 4-22: acBoardEthernetLinkAlarm Alarm Trap (Applicable only to 2000 Series, Mediant 1000, and MediaPack)

Alarm:	acBoardEthernetLinkAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.10
Default Severity:	Critical
Event Type:	equipmentAlarm
Probable Cause:	underlyingResourceUnavailable (56)
Alarm Text:	Ethernet link alarm: <text>
Status Changes:	
Condition:	Fault on single interface
Alarm status:	Major
<text> value:	Redundant link is down
Condition:	Fault on both interfaces
Alarm status:	critical
<text> value:	No Ethernet link
Condition:	Both interfaces are operational
Alarm status:	cleared
Corrective Action:	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 4-23: acBoardCallResourcesAlarm Alarm Trap (Applicable only to Mediant 1000)

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

Table 4-24: acBoardControllerFailureAlarm Alarm Trap (Applicable only to Mediant 1000)

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

Table 4-25: acBoardOverloadAlarm Alarm Trap (Applicable only to Mediant 1000)

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

Table 4-26: acFeatureKeyError Alarm Trap (Applicable only to Digital devices)

Alarm:	acFeatureKeyError
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.6
Default Severity:	Critical
Event Type:	processingErrorAlarm
Probable Cause:	configurationOrCustomizationError (7)
Alarm Text:	Feature key error
Status Changes:	
Note:	Support of this alarm is pending
Corrective Action:	

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 4-27: acSAMissingAlarm Alarm Trap (Applicable only to the 3000 Series devices)

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

4.16.1.4 Component: AlarmManager#0

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

Table 4-28: acActiveAlarmTableOverflow Alarm Trap

Alarm:	acActiveAlarmTableOverflow
OID:	1.3.6.1.4.15003.9.10.1.21.2.0.12
Default Severity:	Major
Event Type:	processingErrorAlarm
Probable Cause:	resourceAtOrNearingCapacity (43)
Alarm Text:	Active alarm table overflow
Status Changes:	
Condition:	Too many alarms to fit in the active alarm table
Alarm status:	Major
Condition:	After raise
Alarm status:	Status stays major until reboot. A clear trap is not sent.
Note:	The status stays 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.
Corrective Action:	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.

4.16.1.5 Component: AudioStaging#0



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

Table 4-29: acAudioProvisioningAlarm Alarm Trap

Alarm:	acAudioProvisioningAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.14
Default Severity:	Critical
Event Type:	processingErrorAlarm
Probable Cause:	configurationOrCustomizationError (7)
Alarm Text:	Unable to provision audio
Status Changes:	
Condition:	Media server times out waiting for a successful audio distribution from the APS (Audio Provisioning Server)
Alarm status:	critical
Condition:	After raise, media server is successfully provisioned with audio from the APS
Alarm status:	critical
	cleared
Corrective Action:	<p>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.</p> <p>For more information regarding the problem, view the Syslogs from the device as well as the APS manager logs.</p>

4.16.1.6 Component: SS7#0



Note: This section is only applicable to AudioCodes' 2000 Series and 3000 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 4-30: acSS7LinkStateChangeAlarm Trap

Alarm:	acSS7LinkStateChangeAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.19
Default Severity:	Major
Event Type:	communicationsAlarm
Probable Cause:	Other
Alarm Text:	*** SS7 *** Link %i is %s \$s
Status Changes:	
Condition:	Operational state of the SS7 link becomes 'BUSY'.
Alarm status:	Major
<text> value:	%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.
Additional Info1 varbind	BUSY
Condition:	Operational state of the link becomes 'IN-SERVICE' or 'OFFLINE'.
Alarm status:	Cleared
Corrective Action:	For full details refer to the SS7 MTP2 and MTP3 relevant standards.

Table 4-31: acSS7LinkCongestionStateChangeAlarm Trap

Alarm:	acSS7LinkCongestionStateChangeAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.22
Default Severity:	Major
Event Type:	communicationsAlarm
Probable Cause:	Other
Alarm Text:	*** SS7 *** Link %i is %s %s
Status Changes:	
Condition:	SS7 link becomes congested (local or remote).
Alarm status:	Major
<text> value:	%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" }
Additional Info1 varbind	CONGESTED
Condition:	Link becomes un-congested (local AND remote).
Alarm status:	Cleared
Corrective Action:	Reduce SS7 traffic on that link.
Note:	This alarm is raised for any change in the remote or local congestion status.

Table 4-32: acSS7LinkInhibitStateChangeAlarm Trap

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

Table 4-33: acSS7LinkBlockStateChangeAlarm Trap

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

Table 4-34: acSS7LinkSetStateChangeAlarm Trap

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

Table 4-35: acSS7RouteSetStateChangeAlarm Trap

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

Table 4-36: acSS7SNSetStateChangeAlarm Trap

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

Table 4-37: acSS7RedundancyAlarm Trap

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

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



Note: The alarm traps discussed in this section applies only to the Mediant 3000 and IPmedia 3000 devices in High Availability mode.

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

Table 4-38: acHASystemFaultAlarm Trap

Trap:	acHASystemFaultAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.33
Default Severity:	critical
Event Type:	qualityOfServiceAlarm
Probable Cause:	outOfService
Trap Text:	No HA! <text>
Status Changes:	
Condition:	HA feature is active but the system is NOT working in HA mode.
Trap status:	Critical
<text> value:	<p>There are many possible values for the text:</p> <ul style="list-style-type: none"> Fatal exception error TCPIP exception error Network processor exception error SW WD exception error HW WD exception error SAT device is missing SAT device error DSP error BIT tests error PSTN stack error Keep Alive error Software upgrade Manual switch over Manual reset Board removal Can't read slot number TER misplaced HW fault. TER in slot 2 or 3 is missing HW fault. TER has old version or is not functional HW fault. invalid TER Type HW fault. invalid TER active/redundant state HW fault. Error reading GbE state Redundant module is missing Unable to sync SW versions Redundant is not connecting Redundant is not reconnecting after deliberate restart No Ethernet Link in redundant module SA module faulty or missing
Condition:	HA feature is active and the redundant module is in start up mode and hasn't connected yet.
Trap status:	Minor
<text> value:	Waiting for redundant to connect
Condition:	HA system is active.
Trap status:	Cleared

Table 4-39: acHASystemConfigMismatchAlarm Trap

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

Table 4-40: acHASystemSwitchOverAlarm Trap

Trap:	acHASystemSwitchOverAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.35
Default Severity:	Critical
Event Type:	qualityOfServiceAlarm
Probable Cause:	outOfService
Trap Text:	Switch-over:
Status Changes:	
Condition:	Switch over has taken place.
Trap status:	Critical
<text> value:	See the acHASystemFaultAlarm table above.
Condition:	10 seconds have passed since the switch over.
Trap status:	cleared

Table 4-41: acBoardTemperatureAlarm Trap

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

The source varbind text for the alarm under this component is:

- If the lost link is from the Active module: Chassis#0/Module#m/EthernetLink#0, where m is the blade's slot number.
- If the lost link is from the Redundant module: Chassis#0/Module#m, where m is the blade's slot number.

Table 4-42: acBoardEthernetLinkAlarm Trap

Trap:	acBoardEthernetLinkAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.10
Default Severity:	Critical
Event Type:	equipmentAlarm
Probable Cause:	underlyingResourceUnavailable (56)
Trap Text:	Ethernet link alarm: <text>
Status Changes:	
Condition:	Fault on single interface of the Active module.
Trap status:	Major
<text> value:	Redundant link (physical link n) is down
Condition:	Fault on both interfaces
Trap status:	critical
<text> value:	No Ethernet link
Condition:	Fault on single interface of the Redundant module.
Trap status:	Major
<text> value:	Redundant link in the redundant module (physical link n) is down
Condition:	Both interfaces are operational
Trap status:	Cleared
Corrective Action:	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
Note:	The alarm behaves differently when coming from the redundant or the active modules of an HA system. The alarm from the redundant will be 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 noHA alarm that follows such a case.

4.16.2 Log Traps (Notifications)

This section details traps that are not alarms. These traps are sent with the severity varbind value of 'indeterminate'. These traps don't 'clear', they don't appear in the alarm history or active tables. One log trap that does send clear is acPerformanceMonitoringThresholdCrossing.

Table 4-43: acKeepAlive Log Trap

Trap:	acKeepAlive
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.16
Default Severity:	Indeterminate
Event Type:	other (0)
Probable Cause:	other (0)
Trap Text:	Keep alive trap
Status Changes:	
Condition:	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'
Trap status:	Trap is sent
Note:	Keep-alive is sent every 9/10 of the time defined in the parameter NatBindingDefaultTimeout.

Table 4-44: acPerformanceMonitoringThresholdCrossing Log Trap

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

Table 4-45: acHTTPDownloadResult Log Trap

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

4.16.3 Other Traps

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

Table 4-46: coldStart Trap

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

Table 4-47: authenticationFailure Trap

Trap Name:	authenticationFailure
OID:	1.3.6.1.6.3.1.1.5.5
MIB:	SNMPv2-MIB

Table 4-48: acBoardEvBoardStarted Trap

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

Table 4-49: AcDChannelStatus Trap (Applicable only to 3000 Series and 2000 Series devices)

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

4.16.4 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.

4.16.5 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.

Reader's Notes

5 Configuration Files

This section describes the configuration *dat* files that are loaded (in addition to the *ini* file) to the device. The configuration files include the following:

You can load the configuration files to the device using one of the following methods:

- Embedded Web Server (refer to 'Auxiliary Files' in the devices *User's Manual*)
- The *ini* file: specify the name of the relevant configuration file in the device's *ini* file and then load the *ini* file to the device (refer to 'Loading the cmp File, Booting the Device' on page 191).

5.1 Configuring the Call Progress Tones File

The Call Progress Tones (and Distinctive Ringing for Analog devices) configuration file used by the device is a binary file (with file extension *dat*). This file contains the following, depending on device:

- **3000 Series and 2000 Series:** contains the definitions of the Call Progress Tones (levels and frequencies) that are detected / generated by the device
- **Analog:** comprised of two sections. The first section contains the definitions of the Call Progress Tones (levels and frequencies) that are detected / generated by the device. The second section contains the characteristics of the distinctive ringing signals that are generated by the device (refer to 'Configuring the Distinctive Ringing Section of the ini File' on page 104).

You can either use one of the supplied device configuration (*dat*) files or create your own file. To create your own configuration file, it's recommended to modify the supplied *usa_tone.ini* file (in any standard text editor) to suit your specific requirements, and to convert the modified *ini* file into binary format using the TrunkPack Downloadable Conversion Utility. For the description of the procedure on how to convert CPT *ini* file into a binary *dat* file, refer to 'Converting a CPT ini File to a Binary dat File' on page 205.

To load the Call Progress Tones (*dat*) file to the device, use the Embedded Web Server (refer to 'Auxiliary Files' in the devices *User's Manual*) or the *ini* file (refer to 'Configuration Files Parameters' in the device's *User's Manual*).



Note: Only the *dat* file can be loaded to the device.

You can create up to 32 different Call Progress Tones, each with frequency and format attributes. The frequency attribute can be single or dual-frequency (in the range of 300 to 1980 Hz), or an Amplitude Modulated (AM). In total, up to 64 different frequencies are supported. Only eight AM tones, in the range of 1 to 128 kHz, can be configured (the detection range is limited to 1 to 50 kHz). Note that when a tone is composed of a single frequency, the second frequency field must be set to zero.

The format attribute can be one of the following:

- **Continuous:** (e.g., dial tone) a steady non-interrupted sound. Only the 'First Signal On time' should be specified. All other on and off periods must be set to zero. In this case, the parameter specifies the detection period. For example, if it equals 300, the tone is detected after 3 seconds (300 x 10 msec). The minimum detection time is 100 msec.
- **Cadence:** A repeating sequence of on and off sounds. Up to four different sets of on / off periods can be specified.
- **Burst:** A single sound followed by silence. Only the 'First Signal On time' and 'First Signal Off time' should be specified. All other on and off periods must be set to zero. The burst tone is detected after the off time is completed.

You can specify several tones of the same type. These additional tones are used only for tone detection. Generation of a specific tone conforms to the first definition of the specific tone. For example, you can define an additional dial tone by appending the second dial tone's definition lines to the first tone definition in the *ini* file. The device reports dial tone detection if either of the two tones is detected.

The Call Progress Tones section of the *ini* file comprises the following segments:

- **[NUMBER OF CALL PROGRESS TONES]:** Contains the following key: 'Number of Call Progress Tones' defining the number of Call Progress Tones that are defined in the file.
- **[CALL PROGRESS TONE #X]:** containing the Xth tone definition (starting from 1 and not exceeding the number of Call Progress Tones defined in the first section) using the following keys:
 - **Tone Type:** Call Progress Tone types:
 - ◆ [1] Dial Tone
 - ◆ [2] Ringback Tone
 - ◆ [3] Busy Tone
 - ◆ [7] Reorder Tone
 - ◆ [8] Confirmation Tone (Applicable only to Analog devices)
 - ◆ [9] Call Waiting Tone (Applicable only to Analog devices)
 - ◆ [15] Stutter Dial Tone (Applicable only to Analog devices)
 - ◆ [16] Off Hook Warning Tone (Applicable only to Analog devices)
 - ◆ [17] Call Waiting Ringback Tone
 - ◆ [23] Hold Tone
 - **Tone Modulation Type:** Either Amplitude Modulated (1) or regular (0).
 - **Tone Form:** The tone's format can be one of the following:
 - ◆ Continuous (1)
 - ◆ Cadence (2)
 - ◆ Burst (3)
 - **Low Freq [Hz]:** frequency (in Hz) of the lower tone component in case of dual frequency tone, or the frequency of the tone in case of single tone. This is not relevant to Amplitude Modulated (AM) tones.
 - **High Freq [Hz]:** frequency (in Hz) of the higher tone component in case of dual frequency tone, or zero (0) in case of single tone (not relevant to AM tones).

- **Low Freq Level [-dBm]:** generation level 0 dBm to -31 dBm in dBm (not relevant to AM tones).
- **High Freq Level:** generation level. 0 to -31 dBm. The value should be set to 32 in the case of a single tone (not relevant to AM tones).
- **First Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the first cadence on-off cycle. For be continuous tones, this parameter defines the detection period. For burst tones, it defines the tone's duration.
- **First Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the first cadence on-off cycle (for cadence tones). For burst tones, this parameter defines the off time required after the burst tone ends and the tone detection is reported. For continuous tones, this parameter is ignored.
- **Second Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.
- **Second Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.
- **Third Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the third cadence ON-OFF cycle. Can be omitted if there isn't a third cadence.
- **Third Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the third cadence ON-OFF cycle. Can be omitted if there isn't a third cadence.
- **Fourth Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the fourth cadence ON-OFF cycle. Can be omitted if there isn't a fourth cadence.
- **Fourth Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the fourth cadence ON-OFF cycle. Can be omitted if there isn't a fourth cadence.
- **Carrier Freq [Hz]:** frequency of the carrier signal for AM tones.
- **Modulation Freq [Hz]:** frequency of the modulated signal for AM tones (valid range from 1 to 128 Hz).
- **Signal Level [-dBm]:** level of the tone for AM tones.
- **AM Factor [steps of 0.02]:** amplitude modulation factor (valid range from 1 to 50. Recommended values from 10 to 25).

**Notes:**

- When the same frequency is used for a continuous tone and a cadence tone, the 'Signal On Time' parameter of the continuous tone must have a value that is greater than the 'Signal On Time' parameter of the cadence tone. Otherwise the continuous tone is detected instead of the cadence tone.
- The tones frequency should differ by at least 40 Hz from one tone to other defined tones.

For example, to configure the dial tone to 440 Hz only, enter the following text:

```
#Dial tone
[CALL PROGRESS TONE #1]
Tone Type=1
Tone Form =1 (continuous)
Low Freq [Hz]=440
High Freq [Hz]=0
Low Freq Level [-dBm]=10 (-10 dBm)
High Freq Level [-dBm]=32 (use 32 only if a single tone is
required)
First Signal On Time [10msec]=300; the dial tone is detected after
3 sec
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=0
Second Signal Off Time [10msec]=0
```

5.2 Configuring the Distinctive Ringing Section of the ini File



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

Distinctive Ringing is only applicable to FXS interface. Using the distinctive ringing section of this configuration file, you can create up to 16 distinctive ringing patterns. Each ringing pattern configures the ringing tone frequency and up to four ringing cadences. The same ringing frequency is used for all the ringing pattern cadences. The ringing frequency can be configured in the range of 10 to 200 Hz with a 5 Hz resolution. Each of the ringing pattern cadences is specified by the following parameters:

- **Burst Ring On Time:** Configures the cadence to be a burst cadence in the entire ringing pattern. The burst relates to On time and the Off time of the same cadence. It must appear between 'First/Second/Third/Fourth' string and the 'Ring On/Off Time' This cadence rings once during the ringing pattern. Otherwise, the cadence is interpreted as cyclic: it repeats for every ringing cycle.
- **Ring On Time:** specifies the duration of the ringing signal.
- **Ring Off Time:** specifies the silence period of the cadence.



Note: In SIP, the distinctive ringing pattern is selected according to Alert-Info header that is included in INVITE message. For example, Alert-Info <Bellcore-dr2>, or Alert-Info<http://.../Bellcore-dr2>. 'dr2' defines ringing pattern # 2. If the Alert-Info header is missing, the default ringing tone (0) is played.

The distinctive ringing section of the *ini* file format contains the following strings:

- **[NUMBER OF DISTINCTIVE RINGING PATTERNS]:** Contains the following key:
 - 'Number of Distinctive Ringing Patterns' defining the number of Distinctive Ringing signals that are defined in the file.
- **[Ringing Pattern #X]:** Contains the Xth ringing pattern definition (starting from 0 and not exceeding the number of Distinctive Ringing patterns defined in the first section minus 1) using the following keys:
 - **Ring Type:** Must be equal to the Ringing Pattern number.
 - **Freq [Hz]:** Frequency in hertz of the ringing tone.
 - **First (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the first cadence on-off cycle.
 - **First (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the first cadence on-off cycle.
 - **Second (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the second cadence on-off cycle.
 - **Second (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the second cadence on-off cycle.
 - **Third (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the third cadence on-off cycle.
 - **Third (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the third cadence on-off cycle.
 - **Fourth (Burst) Ring On Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the fourth cadence on-off cycle.
 - **Fourth (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the fourth cadence on-off cycle.

5.2.1 Examples of Ringing Signals

- Below is an example of a **ringing burst**:

```
#Three ringing bursts followed by repeated ringing of 1 sec on and
3 sec off.
[NUMBER OF DISTINCTIVE RINGING PATTERNS]
Number of Ringing Patterns=1
[Ringing Pattern #0]
Ring Type=0
Freq [Hz]=25
First Burst Ring On Time [10msec]=30
First Burst Ring Off Time [10msec]=30
Second Burst Ring On Time [10msec]=30
Second Burst Ring Off Time [10msec]=30
Third Burst Ring On Time [10msec]=30
Third Burst Ring Off Time [10msec]=30
Fourth Ring On Time [10msec]=100
Fourth Ring Off Time [10msec]=300
```

- Below is an example of **various ringing signals**:

```
[NUMBER OF DISTINCTIVE RINGING PATTERNS]
Number of Ringing Patterns=3

#Regular North American Ringing Pattern
[Ringing Pattern #0]
Ring Type=0
Freq [Hz]=20
First Ring On Time [10msec]=200
First Ring Off Time [10msec]=400

#GR-506-CORE Ringing Pattern 1
[Ringing Pattern #1]
Ring Type=1
Freq [Hz]=20
First Ring On Time [10msec]=200
First Ring Off Time [10msec]=400

#GR-506-CORE Ringing Pattern 2
[Ringing Pattern #2]
Ring Type=2
Freq [Hz]=20
First Ring On Time [10msec]=80
First Ring Off Time [10msec]=40
Second Ring On Time [10msec]=80
Second Ring Off Time [10msec]=400
```

5.3 Prerecorded Tones (PRT) File

The Call Progress Tones mechanism has several limitations such as a limited number of predefined tones and a limited number of frequency integrations in one tone. To work around these limitations and provide tone generation capability that is more flexible, the PRT file can be used. If a specific prerecorded tone exists in the PRT file, it takes precedence over the same tone that exists in the CPT file and is played instead of it.



Note: The Prerecorded tones are used only for generation of tones. Detection of tones is performed according to the CPT file.

The PRT *dat* file contains a set of prerecorded tones to be played by the device during operation. Up to 40 tones (totaling approximately 10 minutes) can be stored in a single file in flash memory. The prerecorded tones (raw data PCM or L8 files) are prepared offline using standard recording utilities (such as CoolEdit™) and combined into a single file using the TrunkPack Downloadable Conversion utility (refer to 'Creating a Loadable Prerecorded Tones File' in the device's *User's Manual*).

The raw data files must be recorded with the following characteristics:

- Coders:** G.711 A-law, G.711 μ -law or Linear PCM
- Rate:** 8 kHz
- Resolution:** 8-bit
- Channels:** mono

The generated PRT file can then be loaded to the device using the BootP/TFTP utility (refer to 'Loading the Auxiliary Files via the ini File' in the device's *User's Manual*) or via the Embedded Web Server (refer to 'Auxiliary Files' in the device's *User's Manual*).

The Pre-recorded tones are played repeatedly. This enables you to record only part of the tone and play it for the full duration. For example, if a tone has a cadence of 2 seconds on and 4 seconds off, the recorded file should contain only these 6 seconds. The PRT module repeatedly plays this cadence for the configured duration. Similarly, a continuous tone can be played by repeating only part of it.

5.4 Voice Prompts File



Note: This section is applicable only to AudioCodes' 3000 Series and 2000 Series devices.

The voice announcement file contains a set of Voice Prompts that can be played by the device during operation. The voice announcements are prepared offline using standard recording utilities and combined into a single file using the TrunkPack Downloadable Conversion Utility.

The generated announcement file can then be loaded to the device using the BootP/TFTP utility (refer to 'Loading the Auxiliary Files via the ini File' in the device's *User's Manual*) or via the Embedded Web Server (refer to 'Auxiliary Files' in the device's *User's Manual*).

If the size of the combined Voice Prompts file is less than 1 MB, it can permanently be stored in flash memory. Larger files, up to 10 MB, are stored in RAM, and should be loaded again (using BootP/TFTP utility) after the device is reset.

The Voice Prompts integrated file is a collection of raw voice recordings and / or *wav* files. These recordings can be prepared using standard utilities such as CoolEdit, Goldwave™ and others. The raw voice recordings must be sampled at 8000 kHz / mono / 8 bit. The *wav* files must be recorded with G.711 μ -Law/A-Law/Linear.

When the list of recorded files is converted to a single *voiceprompts.dat* file, every Voice Prompt is tagged with an ID number, starting with '1'. This ID is used later by the device to start playing the correct announcement. Up to 1,000 Voice Prompts can be used.

AudioCodes provides a professionally recorded English (U.S.) Voice Prompts file.

➤ To generate and load the Voice Prompts file, take these 3 steps:

1. Prepare one or more voice files using standard utilities.
2. Use the TrunkPack Downloadable Conversion Utility to generate the *voiceprompts.dat* file from the pre-recorded voice messages (refer to 'Creating a Loadable Voice Prompts File' in the device's *User's Manual*).
3. Load the *voiceprompts.dat* file to the device either by using a TFTP procedure (refer to 'Loading the Auxiliary Files via the ini File' in the device's *User's Manual*) or via the Embedded Web Server (refer to 'Auxiliary Files' in the device's *User's Manual*).

5.5 CAS Protocol Configuration Files



Note: This section is applicable only to AudioCodes 3000 Series, 2000 Series, and Mediant 1000 devices.

The CAS Protocol Configuration files contain the CAS Protocol definitions that are used for CAS-terminated trunks. You can either use the supplied files or construct your own files. Up to eight files can be loaded and different files can be assigned to different trunks. The CAS files can be loaded to the device using the Embedded Web Server (refer to 'Auxiliary Files' in the device's *User's Manual*) or alternatively, using the device's *ini* file (refer to 'Loading the Auxiliary Files via the ini File' in the device's *User's Manual*).



Note: All CAS files loaded together must belong to the same Trunk Type (either E1 or T1).

5.6 Coefficient Configuration File



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

The *Coeff_FXS.dat* file is used to provide best termination and transmission quality adaptation for different line types for FXS interfaces / *prodtypeDevice*>. This adaptation is performed by modifying the telephony interface characteristics (such as DC and AC impedance, feeding current, and ringing voltage). The *coeff.dat* configuration file is produced specifically for each market after comprehensive performance analysis and testing, and can be modified on request. The current file supports US line type of 600 ohm AC impedance and 40 V RMS ringing voltage for REN = 2.

To load the *coeff.dat* file to the device, use the Embedded Web Server (refer to 'Auxiliary Files' in the device's *User's Manual*) or alternatively, specify the FXS *coeff.dat* file name in the device's *ini* file (refer to 'Loading the Auxiliary Files via the ini File' in the device's *User's Manual*).

The *Coeff.dat* file consists of a set of parameters for the signal processor of the loop interface devices. This parameter set provides control of the following AC and DC interface parameters:

- DC (battery) feed characteristics
- AC impedance matching
- Transmit gain
- Receive gain
- Hybrid balance
- Frequency response in transmit and receive direction

- Hook thresholds
- Ringing generation and detection parameters

This means, for example, that changing impedance matching or hybrid balance doesn't require hardware modifications, so that a single device is able to meet requirements for different markets. The digital design of the filters and gain stages also ensures high reliability, no drifts (over temperature or time) and simple variations between different line types.

In future software releases, it is to be expanded to consist of different sets of line parameters, which can be selected in the *ini* file, for each port.



Note: For MediaPack devices, use the parameter CountryCoefficients (described in 'Analog Telephony Parameters' in the User's Manual) to configure the FXO coefficients.

5.7 Dial Plan File



Note: This section is applicable only to AudioCodes' 3000 Series, 2000 Series, and Mediant 1000 devices.

The source file for the Dial Plan configuration contains a list of known prefixes (e.g. area codes and international telephone number patterns) for the PSTN to which the device is connected. The device uses this information to detect end-of-dialing in certain CAS configurations where the end-indicator (ST) is not used. The device supports up to 8,000 distinct prefixes in the dial-plan file.

The CasTrunkDialPlanName *ini* file parameter determines which Dial Plan (in a Dial Plan file) to use for a specific trunk (refer to 'Configuration Files Parameters' in the device's *User's Manual*). This can also be configured using the Embedded Web Server (refer to 'Trunk Settings' in the device's *User's Manual*).

The following is an example of an *ini* file that includes these definitions. This ini file is converted (using the TrunkPack Conversion Utility) to a binary file, and loaded to the device.

```

; Example of dial-plan configuration.
; This file contains two dial plans: you may specify which
; one to use in CAS configuration.
[ PLAN1 ]
; Define the area codes 02, 03, 04.
; In these area codes, phone numbers have 7 digits.
02,7
03,7
04,7
; Define the cellular/VoIP area codes 052, 054, 050, and 077.
; In these area codes, phone numbers have 8 digits.
052,8
054,8
050,8
077,8
; Define the international prefixes 00, 012, 014.
; The number following these prefixes may
; be 7 to 14 digits in length.
00,7-14
012,7-14
014,7-14
; Define the emergency number 911.
; No additional digits are expected.
911,0
[ PLAN2 ]
; Define the area codes 02, 03, 04.
; In these area codes, phone numbers have 7 digits.
0[2-4],7
; Operator services starting with a star: *41, *42, *43.
; No additional digits are expected.
*4[1-3],0
    
```

The list must be prepared in a textual *ini* file with the following syntax:

- Every line in the file defines a known dialing prefix and the number of digits expected to follow that prefix. The prefix must be separated from the number of additional digits by a comma (',').
- Empty lines are ignored.
- Lines beginning with a semicolon (;) are ignored.
- Multiple dial plans may be specified in one file; A name in square brackets on a separate line indicates the beginning of a new dial plan. Up to eight dial plans can be defined.
- Asterisks (*) and number-signs (#) can be specified as part of the prefix.
- Numeric ranges are allowed in the prefix.
- A numeric range is allowed in the number of additional digits.



Note: The prefixes must not overlap. Attempting to process an overlapping configuration in the TrunkPack Conversion Utility results in an error message specifying the problematic line.

5.8 User Information File

The User Information file maps PBX extensions (connected to the device) to global IP numbers. In this context, a global IP phone number (alphanumerical) serves as a routing identifier for calls in the 'IP World'. The PBX extension uses this mapping to emulate the behavior of an IP phone.



Note: The mapping mechanism is disabled by default and must be activated using the parameter `EnableUserInfoUsage` (described in 'SIP Configuration Parameters' in the device's *User's Manual*).

Each line in the file represents a mapping rule of a single PBX extension (up to 1000 rules can be configured). Each line includes five items separated with commas. The items are described in the table below. An example of a User Information file is shown in the figure below.

Each PBX extension registers separately (a REGISTER message is sent for each entry, only if `AuthenticationMode` is set to `Per Endpoint`) using the IP number in the From / To headers. The REGISTER messages are sent gradually (i.e., initially, the gateway sends requests according to the maximum number of allowed SIP dialogs (configured by the parameter `NumberOfActiveDialogs`), after each received response, the subsequent request is sent). Therefore, no more than `NumberOfActiveDialogs` dialogs are active simultaneously. The username and password are used for SIP Authentication when required.

The calling number of outgoing Tel-to-IP calls is first translated to an IP number and then (if defined), the manipulation rules are performed. The Display Name is used in the From header in addition to the IP number.

The called number of incoming IP-to-Tel calls is translated to a PBX extension only after manipulation rules (if defined) are performed.

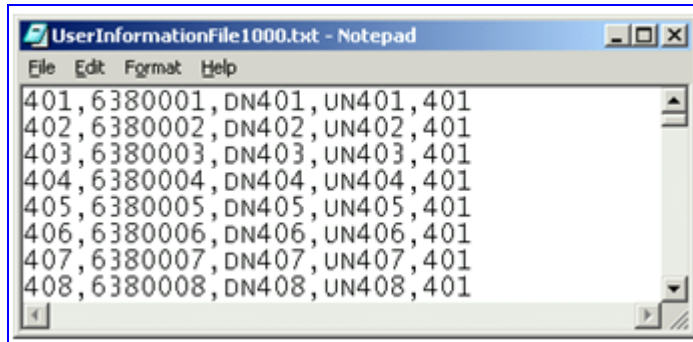
The User Information file is a text file that can be loaded via the *ini* file (`UserInfoFileName`, described in 'Loading the Auxiliary Files via the ini File' in the device's *User's Manual*), the Embedded Web Server (refer to 'Auxiliary Files' in the device's *User's Manual*), or by using the automatic update mechanism (`UserInfoFileURL`, refer to 'Automatic Update Mechanism' in the device's *User's Manual*).

The maximum permissible size of the file is as follows:

- **MediaPack:** 10,800 bytes
- **Mediant 1000:** 10,800 bytes for analog modules and 108,000 bytes for digital modules
- **3000 Series and 2000 Series:** 108,000 bytes

Table 5-1: User Information Items

Item	Description	Maximum Size
PBX extension #	The relevant PBX extension number.	10
Global phone #	The relevant global phone number.	20
Display name	A string that represents the PBX extensions for the Caller ID.	30
Username	A string that represents the username for SIP registration.	20
Password	A string that represents the password for SIP registration.	20

Figure 5-1: Example of a User Information File


```

UserInformationFile1000.txt - Notepad
File Edit Format Help
401,6380001, DN401, UN401, 401
402,6380002, DN402, UN402, 401
403,6380003, DN403, UN403, 401
404,6380004, DN404, UN404, 401
405,6380005, DN405, UN405, 401
406,6380006, DN406, UN406, 401
407,6380007, DN407, UN407, 401
408,6380008, DN408, UN408, 401
    
```


6 Automatic Configuration Options



Note: This section is applicable only to AudioCodes' Analog and 2000 Series devices.

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. Therefore, 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 achieved when the device is deployed in a live network.

This section describes the available options for performing fast, automatic configuration.

6.1 Local Configuration Server with BootP/TFTP



Note: This section is applicable only to AudioCodes' Analog and 2000 Series devices.

Local configuration server with BootP/TFTP provides the most efficient and easy alternative, as described below:

- A computer running BootP and TFTP software is located in a staging warehouse.
- A standard device configuration *ini* file is prepared and placed in the TFTP directory.
- BootP is configured with the MAC address of each device.
- Each device is connected to the network and powered-up.
- The BootP reply contains the *cmp* and *ini* file names in the 'bootfile' field. The device "fetches" these files and stores them in flash.
- 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.
- When the LEDs turn green, the device may be disconnected and shipped to the end customer.
- Local IP addressing at the customer site would normally be provided by DHCP.

This alternative requires configuration to take place at a staging warehouse.

6.2 DHCP-based Configuration Server



Note: This section is applicable only to AudioCodes' Analog and 2000 Series devices.

This alternative is similar to the setup described in 'Local configuration server with BootP/TFTP' on page 113, 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.

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 machine as the DHCP server (the "next-server" directive may be used otherwise).

```
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;
    }
}
```

This alternative requires configuration to take place at a staging warehouse.

6.3 HTTP-based Automatic Updates



Note: This section is applicable only to AudioCodes' Analog and 2000 Series devices.

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 end customer site; HTTP(S) is not sensitive to NAT devices, allowing configuration to take place as needed, without on-site intervention.

For additional security, the URL may contain a different port and a username 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 114 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 115).
- Manually on-site, using the RS-232 port or Web interface.

When the device is deployed at the end 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 MAC address 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.

6.4 Configuration using DHCP Option 67



Note: This section is applicable only to AudioCodes' Analog and 2000 Series devices.

This method is suitable for deployments where DHCP server configuration is feasible at the end 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.

Below is a sample Linux DHCP configuration file (*dhcpd.conf*) illustrating 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;
    }
}

```

This alternative does not require additional servers at the customer premises and is NAT-safe.

6.5 Configuration using FTP or NFS



Note: This section is applicable only to AudioCodes' Analog and 2000 Series devices.

Some networks block access to HTTP(S). The Automatic Update facility provides limited support for FTP/FTPS connectivity. However, note that 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 114 and 'Configuration using DHCP Option 67' on page 115 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.

6.6 TFTP Configuration using DHCP Option 66



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

This particular method is suitable for cases where the end customer 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 looks 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".

The configuration file loaded in this method is a one-time action; the download is not repeated until the device is manually restored to factory defaults (pressing the reset button for 10 seconds while the Ethernet cable is not connected).

This alternative requires a configuration server at the customer premises. In addition, TFTP access to the core network is not NAT-safe.

6.7 Configuration using AudioCodes EMS



Note: This section is applicable only to AudioCodes' Analog and 2000 Series devices.

AudioCodes EMS server functions as a core-network provisioning server. The device should be configured with the IP address of the EMS server as the SNMP manager, using one of the methods detailed in the previous sections.

As soon as a registered device contacts the EMS server via 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.

Reader's Notes

7 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 119).
- 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 122).
- Secured RTP (SRTP) according to RFC 3711, used to encrypt RTP and RTCP transport (refer to 'SRTP' on page 124).
- 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 125).
- Internal Firewall allows filtering unwanted inbound traffic (refer to 'Internal Firewall' on page 129).

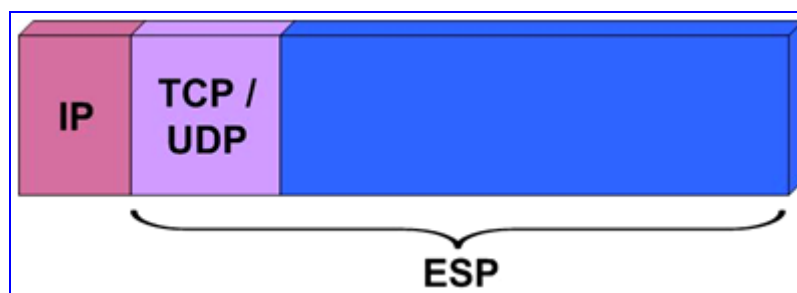
7.1 IPsec and IKE

IPsec and 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 7-1: IPsec Encryption



Note: IPsec doesn't function properly 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.

7.1.1 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

7.1.2 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 an SA already exists for this entry. If it doesn't, the IKE protocol is invoked (refer to 'IKE' on page 120) 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 matches one of the entries of the SPD table and is received un-encrypted is dropped.

IPSec Specifications:

- Transport mode only
- 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

7.1.3 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, Embedded Web Server and SNMP agent to maintain this parameter's confidentiality. In the Embedded Web Server, 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 available. 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's IPSec tables, load a new *ini* file with an empty IPSec table, shown below.

```

[IPSec IKEDB Table]
[\\IPSec_IKEDB_Table]

[IPSEC SPD TABLE]
[\\IPSEC_SPD_TABLE]

```

7.2 SSL / TLS

Secure Socket Layer (SSL), also known as Transport Layer Security (TLS) is the method used to secure the device SIP signaling connections, Embedded Web Server, 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:

- Supported transports: SSL 2.0, SSL 3.0, TLS 1.0
- Supported ciphers: DES, RC4 compatible, Advanced Encryption Standard (AES)
- Authentication: X.509 certificates (CRLs are currently not supported)



Tip: For additional security, consider setting configuration parameter `TLSVersion` to 1 (refer to 'Configuring the General Security Settings' in the device's *User's Manual*). This setting ensures that SSL (which is considered less secure than TLS) is never used. 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**).

7.2.1 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 used, 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*.

7.2.2 Secured HTTPS Embedded Web Server Configuration

For additional security, you can configure the Embedded Web Server to accept only secured (HTTPS) connections by changing the parameter `HTTPSOnly` to 1 (described in Web and 'Telnet Parameters' 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 'Web and Telnet Parameters' in the device's *User's Manual*).

➤ **To use the secured Embedded Web Server, 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
```

7.2.3 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 ]
```

7.3 SRTP

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

SRTP requires a Key Exchange mechanism that is performed according to <draft-ietf-mmusic-sdescriptions-12>. 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.

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 'Security' in the device's *User's Manual*) to select the device's mode of operation that determine 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 'Security' 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.

7.4 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 usernames, 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 username and password. When RADIUS authentication isn't used, the username and password are authenticated with the Embedded Web Server's usernames 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 username and password stored internally in the device's memory. When RADIUS authentication is used, the device doesn't store the username 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 username 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.

7.4.1 Setting Up a RADIUS Server

The following examples refer to FreeRADIUS, a free RADIUS server that can be downloaded from 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 127.

7.4.2 Configuring RADIUS Support

For information on the RADIUS parameters, refer to 'RADIUS Parameters' in the device's *User's Manual*.

➤ **To configure RADIUS support on the device via the Embedded Web Server, take these 13 steps:**

1. Access the Embedded Web Server (refer to 'Accessing the Embedded Web Server' in 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 username and password.
7. In the field 'Local RADIUS Password Cache Timeout', enter a time (in seconds); when this time expires, the username and password verified by the RADIUS server becomes invalid and a username 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 124).
12. Save the changes so they are available after a power fail.
13. Reset the device (refer to Resetting the Device' in the device's *User's Manual*).

After reset, when accessing the Web or Telnet servers, use the username 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.

7.5 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 'Networking Parameters' 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)

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.

The internal firewall can also be configured via the Embedded Web Server (refer to Configuring the 'Firewall Settings' in the device's *User's Manual*).

7.6 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 7-1: 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 HTTPOnly). 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	

7.7 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 125).
(**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 HTTPOnly 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 60).
- 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.

7.8 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' (ey@cryptsoft.com).

8 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 8-1: RTCP-XR Published VoIP Metrics

Metric Name
General
Start Timestamp
Stop Timestamp
Call-ID
Local Address (IP, Port & SSRC)
Remote Address (IP, Port & SSRC)
Session Description
Payload Type
Payload Description
Sample Rate
Frame Duration
Frame Octets

Metric Name
Frames per Packets
Packet Loss Concealment
Silence Suppression State
Jitter Buffer
Jitter Buffer Adaptive
Jitter Buffer Rate
Jitter Buffer Nominal
Jitter Buffer Max
Jitter Buffer Abs Max
Packet Loss
Network Packet Loss Rate
Jitter Buffer Discard Rate
Burst Gap Loss
Burst Loss Density
Burst Duration
Gap Loss Density
Gap Duration
Minimum Gap Threshold
Delay
Round Trip Delay
End System Delay
One Way Delay
Interarrival Jitter
Min Absolute Jitter
Signal
Signal Level
Noise Level
Residual Echo Return Noise
Quality Estimates
Listening Quality R
RLQ Est. Algorithm
Conversational Quality R
RCQ Est. Algorithm
External R In
Ext. R In Est. Algorithm

Metric Name
External R Out
Ext. R Out Est. Algorithm
MOS-LQ
MOS-LQ Est. Algorithm
MOS-CQ
MOS-CQ Est. Algorithm
QoE Est. Algorithm

Reader's Notes

9 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.

9.1 Payload Types Defined in RFC 3551

Table 9-1: Packet Types Defined in RFC 3551

Payload Type	Description	Basic Packet Rate (msec)
0	G.711 μ -Law	10,20
2	G.726-32	10, 20
3	GSM-FR Note: 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 Note: Only applicable to 3000 Series.	20
12	QCELP Note: 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	

9.2 Defined Payload Types

Table 9-2: Defined Payload Types

Payload Type	Description	Basic Packet Rate (msec)
3	MS-GSM Note: Only applicable to 2000 Series and 3000 Series.	40
3	GSM-EFR Note: Only applicable to 2000 Series and 3000 Series.	20
22	G.726-24 Note: Only applicable to 2000 Series and 3000 Series.	20
23	G.726-16 Note: Only applicable to 2000 Series and 3000 Series.	20
38	G.726-40 Note: Only applicable to 2000 Series and 3000 Series.	20
51	NetCoder 6.4 kbps Note: Only applicable to 2000 Series and 3000 Series.	20
52	NetCoder 7.2 kbps Note: Only applicable to 2000 Series and 3000 Series.	20
53	NetCoder 8.0 kbps Note: Only applicable to 2000 Series and 3000 Series.	20
54	NetCoder 8.8 kbps Note: Only applicable to 2000 Series and 3000 Series.	20
56	Transparent PCM Note: Only applicable to 2000 Series and 3000 Series.	20
60	EVRC Note: Only applicable to 2000 Series and 3000 Series.	20
64	AMR Note: Only applicable to 2000 Series and 3000 Series.	20
64	AMR-WB Note: Only applicable to 3000 Series.	20
65	iLBC Note: Only applicable to 2000 Series and 3000 Series.	20, 30

Payload Type	Description	Basic Packet Rate (msec)
68	EVRC-B (4GV) Note: Only applicable to 3000 Series.	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	

9.3 Default RTP / RTCP / T.38 Port Allocation

The following table describes device default RTP/RTCP/T.38 port allocation.

Table 9-3: Default RTP/RTCP/T.38 Port Allocation

	Channel Number	RTP Port	RTCP Port	T.38 Port
	1	6000	6001	6002
	2	6010	6011	6012
	3	6020	6021	6022
	4	6030	6031	6032
	5	6040	6041	6042
	6	6050	6051	6052
	7	6060	6061	6062
	8	6070	6071	6072
	:	:	:	:
	n	6000 + 10(n-1)	6001 + 10(n-1)	6002 + 10(n-1)
Maximum:	:	:	:	:
MediaPack	24	6230	6231	6232
Mediant 1000	120	7190	7191	7192
3000 and 2000 Series	480	10790	10791	10792



Notes:

- 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 blades -- TP-1610 / IPM-1610 -- or with a TP-260 / IPM-260).

10 CAS Protocol Table



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

10.1 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.

10.2 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

10.2.1 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.

10.2.2 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.

10.2.3 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.

10.2.4 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 10-1: 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
INIT_DTMF_DIAL	Defines the On-time and Off-time for the DTMF digits generated towards the PSTN.
INIT_COMMA_PAUSE_TIME	Defines the delay between each digit when a comma is used as part of the dialed number string (refer to acPSTNPlaceCall for details).
INIT_DTMF_DETECTION	Defines the minimum/maximum On-time for DTMF digit dialing detection.
INIT_PULSE_DIAL_TIME	Not supported by the current stack version. Defines the Break and Make time for pulse dialing.
INIT_PULSE_DIAL	Not supported by the current stack version. Defines the Break and Make ABCD bits for pulse dialing.
INIT_DEBOUNCE	Defines the interval time of CAS to be considered (a stable one).
INIT_COLLECT_ANI	Enables or Disables reception of ANI in a specific protocol.

Parameter	Description
INIT_DIGIT_TYPE	<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>
INIT_NUM_OF_EVENT_IN_STATE	<p>Inserted for detection on TOTAL_NUMBER_OF_EVENTS_IN_STATE (CASSetup.h).</p>
INIT_INIT_GLOBAL_TIMERS	<p>Initiates specific timers; it is used with Parameter#1 for metering pulse timer duration.</p>
INIT_PULSE_DIAL_ADDITIONAL_PARAMS	<p>Not used.</p>
INIT_RINGING_TO_ANALOGUE	<p>When using analogue gateway option, it defines the CAS value of ringing (#1) CAS value of silence (#2) and CAS value of polarity relevsal(#3).</p>
INIT_DIGIT_TYPE_1	<p>Defines the signaling system used to send operator service.</p>
INIT_REJECT_COLLECT	<p>Defines the method for reject collect calls: <i>disabled</i>, <i>using Line signaling</i>, or <i>using register signaling</i>.</p>
INIT_VERSION	<p>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).</p>
INIT_SIZE_OF_TABLE_PARAM	<p>Users must insert the definition of TOTAL_NUMBER_OF_EVENTS_IN_STATE from CASSetup.h.</p>

10.3 Reserved Words

For reserved words such as DO, NO_STATE, etc., refer to the detailed list in CASSetup.h.

10.4 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

10.5 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).

10.5.1 User Command Oriented Action / Event

Table 10-2: 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.

10.5.2 CAS Change Oriented Events

Table 10-3: 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).
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.

10.5.3 Timer Oriented Events

Table 10-4: 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.

10.5.4 Counter Oriented Events

Table 10-5: 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.

10.5.5 IBS Oriented Events

Table 10-6: 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.
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.

10.5.6 DTMF/MF Oriented Events

Table 10-7: 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.

Event	Explanation
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.
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 10-8: 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.
EV_ACCEPT_SPARE_MF10	When acCASAacceptCall is used with SPARE_MF10 as its reason parameter.
EV_ACCEPT_SPARE_MF11	When acCASAacceptCall is used with SPARE_MF11 as its reason parameter.
EV_ACCEPT_SPARE_MF12	When acCASAacceptCall is used with SPARE_MF12 as its reason parameter.
EV_ACCEPT_SPARE_MF13	When acCASAacceptCall is used with SPARE_MF13 as its reason parameter.
EV_ACCEPT_SPARE_MF14	When acCASAacceptCall is used with SPARE_MF14 as its reason parameter.
EV_ACCEPT_SPARE_MF15	When acCASAacceptCall is used with SPARE_MF 15 as its reason parameter.
EV_REJECT_BUSY	When acCASAacceptCall is used with CALLED_BUSY as its reason parameter.
EV_REJECT_CONGESTION	When acCASAacceptCall is used with CALLED_CONGESTION as its reason parameter.
EV_REJECT_UNALLOCATED	When acCASAacceptCall is used with CALLED_UNALLOCATED as its reason parameter.
EV_REJECT_SIT	When acCASAacceptCall is used with SIT as its reason parameter.
EV_REJECT_RESERVE1	When acCASAacceptCall is used with CALLED_RESERVE1 as its reason parameter.
EV_REJECT_RESERVE2	When acCASAacceptCall is used with CALLED_RESERVE2 as its reason parameter.

10.5.7 Operator Service Events (up to GR-506)

Table 10-9: Operator Service Events (Up to GR-506)

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

10.6 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.

10.7 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 10-10: Available User Functions and Corresponding Parameters

User Function	User Function Parameters and Descriptions
SET_TIMER	(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.
SEND_CAS	(AB value, CD value). ABCD bits are sent as line signaling for the specific channel when the call is setup.
GENERATE_CAS_EV	Check the ABCD bits value, and send a proper event to the state machine.
SEND_EVENT	(Event type, cause). The specific event type is sent to the host/user and retrieved by applying the function acGetEvent().
SEND_DEST_NUM	En-bloc dialing: refers to the digits string located in function acPSTNPlaceCall. Three types are available: (1) DestPhoneNum (2) InterExchangePrefixNum (3) SourcePhoneNum.
DEL_TIMER	(Timer number). Deletes a specific timer or all the timers (0 represents all the timers) for the B-channel.

User Function	User Function Parameters and Descriptions
START_COLLECT	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.
STOP_COLLECT	Refer to START_COLLECT.
SET_COUNTER	(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.
DEC_COUNTER	(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).
RESTRICT_ANI	Indicate the incoming side to hide the ANI from the Far-end user.
SEND_MF	(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 10-11: Parameters Associated with Sending Digits

Parameter	Description
AddressVector and ANIDigitVector	<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"> ▪ ADDRESS: Sends the digit from the address vector (destination number) according to the index requested. Refer to the Index definition. ▪ ANI: Sends the digit from the ANI vector (source number) according to the requested index. ▪ SPECIFIC: Sends the MF digit specified in the cell Parameter #2. ▪ SOURCE_CATEGORY: Sends the predefined source category MF digit. 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. ▪ TRANSFER_CAPABILITY: 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.
Index	<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"> ▪ Index 1: Sends the next digit in the vector. ▪ Index -n: Sends the last n digit. Underflow can occur if n is greater than the number of digits sent so far. ▪ Index 0: Sends the last sent digit. ▪ Index SEND_FIRST_DIGIT: Starts sending the digits vector from the beginning (refer to CASSetup.h).

Parameter	Description
MF Send Time	This send time parameter specifies the maximum transmission time of the MF. <ul style="list-style-type: none"> ▪ STOP_SEND_MF: Stops sending the current MF. ▪ SEND_PROG_TON: Operation, Tone or NONE.

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
CHANGE_COLLECT_TYPE	(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: <ul style="list-style-type: none"> ▪ ADDRESS: The user waits for receipt of address digits. ▪ ANI: The user waits for receipt of ANI digits. ▪ SOURCE_CATEGORY: The user waits for receipt of the source category. ▪ TRANSFER_CAPABILITY: The user waits for receipt of the source transfer capability (line category).

10.8 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.

10.9 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 l 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.

10.9.1 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 digit
This 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 `acCASAcceptCall()` with a certain reason parameter. Six reason parameters are available:

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

11 SS7 Tunneling



Note: This section is applicable only to AudioCodes' 3000 Series 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 11-1: M2UA Architecture

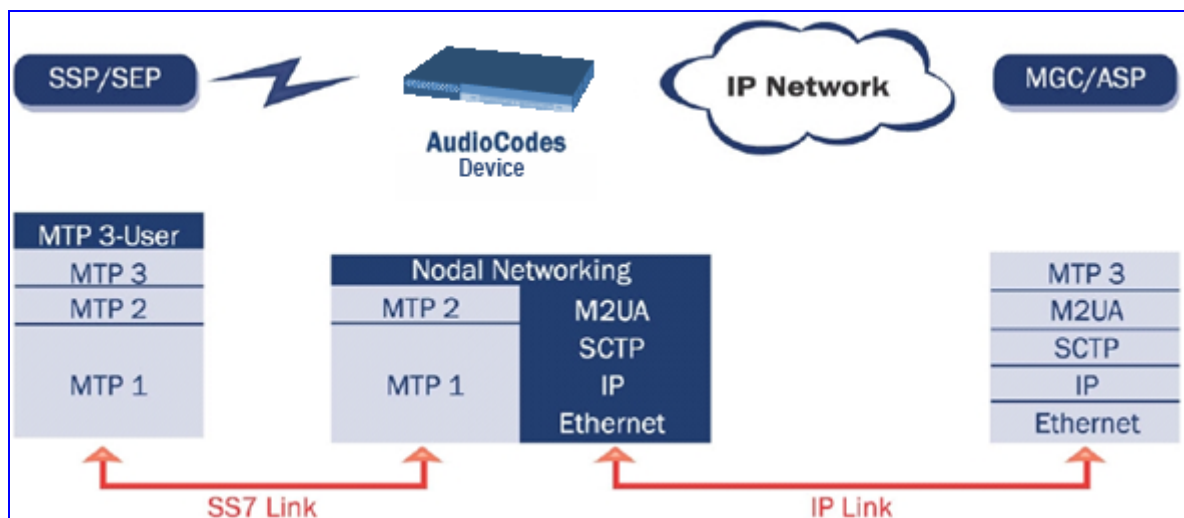
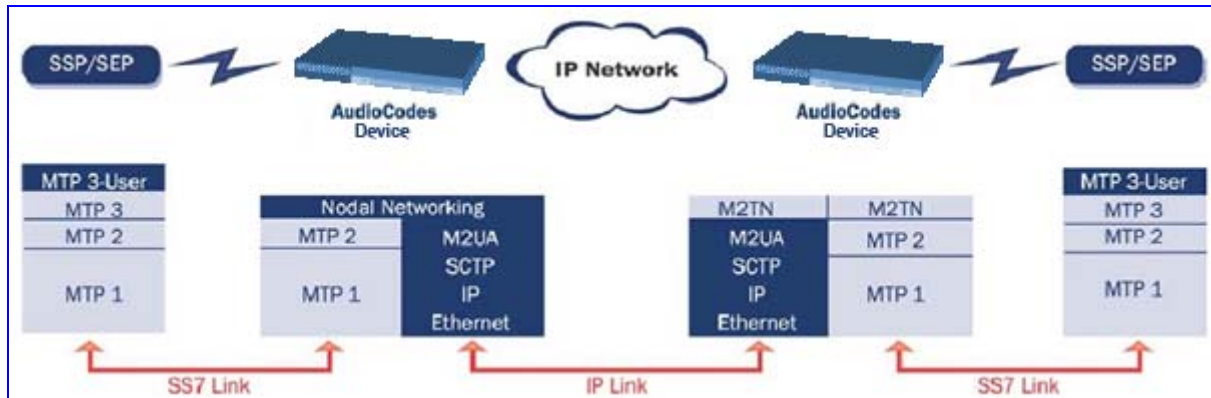


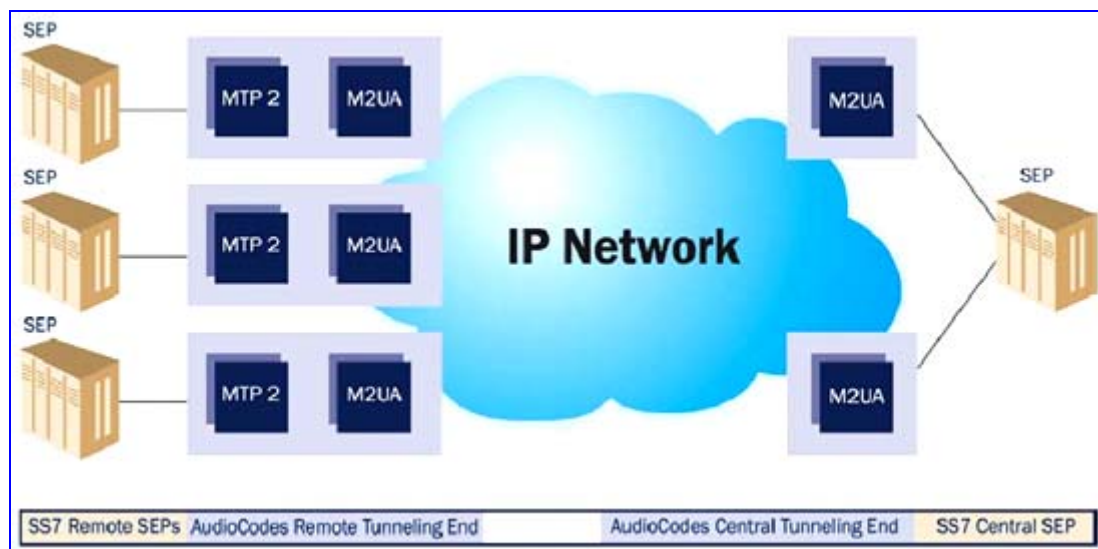
Figure 11-2: M2TN Architecture



11.1 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 11-3: Protocol Architecture for MTP2 Tunneling



11.2 SS7 Characteristics

- 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 gateway).
- 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.

11.3 SS7 Parameter Tables

The SS7 parameter tables can be configured using the Web interface ('Configuring SS7 Tunneling' on page 171) or the *ini* file parameters ('SS7 Parameters' on page 160). For detailed information on *ini* file parameter tables, refer to 'Configuring Parameter Tables Using the ini File' in the device's *User's Manual*.

11.4 SS7 Parameters

Table 11-1: SS7 Parameters

<i>ini</i> File Name	Valid Range and Description
SS7 MTP2 Parameter Table	
SS7Mtp2Parms (MTP2 Table)	Configures the SS7 MTP2 table parameters (refer to 'Configuring MTP2 Attributes' on page 172 for configuring via the Embedded Web Server). [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] For example: [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]
SS7Mtp2Parms_AermTie	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SS7Mtp2Parms_AermTin	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SS7Mtp2Parms_ErrorCorrectionMethod	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SS7Mtp2Parms_lacCp	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SS7Mtp2Parms_LinkRate	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SS7Mtp2Parms_LSSULen gth	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SS7Mtp2Parms_OctetCou nting	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SS7Mtp2Parms_SuermSu D	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SS7Mtp2Parms_SuermT	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SS7Mtp2Parms_PcrN2	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.

<i>ini</i> File Name	Valid Range and Description
SS7Mtp2Parms_T1	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SS7Mtp2Parms_T2	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SS7Mtp2Parms_T3	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SS7Mtp2Parms_T4E	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SS7Mtp2Parms_T4N	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SS7Mtp2Parms_T5	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SS7Mtp2Parms_T6	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SS7Mtp2Parms_T7	For a description of this parameter, refer to 'Configuring MTP2 Attributes' on page 172.
SIGTRAN Interface Groups Table	
SS7_SIG_IF_GROUP_TABLE	Configures the Sigtran Interface Group table (refer to 'Configuring Sigtran Group IDs' on page 184 for configuring via the Embedded Web Server). [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_IP, SS7_MGC_MX_IN_STREAM, SS7_MGC_NUM_OUT_STREAM; [SS7_SIG_IF_GROUP_TABLE]
SS7_SIG_IF_GR_INDEX	For a description of this parameter, refer to 'Configuring Sigtran Group IDs' on page 184.
SS7_IF_GR_ID	For a description of this parameter, refer to 'Configuring Sigtran Group IDs' on page 184.
SS7_SIG_SG_MGC	For a description of this parameter, refer to 'Configuring Sigtran Group IDs' on page 184.
SS7_SIG_LAYER	For a description of this parameter, refer to 'Configuring Sigtran Group IDs' on page 184.
SS7_SIG_TRAF_MODE	For a description of this parameter, refer to 'Configuring Sigtran Group IDs' on page 184.
SS7_SIG_T_REC	For a description of this parameter, refer to 'Configuring Sigtran Group IDs' on page 184.
SS7_SIG_T_ACK	For a description of this parameter, refer to 'Configuring Sigtran Group IDs' on page 184.
SS7_SIG_T_HB	For a description of this parameter, refer to 'Configuring Sigtran Group IDs' on page 184.

<i>ini</i> File Name	Valid Range and Description
SS7_SIG_MIN_ASP	For a description of this parameter, refer to 'Configuring Sigtran Group IDs' on page 184.
SS7_SIG_BEHAVIOUR	For a description of this parameter, refer to 'Configuring Sigtran Group IDs' on page 184.
SS7_LOCAL_SCTP_PORT	For a description of this parameter, refer to 'Configuring Sigtran Group IDs' on page 184.
SS7_SIG_NETWORK	For a description of this parameter, refer to 'Configuring Sigtran Group IDs' on page 184.
SS7_DEST_SCTP_PORT	For a description of this parameter, refer to 'Configuring Sigtran Group IDs' on page 184.
SS7_DEST_IP	For a description of this parameter, refer to 'Configuring Sigtran Group IDs' on page 184.
SS7_MGC_MX_IN_STREAM	For a description of this parameter, refer to 'Configuring Sigtran Group IDs' on page 184.
SS7_MGC_NUM_OUT_STREAM	For a description of this parameter, refer to 'Configuring Sigtran Group IDs' on page 184.
SIGTRAN Interface IDs Table	
SS7_SIG_INT_ID_TABLE (SIGTRAN Interface IDs table)	<p>Configures the Sigtran Interface IDs table (refer to 'Configuring Sigtran Interface IDs' on page 186 for configuring via the Embedded Web Server).</p> <p>[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>
SS7_SIG_IF_ID_INDEX	For a description of this parameter, refer to 'Configuring Sigtran Interface IDs' on page 186.
SS7_SIG_IF_ID_VALUE	For a description of this parameter, refer to 'Configuring Sigtran Interface IDs' on page 186.
SS7_SIG_IF_ID_NAME	For a description of this parameter, refer to 'Configuring Sigtran Interface IDs' on page 186.
SS7_SIG_IF_ID_OWNER_GROUP	For a description of this parameter, refer to 'Configuring Sigtran Interface IDs' on page 186.
SS7_SIG_IF_ID_LAYER	For a description of this parameter, refer to 'Configuring Sigtran Interface IDs' on page 186.
SS7_SIG_IF_ID_NAI	For a description of this parameter, refer to 'Configuring Sigtran Interface IDs' on page 186.
SS7_SIG_M3UA_SPC	For a description of this parameter, refer to 'Configuring Sigtran Interface IDs' on page 186.

<i>ini</i> File Name	Valid Range and Description
SS7 Signaling Link Table	
SS7_LINK_TABLE (SS7 Link table)	Configures the SS7 Links table (refer to 'Configuring Links' on page 180 for configuring via the Embedded Web Server). [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_TABLE] 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]
SS7_LINK_INDEX	For a description of this parameter, refer to 'Configuring Links' on page 180.
SS7_LINK_ACTION	Determines the management field for actions. <ul style="list-style-type: none"> ▪ [0] = acSS7LINK_PS_ACTION_NONE (default) ▪ [1] = acSS7LINK_PS_ACTION_OFFLINE ▪ [2] = acSS7LINK_PS_ACTION_INSERTSERVICE ▪ [3] = acSS7LINK_PS_ACTION_ACTIVATE ▪ [4] = acSS7LINK_PS_ACTION_DEACTIVATE ▪ [5] = acSS7LINK_PS_ACTION_INHIBIT ▪ [6] = acSS7LINK_PS_ACTION_UNINHIBIT
SS7_LINK_ACTION_RESULT	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.
SS7_LINK_NAME	For a description of this parameter, refer to 'Configuring Links' on page 180.
SS7_LINK_OPERATIONAL_STATE	For a description of this parameter, refer to 'Configuring Links' on page 180.
SS7_LINK_ADMINISTRATIVE_STATE	For a description of this parameter, refer to 'Configuring Links' on page 180.
SS7_LINK_TRACE_LEVEL	For a description of this parameter, refer to 'Configuring Links' on page 180.
SS7_LINK_L2_TYPE	For a description of this parameter, refer to 'Configuring Links' on page 180.
SS7_LINK_L3_TYPE	For a description of this parameter, refer to 'Configuring Links' on page 180.
SS7_LINK_TRUNK_NUMBER	For a description of this parameter, refer to 'Configuring Links' on page 180.

<i>ini</i> File Name	Valid Range and Description
SS7_LINK_TIMESLOT_NUMBER	For a description of this parameter, refer to 'Configuring Links' on page 180.
SS7_LINK_MTC_BUSY	For a description of this parameter, refer to 'Configuring Links' on page 180.
SS7_LINK_LAYER2_VARIANT	For a description of this parameter, refer to 'Configuring Links' on page 180.
SS7_LINK_MTP2_ATTRIBUTES	For a description of this parameter, refer to 'Configuring Links' on page 180.
SS7_CONGESTION_LOW_MARK	For a description of this parameter, refer to 'Configuring Links' on page 180.
SS7_CONGESTION_HIGH_MARK	For a description of this parameter, refer to 'Configuring Links' on page 180.
SS7_LINK_M2UA_IF_ID	For a description of this parameter, refer to 'Configuring Links' on page 180.
SS7_LINK_GROUP_ID	For a description of this parameter, refer to 'Configuring Links' on page 180.
SS7_LINK_TNL_MGC_LINK_NUMBER	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.
SS7_LINK_TNL_ALIGNMENT_MODE	Determines the MTP2 Tunneling: Alignment mode of signaling links in tunnel. <ul style="list-style-type: none"> ▪ [0] = M3B_ALIGNMENT_NORMAL ▪ [1] = M3B_ALIGNMENT_EMERGENCY (default)
SS7_LINK_TNL_CONGESTION_MODE	Determines the MTP2 Tunneling: Congestion mode of signaling links in tunnel. <ul style="list-style-type: none"> ▪ [0] = M3B_CONGESTION_ACCEPT (default) ▪ [1] = M3B_CONGESTION_DISCARD
SS7_LINK_TNL_WAIT_START_COMPLETE_TIMER	Determines the MTP2 Tunneling Timer: wait start complete. The valid range is 500 to 0xFFFFFFFF. The default value is 30000.
SS7_LINK_TNL_OOS_START_DELAY_TIMER	Determines the MTP2 Tunneling Timer: OOS start delay. The valid range is 500 to 0xFFFFFFFF. The default value is 5000.
SS7_LINK_TNL_WAIT_OTHER_SIDE_INSV_TIMER	Determines the MTP2 Tunneling Timer: wait other side inservice. The valid range is 500 to 0xFFFFFFFF. The default value is 30000.
SS7 Link-Set Parameters	
SS7_LINKSET_SN_INDEX [SN Number]	Determines the first index field for line. The valid range is 0 to 1. The default value is 0.
SS7_LINKSET_LINKSET_INDEX [Link-set Number]	Determines the second index field for line. The valid range is 0 to 83. The default value is 0.
SS7_LINKSET_ROWSTATUS	Determines the RowStatusField for line. The valid range is acPARAMSET_ROWSTATUS_DOESNOTEXIST to acPARAMSET_ROWSTATUS_DESTROY. The default value is acPARAMSET_ROWSTATUS_DOESNOTEXIST.

<i>ini</i> File Name	Valid Range and Description
SS7_LINKSET_ACTION	Determines the management field for actions. <ul style="list-style-type: none"> ▪ [0] = acSS7LINKSET_PS_ACTION_NONE (default) ▪ [1] = acSS7LINKSET_PS_ACTION_OFFLINE ▪ [2] = acSS7LINKSET_PS_ACTION_INSERTSERVICE ▪ [3] = acSS7LINKSET_PS_ACTION_ACTIVATE ▪ [4] = acSS7LINKSET_PS_ACTION_DEACTIVATE
SS7_SN_ACTION_RESULT	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.
SS7 Signaling Node Timers Parameter Table	
SS7_SN_TIMERS_TABLE (SS7 Signaling Node Timers table)	Configures the SS7 Signaling Node Timers table parameters (refer to 'Configuring SS7 Signaling Node Timers' on page 174 for configuring via the Embedded Web Server). [SS7_SN_TIMERS_TABLE] FORMAT SS7_SNTIMERS_INDEX = SS7_SNTIMERS_NAME, SS7_SNTIMERS_T6, SS7_SNTIMERS_T8, SS7_SNTIMERS_T10, SS7_SNTIMERS_T11, SS7_SNTIMERS_T15, SS7_SNTIMERS_T16, SS7_SNTIMERS_T22_ANSI, SS7_SNTIMERS_T23_ANSI, SS7_SNTIMERS_T24_ANSI, SS7_SNTIMERS_T25_ANSI, SS7_SNTIMERS_T26_ANSI, SS7_SNTIMERS_T28_ANSI, SS7_SNTIMERS_T29_ANSI, SS7_SNTIMERS_T30_ANSI, SS7_SNTIMERS_T18_ITU, SS7_SNTIMERS_T19_ITU, SS7_SNTIMERS_T20_ITU, SS7_SNTIMERS_T21_ITU, SS7_SNTIMERS_T24_ITU; [SS7_SN_TIMERS_TABLE] 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]
SS7_SNTIMERS_INDEX	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174.
SS7_SNTIMERS_NAME	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174.
SS7_SNTIMERS_T6	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174.
SS7_SNTIMERS_T8	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174.
SS7_SNTIMERS_T10	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174.
SS7_SNTIMERS_T11	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174.
SS7_SNTIMERS_T15	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174.

<i>ini</i> File Name	Valid Range and Description
SS7_SNTIMERS_T16	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174 .
SS7_SNTIMERS_T18_ITU	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174 .
SS7_SNTIMERS_T19_ITU	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174 .
SS7_SNTIMERS_T20_ITU	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174 .
SS7_SNTIMERS_T21_ITU	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174 .
SS7_SNTIMERS_T24_ITU	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174 .
SS7_SNTIMERS_T22_ANS I	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174 .
SS7_SNTIMERS_T23_ANS I	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174 .
SS7_SNTIMERS_T24_ANS I	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174 .
SS7_SNTIMERS_T25_ANS I	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174 .
SS7_SNTIMERS_T26_ANS I	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174 .
SS7_SNTIMERS_T28_ANS I	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174 .
SS7_SNTIMERS_T29_ANS I	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174 .
SS7_SNTIMERS_T30_ANS I	For a description of this parameter, refer to 'Configuring SS7 Signaling Node Timers' on page 174 .

<i>ini</i> File Name	Valid Range and Description
SS7 Link-Set Timers Parameter Table	
SS7_LINKSET_TIMERS_TABLE (SS7 Link Set Timers table)	Configures the SS7 Link Set Timers table parameters (refer to 'Configuring Link-Set Timers' on page 178 for configuring via the Embedded Web Server). [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] 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]
SS7_LKSETTIMERS_INDEX	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.
SS7_LKSETTIMERS_NAME	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.
SS7_LKSETTIMERS_T2SLT	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.
SS7_LKSETTIMERS_T1	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.
SS7_LKSETTIMERS_T2	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.
SS7_LKSETTIMERS_T3	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.
SS7_LKSETTIMERS_T4	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.
SS7_LKSETTIMERS_T5	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.
SS7_LKSETTIMERS_T7	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.
SS7_LKSETTIMERS_T12	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.
SS7_LKSETTIMERS_T13	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.
SS7_LKSETTIMERS_T14	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.
SS7_LKSETTIMERS_T17	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.

<i>ini</i> File Name	Valid Range and Description
SS7_LKSETTIMERS_T22_ITU	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.
SS7_LKSETTIMERS_T23_ITU	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.
SS7_LKSETTIMERS_T20_ANSI	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.
SS7_LKSETTIMERS_T21_ANSI	For a description of this parameter, refer to 'Configuring Link-Set Timers' on page 178.

11.5 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 M ARK, 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 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, ROTERDAM , 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, ROTERDAM , 4, 4, 2, 0;
SS7 SIG INT ID TABLE 10 = 1, GAUDA , 4, 4, 3, 0;
[ \SS7 SIG INT ID TABLE ]
    
```

11.6 Configuring SS7 Tunneling

You can configure SS7 in the Embedded Web Server from the **Advanced Configuration** menu:

- Configure M2P2 Attributes (refer to 'Configuring M2P2 Attributes' on page [172](#))
- Configure SS7 Signaling Node Timers ('Configuring SS7 Signaling Node Timers' on page [174](#))
- Configure SS7 Link-Set Timers ('Configuring Link-Set Timers' on page [178](#))
- Configure Links (refer to 'Configuring Links' on page [180](#))
- Configure SS7 Signaling Nodes ('Configuring SS7 Signaling Nodes' on page [182](#))
- Configure Sigtran Group IDs (refer to 'Configuring Sigtran Group IDs' on page [184](#))
- Configure Sigtran Interface IDs (refer to 'Configuring Sigtran Interface IDs' on page [186](#))

11.6.1 Configuring MTP2 Attributes

The 'MTP2 Attributes' screen 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 160).

For a detailed description of MTP2, refer to 'MTP2 Tunneling Technology' on page 158.

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

1. Open the 'MTP2 Attributes' screen (**Advanced Configuration** menu > **SS7 Configuration** > **MTP2 Attributes** option); the 'MTP2 Attributes' screen is displayed.

Figure 11-4: MTP2 Attributes Screen

MTP2 Attributes	
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 or modify the parameters according to the table below.

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

Table 11-2: MTP2 Parameters

Parameter	Description
Link Rate [SS7Mtp2Parms_LinkRate]	Defines the SS7 SLI Link Rate. Choose either: <ul style="list-style-type: none"> ▪ [0] 0 = 64 kbps (default) ▪ [A] A = 64 kbps ▪ [D] D = 56 kbps
Error Correction Method [SS7Mtp2Parms_ErrorCorrectionMethod]	Defines the SLI error correction method. <ul style="list-style-type: none"> ▪ [0] 0 = Basic (default) ▪ [B] B = Basic ▪ [P] P = PCR (Preventive Cyclic Retransmission)
IAC CP [SS7Mtp2Parms_lacCp]	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.
SUERM T [SS7Mtp2Parms_SuermT]	Defines the SS7 SUERM (Signal Unit Error Rate Monitor) T threshold. The valid range is 0 to 256. The default value is 64.
AERM TIN [SS7Mtp2Parms_AermTin]	Defines the SS7 alignment normal error rate threshold. The valid range is 0 to 20. The default value is 4.
AERM TIE [SS7Mtp2Parms_AermTie]	Defines the SS7 alignment emergency error rate threshold. The valid range is 0 to 10. The default value is 1.
SUERM SU D [SS7Mtp2Parms_SuermSuD]	Defines the SS7 Signal Unit error rate monitor D threshold. The valid range is 0 to 256. The default value is 256.
Octet Counting [SS7Mtp2Parms_OctetCounting]	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.
LSSU Length [SS7Mtp2Parms_LSSULength]	Defines the SS7 MTP2 LSSU length as 1 or 2 (bytes). The valid range is 1 to 2. The default value is 1.
PCR N2 [SS7Mtp2Parms_PcrN2]	Number of message signal unit octets available for retransmission. The valid range is 0 to 512. The default value is 200.
T1 [SS7Mtp2Parms_T1]	Defines the SS7 MTP2 T1 alignment ready timer (in msec). The valid range is 0 to 100000. The default value is 50000.
T2 [SS7Mtp2Parms_T2]	Defines the SS7 MTP2 T2 unaligned timer (in msec). The valid range is 0 to 200000. The default value is 150000.
T3 [SS7Mtp2Parms_T3]	Defines the SS7 MTP2 T3 timer aligned. The valid range is 0 to 20000. The default value is 2000.
T4N [SS7Mtp2Parms_T4N]	Defines the SS7 MTP2 T4n Nominal proving period timer. The valid range is 0 to 15000. The default value is 8200.

Parameter	Description
T4E [SS7Mtp2Parms_T4E]	Defines the SS7 MTP2 T4e Emergency proving period timer (msec). The valid range is 0 to 5000. The default value is 500.
T5 [SS7Mtp2Parms_T5]	Defines the SS7 MTP2 Sending SIB timer. The valid range is 0 to 2400. The default value is 120.
T6 [SS7Mtp2Parms_T6]	Defines the SS7 MTP2 Remote Congestion timer (in msec). The valid range is 0 to 10000. The default value is 6000.
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.

11.6.2 Configuring SS7 Signaling Node Timers

The 'SS7 Signaling Node Timers' screen 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 160).

- **To configure the Signaling Node Timers parameters, take these 4 steps:**
 1. Open the 'SS7 Signaling Node Timers' screen (**Advanced Configuration** menu > **SS7 Configuration** > **SN Timers** option); the 'SS7 Signaling Node Timers' screen is displayed.

Figure 11-5: SS7 Signaling Node Timers

SS7 Signaling Node Timers	
SN Timers Number	0 <input type="button" value="v"/>
Name	SN_Timers
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	3000
T28 ANSI	3000
T29 ANSI	60000
T30 ANSI	30000

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

Table 11-3: 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'.
Common Timers	
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.
ITU Specific Timers	
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.
ANSI Specific Timers	
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.

11.6.3 Configuring Link-Set Timers

The procedure below describes how 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' screen (**Advanced Configuration** menu > **SS7 Configuration** > **Link Set Timers** option); the 'SS7 Link-set Timers' screen is displayed.

Figure 11-6: SS7 Link-set Timers Screen

SS7 Link-set Timers	
Link-set Timers Number	1 <input type="button" value="v"/>
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 or modify the parameters according to the table below.

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

Table 11-4: 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'.
Common Timers	
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.
ITU - Specific Timers	
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.
ANSI - Specific Timers	

Parameter	Description
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.


11.6.4 Configuring Links

The 'Links' screen 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 160).

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

1. Open the 'Links' screen (**Advanced Configuration** menu > **SS7 Configuration** > **Links** option); the 'Links' screen is displayed.

Figure 11-7: Links Screen



Link Number	Link Status
0	🟢
1	🟢
2	🟢
3	🟢
4	🟢
5	🟢
6	🟢
7	🟢
8	🟢
9	🟢
10	🟢
11	🟢
12	🟢
13	🟢
14	🟢
15	🟢
16	🟢
17	🟢
18	🟢
19	🟢
20	🟢
21	🟢
22	🟢
23	🟢
24	🟢
25	🟢
26	🟢
27	🟢
28	🟢
29	🟢
30	🟢
31	🟢

Administrative State	Operative State
In service	In service

Link Number	0
Name	link_0_SP_A
Trace	1
Variant	ITU-T
Local Busy	0
Inhibition	Uninhibited
Link Type	
Layer 2 Type	MTP2
Layer 3 Type	MTP3
MTP2 Fields	
Trunk Number	9
Timeslot Number	16
MTP2 Attributes Index	0
Congestion Low Watermark	5
Congestion High Watermark	20

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 11-5: 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"> ▪ [0] = NET_VARIANT_OTHER ▪ [1] ITU-T= NET_VARIANT_ITU (default) ▪ [2] ANSI = NET_VARIANT_ANSI ▪ [3] CHINA = NET_VARIANT_CHINA
Operative State [SS7_LINK_OPERATIONAL_STATE]	Determines the operational state of a signaling link. <ul style="list-style-type: none"> ▪ [0] Offline = L3_OFFLINE (default) ▪ [1] Busy = L3_BUSY, ▪ [2] In service = L3_INSERTSERVICE
Layer 2 Type [SS7_LINK_L2_TYPE]	Determines the link layer type - defines level 2 media of signaling link. <ul style="list-style-type: none"> ▪ [0] None = SS7_SUBLINK_L2_TYPE_NONE (default) ▪ [1] MTP2 = SS7_SUBLINK_L2_TYPE_MTP2 ▪ [2] M2UA MGC = SS7_SUBLINK_L2_TYPE_M2UA_MGC ▪ [3] SAAL = SS7_SUBLINK_L2_TYPE_SAAL
Layer 3 Type [SS7_LINK_L3_TYPE]	Determines the link high layer type - defines level 3 or L2 high layer of signaling link. <ul style="list-style-type: none"> ▪ [0] None = SS7_SUBLINK_L3_TYPE_NONE (default) ▪ [1] M2UA SG = SS7_SUBLINK_L3_TYPE_M2UA_SG ▪ [2] MTP3 = SS7_SUBLINK_L3_TYPE_MTP3 ▪ [3] MTP2 Tunneling = SS7_SUBLINK_L3_TYPE_MTP2_TUNNELING
MTP2 Layer 2	
Trunk Number [SS7_LINK_TRUNK_NUMBER]	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 [SS7_LINK_TIMESLOT_NUMBER]	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 [SS7_LINK_MTP2_ATTRIBUTES]	Determines the MTP2 attributes of signaling link (TDM). The valid range is 0 to MAX_C7_MTP2_PARAMS_INDEX. The default value is 3.

Parameter	Description
Congestion Low Watermark [SS7_CONGESTION_LOW_MARK]	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 [SS7_CONGESTION_HIGH_MARK]	Determines the link congestion high mark of signaling link (TDM). The valid range is 0 to 255. The default value is 20.
M2UA MGC Layer 2	
Group ID [SS7_LINK_GROUP_ID]	Determines the group ID (M3UA) of signaling link. The valid range is 0 to 0xFFFF. The default value is 0.
Interface ID [SS7_LINK_M2UA_IF_ID]	Determines the interface ID (M2UA) of signaling link. The valid range is 0 to 4294967295. The default value is 0.
Local Busy [SS7_LINK_MTC_BUSY]	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.

11.6.5 Configuring SS7 Signaling Nodes

The procedure below describes how to configure SS7 Signaling Nodes.

- **To configure the SS7 Signaling Nodes parameters, take these 4 steps:**
 1. Open the 'Links' screen (**Advanced Configuration** menu > **SS7 Configuration** > **SNs** option); the 'SS7 Signaling Nodes' screen is displayed.

Figure 11-8: SS7 Signaling Nodes Screen



2. Configure or modify 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 11-6: 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"> ▪ [1] ITU-T (default) ▪ [2] ANSI ▪ [3] CHINA
Trace [SS7_SN_TRACE_LEVEL]	Trace level of signaling node (level 3). <ul style="list-style-type: none"> ▪ [0] 0 (default) ▪ [1] 1
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"> ▪ [0] International (default) ▪ [1] International(Spare) ▪ [2] National ▪ [3] National(Spare)
STP Function [SS7_SN_SP_STP]	Routing function of signaling node. <ul style="list-style-type: none"> ▪ [0] SP (default) ▪ [1] STP
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.
Layer 4 Applications	
ISUP [SS7_SN_ISUP_APP]	Level 4 application that handles ISUP traffic for this signaling node. <ul style="list-style-type: none"> ▪ [0] None = NIL (default) ▪ [4] UAL = UAL
SCCP [SS7_SN_SCCP_APP]	Level 4 application that handles SCCP traffic for this signaling node. <ul style="list-style-type: none"> ▪ [0] None = NIL (default) ▪ [4] UAL = UAL
BISUP [SS7_SN_BISUP_APP]	Level 4 application that handles BISUP traffic for this signaling node. <ul style="list-style-type: none"> ▪ [0] None = NIL (default) ▪ [4] UAL = UAL

Parameter	Description
ALCAP [SS7_SN_ALCAP_APP]	Level 4 application that handles ALCAP traffic for this signaling node. <ul style="list-style-type: none"> ▪ [0] None = NIL (default) ▪ [4] UAL = UAL ▪ [5] ALCAP

11.6.6 Configuring Sigtran Group IDs

The 'SS7 Sigtran Group IDs' screen allows you to configure 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 160).

➤ **To configure the Sigtran Group IDs parameters, take these 4 steps:**

1. Open the 'SS7 Sigtran Group IDs' screen (**Advanced Configuration** menu > **SS7 Configuration** > **Sigtran Group IDs** option); the 'SS7 Sigtran Group IDs' screen is displayed.

Figure 11-9: SS7 Sigtran Group IDs Screen

SS7 Sigtran Group IDs	
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
Ta - Group Acknowledge Timer	2000
Th - Group Heartbeat Timer	30000

2. Configure or modify 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 11-7: Sigtran Group IDs Parameters

Parameter	Description
Group Number [SS7_SIG_IF_GR_INDEX]	Indicates the SS7 interface group index for a line. The valid range is 0 to 7.
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.
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"> ▪ [0] = no_layer (default) ▪ [1] IUA = iua ▪ [2] M2UA = m2ua ▪ [3] M3UA = m3ua ▪ [4] M2Tunnel = m2tunnel ▪ [5] DUA = V5ua
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.

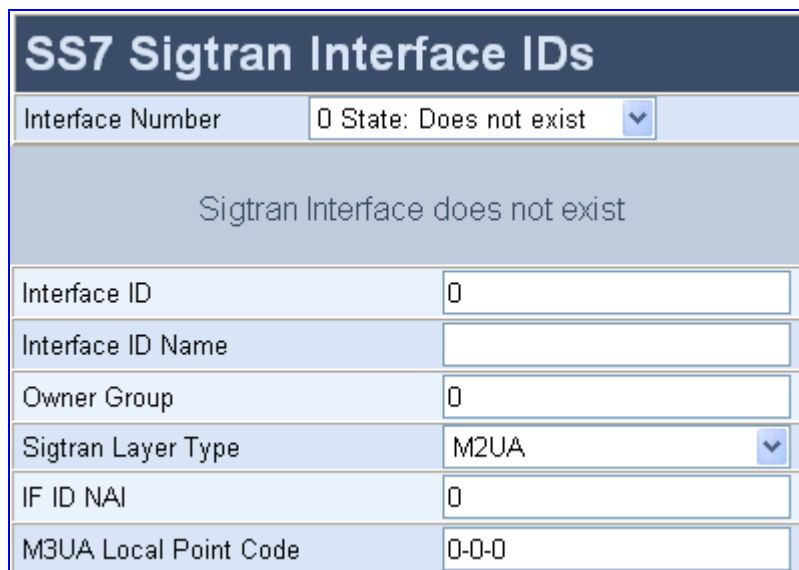
Parameter	Description
Interface Group Timers	
Tr - Group Recovery Timer [SS7_SIG_T_REC]	Determines the SIGTRAN group T recovery. The valid range is 0 to 10000000. The default value is 2000.
Ta - Group Acknowledge Timer [SS7_SIG_T_ACK]	Determines the SIGTRAN group T Ack (in msec). The valid range is 0 to 10000000. The default value is 2000.
Th - Group Heartbeat Timer [SS7_SIG_T_HB]	Determines the SIGTRAN group T Hb (in msec). The valid range is 0 to 10000000. The default value is 2000.

11.6.7 Configuring Sigtran Interface IDs

The 'SS7 Sigtran Interface IDs' screen 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 160).

- **To configure the Sigtran Interface IDs parameters, take these 4 steps:**
 1. Open the 'SS7 Sigtran Interface IDs' screen (**Advanced Configuration** menu > **SS7 Configuration** > **Sigtran Interface IDs** option); the 'SS7 Sigtran Interface IDs' screen is displayed.

Figure 11-10: SS7 Sigtran Interface IDs



SS7 Sigtran Interface IDs	
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
M3UA Local Point Code	0-0-0

2. Configure or modify 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 11-8: Sigtran Interface IDs Parameters

Parameter	Description
Interface Number [SS7_SIG_IF_ID_INDEX]	Determines the SS7 interface ID index, for a line. The valid range is 0 to 15. The default value is 1.
Interface ID [SS7_SIG_IF_ID_VALUE]	Determines the SIGTRAN interface ID value. The valid range is 0 to 4294967294. The default value is 0.
Interface ID Name [SS7_SIG_IF_ID_NAME]	Determines the SIGTRAN interface ID (text string). The default string is 'INT_ID'.
Owner Group [SS7_SIG_IF_ID_OWNER_GROUP]	Determines the SIGTRAN interface ID owner group. The valid range 0 to 65534. The default value is 0.
Sigtran Layer Type [SS7_SIG_IF_ID_LAYER]	Determines the SIGTRAN group layer (IUA/M2UA/M3UA). <ul style="list-style-type: none"> ▪ [0] None = no layer (default) ▪ [1] IUA ▪ [2] M2UA ▪ [3] M3UA ▪ [4] MTP2 Tunnel ▪ [5] v5ua
IF ID NAI [SS7_SIG_IF_ID_NAI]	Determines the SIGTRAN interface ID NAI. The valid range 0 to 65534. The default value is 65534.
M3UA Local Point Code [SS7_SIG_M3UA_SPC]	Determines the SIGTRAN M3UA SPC. The valid range 0 to 4294967294. The default value is 0.

Reader's Notes

12 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 189)
- **AudioCodes TrunkPack Downloadable Conversion Utility** (DConvert) (refer to 'TrunkPack Downloadable Conversion Utility' on page 204)
- **AudioCodes Call Progress Tones Wizard** (applicable only to Analog devices) (refer to Call Progress Tones Wiz'ard on page 216)

12.1 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. 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.

12.1.1 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.



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.

12.1.2 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.

12.1.3 Key Features

the BootP/TFTP program offers the following key features:

- Internal BootP supporting hundreds of entities
- Internal TFTP
- Contains all required data for our 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, Windows™ NT, Windows™ 2000 and Windows™ XP

12.1.4 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

12.1.5 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 click the file *Setup.exe*.
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 the user preferences. Refer to 'Setting the Preferences' on page 194 for information on setting the preferences.

12.1.6 Loading the cmp File, Booting the Device

Once the application is running and the preferences are defined (refer to 'Setting the Preferences' on page 194) for each device that is to be supported, enter parameters into the tool to set up 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 196.

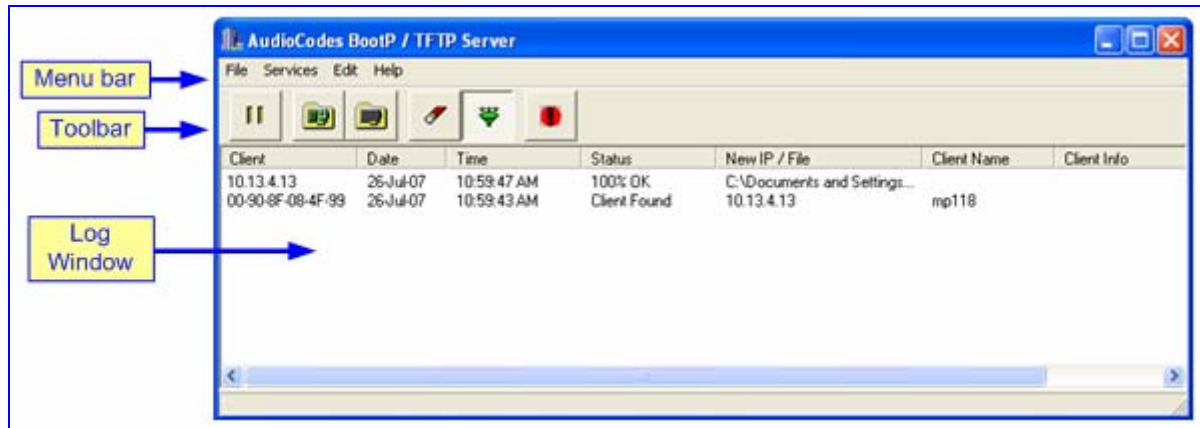
➤ **To load the software and configuration files, take these 4 steps:**

1. Create a folder on your computer that contains all 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 194).
3. Add a new client for the device that you want to initialize by the BootP (refer to 'Adding Clients' on page 199).
4. Reset the device, either physically or remotely. This causes the device to use BootP to access the network and configuration information.

12.1.7 BootP/TFTP Application User Interface

The figure below shows the main window of the BootP/TFTP utility.

Figure 12-1: Main Screen



12.1.8 Function Buttons on the Main Screen

The buttons on the toolbar are described in the table below:







Button	Name	Description
	Pause	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 Pause button provides a depressed graphic when the feature is active.
	Edit Clients	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 196.
	Edit Templates	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 202.
	Clear Log	Clears all entries from the Log window portion of the main window. Details on the Log window are provided in 'Log Window' on page 193.
	Filter Unknown Clients	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.
	Reset	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 12-2: 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 on the same subnet as the controlled device.

12.1.9 Log Window

The Log window (refer to 'BootP/TFTP Application User Interface' on page 192) 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 201).
 - *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.

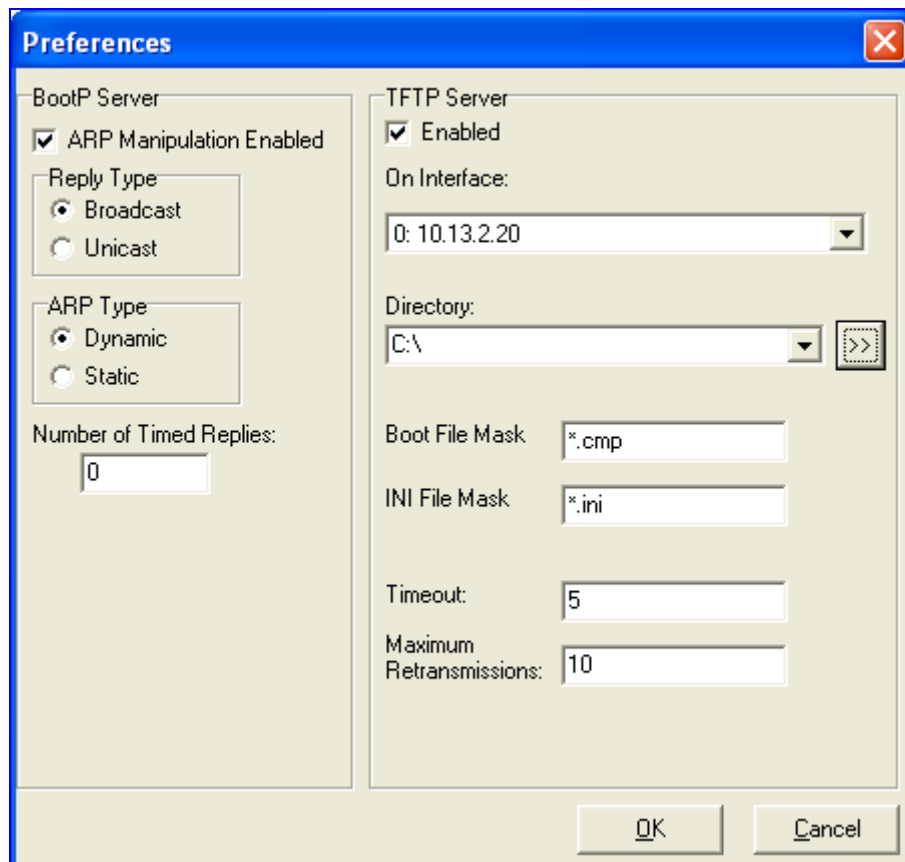
Right-click a line in the Log window to open a pop-up window with the following options:

- **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.

12.1.10 Setting the Preferences

The Preferences window (**Edit** menu > **Preferences**), as shown below is used to configure the BootP / TFTP parameters.

Figure 12-3: Preferences Screen



12.1.10.1 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, in order 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 device 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.

12.1.10.2 TFTP Preferences

The Preferences window (**Edit** menu > **Preferences**) allows you to define the following TFTP preferences:

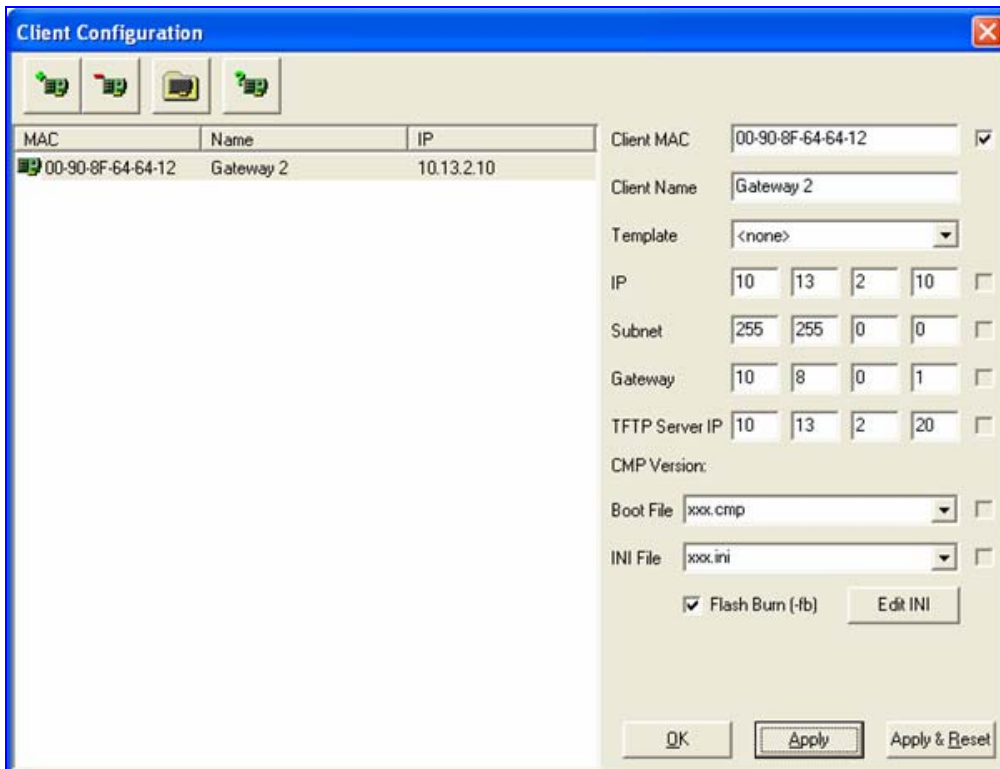
- **Enabled:** Select this check box to enable the TFTP functionality of the BootP/TFTP utility. If you want to use another TFTP application other than the one included with the BootP/TFTP utility, clear this check box.
- **On Interface:** From the drop-down list, select the network interface available on your PC that you want to use for the TFTP. (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. This can be left at the default value of 5 (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).

12.1.11 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 12-4: Client Configuration Screen



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)

12.1.11.1 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. 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 apply to the device. Use the normal dotted decimal format.
- **Subnet:** Enter the subnet mask you want to apply to the device. Use the normal dotted decimal format. 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:** Specifies 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 198.

**Notes:**

- Once the software file loads to the device, the device begins functioning 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 your 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 version of software 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. One, two, or no file names can appear in the **Boot File** field. 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:** Specifies 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.

12.1.11.2 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 the file name defined in the field as is (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 12-1: Command Line Switch Descriptions


Switch	Description	
-fb	Burns <i>ram.cmp</i> in flash (only for <i>cmp</i> files). Note: Instead of using this switch, you can simply select the Flash Burn check box.	
-em #	Use this switch to set the Ethernet mode: <ul style="list-style-type: none"> ▪ 0 = 10 Base-T half-duplex ▪ 1 = 10 Base-T full-duplex ▪ 2 = 100 Base-TX half-duplex ▪ 3 = 100 Base-TX full-duplex ▪ 4 = auto-negotiate (default) For detailed information on Ethernet interface configuration, refer to 'Ethernet Interface Configuration' in the device's <i>User's Manual</i> .	
-br	This parameter is used to perform the following: Set 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"> ▪ 1 = 1 BootP retry, 1 second ▪ 2 = 2 BootP retries, 3 seconds ▪ 3 = 3 BootP retries, 6 seconds ▪ 4 = 10 BootP retries, 30 seconds ▪ 5 = 20 BootP retries, 60 seconds ▪ 6 = 40 BootP retries, 120 seconds ▪ 7 = 100 BootP retries, 300 seconds ▪ 15 = BootP retries indefinitely 	Set 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"> ▪ 1 = 4 DHCP packets ▪ 2 = 5 DHCP packets ▪ 3 = 6 DHCP packets (default) ▪ 4 = 7 DHCP packets ▪ 5 = 8 DHCP packets ▪ 6 = 9 DHCP packets ▪ 7 = 10 DHCP packets ▪ 15 = 18 DHCP packets
	Note: This switch takes effect only from the next device reset.	

Switch	Description
-bd	<p>BootP delays. Sets the interval between the device's startup and the first BootP/DHCP request that is issued by the device. The switch only takes effect from the next reset of the device.</p> <ul style="list-style-type: none"> ▪ 1 = 1 second delay (default) ▪ 2 = 10 second delay ▪ 3 = 30 second delay ▪ 4 = 60 second delay ▪ 5 = 120 second delay
-bs	<ul style="list-style-type: none"> ▪ -bs 1: enables the Selective BootP mechanism ▪ -bs 0: disables the Selective BootP mechanism <p>The Selective BootP mechanism (available from Boot version 1.92) enables the device's integral BootP client to filter unsolicited BootP/DHCP replies (accepts only BootP replies that contain the text 'AUDC' in the vendor specific information field). This option is useful in environments where enterprise BootP/DHCP servers provide undesired responses to the device's BootP requests.</p>
-be	<p>Use -be 1 for the device to send device-related initial startup information (such as blade 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 192). For a full list of the vendor specific Information fields, refer to 'Vendor Specific Information Field' on page 19.</p> <p>Note: This option is not available on DHCP servers.</p>

12.1.11.3 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 196.
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 202.

➤ 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.

12.1.11.4 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.

12.1.11.5 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**.

12.1.11.6 Testing the Client

There must only be one BootP utility supporting any particular client MAC active on the network at any given time.

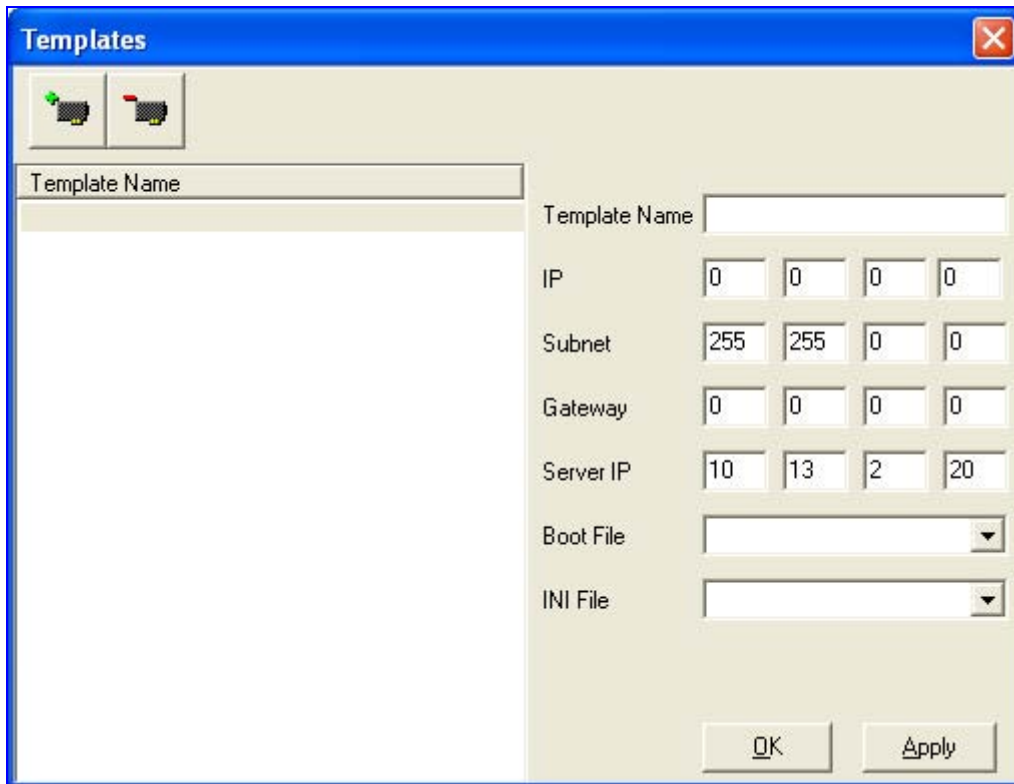
➤ **To test if other BootP utilities support this client, take these 4 steps:**

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.


12.1.12 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 the same.

Figure 12-5: Templates Screen




➤ **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.

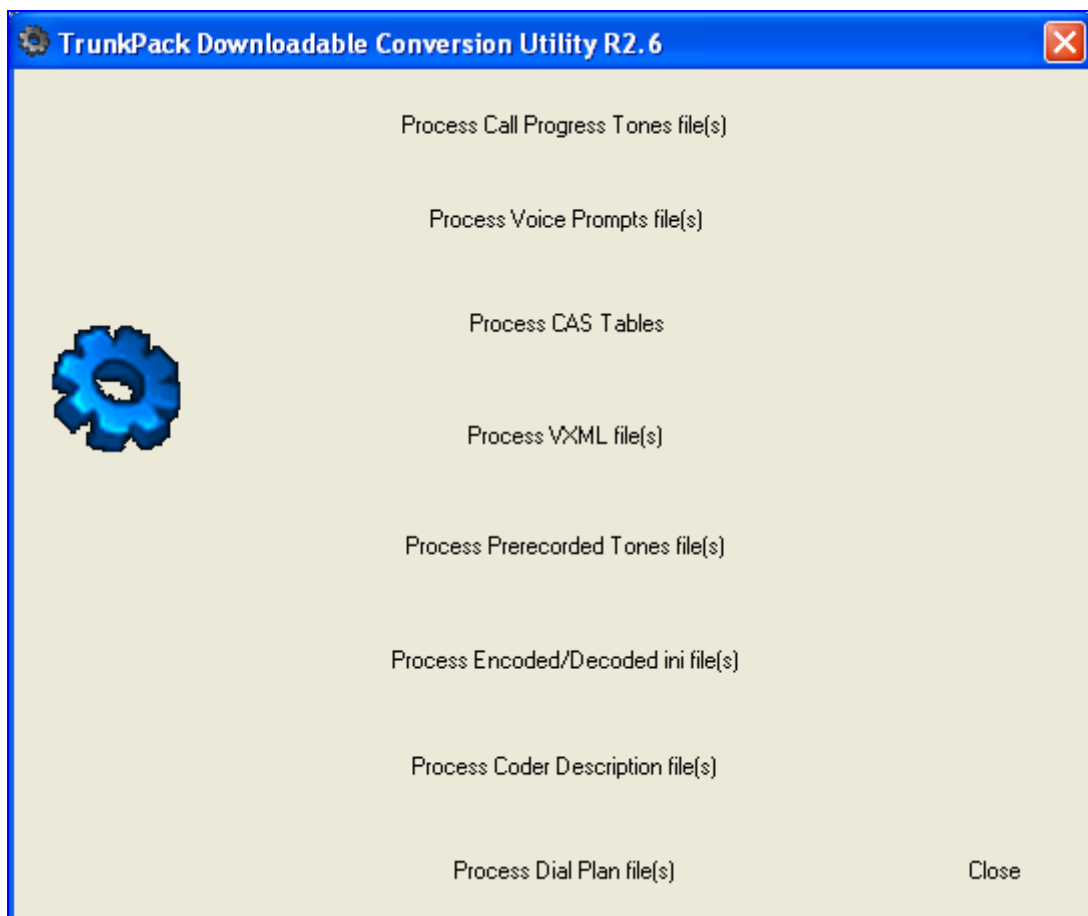
12.2 TrunkPack Downloadable Conversion Utility

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 205)
- 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 206)
- Create a loadable CAS protocol table file (Only applicable to Digital devices) (refer to 'Creating a loadable CAS Protocol Table' on page 208)
- Create Dial Plan file(s) (Only applicable to Digital devices)
- Encode / decode an *ini* file (refer to 'Encoding / Decoding an ini File' on page 212)
- Create a loadable Prerecorded Tones file (refer to 'Creating a Loadable Prerecorded Tones File' on page 214)

The TrunkPack Downloadable Conversion Utility is run by clicking the file *DConvert.exe*, which is supplied with your software package.

Figure 12-6: TrunkPack Downloadable Conversion Utility Main Screen



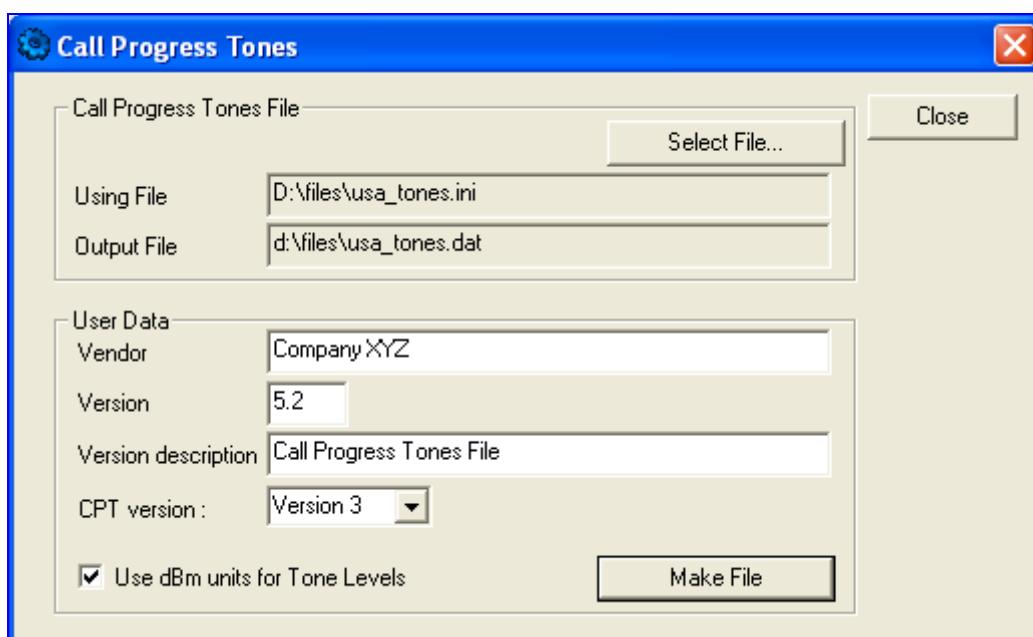
12.2.1 Converting a CPT ini File to a Binary dat File

For detailed information on creating a Call Progress Tones (CPT) *ini* file, refer to 'Configuring the Call Progress Tones and Distinctive Ringing File' on page 101.

➤ **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 204).
2. Click the **Process Call Progress Tones File(s)** button; the Call Progress Tones screen opens, shown in the figure below.

Figure 12-7: 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.

12.2.2 Creating a Loadable Voice Prompts File

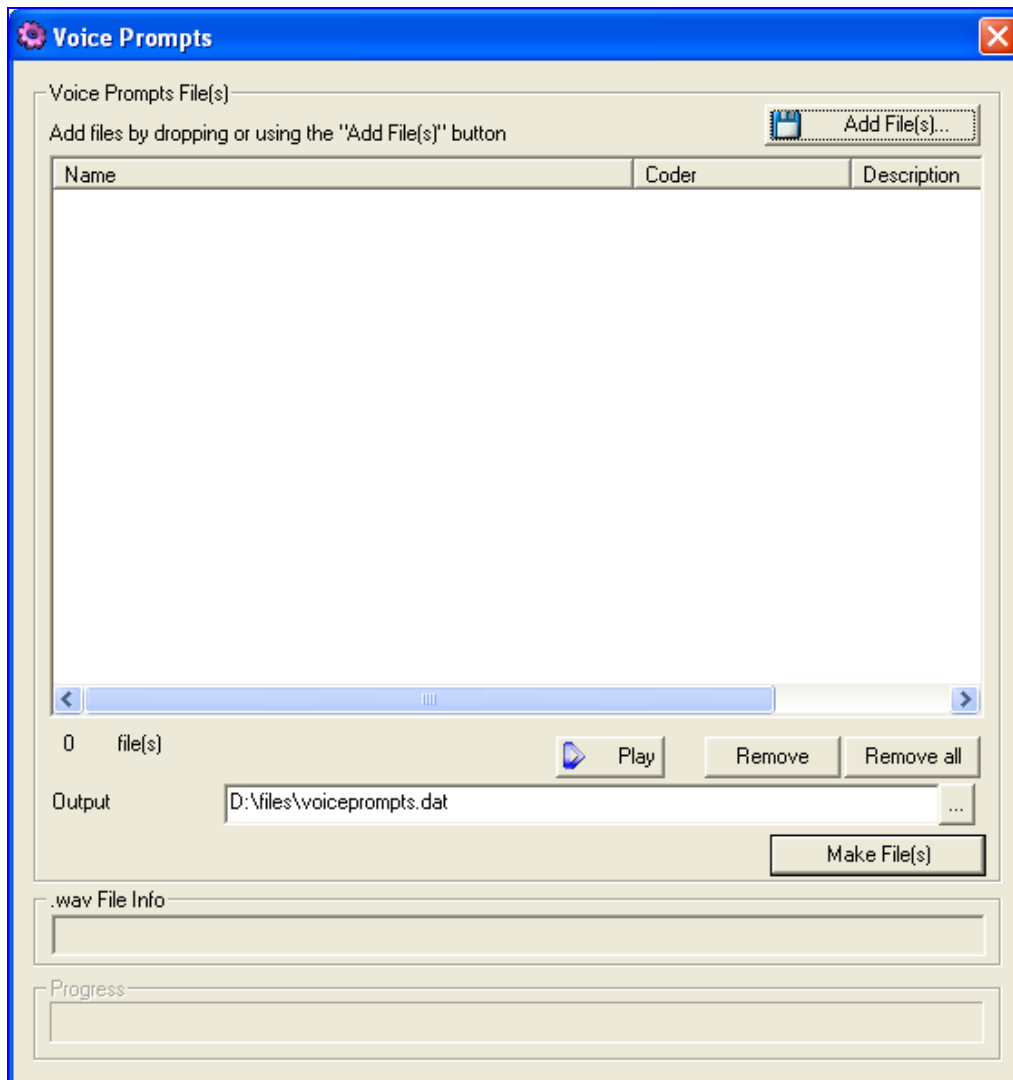


Note: This section is only applicable to AudioCodes 3000 Series and 2000 Series devices.

For detailed information on the Voice Prompts file, refer to 'Voice Prompts File' on page 107.

- **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 204).
 2. Click the **Process Voice Prompts File(s)** button; the Voice Prompts screen opens.

Figure 12-8: 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.

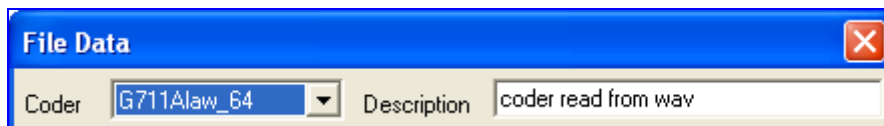
**Tips:**

- Use the **Play** button to listen to the *wav* files.
- Use the **Remove** and **Remove all** buttons to delete files from the list.

5. 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 12-9: File Data Window



6. 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*).
7. Click the **Make File(s)** button; the Voice Prompts loadable file is produced.

12.2.3 Creating a Loadable CAS Protocol Table File

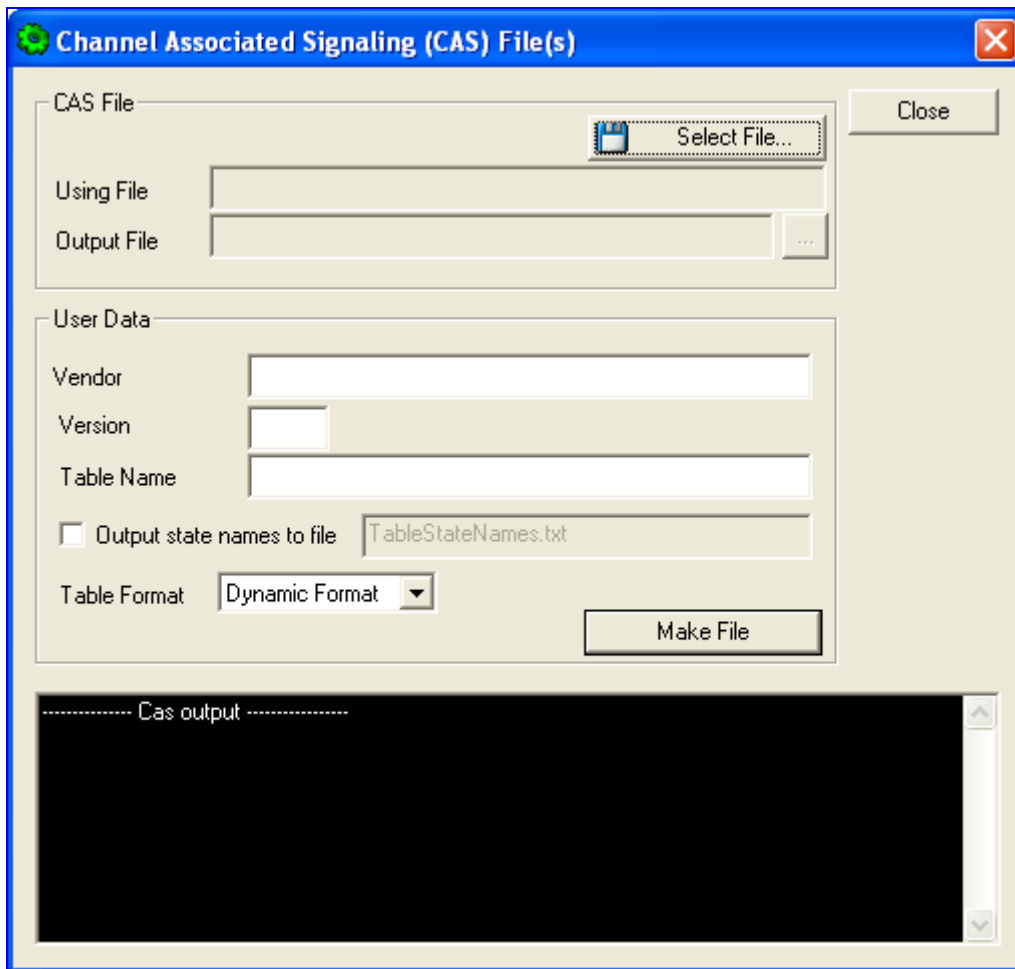


Note: This section is only applicable to AudioCodes Digital devices.

- **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 204).

4. Click **Process CAS Tables**; the Channel Associated Signaling (CAS) screen opens, shown in the figure below.

Figure 12-10: 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.

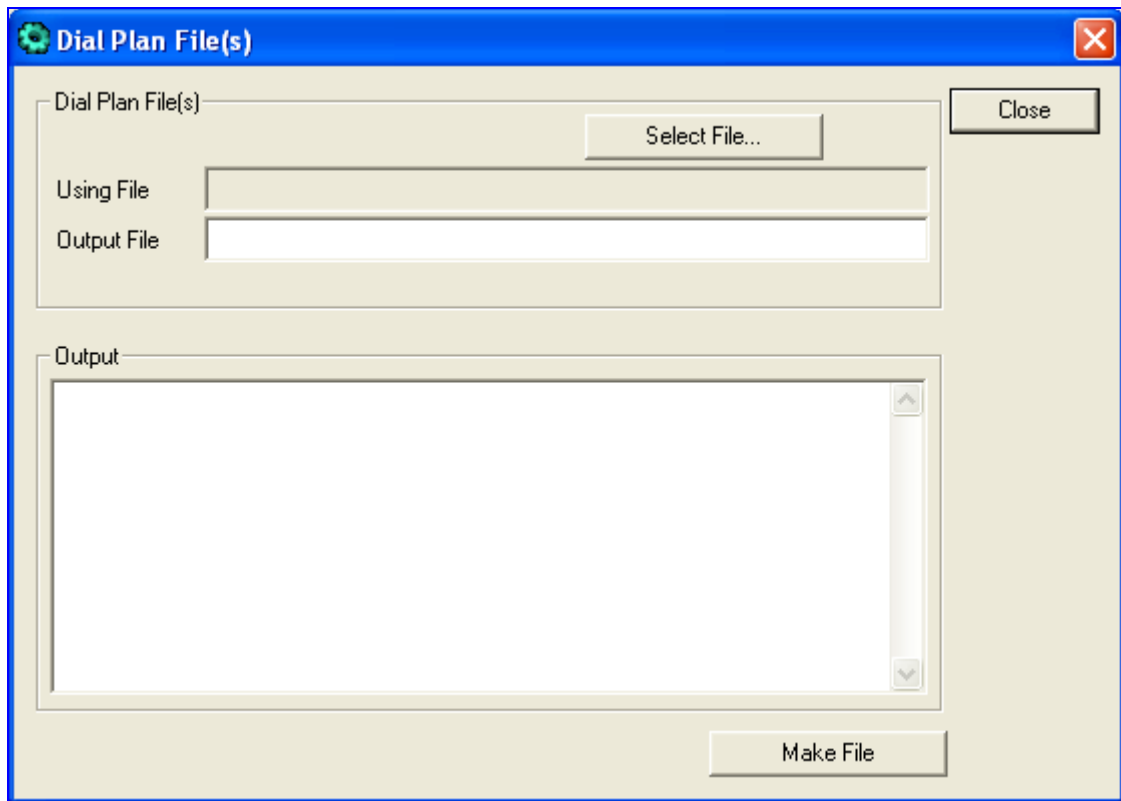
12.2.4 Creating a Dial Plan File



Note: This section 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' on page 109.
 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 12-11: 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.

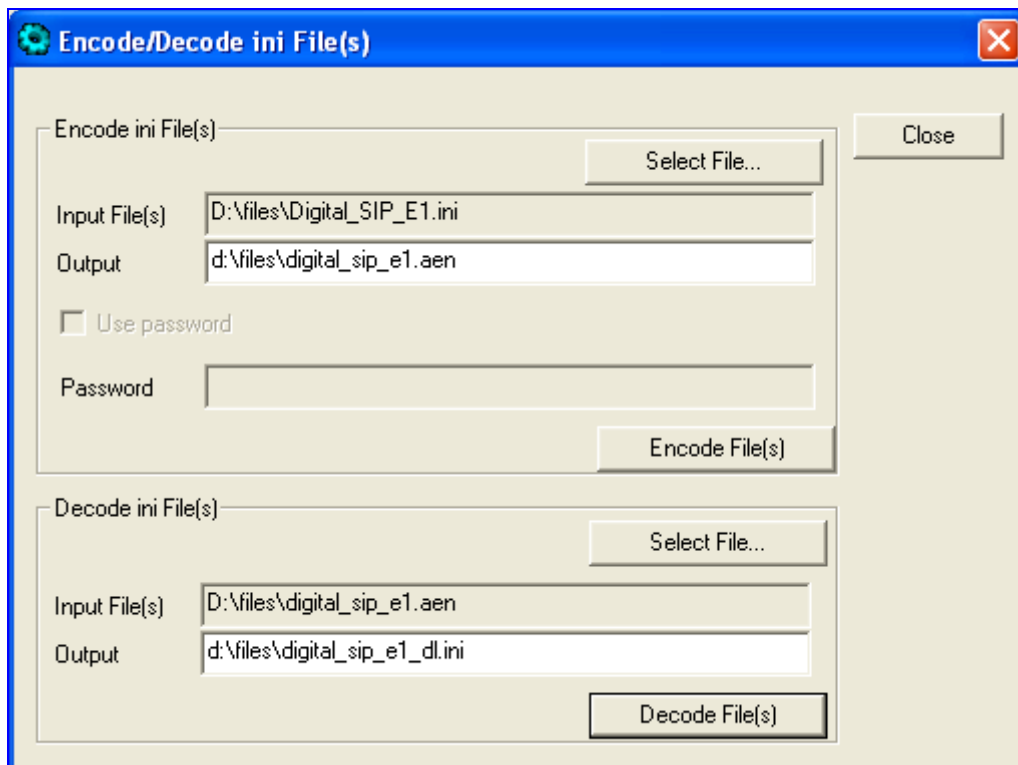
12.2.5 Encoding / Decoding an ini File

For detailed information on secured *ini* file, refer to Secured ini File.

➤ **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 204).
2. Click the **Process Encoded/Decoded ini file(s)** button; the 'Encode/Decode *ini* File(s)' screen, shown below, opens.

Figure 12-12: 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.



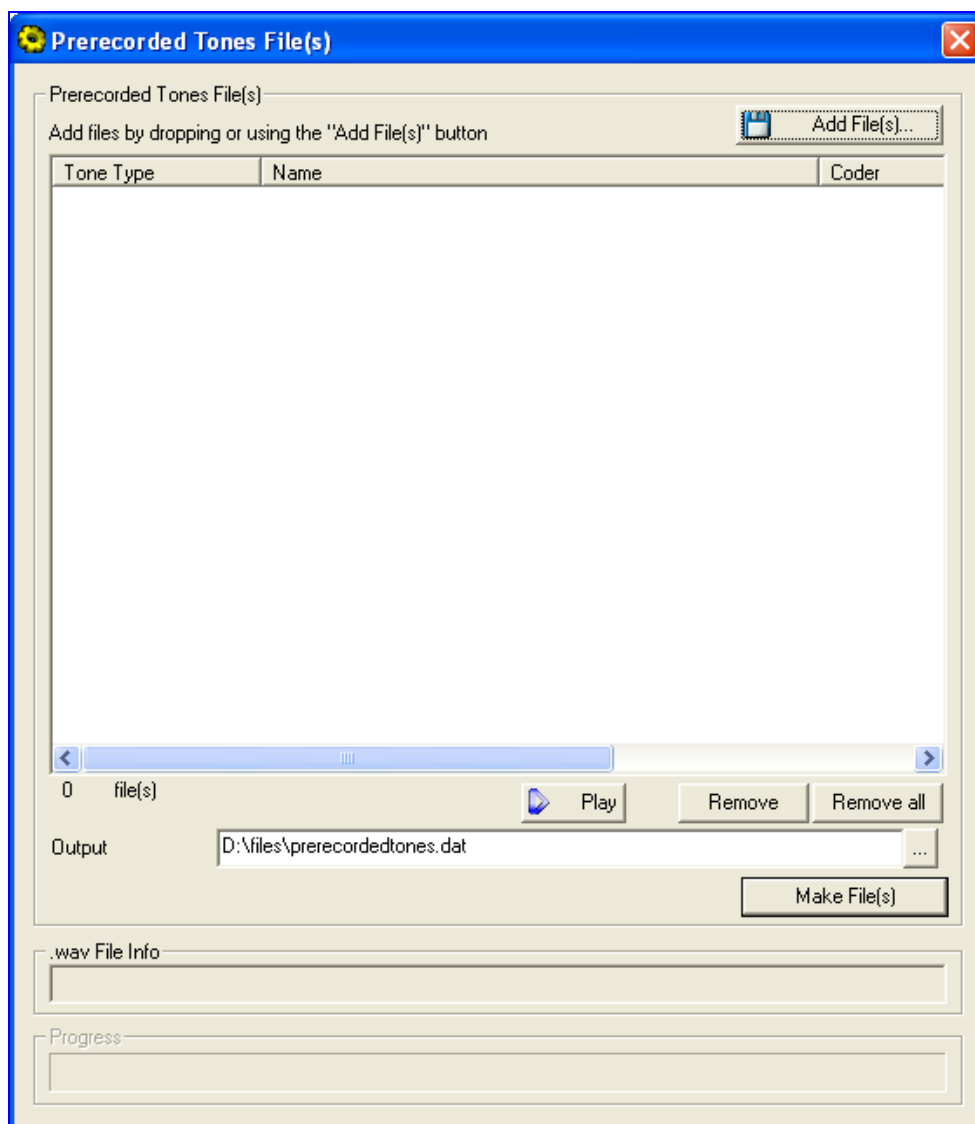
Note: The decoding process verifies the input file for validity. Any change made to the encoded file causes an error and the decoding process is aborted.

12.2.6 Creating a Loadable Prerecorded Tones File

For detailed information on the Prerecorded Tones (PRT) file, refer to 'Prerecorded Tones (PRT) File' on page 106.

- **To create a loadable PRT *dat* file from your raw data files, take these 7 steps:**
 1. Prepare the prerecorded tones (raw data PCM or L8) files 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 204).
 3. Click the **Process Prerecorded Tones File(s)** button; the Prerecorded Tones File(s) screen opens.

Figure 12-13: Prerecorded Tones Screen



4. To add the prerecorded tone 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 Prerecorded Tone files and 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 12-14: 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.

12.3 Call Progress Tones Wizard



Note: This section 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 generated by any PBX or telephone network. The CPTWizard creates a basic Call Progress Tones *ini* file and *dat* files, providing a good starting point when configuring an FXO device. The *ini* file contains definitions for all relevant Call Progress Tones; 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 205) 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: For CPTWizard to operate correctly, firmware Version 4.2 or later is required on the device.

12.3.1 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.

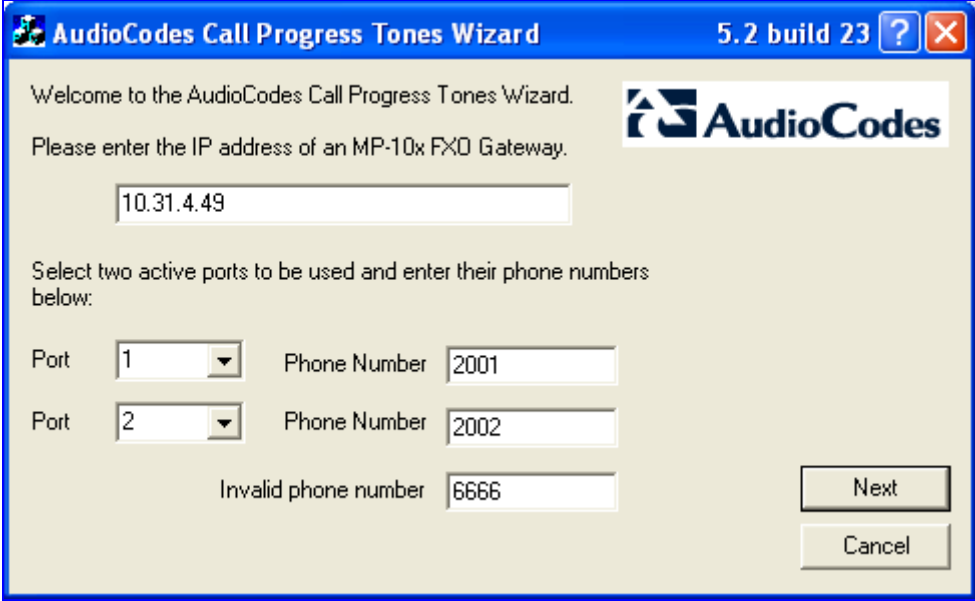
To install the CPTWizard, copy the files from the supplied installation kit to any folder on your PC. No further setup is required (approximately 5 MB of hard disk space is required).

12.3.2 Initial Settings

➤ **To start the CPTWizard, take these 5 steps:**

1. Run the *CPTWizard.exe* file; the wizard's initial settings screen is displayed.

Figure 12-15: Initial Settings Screen



Welcome to the AudioCodes Call Progress Tones Wizard. Please enter the IP address of an MP-10x FXO Gateway.

10.31.4.49

Select two active ports to be used and enter their phone numbers below:

Port	1	Phone Number	2001
Port	2	Phone Number	2002

Invalid phone number: 6666

Next

Cancel

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**.

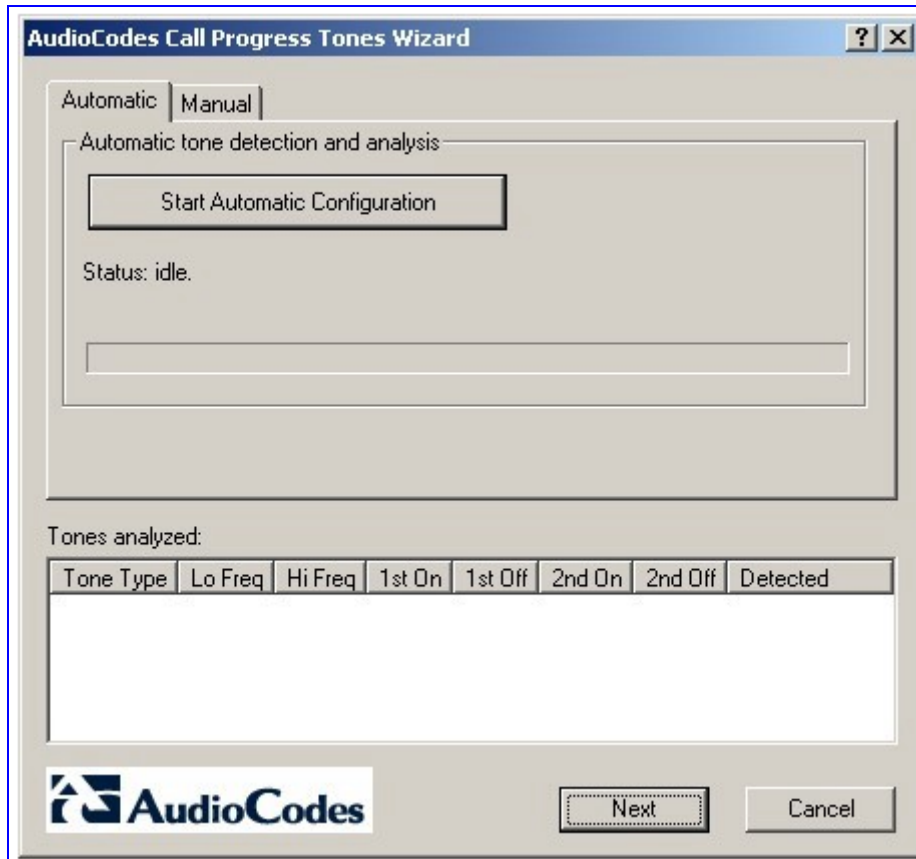


Note: The CPTWizard communicates with the FXO device via TPNC (TrunkPack Network Control Protocol). If this protocol has been disabled in the device configuration, the CPTWizard doesn't display the next screen and an error is reported.

12.3.3 Recording Screen - Automatic Mode

Once the connection between the CPTWizard and the FXO device is established, the recording screen is displayed:

Figure 12-16: Recording Screen - Automatic Mode

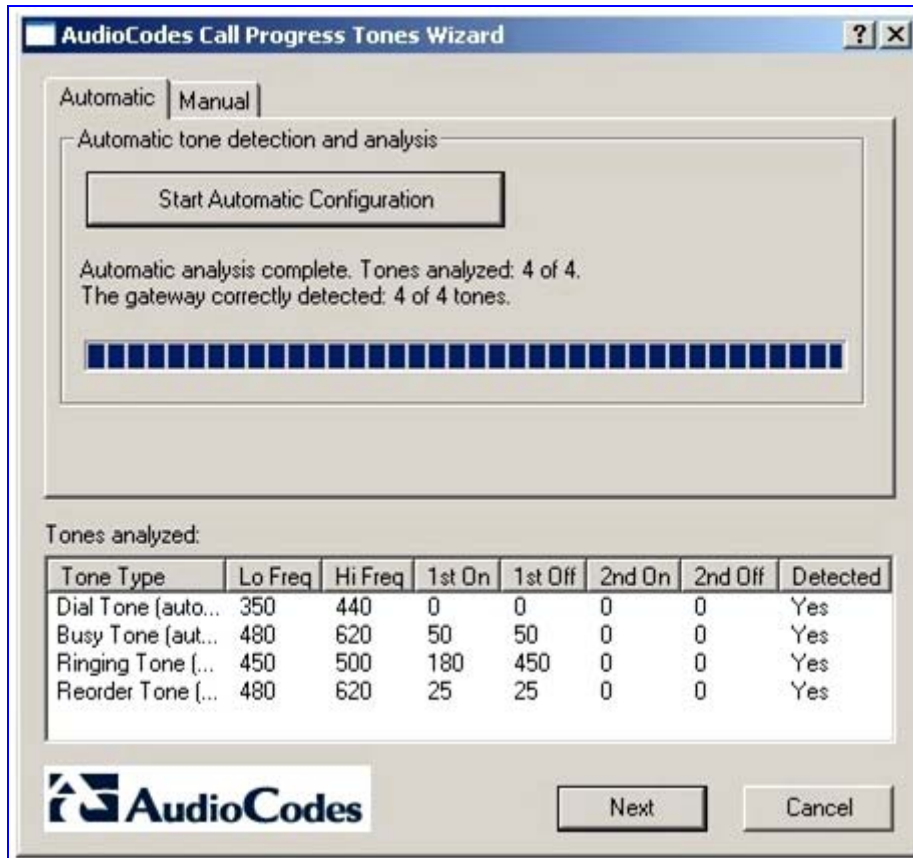


➤ **To start recording in automatic mode, take these 4 steps:**

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.

- 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 12-17: Recording Screen after Automatic Detection



- 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*
- 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.

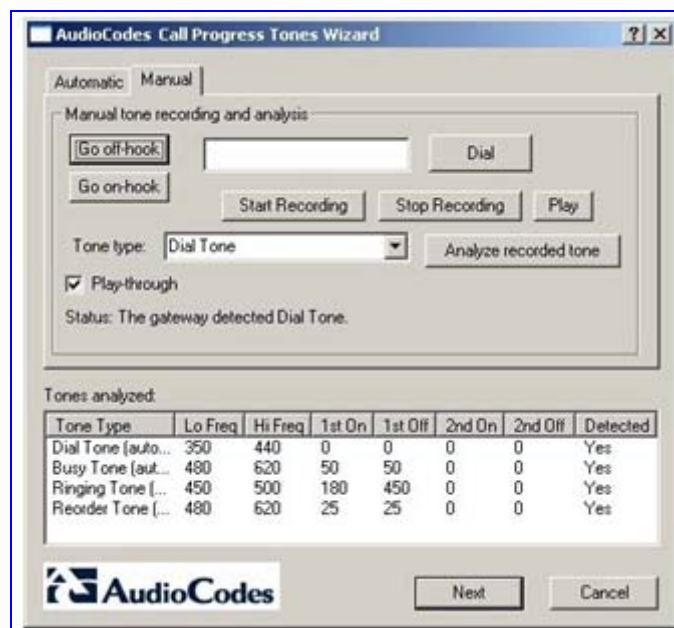
12.3.4 Recording Screen - Manual Mode

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 12-18: 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.

12.3.5 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 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.

12.3.6 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 204).
11. Load the new file to the gateway, and then reset the gateway.

Reader's Notes

13 Installing and Configuring Apache HTTP Server

This section describes the installation and configuration of Apache's HTTP server with Perl script environment (required for recording).

13.1 Windows 2000/XP Operation Systems

The procedure below describes how to configure the Apache HTTP server. Additional software is required: an uploading script (put.cgi), supplied with the software package.



Note: For detailed installation information, refer to ['http://perl.apache.org/docs/2.0/os/win32/config.html'](http://perl.apache.org/docs/2.0/os/win32/config.html).

➤ **To configure the Apache HTTP server and mod_perl version 2.0 software, take these 9 steps:**

1. Download the third party, Perl-5.8-win32-bin.exe installation file from the following link: www.apache.org/dist/perl/win32-bin/Perl-5.8-win32-bin.exe.

The installation file includes: Apache 2.0.46, Perl 5.8.0 and mod_perl-1.99 (the content of the file and the software version are subject to modification and changes).

For full installation instructions, refer to www.apache.org/dist/perl/win32-bin/Perl-5.8-win32-bin.readme.

2. To start the installation wizard, run the Perl-5.8-win32-bin.exe file.
3. During installation, you are prompted to define the destination folder under which the package is installed. It's recommended to provide a non-spaced path (such as: C:\directory_name_without_spaces).
4. In the following screen (Configuration), clear the 'Build html docs' and 'Configure CPAN pm' check boxes. If you are prompted to bring "nmake", respond by clicking **No**.
5. After the installation is complete, add the '/path/perl/bin' and '/path/apache2/bin' (path represents the path that was previously specified in the 'Destination Folder') directories to the system known path. Open the **Control Panel > System > Advanced > Environment Variables**, and then in the System Variables dialog box, choose 'Path' and click **Edit**; in the opened 'Variable Value' check box, append both of the paths to the existing list. Restart Windows to activate the new paths.

6. Open the Apache2/conf/httpd.conf file for editing and set the parameter MaxKeepAliveRequests to 0 (enables an unlimited number of requests during a persistent connection, which is required for multiple consecutive HTTP PUT requests for uploading the file).

```
Alias /perl/ "C:/Apache2/perl/  
<Location /perl>  
SetHandler perl-script  
PerlResponseHandler ModPerl::Registry  
Options +ExecCGI  
PerlOptions +ParseHeaders  
</Location>
```

7. Open the Apache2/conf/perl.conf file for editing and add the line 'Script PUT /perl/put.cgi' after the last line in the following section (note that if the following section is omitted or different in the file, insert it into the file or change it accordingly).
8. Locate the file 'put.cgi' in the supplied software package and copy it into the Apache2\perl\ directory. Change the first line in this file from 'c:/perl/bin/perl' to your perl executable file (this step is required only if mod_perl is excluded from your Apache installation).
9. In the apache2\bin directory, at a DOS prompt, type the following commands:
Apache.exe -n Apache2 -k **install**
Apache.exe -n Apache2 -k **start**

The installation and configuration are complete. You are now ready to start using the HTTP server.

13.2 Linux Operation Systems

The procedure below describes the Apache HTTP server configuration. Additional required software: an uploading script (put.cgi), supplied with the software package.



Note: It's assumed that the Linux installation already includes Apache server (for example, Apache 1.3.23), perl, and mod_perl (for example, mod_perl 1.26).

➤ **To configure Apache HTTP server, take these 4 steps:**

1. Inside the Apache directory, create the directory /perl (for example /var/www/perl). Locate the file put.cgi in the supplied software package and copy it to this directory.
2. In the put.cgi script, change the first line from c:/perl/bin/perl to your perl executable file (this step is required only if mod_perl is not included in your Apache installation).
3. Enable access to the following directories and files, by typing the following:

```
>chmod 777 perl
>chmod 755 put.cgi
>chmod 777 html (name of the server's shared files directory)
```
4. Configure the Apache server:
 - a. Open etc/httpd/conf/httpd.conf (or a similar file) for editing.
 - b. Set the KeepAlive parameter to true.
 - c. Set the MaxKeepAliveRequests parameter to 0 (enables an unlimited number of requests during a persistent connection – required for multiple consecutive HTTP POST requests for uploading the file).
 - d. Set MaxClients to 250.
 - e. Change the mod_perl module lines to:

```
<IfModule mod_perl.c>
  Alias /perl/ /var/www/perl/
  <Directory /var/www/perl>
    SetHandler perl-script
    PerlHandler Apache::Registry
    Options +ExecCGI
    PerlSendHeader On
  </Directory>
</IfModule>
Script PUT /perl/put.cgi
```

Reader's Notes

14 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 Alarm LEDs and Alarm Cut-Off (ACO))
- Self-Testing on hardware initialization (refer to Section 'Self-Testing' on page 229)
- FXS Line testing (refer to Section 'FXS Line Testing' on page 230) -- Only applicable 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 233) or in the 'Message Log' screen in the Embedded Web Server (refer to Activating the Internal Syslog Viewer). Note that the 'Message Log' screen is not recommended for prolong debugging.
 - **RS-232 terminal** (for establishing a serial communications link with the device, refer to 'Establishing a Serial Communications Link' in the User's Manual). Only applicable to Analog devices.
- Debug Recording using CLI (refer to 'Debug Recording (DR)' on page 37)

14.1 Self-Testing

The device features the following self-testing modes used 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.
 - **Rapid Test:** performed every time the device starts up. It is executed each time the device completes its initialization process. This is a short test phase in which the only error detected and reported is failure in initializing hardware components. If an error is detected, an error message is sent to the Syslog.
 - **Enhanced Test** (only applicable to 3000 Series and 2000 Series): performed every time the device starts up. The following hardware components are tested:
 - ◆ **For 3000 Series and 2000 Series:**
 - ✓ TSA (Time Slot Assigner)
 - ✓ PSTN framers (when used)
 - ✓ Missing DSP's
 - ✓ Conference channels (where supported)
 - ◆ **For 3000 Series:**
 - ✓ Lattice (TPM and TER)
 - ✓ GB Ethernet
 - ✓ Voice path (only for the active blade)

- **Detailed Test:** initiated by the user when the device is offline (isn't 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 is completed and if the parameter EnableDiagnostics is set to 1 or 2. In this mode, the device tests its DSPs, and RAM and flash memories. When EnableDiagnostics is set to 1, flash is tested thoroughly; when set to 2, flash is only partially tested. (For Analog devices, the **Ready** and **Fail** LEDs are lit while the Detailed test is running.)



Warning: To continue regular operation, the Detailed test must be disabled. Set the parameter EnableDiagnostics to 0, and then reset the device.



Note: When the Detailed test is enabled, errors sent to the Syslog server must be ignored.

- **Run-time (Periodic) Test** (only applicable to 3000 Series and 2000 Series): monitors the device during run-time. This test is performed every hour after startup, even when there is full traffic on the device (quality is not degraded).

The following hardware components are tested:

- **For 3000 Series and 2000 Series:**
 - ◆ TSA
 - ◆ PSTN framers (when used)
 - ◆ Missing DSP's
 - ◆ Conference channels (where supported)
- **For 3000 Series:**
 - ◆ Lattice (TPM & TER)
 - ◆ GB Ethernet
 - ◆ Voice path (only for the redundant blade)

If an error is detected, an error message is sent to the Syslog.

14.2 FXS Line Testing



Note: This section is applicable only to AudioCodes Analog devices.

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 - onhook; 1 - offhook)
- 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)



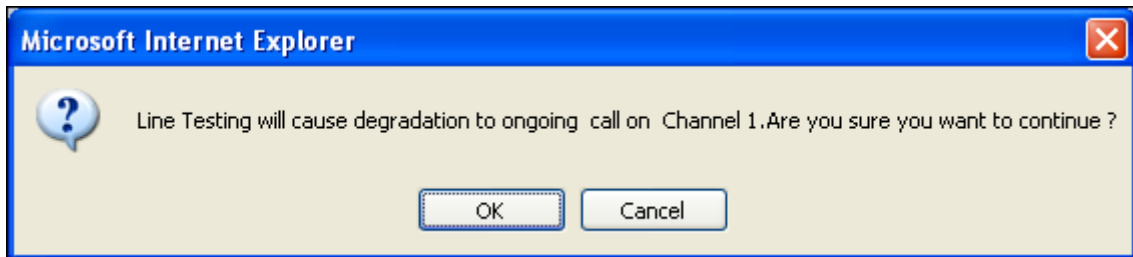
Notes:

- For Mediant 1000, Line testing is executed using SNMP only (acAnalogFxoLineTestTable SNMP table for FXO and acAnalogFxsLineTestTable SNMP table for FXS)
- The line testing mechanism must be used only for monitoring and never when there are calls in progress.

Analog line testing in the MediaPack Series gateways can be performed using the Embedded Web Server, as described in the procedure below:

- **To perform analog line testing using the Embedded Web Server, take these 3 steps:**
- 1. Click the **Analog Line Testing** submenu (**Status & Diagnostics** menu > **Analog Line Testing**); the Analog Line Testing confirmation box is displayed.

Figure 14-1: Analog Line Testing Confirmation Box



- 2. Click **OK** to confirm that you want to start the test; the 'FXS Line Testing For Channel 1' screen appears.

Figure 14-2: FXS Line Testing for Channel 1

FXS Line Testing For Channel 1	
Hook State :	On Hook
Ring State :	Ring On
Polarity Status :	Reverse On
Message Waiting Indication :	Message Waiting Indication On
Current Reading[10uA] :	85
Voltage Reading[10mV] :	-5406
Analog Voltage Reading[10mV] :	331
Ring Voltage Reading[10mV] :	-10856
Long Line Current Reading[10uA] :	195

- 3. To perform the test again, click the **ReTest** button.

14.3 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 described above 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 3000 Series and 2000 Series: When NTP is enabled, a timestamp string [hour:minutes:seconds] is added to all Syslog messages (for information on NTP, refer to 'Simple Network Time Protocol Support' in the *User's Manual*).
- For 3000 Series: All High Availability main operations and events are sent to the Syslog with the following prefix: 'M3K_HA'. All Syslog messages and events of the redundant IPM-6310 / TP-6310 blade are sent to the Syslog by the active IPM-6310 / TP-6310 blade with the appropriate message prefix.

14.3.1 Syslog Servers

You can use the supplied AudioCodes proprietary Syslog server ACSyslog, or any other third-party Syslog server. 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
- TriAction Software: <http://www.triaction.nl/Products/SyslogDaemon.asp>
- Netal SL4NT 2.1 Syslog Daemon: <http://www.netal.com>

14.3.2 Operation

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 Embedded Web Server ('Configuring the Management Settings' in the *User's Manual*) or *ini* file.

➤ **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 -- refer to 'Networking Parameters' in the *User's Manual*).
2. Define the IP address of the Syslog server (*ini* file parameter SyslogServerIP -- refer to 'Networking Parameters' in the *User's Manual*).
3. Define the port number of the Syslog server (*ini* file parameter SyslogServerPort -- refer to 'Networking Parameters' in the *User's Manual*).
4. Define the Syslog logging level (*ini* file parameter GWDebugLevel -- refer to 'System Parameters' in the *User's Manual*; Embedded Web Server: **Protocol Management** menu > **Advanced Parameters** > **General Parameters**).
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 (described in 'System Parameters' in the *User's Manual*) or the Embedded Web Server ('Configuring the Management Settings' in the *User's Manual*).

Reader's Notes

Reference Manual for SIP Gateways and Media Servers

Version 5.2

