

Mediant™ 2000 VoP Media Gateway

Mediant 2000 User's Manual For Nortel H.248 Applications

Version 4.2

Document #: LTRT00727 Rev 002



The following AudioCodes manuals should be used for additional information:

“Mediant 2000 H.248 Release Notes”, Document #: LTRT-00728, for Release 4.2

“VoPLib Reference Library User’s Manual”, Document #: LTRT-00800, for Release 4.2

Note: Electronic copies of this manual, whether on CD or on Web site, provide all cross-references with hyperlink. To reach the individual cross-referenced item directly, you may click on the hyperlinked item, i.e. a page number or section numbers.

Structure of this Manual

This **Mediant 2000** User's Manual has the following sections:

General, Notices, Warranty	3
Contents, Figures and Tables	5
1. Overview of the Mediant 2000	13
2. Hardware Installation	21
3. Software Installation	33
5. Device Management	55
6. Mediant 2000 Provisioning	81
7. Diagnostics	107
8. Functional Specifications	113
Appendix A. Mediant 2000 Configuration Guide for Nortel H.248 Application	119
Appendix B. Example of the ini File	133
Appendix C. AudioCodes BootP/TFTP Server Utility	137
Appendix D. RTP/RTCP Payload Types	145
Appendix E. DTMF, Fax and Modem Transport Modes	149
Appendix F. DHCP Server Configuration	153
Appendix G. BootP Server Configuration	161
Appendix H. TFTP Server Configuration	171
Appendix I. CAS Protocol Table	173
Appendix J. PSTN Trace Utilities	181
Appendix K. MEGACO Compliance	183
Appendix L. List of Abbreviations	197

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.

Trademarks

AC, AudioCoded, AudioCodes, AudioCodes logo, IPmedia, Mediant, MediaPack, NetCoder, Stretto, TrunkPack and VoicePacketizer, are trademarks or registered trademarks of AudioCodes Limited. All other products or trademarks are property of their respective owners.

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.

Notice

This User's Manual describes the installation and use of AudioCodes' **Mediant 2000** for Nortel H.248 Applications. 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 the accuracy of printed material after the Date Published nor can it accept responsibility for errors or omissions.

For **Technical Support** please contact:

E-mail: support@audiocodes.com

In the USA, fax 408-577-0492

In other countries, fax + 972-3-539 4040

© 2004 AudioCodes Ltd. All rights reserved.

This document is subject to change without notice.

Please refer to the release notes that may be included with your documentation or hardware delivery.

Date Published: February 5, 2004

Date Printed: February 25, 2004

Hardware & Software Designers are invited to contact
support@audiocodes.com
for possible updates before finalizing a design.

Table of Contents

1 Overview of the Mediant 2000	13
1.1 General Features	14
1.2 The Mediant 2000 Chassis	15
1.3 Mediant 2000 Diagrams	16
1.4 TP-1610 Software Overview	17
1.5 Mediant 2000 Applications	17
1.5.1 Available Configurations	18
1.6 Benefits	18
1.7 Functional Block Diagram	19
2 Hardware Installation	21
2.1 Unpacking	21
2.2 Installing	22
2.2.1 Installing the Mediant 2000 Chassis	22
2.2.2 Connecting the E1/T1 Trunk Interface	23
2.3 Chassis LED Indicators	26
2.4 The TP-1610 Board	26
2.4.1 TP-1610 Board Front Panel LED Indicators	30
2.4.2 Optional CPU Board	31
2.4.3 Board Replacement	31
3 Software Installation	33
3.1 Software Installation	33
3.1.1 Unzipping the Package	33
3.2 Initialization Process	33
3.2.1 Mediant 2000 with a CPU Board	33
3.2.2 BootP/DHCP-based Initialization Process	34
3.3 BootP & TFTP Server Installation and Configuration	35
3.3.1 Mediant 2000 without CPU Board	35
3.4 Software Directory Structure	36
4 Operation & Configuration	37
4.1 General	37
4.2 MEGACO (Media Gateway Control) Protocol	37
4.2.1 MEGACO Overview	37
4.2.2 Operation	37
4.2.2.1 Executing MEGACO Commands	37
4.2.2.2 Setting MEGACO Call Agent IP Address and Port	38
4.2.2.3 Handling Events	38
4.2.2.4 Playing Signals	38
4.2.2.5 Mediation	39
4.2.2.6 CAS/R2 support in MEGACO	40
4.2.3 SDP Support in MEGACO	41
4.2.4 Supported MEGACO Packages	44
4.2.4.1 Generic Media Package - G	44
4.2.4.2 Base Root Package - ROOT	44
4.2.4.3 Tone Generator Package - ToneGen	45
4.2.4.4 Tone Detection Package - ToneDet	45
4.2.4.5 DTMF Generator Package - DG (Extends ToneGen)	45
4.2.4.6 DTMF Detection Package - DD (Extends ToneDet)	46
4.2.4.7 Call Progress Tones Generator Package - CG (Extends ToneGen) ..	46
4.2.4.8 Call Progress Tones Detection Package - CD (Extends ToneDet)	47

4.2.4.9	Basic Continuity Package - CT	47
4.2.4.10	Network Package - NT	47
4.2.4.11	RTP Package - RTP (Extends - NT)	48
4.2.4.12	TDM Circuit Package – TDMC (Extends - NT)	48
4.2.4.13	Generic Announcement Package - AN	48
4.2.4.14	Expanded Call Progress Tones Generator Package - XCG (Extends - ToneGen)	49
4.2.4.15	Basic Service Tones Generation Package - SRVTN (Extends - ToneGen)	49
4.2.4.16	Expanded Services Tones Generation Package - XSRVTN (Extends - ToneGen)	49
4.2.4.17	Basic CAS Package - CAS	50
4.2.4.18	R2 CAS Package – R2.....	50
4.2.4.19	MF Generator Package - MFG (Extends - ToneGen).....	50
4.2.4.20	MF Detection Package - MFD (Extends - ToneDet)	51
4.2.5	MEGACO Terminations Naming	52
5	Device Management	55
5.1	SNMP Management	55
5.1.1	SNMP Overview.....	55
5.1.2	SNMP Message Standard.....	55
5.1.3	SNMP MIB Objects	56
5.2	Embedded Web Server.....	58
5.2.1	Password Control.....	59
5.2.1.1	The Embedded Web Server Username Password	59
5.2.1.2	Authentication Method	59
5.2.2	Web Configuration	59
5.2.2.1	Read-only Mode.....	59
5.2.2.2	Disable/Enable Embedded Web Server.....	59
5.3	Accessing the Embedded Web Server	60
5.4	Protocol Management	61
5.4.1	Protocol Definition	62
5.5	Advanced Configuration Menu.....	62
5.5.1	Network Settings	63
5.5.2	Channel Settings.....	64
5.5.3	Trunk Settings	65
5.5.4	TDM Bus Settings	67
5.5.5	Configuration File.....	68
5.5.6	Regional Settings	69
5.5.7	Change Password.....	70
5.6	Status and Diagnostic Menu.....	71
5.6.1	Trunk and Channel Status	72
5.6.2	Message Log.....	74
5.6.3	Versions	74
5.7	Software Update	75
5.7.1	TFTP Download	75
5.7.2	HTTP Download.....	76
5.7.3	License.....	77
5.8	Save Configuration.....	79
5.9	Reset Button	79
6	Mediant 2000 Provisioning	81
6.1	Provisioning of the Mediant 2000 for Call Control Protocol.....	81
6.2	Provisioning the Mediant 2000 for MGCP & MEGACO.....	87
6.3	ini File Structure	96
6.3.1	ini File Structure Rules	96
6.3.2	ini File Example	97
6.4	Using Call Progress Tones.....	99
6.4.1	Format of the Call Progress ini File.....	99
6.4.2	Default Template for Call Progress Tones	100
6.4.3	Call Progress Tone Generation and Download Procedure	104

7	Diagnostics	107
7.2.1	Reinitializing the TP-1610	107
7.3	Alarms.....	108
7.3.1	LED Visual Indicator Alarms	108
7.3.2	TP-1610 Self-Testing	109
7.4	Control Protocol Reports.....	110
7.4.1	Error Indications through the Network Interface.....	110
7.5	SysLog Support.....	110
7.5.1	SysLog Overview	110
7.5.2	SysLog Operation	110
7.5.2.1	Sending the SysLog Messages.....	110
7.5.2.2	Setting the SysLog Server IP Address.....	111
7.5.2.3	Controlling the Activation of the SysLog Client	111
7.5.2.4	ini File Example for SysLog.....	111
7.6	Solutions to Problems.....	111
7.6.1	Possible Common Problems.....	111
7.6.2	Possible Voice Problems	112
8	Functional Specifications	113
8.1	Mediant 2000 Selected Specifications	113
Appendix A Mediant 2000 Configuration Guide for Nortel H.248		
Application		
		119
A.1	Purpose.....	119
A.2	Introduction	119
A.3	Overview	119
A.4	Nortel Default Initialization File Parameters	120
A.5	Locality-Specific Parameters	123
A.6	Setting up Trunk Configurations	126
A.6.1	ISDN Relay.....	127
A.6.1.1	Example 1: Configuring ISDN Relay for a Single T1 (User-side).	127
A.6.1.2	Example 2: Configuring ISDN Relay for Multiple Trunks (2 T1s-User-side).....	128
A.6.1.3	Example 3: Configuring ISDN Relay for NFAS Trunks (4 T1s, Single D-channel, User-side).....	128
A.6.1.4	Example 4: Configuring ISDN relay for NFAS trunks with Backup D-channel (4-T1s, User-side).....	129
A.7	T.38 & Bypass Fax Support.....	130
Appendix B Example of the ini File		
		133
Appendix C AudioCodes BootP/TFTP Server Utility		
		137
C.1	Introduction	137
C.1.1	Key Features	137
C.1.2	Specifications	137
C.1.3	BootP/TFTP Server Utility Installation	138
C.1.4	Logging Screen	138
C.1.5	Preferences Screen.....	139
C.1.6	Client Configuration Screen.....	139
C.1.7	Template Screen	139
C.2	Screen Details	140
C.2.1	Main Screen	140
C.2.2	Preferences Screen.....	141
C.2.3	Client Configuration Screen.....	142
C.2.4	Templates Screen	143
Appendix D RTP/RTCP Payload Types		
		145
D.1	Packet Types Defined in RFC 1890	145
D.2	AudioCodes Defined Payload Types.....	146
D.3	Mediant 2000 Default RTP/RTCP/T.38 Port Allocation.....	147

Appendix E	DTMF, Fax and Modem Transport Modes	149
E.1	DTMF/MF Relay Settings	149
E.2	Fax/Modem Settings	149
E.3	Configuring Fax Relay Mode	149
E.4	Configuring Fax/Modem ByPass Mode	150
E.5	Supporting V.34 Faxes	150
E.5.1	Using Bypass Mechanism for V.34 Fax Transmission:	150
E.5.2	Using Events Only Mechanism for V.34 Fax Transmission	151
E.5.3	Using Relay mode for all kind of fax machines (T.30 and V.34)	151
Appendix F	DHCP Server Configuration	153
Appendix G	BootP Server Configuration	161
G.1	BootP Server Configuration	161
G.2	Setting Call Agent Address Option	168
G.3	Changing MAC Hardware Address	168
Appendix H	TFTP Server Configuration	171
Appendix I	CAS Protocol Table	173
I.1	Constructing a CAS Protocol Table	173
I.2	Table Elements	173
I.3	INIT variables	173
I.4	Actions	173
I.5	Functions	174
I.6	States	174
I.7	Reserved Words	175
I.8	State's Line Structure	175
I.9	Action/Event	175
I.10	Function	178
I.11	Parameters	178
I.12	Next State	180
Appendix J	PSTN Trace Utilities	181
J.1	About PSTN Trace Utilities	181
J.2	Enabling PSTN Trace	181
Appendix K	MEGACO Compliance	183
K.1	MEGACO Compliance Matrix	183
Appendix L	List of Abbreviations	197

List of Figures

Figure 1-1: Front View of Mediant 2000 Populated with AudioCodes' TP-1610 Board	16
Figure 1-2: AudioCodes' TP-1610 Board	16
Figure 1-3: View of AudioCodes' TP-1610 RTM Board	17
Figure 1-4: Functional Block Diagram	19
Figure 1-5: Typical Mediant 2000 VoIP Application	20
Figure 2-1: Mediant 2000 Front Details	22
Figure 2-2: Mediant 2000 Rear Connection Details (8 Spans)	23
Figure 2-3: Mediant 2000 Rear Connection Details (16 Spans)	23
Figure 2-4: 50-pin Female Telco Board-mounted Connector	24
Figure 2-5: RJ-48c Trunk Connectors	25
Figure 2-6: RJ-45 LAN/Ethernet Network Connectors	25
Figure 2-7: DC Power Connector	26
Figure 2-8: TP-1610 Board, Front Panel View	27
Figure 2-9: RTM Panel with 2 Telco Connectors	27
Figure 2-10: RTM Panel with 8 RJ-48c Trunk Connectors	29
Figure 4-1: MEGACO-R2 Call Flow Diagram	41
Figure 5-1: Enter Network Password screen	60
Figure 5-2: Quick Setup Screen	61
Figure 5-3: Protocol Definition Screen	62
Figure 5-4: Advanced Configuration Parameters	63
Figure 5-5: Network Settings Screen	64
Figure 5-6: Channel Settings Screen	65
Figure 5-7: Trunk Settings Screen	66
Figure 5-8: TDM Bus Settings Screen	68
Figure 5-9: Configuration File Screen	69
Figure 5-10: Regional Settings Screen	70
Figure 5-11: Change Password Screen	71
Figure 5-12: Status and Diagnostic Menu Screen	72
Figure 5-13: Trunk and Channel Status Screen	73
Figure 5-14: Message Log Screen	74
Figure 5-15: Versions Screen	75
Figure 5-16: TFTP Download Screen	76
Figure 5-17: HTTP Download Screen	77
Figure 5-18: License Screen	78
Figure 5-19: Save Configuration Screen	79
Figure 5-20: Reset Screen	80
Figure 6-1: ini File Structure	96
Figure 6-2: Example of an ini File	97
Figure 6-3: ini File Example for MEGACO	98
Figure 6-4: Default Call Progress Tone Template (continued to page 104)	101
Figure 6-5: Download Selection Screen	104
Figure 6-6: File Selection Screen	105
Figure 7-1: The ini File Example for SysLog	111
Figure A-1: Protocol Definition	121
Figure A-2: Network Settings	122
Figure A-3: Channel Settings	123
Figure A-4: Protocol Definition	124
Figure A-5: Network Settings	125
Figure A-6: Configuration File	126
Figure A-7: Trunk Settings	128
Figure A-8: Transparent Trunk	129

Figure A-9: T1_IUA trunk for a Backup D-channel	130
Figure C-1: Main Screen	140
Figure C-2: Preferences Screen	141
Figure C-3: Client Configuration Screen	142
Figure C-4: AudioCodes BootP/TFTP Templates Screen	143
Figure F-1: Scope Menu	153
Figure F-2: Reserved Clients Menu	154
Figure F-3: Scope Menu, Active Leases Option	155
Figure F-4: Active Leases Selection Box	155
Figure F-5: Client Properties Screen	156
Figure F-6: DHCP Options - Reservation Screen	156
Figure F-7: DHCP Options - Active Options Displayed	157
Figure F-8: IP Address Array Editor Screen	157
Figure F-9: Host Name Screen	158
Figure F-10: Boot Server Host Name Screen	158
Figure F-11: Bootfile Name Screen	159
Figure G-1: Properties Screen	161
Figure G-2: Clients Screen	162
Figure G-3: Options Screen	162
Figure G-4: New Hardware Address	163
Figure G-5: Configured Options Screen	164
Figure G-6: Boot File Screen	165
Figure G-7: Default Gateway Screen	165
Figure G-8: Mediant 2000 IP Address Screen	166
Figure G-9: Call Agent IP Screen	167
Figure G-10: Status Screen	167
Figure G-11: Call Agent (IP Address Array) Screen	168
Figure G-12: MAC (Multiple Bytes) Screen	169
Figure G-13: MAC (Options) Screen	169
Figure G-14: MAC Address Screen	170
Figure G-15: Status Validation Screen	170
Figure H-1: TFTP Initial Screen	171
Figure H-2: Service Properties Screen	171
Figure H-3: TFTP General Screen	172
Figure H-4: TFTP Final Status Screen	172
Figure J-1: Trunk Traces	182
Figure J-2: UDP2File Utility	182

List of Tables

Table 2-1:Connections on each 50-pin Telco Connector	24
Table 2-2:Chassis Indicators	26
Table 2-3:Board Status LED Indicators.....	30
Table 2-4:Trunks Status LED Indicators	30
Table 2-5:Ethernet LED Indicators	30
Table 2-6:Auxiliary LED Indicators.....	30
Table 4-1:MEGACO Mapping Payload Numbers to Coders (continued to page 43).....	42
Table 4-2:Generic Media Package - G	44
Table 4-3:Base Root Package - ROOT	44
Table 4-4:Tone Generator Package - ToneGen	45
Table 4-5:Tone Detection Package - ToneDet.....	45
Table 4-6:DTMF Generator Package - DG (continued on page 45).....	45
Table 4-7:DTMF Detection Package - DD (continued on page 46).....	46
Table 4-8:Call Progress Tones Generator Package - CG	46
Table 4-9:Call Progress Tones Detection Package - CD	47
Table 4-10:Basic Continuity Package - CT	47
Table 4-11:Network Package - NT.....	47
Table 4-12:RTP Package - RTP	48
Table 4-13:TDM Circuit Package - TDMC	48
Table 4-14:Generic Announcement Package	48
Table 4-15:Expanded Call Progress Tones Generator Package - XCG.....	49
Table 4-16:Basic Service Tones Generation Package - SRVTN	49
Table 4-17:Expanded Services Tones Generation Package - XSRVTN	49
Table 4-18:Basic CAS Signal/Events.....	50
Table 4-19:R2 CAS Signal/Events Table	50
Table 4-20:MF Generator Package - MFG (continues on page 51).....	50
Table 4-21:MF Generator Package - MFG	51
Table 4-22:MEGACO Endpoint Names (continues on page 53 to page 54).....	52
Table 5-1:Trunk Status Color Indicator Key	66
Table 5-2:Trunk and Channel Status Color Indicator Key.....	73
Table 6-1:Mediant 2000 Basic Configuration Parameters (continued to page 87).....	82
Table 6-2:MGCP Specific Parameters (continued on page 88 to page 89).....	87
Table 6-3:MEGACO Specific Parameters (continued on page 90)	89
Table 6-4:MGCP/MEGACO Common Parameters (continued to page 92).....	91
Table 6-5:MGCP/MEGACO Default Channel Parameters (continued on page 94 to page 96)	92
Table 7-1:Chassis Indicators	108
Table 7-2:Board Status LED Indicators.....	108
Table 7-3:Trunks Status LED Indicators	108
Table 7-4:Ethernet LED Indicators	109
Table 7-5:Auxiliary LED Indicators.....	109
Table 7-6:Solutions to Common Problems (continued on page 112)	111
Table 7-7:Solutions to Possible Voice Problems.....	112
Table 8-1:Selected Specifications (continues until page 117).....	113
Table D-1:Packet Types Defined in RFC 1890	145
Table D-2:AudioCodes Defined Payload Types	146
Table D-3:Default RTP/RTCP/T.38 Port Allocation.....	147
Table E-1:V.34Fax to V.34Fax – Bypass Mode	151
Table E-2:V.34Fax to V.34Fax – Events Only Mode	151
Table E-3:V.34Fax to V.34Fax – Relay Mode	152
Table I-1:ST_DIAL: Table Elements.....	174

Table I-2:CAS Parameters	178
Table K-1:MEGACO Compliance Table (continues until page 195)	183
Table L-1:List of Abbreviations (continues on page 197 to page 199)	197

1 Overview of the Mediant 2000

The **Mediant 2000** is an ideal cost-effective, entry-level member in the AudioCodes family of market-ready, standards-compliant, media server systems. Intelligently packaged in a 1U chassis especially designed for small-scale deployments and smaller locations in the packet network, the Mediant 2000 is the right-sized solution for small-scale needs. Incorporating AudioCodes' leading Voice over Packet technology and based on field proven media technology, the Mediant 2000 enables Network Equipment Providers (NEPs) rapid time-to-market and reliable cost-effective deployments of enhanced voice services in VoIP telephony networks. This compact device, designed to be installed either as a desktop unit or installed in a 19-inch rack.

The Mediant 2000 contains AudioCodes' TP-1610 cPCI VoIP communication board is an ideal building block for deploying high-density, high availability Voice over IP (VoIP) and wireless enterprise systems.

The TP-1610 is suitable for VoIP gateways, IP-enabled call centers, large Telcos and next generation DLCs. Offering integrated voice gateway functionality capable of delivering up to 480 simultaneous calls, the TP-1610 supports all necessary functions for voice and fax streaming over IP networks.

The TP-1610 supports a broad selection of voice processing related algorithms, including G.711, G.723.1 and G.729A Vocoders, G.168-2000 compliant echo cancellation, T.38 Real-time Fax over IP, a wide selection of In-band and Out-of-band tone detection and generation, as well as signaling protocol support including ISDN PRI, SIGTRAN (M2UA, M3UA, IUA) and CAS.

The Mediant 2000 incorporates 1, 2, 4, 8 or 16 E1 or T1 spans for connection, either directly to PSTN telephony trunks or to an enterprise PBX, and two 10/100 Base-T Ethernet ports for connection to the LAN. Two packet processors handle packet-streaming functions through two redundant integral 10/100 Base-T interfaces. Each processor implements the industry-standard RTP/RTCP packet-streaming protocol, advanced adaptive jitter buffer management and T.38 fax relay over IP. The basic configuration includes an H.110 interface. An E1/T1 trunk interface module is provided for 16-T1 or 16-E1 trunks allowing for full gateway streaming functions in a single cPCI slot.

The TP-1610 should be considered as two logical Media Gateway entities each with its own PSTN interface. Physically, the TP-1610 takes up one slot of a cPCI chassis, but should be considered as two logical sides, each with its own MAC address, its own IP address, its own board handle, etc.

AudioCodes' TP-1610 board complies with many industry-standard network control protocols including MEGACO (H.248), TPNC (AudioCodes' proprietary TrunkPack Network control protocol) and optionally SIP (RFC 3261) or H.323 (ITU ver. 4). These allow for the implementation of a distributed media gateway or server architecture that separates call processing functions from media processing functions, resulting in better redundancy, scalability and higher system availability.

The TP-1610 is a member of AudioCodes' 4th generation of the widely deployed TrunkPack Media Gateway-on-a-blade family. Like the other members of the TrunkPack family, the TP-1610 supports AudioCodes' API, which enables software download, provisioning and control. Maintaining essential API backward compatibility of future releases to protect the customers' investment is an important feature of AudioCodes' software update/upgrade program.

Enabling accelerated design cycles with higher density and reduced costs, the TP-1610 is an ideal building block for scalable, reliable VoIP solutions. With the Mediant 2000's comprehensive feature set, customers can quickly design a wide range of solutions for PSTN and VoIP networks.

AudioCodes is a world leading provider of new voice infrastructure network technologies, and AudioCodes' commitment to innovation yields consistently high-quality voice processing products that meet our customers' demand for higher levels of integration.

1.1 General Features

The AudioCodes Mediant 2000 has the following features:

- **Vocoder configuration options:**
 - PCM/ADPCM, G.723, G.729A, GSM FR and NetCoder

Note 1: G.729 and G.723 should not be used simultaneously on the same board.

Note 2: G.728 coder can be supported; for additional information, contact your AudioCodes representative.

- 480 voice/fax/data with 16 E1 or 16 T1 integral interfaces
- Up to 16 E1/T1 digital spans
- Independent vocoder selection per channel
- Extensive media processing functions
- RTP stream multiple destination connection (i.e., to TDM, other RTP channels and PCI channels (for recording))
- Packet telephony standard compliant
- PSTN protocol termination support
- Expansion slot for application hosting
- Open architecture
- Flexible deployment and multiple density options
- NEBS Level 3 compliant
- Superior, high quality Voice and Fax over IP calls
- Interchangeable IP/RTP or PSTN or H.110 Endpoints
- VoIP packet streaming (RTP/ RTCP) per RFC 1889/1890
- TPNCN (TrunkPack NCP) AudioCodes' proprietary call control MEGACO (H.248); and standard control protocols
- Real-time Fax over IP/T.38 with superior performance (round trip delay of up to 9 sec)
- Integral Announcement support towards PSTN/TDM and IP
- IP to IP Mediation capabilities
- IP to IP Transcoding (G.711 to and from LBR)
- Tone detection and generation (MF, DTMF, RFC 2833)
- Packet interface: Dual 10/100 Base-T link ports or cPSB back plane (PICMG 2.16) interface
- G.168-2000 compliant Echo Cancellation with a 32, 64 or 128 msec tail (may reduce channel density)

- Silence Suppression supporting VAD (Voice Activity Detection) and CNG (Comfort Noise Generation)
- Automatic Fax Bypass modes
- DTMF Detection and Generation according to TIA 464B
- DTMF Relay according RFC 2833
- PSTN Signaling: CAS, ISDN PRI
- Transport of SS-7 signaling, with the use of SIGTRAN. MTP-3 and higher layer messages are relayed using M2UA , M3UA over SCTP over IP
- MF-R1, MFC-R2 and Call Progress Tone detection and generation
- PICMG 2.1 for Hot-swap support
- PICMG 2.5 for H.110 support
- cPSB (PICMG 2.16) support
- Rear transition board
- Redundant 10/100 Base-T packet interface
- API Control via PCI or IP ('hostless' mode)
- Management Interfaces: SNMP V2, Embedded Web Server
- Single-slot compactPCI™ (cPCI) 6U card
- Compact, rugged 19-inch rack mount unit, one U high (1.75" or 44.5 mm), with two compactPCI™ (cPCI) slots
- Optional cPCI slot for optional 3rd party CPU board
- TP-1610 Hot swappable cPCI board

1.2 The Mediant 2000 Chassis

AudioCodes' **Mediant 2000** Media Gateway comprises a 19-inch 1U chassis with a 110/220 VAC power supply. The **Mediant 2000** is populated by a single AudioCodes compactPCI™ board, the **TP-1610**, and its Rear Transition Module (either 1-span, 2-span, 4-span, 8-span or 16-span) on which both PSTN trunks and the Ethernet interface are located.

The **Mediant 2000** chassis' front cage, slot #1 – the lower slot, houses the **TP-1610**.

The **Mediant 2000** chassis' rear cage, slot #1 – the lower slot, houses the **TP-1610** RTM.

Slot # 2 in the **Mediant 2000** chassis' front and rear cages can optionally be used by Customers for a CPU board. The following boards were tested for compliancy with the **Mediant 2000** chassis:

Sun™: CP2080 + PMC-233 (Ramix™ disk on board), or

Motorola™: CPV5375-700-02 (this includes an on-board HDD) + CPTM-01 (RTM).

1.3 Mediant 2000 Diagrams

Figure 1-1: Front View of Mediant 2000 Populated with AudioCodes' TP-1610 Board



The **Mediant 2000** can be provided with AudioCodes' 1-Span, 2-Span, 4-Span, 8-Span or 16-Span Rear Transition Module (RTM) boards

Figure 2-2, on page 23 shows AudioCodes' **Mediant 2000** chassis populated with AudioCodes' **8** -span RTM. Figure 2-2, on page 23 shows AudioCodes' **Mediant 2000** chassis populated with AudioCodes' **16** -span RTM (featuring 2 Telco connectors and 2 RJ-45 connectors).

The physical difference between AudioCodes' 1-Span, 2-Span and 4-Span RTMs, and the 8-span RTM (shown in Figure 2-2, on page 23), is that the RJ-48c ports are depopulated correspondingly.

Figure 1-2 displays AudioCodes' TP-1610 board. Figure 1-3 illustrates the corresponding RTM board.

Figure 1-2: AudioCodes' TP-1610 Board

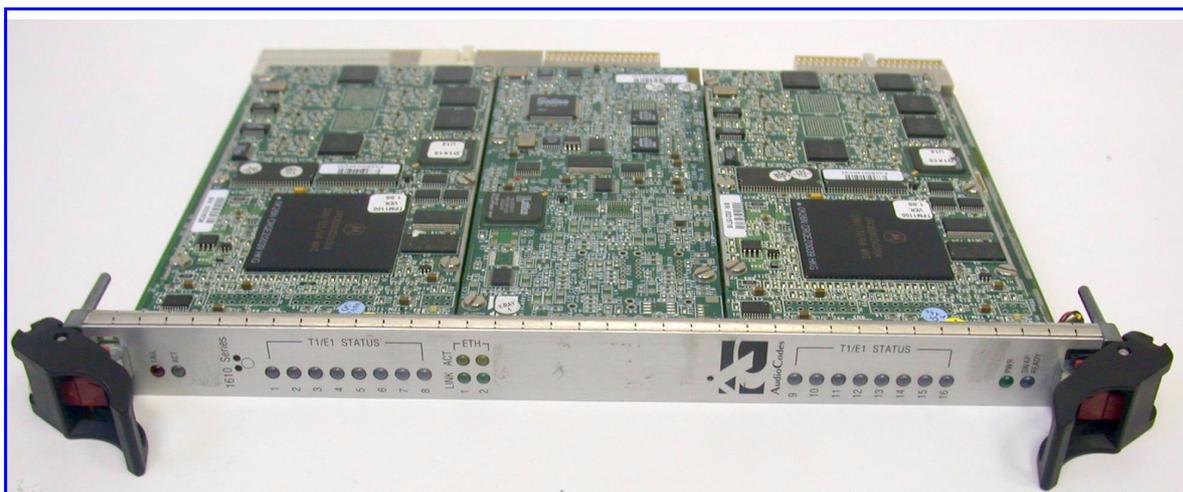
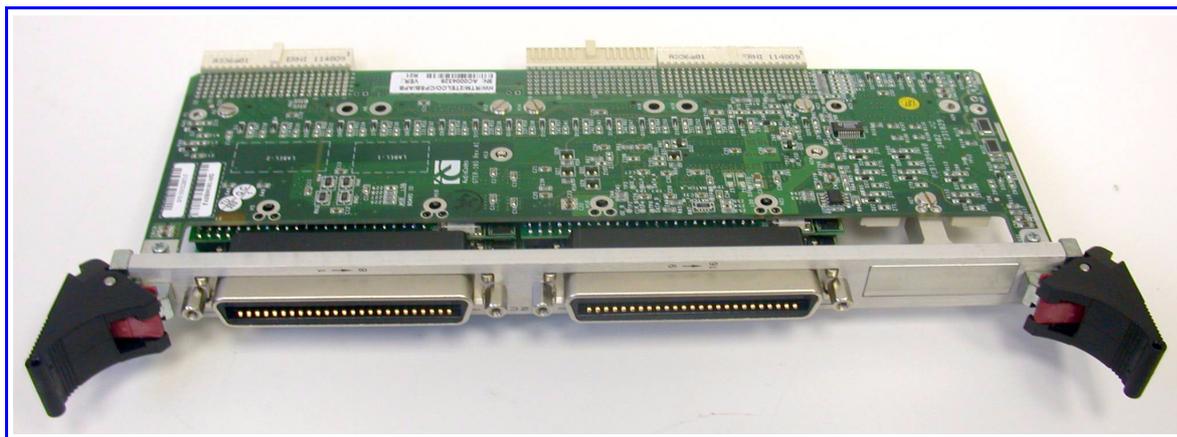


Figure 1-3: View of AudioCodes' TP-1610 RTM Board

1.4 TP-1610 Software Overview

The TP-1610 should be considered as two separate Media Gateway modules. Physically the TP-1610 takes up one slot of a cPCI chassis, but are as two independent entities, each with its own MAC address, IP address and board handle (when using the acOpenRemoteBoard API function), etc. In addition, each module with its own PSTN interface. Both modules share a redundant LAN connection via an internal Ethernet switch.

The TP-1610 is supplied with a multi platform/OS, VoP-compliant, media streaming driver package. This package is composed of the following items:

- **Low level PCI driver.** Device driver software for PCI/cPCI boards.
- **VoP-compliant media-streaming API.** The VoP Library (VoPLib) enables the User to control the way the board generates and processes a VoIP compliant stream. This API is similar for all TrunkPack-VoIP series boards, enabling Users to switch easily between different boards of this family without needing to modify the application.
- **PSTN API.** The PSTN Library (PSTNLib) uses a unified API to control all digital PSTN protocols (ISDN PRI, CAS trunks, MFR2 trunks etc.). It enables the User to benefit from the onboard dual E1/T1/J1 digital PSTN interface.

Additionally, demo applications are provided. These applications can serve as a reference for developing applications using the VoPLib/PSTNLib API.

1.5 Mediant 2000 Applications

AudioCodes' Mediant 2000 can be used in a variety of applications, which exploit its unique advantages of compressing PCM voice channels to IP packets according to international and ITU standards

Examples include:

- **Next Generation Switches**
- **IP Services Platforms**
- **VoIP Access Gateways**
- **Carrier Grade Trunking Gateways**
- **IP-enabled Call Centers**
- **Cable Telephony Gateways**

1.5.1 Available Configurations

AudioCodes' Mediant 2000 comes in a variety of channel densities and rear I/O options. Most of the descriptions and illustrations in this manual refer to the full capacity board.

1.6 Benefits

- **IP-enabled, cost-effective technology**
- **Low to high channel density, high performance board**
- **Concurrent toll quality voice and fax support**
- **Wide range of PSTN signaling protocols**
- **Fast time-to-market**
- **Flexible and easy migration to VoIP networks**
- **Extensive VoIP experience**
- **All-in-one integrated board - Reduced inventory**
- **Scalable distributed architectures**
- **Shorter development cycle**

1.7 Functional Block Diagram

Figure 1-4 is the block diagram illustrating the functionality of the .

Figure 1-4: Functional Block Diagram

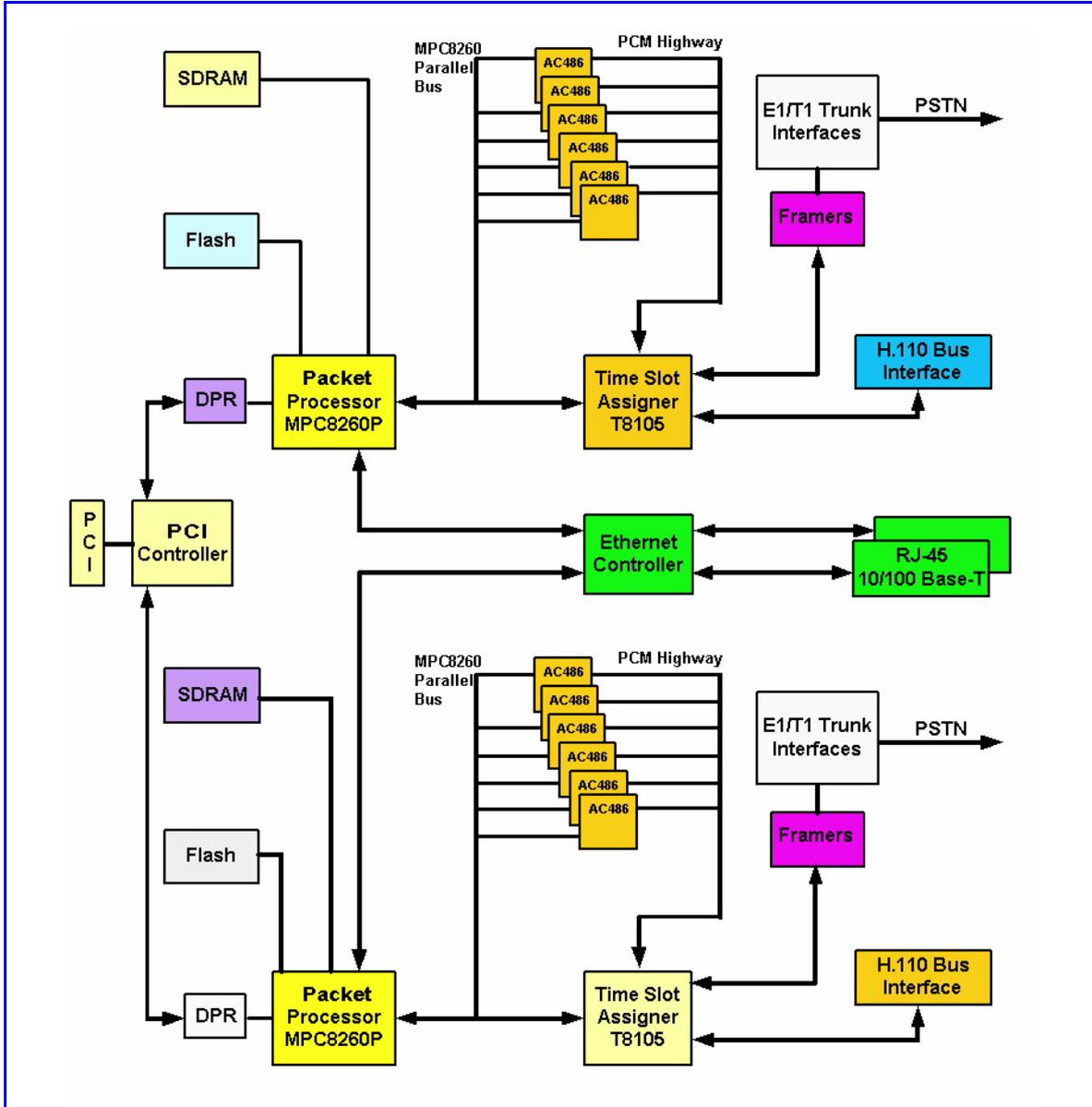
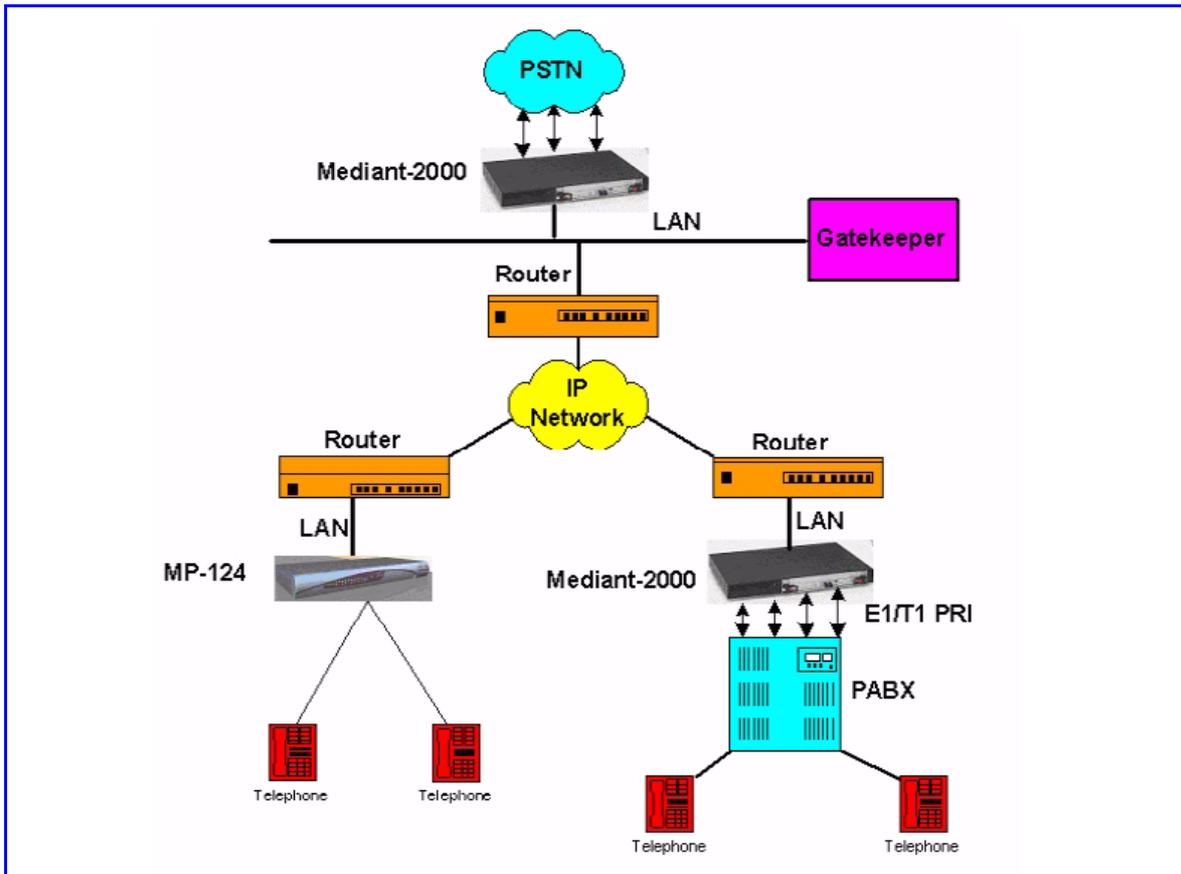


Figure 1-5, illustrates a typical Mediant 2000 VoIP application.

Figure 1-5: Typical Mediant 2000 VoIP Application



2 Hardware Installation

Caution

Prior to installation of any AudioCodes' board in a chassis, always correctly connect the chassis to a safety ground according to the laws and regulations of the country in which the installation is performed.

This section describes the hardware installation procedures for the **Mediant 2000** system. The **Mediant 2000** hardware installation can be for either for a desktop or a chassis system.

The **TP-1610** board is hot-swappable, so that the board can be inserted and removed when the cPCI system is running. The software library provided with the board, in any of these events, notifies the User's application so that the board can be inserted or removed at the correct time.

When inserting the **TP-1610** board into the cPCI chassis, press the two red latches and insert the board. Wait for the blue LED to light, indicating that the board has been inserted correctly.

When removing the **TP-1610** board from the cPCI chassis, press the two red latches and wait for the blue LED to light for a couple of seconds, indicating that the board can be removed.

There are no DIP switches to be set.

The **TP-1610** board supports Ethernet redundancy using its dual Ethernet interfaces. If interface #1 is not functioning, the board automatically switches to use interface #2.

The **TP-1610** board is a fully 'Plug and Play' device, in which the PC's boot-up sequence determines its I/O addresses and interrupts.

2.1 Unpacking

➤ **To unpack the Mediant 2000 take the following 6 steps:**

- Step 1.** Open the carton and remove the packing materials.
- Step 2.** Remove the **Mediant 2000** from the carton.
- Step 3.** Check that there is no equipment damage.
- Step 4.** Check, retain and process any documents.
- Step 5.** Notify AudioCodes of any damage or discrepancies.
- Step 6.** Retain any diskettes or CDs.

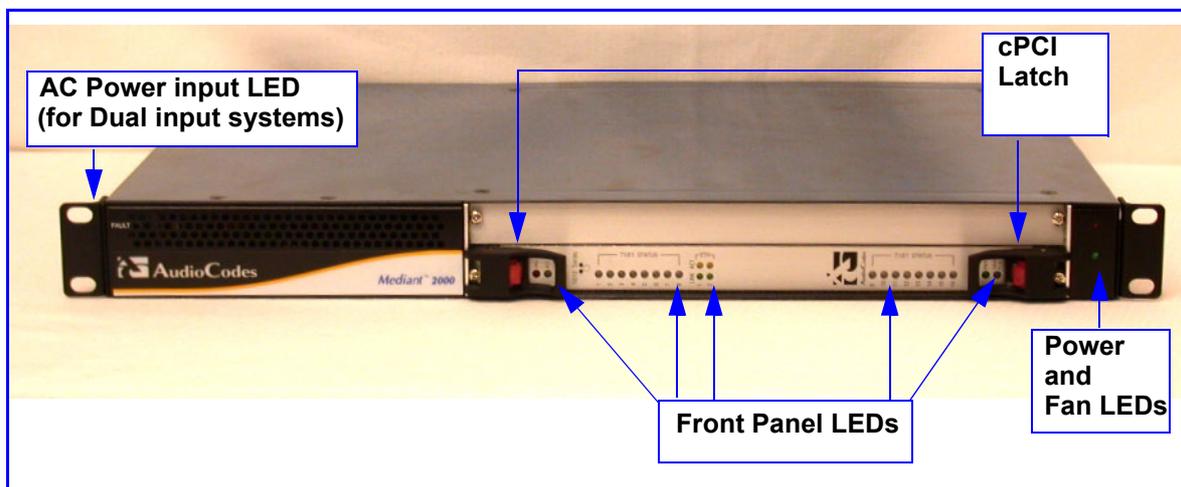
2.2 Installing

Caution

- 1 Electronic components on printed circuit boards are extremely sensitive to static electricity. Normal amounts of static electricity generated by clothing can damage electronic equipment. To reduce the risk of damage due to electrostatic discharge when installing or servicing electronic equipment, it is recommended that anti-static grounding straps and mats be used.
- 2 The equipment must only be installed or serviced by qualified service personnel.

2.2.1 Installing the Mediant 2000 Chassis

Figure 2-1: Mediant 2000 Front Details



➤ **To install the Mediant 2000 chassis, take the following 6 steps:**

Note: For a desktop or on a shelf installation, see Steps 1 and 2.
For a rack mount installation, go to Step 4.

For a desktop or on a shelf installation:

- Step 1.** Insert the small rubber feet into spaces provided on the underside of the **Mediant 2000** chassis.
- Step 2.** Position the **Mediant 2000** chassis on a desktop or shelf.
- Step 3.** **Go to Step 5.**

For a rack mount installation:

- Step 4.** Insert the **Mediant 2000** into the 19-inch rack and fasten the left-hand and right-hand brackets to the vertical tracks of the 19-inch rack, using standard 19-inch rack bolts (not provided).

Step 5. Connect the power supply using one of the following:

- 110/220 VAC power supply
- -48 VDC power supply

If you are using the 110/220 VAC power supply:

Attach a certified 100/240 VAC power cable to the rear AC socket (option dual AC sockets) and connect to the correct AC power supply.

If you are using the -48 VDC power supply:

When the optional DC power supply is used, the Gateway is equipped with a DC power inlet (type MSTB2.5/2-STF of Phoenix Contact). Use #14-16 AWG cables for the DC connection. The DC input is floating and the maximum input current is 4 amp.

Step 6. Go to Section 2.2.2 on page 23.

2.2.2 Connecting the E1/T1 Trunk Interface

Figure 2-2: Mediant 2000 Rear Connection Details (8 Spans)

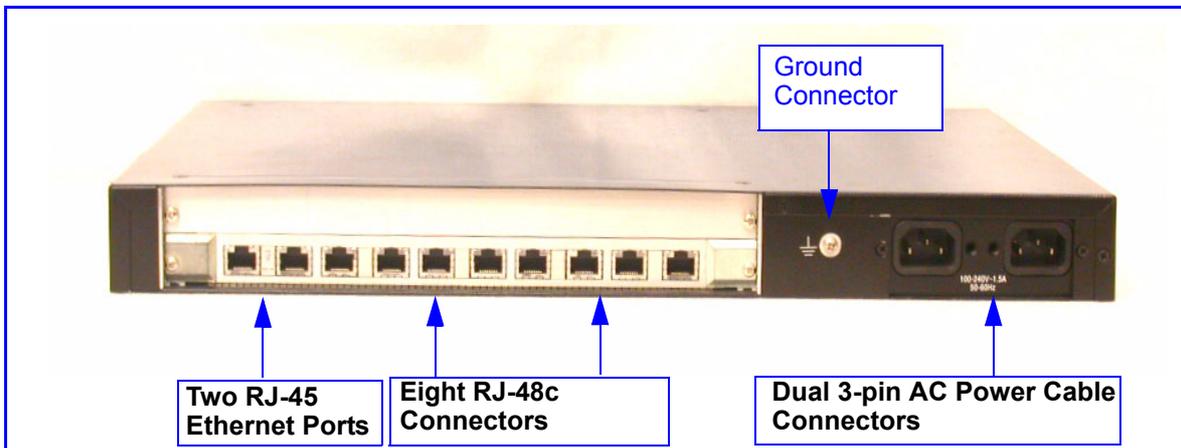
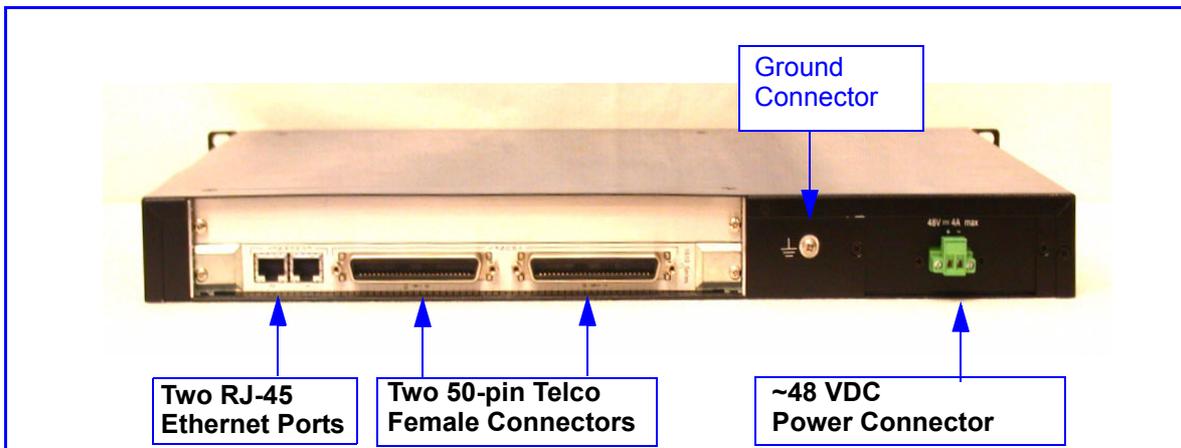


Figure 2-3: Mediant 2000 Rear Connection Details (16 Spans)



➤ **To connect the E1/T1 trunk interfaces, take the following 6 steps:**

Step 1. Connect to the E1/T1 trunk Interface using one of the following:

- **50-pin Telco connectors-- (for 16 Spans)**
- **RJ-48c connectors (for 8 Spans)** (Go to page 25.)

If you are using 50-pin Telco Connectors:

Connect the E1/T1 trunks cables to the **Mediant 2000** E1/T1 interfaces on the RTM panel. These are two 50-pin female Telco connectors (DDK 57AE-40500-21D) labeled Trunks 1-8 and Trunks 9-16 on the RTM shown in Figure 1-3, on page 17, and detailed in Figure 2-4 on page 24. The pin out of the 50-pin connectors is shown in Table 2-1 on page 24. **Now continue with Step 2.**

Note: The User's 50-pin **male** connector of Trunks 1 to 8 is connected to the 50-pin female connector labeled **E1/T1 1 to 8**. The User's 50-pin **male** connector of Trunks 9 to 16 is connected to the 50-pin female connector labeled **E1/T1 9 to 16**. The 2 male connectors **must be** wired identically, according to Table 2-1 and Figure 2-4 shown below.

Figure 2-4: 50-pin Female Telco Board-mounted Connector

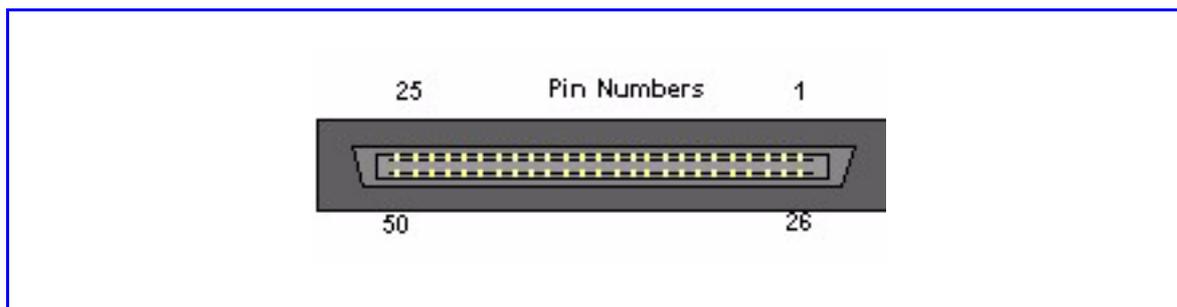


Table 2-1: Connections on each 50-pin Telco Connector

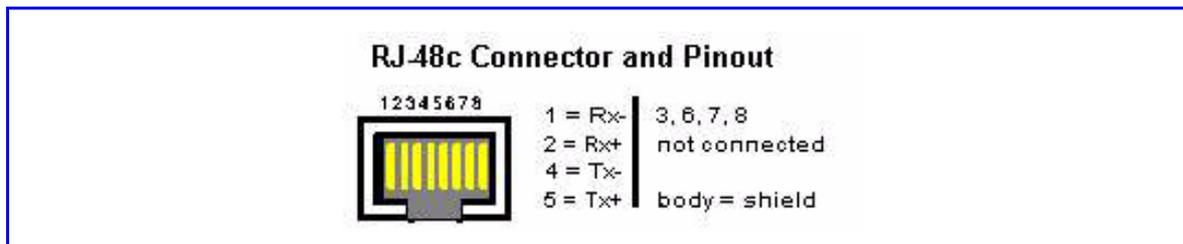
E1 Number		Tx Pins (Tip/Ring)	Rx Pins (Tip/Ring)
1 to 8	9 to 16		
1	9	27/2	26/1
2	10	29/4	28/3
3	11	31/6	30/5
4	12	33/8	32/7
5	13	35/10	34/9
6	14	37/12	36/11
7	15	39/14	38/13
8	16	41/16	40/15

If you are using RJ-48c Connectors:

Connect the E1/T1 trunk cables to the **Mediant 2000** E1/T1 interfaces on the RTM panel (refer to Figure 2-5 on page 25). These are eight RJ-48c connectors labeled Trunks 1 to 8 on the RTM shown in Figure 2-10, on page 29. The RJ-48c connectors are wired according to Figure 2-5 shown below. **Now continue with Step 2.**

The RJ-48c connector is wired according to Figure 2-5.

Figure 2-5: RJ-48c Trunk Connectors



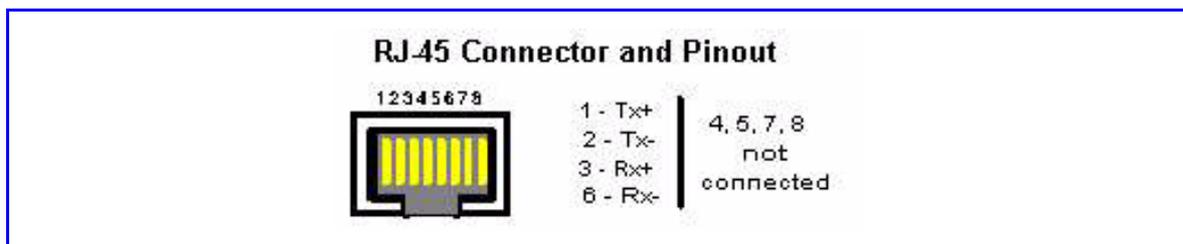
- Step 2.** On the RTM (shown in Figure 2-9 on page 27 or Figure 2-10 on page 29), connect the Category 5, LAN cables to the Ethernet1 and optionally Ethernet 2, RJ-45 interfaces. Connect the other end of the Category 5 LAN cable to your IP network.

The RJ-45 connectors labeled **Ethernet 1** and **Ethernet 2** are wired according to Figure 2-6 shown below.

- Step 3.** For redundant operation, connect **Ethernet 1** to Ethernet Switch #1 and **Ethernet 2** to Ethernet Switch #2.

The RJ-45 connectors labeled **Ethernet 1** and **Ethernet 2** are wired according to Figure 2-6.

Figure 2-6: RJ-45 LAN/Ethernet Network Connectors



- Step 4.** Connect the ground. Refer to Figure 2-2 and Figure 2-3, on page 23.
- Step 5.** Connect the AC or DC power cables to the rear of the chassis and connect the power source. The AC version can be provided with single or dual power cable connectors. For the DC power connector, refer to Figure 2-7

Figure 2-7: DC Power Connector



Step 6. Restart the applications and run the system.

The Mediant 2000 hardware installation is now complete.

2.3 Chassis LED Indicators

Table 2-2 details the LED indicators on the front panel of the chassis.

Table 2-2: Chassis Indicators

Placement	Color	Function
Right side of front panel	Green	Power is on
Right side of front panel	Red	Fan failure - indicates that any of the internal fans' speed has significantly reduced or has gotten stuck
Left side of front panel	Red	Power supply failure - indicates that one of the two AC redundant power supplies is faulty

2.4 The TP-1610 Board

The **TP-1610** cPCI Board is the main component of the **Mediant 2000**. It is supplied within the **Mediant 2000** Gateway shown in Figure 2-1, on page 22.

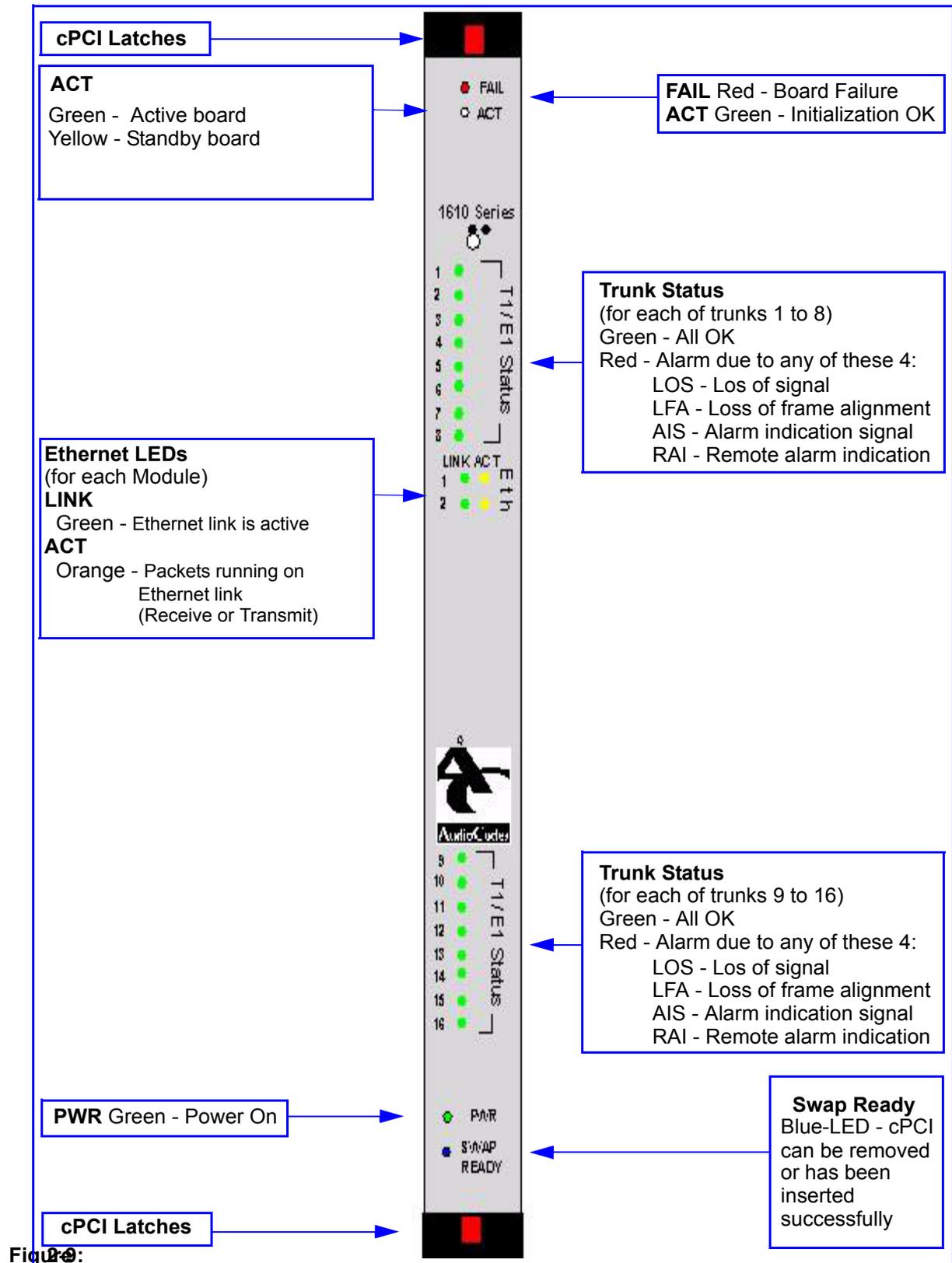
The front panel of the **TP-1610** board is shown in Figure 2-6 on page 25. The panel options for the **TP-1610** RTM are shown in Figure 2-9 on page 27 or Figure 2-10 on page 29. Section 2.2.2 on page 23 provides details about the LED indicators and additional information.

The **Mediant 2000** Media Gateway is provided with two types of RTM rear panels:

- RTM rear panel with two 50-pin Telco connectors, shown in Figure 2-4 on page 24. This RTM is required for a Media Gateway equipped with up to 16 E1/T1 spans.
- RTM rear panel with eight RJ-48c connectors, shown in Figure 2-6, on page 25. This RTM can be used for a Gateway with up to 8 E1/T1 spans (1, 2, 4 or 8 E1/T1 spans).

Consult an AudioCodes representative for more information on the available configurations.

Figure 2-8: TP-1610 Board, Front Panel View



RTM Panel with 2 Telco Connectors

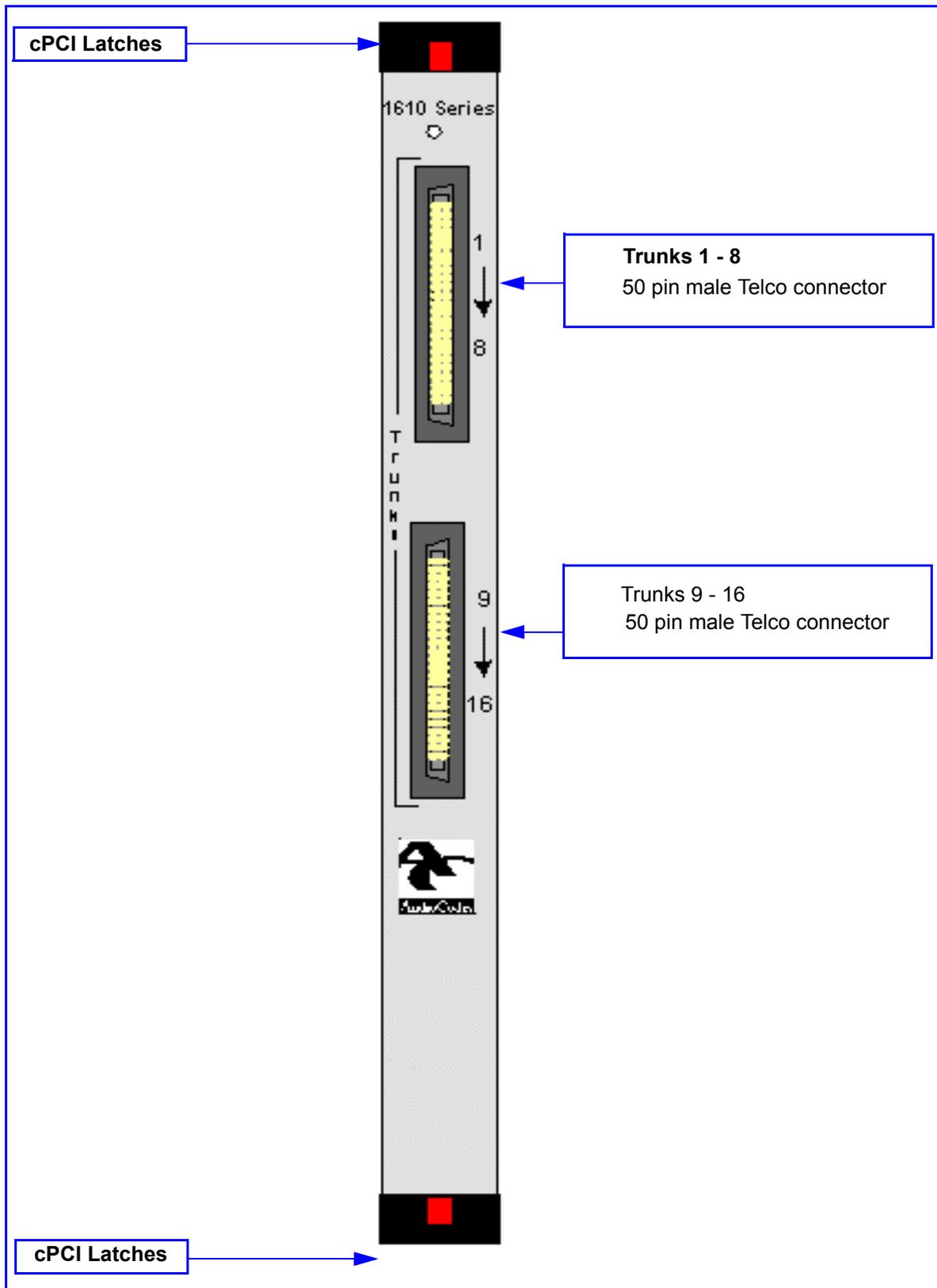
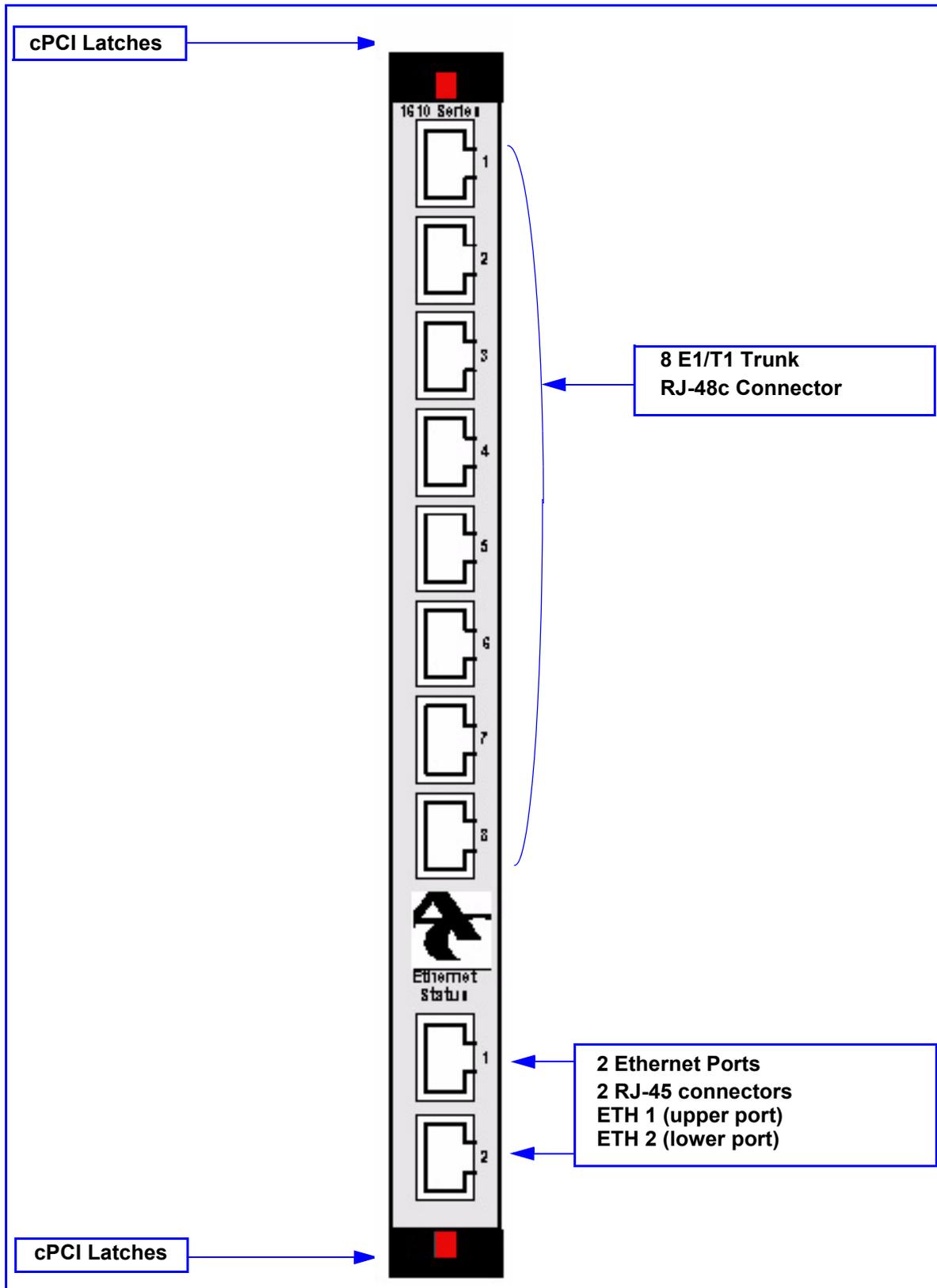


Figure 2-10: RTM Panel with 8 RJ-48c Trunk Connectors



2.4.1 TP-1610 Board Front Panel LED Indicators

Refer to Table 2-3 through to for LED indicator definitions.

Table 2-3: Board Status LED Indicators

Label	Color	Function
FAIL	Red	Normally OFF; Red shows board failure (fatal error)
ACT	Green	Board initialization sequence terminated OK
	Yellow	N/A

The FAIL LED is normally OFF, and turns to Red to indicate board failure.

The ACT LED is lit Green as soon as download is completed successfully.

Table 2-4: Trunks Status LED Indicators

Label	Color	Signal Description
(Currently not in use)	Green	Trunk is synchronized (normal operation)
	Red	Loss due to one of the following 4 signals:
	LOS	Loss of Signal
	LFA	Loss of Frame Alignment
	AIS	Alarm Indication Signal (The blue alarm)
	RAI	Remote Alarm Indication (The yellow alarm)

During normal **Mediant 2000** operation, the E1/T1 bicolor LED is lit Green for each Trunk. Any other condition, either in the E1/T1 cable, in the **Mediant 2000** or at the remote side, causes the E1/T1 bicolor LED to be lit Red, indicating a loss due to any of the 4 signals shown above.

Table 2-5: Ethernet LED Indicators

Label	Color	Function
LINK	Green	Link all OK
ACT	Yellow	Transmit/Receive Activity

Table 2-6: Auxiliary LED Indicators

Label	Color	Function
PWR	Green	Power is supplied to the board
SWAP READY	Blue	The board can be removed or inserted

Before removing the board, wait for the Blue LED to flash ON and then stay OFF

2.4.2 Optional CPU Board

The **Mediant 2000** provides an optional second cPCI slot that can be used for a Customer's CPU board. This CPU board can be used for general applications such as Softswitch, Application Server or other. The following CPU boards were tested for compliancy with the **Mediant 2000** chassis:

- **Sun™: CP2080 + PMC-233 (Ramix™ disk on board).**
- **Motorola™: CPV5375-700-02 (with integral HDD) + CPTM-01 (RTM).**

2.4.3 Board Replacement

The **TP-1610** cPCI board is hot-swappable, meaning that the board can be inserted and removed while the **Mediant 2000** chassis is under power.

For the optional CPU board, refer to the directions accompanying the board.

➤ **To remove a front TP-1610 board from the chassis, take the following 3 steps:**

- Step 1.** Unfasten the screws on the plate of the board.
- Step 2.** Press the red ejector buttons on the two black ejector/injector latches on both ends and wait for the Hot Swap blue LED to light, indicating that the board can be removed.
- Step 3.** Pull on the two ejector/injector latches and ease out the board from the slot.

➤ **To remove the TP-1610 RTM from the chassis, take the following 4 steps:**

- Step 1.** Remove the cables attached to the RTM.
- Step 2.** Unfasten the screws on brackets at both ends of the panel that secure the board to the chassis.
- Step 3.** Press the red ejector buttons on the two black ejector/injector latches on both ends.
- Step 4.** Grasp the panel and ease the board out of the slot.

➤ **To insert a TP-1610 board into the cPCI chassis, take the following 9 steps:**

- Step 1.** Hold the board horizontally.
- Step 2.** With the black ejector/injector latches in the open (pulled out) position, insert the board in the slot, aligning the board on the groves on each end.
- Step 3.** Ease the board all the way into the slot until the ejector/injector latches touch the chassis. The Blue hotswap LED is lit.
- Step 4.** Press the two black ejector/injector latches on both ends inward, toward the middle until you hear a click.
- Step 5.** Wait for the hotswap blue LED to turn off.
- Step 6.** Fasten the screws on the front plate of the board to secure the board to the chassis and to ensure that the board has a ground connection to the chassis.
- Step 7.** Reattach the cables.

➤ **To insert a TP-1610 RTM board into the cPCI chassis, take the following 6 steps:**

- Step 1.** Hold the board horizontally.
- Step 2.** With the black ejector/injector latches in the open (pulled out) position, insert the board in the slot, aligning the board on the grooves on each end.
- Step 3.** Ease the board all the way into the slot until the ejector/injector latches touch the chassis.
- Step 4.** Press the two black ejector/injector latches on both ends inward, toward the middle until you hear a click.
- Step 5.** Fasten the screws on the front plate of the board to secure the board to the chassis and to ensure that the board has a ground connection to the chassis.
- Step 6.** Reattach the cables.

3 Software Installation

3.1 Software Installation

3.1.1 Unzipping the Package

➤ **When using a zip file in Solaris™, take these 3 steps:**

Step 1. List the contents of the zip file.

```
unzip Sample_M2K_MEGACO_4_20_xxx_y -l
```

Step 2. Extract all files into the current directory (maintains directory hierarchy of zip).

```
unzip Sample_M2K_MEGACO_4_20_xxx_y
```

Step 3. Extract all files into a specified root directory.

```
unzip Sample_M2K_MEGACO_4_20_xxx_y -d /tmp
```

Note: Remember that you must have permission to create/modify in the directory into which you want to unzip.

Use a WinZip compatible utility to unzip the files into the desired location.

3.2 Initialization Process

3.2.1 Mediant 2000 with a CPU Board

This section describes the process that takes place after the **TP-1610** has been reset or powered up. The following major initialization sequences are supported:

- **BootP/DHCP-based Initialization** – in this mode, the **TP-1610** is configured through the network interface only, and no interaction takes place with the hosting cPCI machine. This method is a proprietary extension of the well-known BootP/DHCP protocol. The board starts applying this method only if it has detected that it wasn't initialized by the hosting cPCI machine. The **TP-1610** performs BootP/DHCP from each of its modules, with two different MAC addresses. This mode is described in Section 3.2.2 on page 34.
- **Booting the last configuration stored in the integral non-volatile memory** - In this case, the **TP-1610** uses the firmware, configuration parameters (including IP address, IP subnet mask, etc.), and tables stored in the non-volatile memory by the previous configuration process. The **TP-1610** does not attempt to connect to BootP/DHCP servers. In this mode, the User can determine whether the board uses only the stored firmware, the stored configuration files, or both. This option is very useful for operational boards installed in the field, and can be used in environments in which the IP addresses are static.

Note: When working in hostless mode (with BootP/TFTP) the IP address corresponding to each board's MAC address should be preconfigured in the BootP server to avoid network conflicts. When working in host mode (PCI) the IP address is part of the boardparams structure that should be filled before opening the board.

3.2.2 BootP/DHCP-based Initialization Process

In BootP/DHCP-based mode, when the system is initialized (and after it determines that the local cPCI machine based VoP library did not activate the acOpenBoard() function,) the TP-1610 searches for an external BootP/DHCP server. If a BootP/DHCP server is located, a configuration process takes place. If no server is located, the board reverts to the last valid configuration stored in the integral non-volatile memory. The BootP/DHCP server can be used to provision the following parameters:

- **IP address, IP subnet mask** – these parameters are mandatory and are supplied by the server to the module every time the BootP/DHCP process takes place.
- **Default Gateway IP address** – this configuration parameter is optional and is supplied by the BootP/DHCP server only if the field was defined in the server configuration.
- **Firmware file name** – when the module detects that this optional parameter was specified in the BootP/DHCP server, it initiates a TFTP process to download this file.

- **Command Line Switches**

The following Command Line Switches are available:

- fb Burn ram.cmp in flash (only for cmp files)
- em #Set Ethernet mode
- br BootP retries
- bd BootP delays

Examples of use:

“ramxxx.cmp -fb” to burn flash memory.

“ramxxx.cmp -fb -em 4” to burn flash memory and for Ethernet Mode 4 (auto-negotiate).

Note: Such Command Line Switches can only be added in the BootP file name location. The actual file name remains (e.g., ramxxx.cmp) and does not change.

- When adding the ‘-fb’ command line switch the image file is stored in the integral non-volatile memory. When the regular *.hex extension is used, the downloaded image is the one used by the board, but it is not stored in the integral non-volatile memory (even when setting the ‘-fb’ command line switch). If the firmware file name has not been specified in the BootP/DHCP server, the board uses the last image stored in its non-volatile memory.
- **Example:** In the BootP server (in the Client Configuration screen), set the **Boot File** field:

ramxxxxx.cmp -fb; TP1610.ini

Note: The file *ram.hex* is a downloadable firmware file. The file *ram.cmp* is a compressed version of the *ram.hex*, enabling faster download and reduced file size. The *ram.cmp* is the only version that can be burned to the flash memory.

- **Configuration (*ini*) file name** – This parameter is the name of an AudioCodes proprietary configuration file, with an *ini* extension, capable of containing various configuration parameters and tables. For additional information on the format of this file, refer to section “BootP/DHCP Configuration” in AudioCodes’ “VoPLib Reference Library User’s Manual”, Document #: LTRT-00744. When the **TP-1610** board detects that this optional configuration parameter was specified in the BootP/DHCP server, it initiates a TFTP process to download the file. The new configuration contained within the *ini* file is stored in the integral non-volatile memory, to be used whenever an *ini* file name is specified by the BootP/DHCP server (or when a BootP/DHCP server is not found or not searched for).

3.3 BootP & TFTP Server Installation and Configuration

Users can use any commercially available BootP and TFTP server, provided compliant with the relevant RFCs.

AudioCodes’ **Mediant 2000** BootP client fully complies with IETF RFC 951, RFC 1542 and RFC 2132.

AudioCodes’ **Mediant 2000** TFTP client fully complies with IETF RFC 906 and RFC 1350.

Note: Instructions for BootP and TFTP Server Installation & Configuration are detailed, with examples of the individual screens, in AudioCodes’ “Software Utilities Manual”, Document #: LTRT-00702.

3.3.1 Mediant 2000 without CPU Board

This section describes the process that occur after the **Mediant 2000** is reset or powered up. The following major initialization sequences are supported.

BootP/DHCP-based Initialization – in this mode, the **Mediant 2000** is configured through the network interface only and no interaction takes place with the hosting cPCI machine. This method is a proprietary extension of the well-known BootP/DHCP. The board starts applying this method only if it detects that it wasn’t initialized by the hosting cPCI machine. The **Mediant 2000** performs BootP/DHCP from each of its modules, with two different MAC addresses. This mode is described in Section 4.1.3

For detailed information, refer to the AudioCodes “VoPLib Reference Library User Manual”, Document # LTRT-00744.

3.4 Software Directory Structure

The file structure of the CD is as follows:

```
/
/Tones
/Tones/Tones_IniFiles
/Utilities
```

Root directory:

- CD Contents text file
- Mediant 2000 User's Guide
- Mediant 2000 Fast Track Installation Guide
- Default Configuration INI file
- Software Load (.cmp) file
- Version Description Document (Release Notes)

Tones subdirectory :

- Country specific tone DAT files (created from tone INI files)
- Tones/Tones_IniFiles subdirectory:
- Country specific tone INI files

Utilities subdirectory:

- audcVer.pl (displays version of .cmp and .dat files)
- README_AudcVer.txt (readme file for version tool)
- bootp server installation zip file
- conversion tool for converting tone INI files to dat files
- syslog daemon installation package

4 Operation & Configuration

4.1 General

AudioCodes' **Mediant 2000** can be controlled from a Media Gateway Controller (MGC)/Call Agent using the standard MEGACO (Media Gateway Control) protocol.

4.2 MEGACO (Media Gateway Control) Protocol

4.2.1 MEGACO Overview

MEGACO (MEDIA GATEWAY CONTROL) Protocol is a standard, network-based control protocol (based on IETF RFC 3015 and ITU-T recommendation H.248 V1). MEGACO assumes a call control architecture where the call control intelligence is outside the Media Gateway and handled by an external Media Gateway Controller (MGC). MEGACO is a master/slave protocol, where the gateway is expected to execute commands sent by the Call Agent (another name for MGC).

The connection is handled using two elements: **Terminations** and **Contexts**. **Termination** is the basic element of the call. There is a physical Termination representing a physical entity (e.g., analog line), and an ephemeral Termination representing the generated stream. To create a connection, a **Context** is used. A **Context** contains one or more Terminations, and describes the topology between the Terminations. A typical connection creation command creates a new Context and adds into it one physical Termination and one new (ephemeral) Termination. The ephemeral Termination parameters describe the media type and the stream direction (SendReceive, SendOnly or ReceiveOnly).

Since this is a standard control protocol, AudioCodes does not provide any special software library to enable Users to construct their own Call Agent (many such stacks are available in the market and the User is able to choose any one of them).

4.2.2 Operation

4.2.2.1 Executing MEGACO Commands

MEGACO commands, received from an external Call Agent through the IP network, are decoded and executed in the board. Both text encoding and binary encoding are supported. Commands can create new connections, delete connections, or modify the connection parameters.

Several commands that support the basic operations required to control a Media Gateway:

Status change command: The command ServiceChange allows changing the status of one or more Terminations. When used with a special Termination, called the ROOT Termination, it affects the entire Media Gateway.

Connection commands: The commands Add, Move, Modify and Subtract allow the creation and deletion of a call connection inside the Media Gateway. These commands allow the application to create new connections, delete existing connections, and modify the connection parameters.

Notify command: The Notify command is used by the gateway to inform the Call Agent of events occurring on one of the Terminations.

Audit commands: The AuditCapabilities and AuditValue commands are used to query the gateway about Termination configuration and state. This information helps in managing and controlling the gateway.

A MEGACO-configured Media Gateway starts by sending a ServiceChange command to its primary MGC. If no response is received from it, the gateway goes on to the next MGC in its list. When an MGC accepts the gateway registration, the session can start. From now on, the gateway responds to MGC commands. Even notifications are sent only if the MGC requests them specifically. If the KeepAlive mechanism is enabled, the gateway sends a NOP ServiceChange command when it detects a long period without commands from the MGC (the default period is 12 seconds.) If no response is received from the MGC, the retransmission mechanism starts and eventually causes a new ServiceChange command to be sent to the next available MGC.

4.2.2.2 Setting MEGACO Call Agent IP Address and Port

Users can provide the gateway with up to 5 IP addresses of the MEGACO Call Agents using the parameters 'ProvisionedCallAgents' and 'ProvisionedCallAgentsPorts' .

The first Call Agent in the list is the primary one. In the case of a loss of connection, the gateway tries to connect with the next on the list, and it will continue trying until one of the Call Agents accepts the registration request. If the current connection is with a secondary MGC, the gateway starts again from the primary MGC. The current Call Agent can override this setting by sending a ServiceChange command with a new IP address (not necessarily in the original list) and a HandOff method. If no CallAgent IP address exists, MEGACO will not become operational.

Instead of defining an IP address, Users can use a domain name for the Call Agent using the 'CallAgentDomainName' parameter. When using it, define also the 'DNSPRISERVERIP' and 'DNSSECSERVERIP' parameters. When using a domain name, the gateway resolves the name on each disconnection, allowing the User to switch to another Call Agent.

4.2.2.3 Handling Events

Events are declared in an EventsDescriptor, that has an ID and a list of events on which the Call Agent requires notification. Up to 16 events can be defined in the descriptor. Wildcards are permitted in the events names. For example, if the list includes 'dd/*', and the User presses the number 1, the Call Agent receives notification when the digit starts (dd/std{tl=d1}) and when it ends (dd/etd{tl=d1}). The event dd/d1 is not sent as it is included in the other two. An event can have parameters, for example the KeepActive flag. When the event having the KeepActive flag is received, it does not stop the currently played signals.

An event can have an embedded descriptor in it. It can be a SignalsDescriptor (refer to Section 4.2.2.4), a new EventDescriptor, or both. The embedded descriptor replaces the current descriptor.

4.2.2.4 Playing Signals

Signals in MEGACO reside in a SignalDescriptor. Only one signal is allowed in the descriptor as the boards cannot play more than one signal at a time. However, this one signal can be of the SignalList type, in which case, there can be up to 30 signals in the list, and they are played sequentially until the list ends or the execution is interrupted. Interrupting the execution can be one of the following:

- **Event** – Only events required by the Call Agent stop the execution, and only if they don't have the KeepActive flag.
- **New Signals Descriptor** – Stops the execution, unless the same signal is received, and it has a KeepActive flag. If the old signal and the new signal are both signal lists and have the same ID, the new signal is ignored.
- **Subtracting the termination from the call.**

When a signal is ended, a signal completion notification is sent only if:

- The signal has the NotifyCompletion parameter and the completion reason (TimeOut, Interrupted by Signal, Interrupted by Event) matches one of the NotifyCompletion parameters.
- The events descriptor contains the signal completion event (g/sc).

The notification includes the ID of the signal which was ended and the signal list ID if it was a signal list.

Signal duration can be defined as a parameter in the signal. If omitted, a default value is used. (Refer to the packages description).

Call Progress Tones must be defined by the User in a Call Progress Tones (CPT ID) file. An off-line utility is supplied to convert this file to a binary file. Each tone has a toneld in the file, used by MEGACO when playing the signal. For the correlation between signal names and CPT file IDs, refer to the column "Map to CPT File" of Table 4-2, on page 44. When a CPT file is missing, the board defines default values only for the following signals:

- Dial tone
- Ringing tone
- Busy tone

Announcements should also be prepared offline by Users.

The following example shows a command that plays a list of announcements. When the list is finished, a notify command is sent:

```
MEGACO/1 [172.16.8.88]
T=207{
C = 1 {
Modify = gws0c1 {
  SG{SL=1234{an/apf{an=2},an/apf{an=3},an/apf{an=1,NC={TO,IBS}}}},
  E=1001 {g/sc}}}
```

And the Notify request:

```
MEGACO/1 [10.2.229.18]:2944
T=2015{
C = 1 {
O-N=gws0c1{
OE=1001{19700101T00003542: g/sc{
Meth=TO, SigId=an/apf, SLID=1234}}}}
```

4.2.2.5 Mediation

Mediation in MEGACO is connecting two ephemeral terminations. This operation can be used by a Call Agent to connect Users with different coders or to connect two types of Users, such as ATM and RTP. The mediation operation requires up to two DSPs according to the following rules:

- When both Users use the same coder, no DSP is allocated.
- When one User uses a G.711, one DSP is allocated for the other User.
- When both users use non-G.711 and different coders, two DSPs are allocated.

The mediation is created with a simple MEGACO ADD command, with two ephemeral terminations, as shown in the following example:

```
MEGACO/1 [10.10.0.70]; Connect the streams,
Transaction = 2 {
  Context = $ {
    Add = $ {
      Media {
```

```

        LocalControl {
            Mode = SendReceive,
            rtp/jit=70 },
        Local {
v=0
m=audio $ RTP/AVP 0
c=IN IP4 $
        },
        Remote {
v=0
m=audio 4000 RTP/AVP 0
c=IN IP4 10.2.229.19
        }
        },
        Add = $ {
            Media {
                LocalControl {
                    Mode = SendReceive,
                    rtp/jit=70 },
                Local {
v=0
m=audio $ RTP/AVP 4
c=IN IP4 $
                },
                Remote {
v=0
m=audio 4010 RTP/AVP 4
c=IN IP4 10.2.229.19
                }
            }
        }
    }
}

```

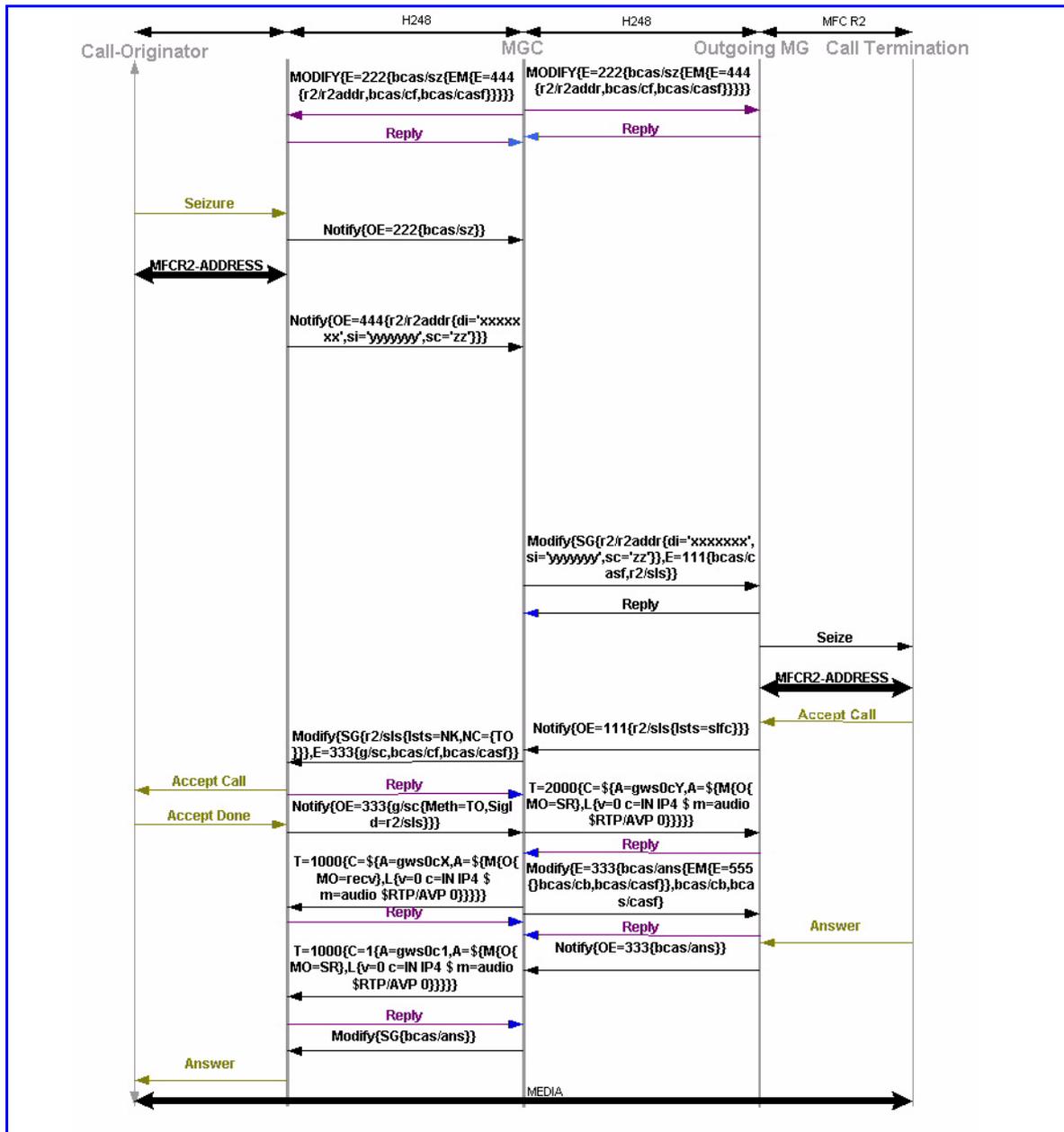
This example connects two RTP streams, one uses the G.711 coder and the other uses the G.723 coder.

4.2.2.6 CAS/R2 support in MEGACO

The CAS/R2 trunk protocols are supported in MEGACO by using the 'bcas' package and the 'r2' package (draft-ietf-megaco-r2package-02).

This protocol is described here as this is the only case where MEGACO shares the control of the channel with the MFCR2 application. This special case dictates a time synchronization between the two protocols, so that MEGACO commands will not cause a damage for the MFCR2 negotiation. For example, The MFCR2 must work with echo canceller in OFF state, otherwise the MF will not be received correctly. So if the MEGACO will get a command to open the channel with echo canceller on, and the MFCR2 is not finished, the whole negotiation may be damaged. Therefore, the following call flow should be used in order to avoid such a collision:

Figure 4-1: MEGACO-R2 Call Flow Diagram



4.2.3 SDP Support in MEGACO

The MEGACO supports basic SDP (Session Description Protocol), as defined in RFC 2327. (Refer to the document at www.ietf.org/rfc/, 'RFC Index'.) It supports also the SDP-ATM, as defined in RFC 3108. Supported SDP attributes are:

- **SILENCESUPP:VAL**
(VAL=on or off) – To turn silence suppression on or off (Defined in RFC 3108).
- **RTPMAP**
Used for dynamic payload mapping, to map the number to the coder. The format is:
a=rtptime:97 G723/8000/1
Where: 97 is the payload number to be used
G723 is the encoding name
8000 is the clock rate (optional)
1 is the number of channels (optional)

■ **FMTTP**

Used for dynamic payload mapping, to define coder specific parameters. The format is:

a=fmtp:97 bitrate=5.3

Where: : 97 is the payload number to be used

bitrate is a G.723 coder parameter.

Other supported parameters are:

mode-set – Defines for the AMR and the X-NETCODER coder which mode is used.

(0-7)

annexa – Defines for G.723 if silence suppression is on (yes or no)

annexb – Defines for G.729 if silence suppression in on (yes or no)

■ **PTIME**

Defines the paketization time for the session. The format is:

a=ptime:20

4.2.3.1 Mapping Payload Numbers to Coders

Table 4-1 shows the default mapping between payload numbers and coders when the dynamic payload assignment **is not used**. Note that this is a general table and only the DSP template that is loaded to a board defines which coder is supported on this board.

Table 4-1: MEGACO Mapping Payload Numbers to Coders (continued to page 43)

Default PayLoad Number	Encoding Name	Coder
0	"PCMU"	G711Mulaw
2	"G726-32"	G726_32
3	"GSM"	GSM
84	"GSM-EFR"	GSM-EFR
4	"G723"	G723 (High)
80	"G723"	G723 (Low)
8	"PCMA"	G711Alaw_64
15	"G728"	G728
18	"G729"	G729
35	"G726-16"	G726_16
36	"G726-24"	G726_24
38	"G726-40"	G726_40
39	"X-G727-16"	G727_16
40	"X-G727-24-16"	G727_24_16
41	"X-G727-24"	G727_24
42	"X-G727-32-16"	G727_32_16
43	"X-G727-32-24"	G727_32_24
44	"X-G727-32"	G727_32
45	"X-G727-40-16"	G727_40_16
46	"X-G727-40-24"	G727_40_24

Table 4-1: MEGACO Mapping Payload Numbers to Coders (continued to page 43)

Default PayLoad Number	Encoding Name	Coder
47	"X-G727-40-32"	G727_40_32
49	"X-NETCODER"	NetCoder_4_8
50	"X-NETCODER"	NetCoder_5_6
51	"X-NETCODER"	NetCoder_6_4
52	"X-NETCODER"	NetCoder_7_2
53	"X-NETCODER"	NetCoder_8
54	"X-NETCODER"	NetCoder_8_8
55	"X-NETCODER"	NetCoder_9_6
56	"X-CCD"	Transparent
60	"EVRC"	EVRC
81	"X-EVRC-TFO"	EVRC (TFO)
61	"X-QCELP-8"	QCELP_8
82	"X-QCELP-8-TFO"	QCELP_8_TFO
62	"QCELP"	QCELP_13
83	"X-QCELP-TFO"	QCELP_13_TFO
63	"G729E"	G.729E
64	"AMR"	AMR (4.75)
65	"AMR"	AMR (5.15)
66	"AMR"	AMR (5.9)
67	"AMR"	AMR (6.7)
68	"AMR"	AMR (7.4)
69	"AMR"	AMR (7.95)
70	"AMR"	AMR (10.2)
71	"AMR"	AMR (12.2)
96	"telephone-event"	RFC 2833
104	"RED"	Redundancy per RFC 2198
13	"CN"	Comfort Noise

Note: When using dynamic payloads, do not use the board default payloads for RFC 2833 (96) and RFC 2198 (104). If these values must be used, the default values for the two RFC's should be changed in the INI file.

4.2.4 Supported MEGACO Packages

Events, signals, properties and statistics are grouped in packages. A package can be extended by a new package. In this case, the basic package becomes a part of the new package.

The TrunkPack series MEGACO supports the basic set of packages as defined in Annex E of RFC 3015 (Refer to the document at www.ietf.org/rfc/, 'RFC Index'), according to the board type. For example, the Analog Line package is supported only for the analog boards.

Note: Unlike MGCP, the MGC must define ALL events for which it requires notification. There are NO persistent events in MEGACO.

4.2.4.1 Generic Media Package - G

Table 4-2: Generic Media Package - G

Symbol	Definition	Type
cause	General failure report	Event
sc	Signal completion	Event

Notes on Tables for MEGACO Packages (These notes refer to Table 4-2, on page 44 to Table 4-18, on page 50):

- S:** The signal type; the following symbols identify the type of signal:
- OO signal:** The On/Off signal is turned ON until commanded by the Call Agent to turn it OFF, and vice versa.
- TO signal:** The Timeout signal lasts for a given duration unless it is superceded by a new signal.
- BR signal:** The Brief signal event has a short, known duration.
- Duration:** Specifies the duration of TO signals.

4.2.4.2 Base Root Package - ROOT

Table 4-3: Base Root Package - ROOT

Symbol	Definition	Type
maxNumberOfContexts	Maximal number of Contexts in the Media Gateway	Property
maxTerminationsPerContext	Maximal Terminations in a Context.	Property
normalMGExecutionTime	Timer for Retransmission	Property
normalMGCExecutionTime	Timer for Retransmission	Property
MGProvisionalResponseTimerValue	Timer for Retransmission	Property
MGCProvisionalResponseTimerValue	Timer for Retransmission	Property

4.2.4.3 Tone Generator Package - ToneGen

Table 4-4: Tone Generator Package - ToneGen

Symbol	Definition	Type	S	Duration
pt	Plays audio tone	Signal	TO	

4.2.4.4 Tone Detection Package - ToneDet

Table 4-5: Tone Detection Package - ToneDet

Symbol	Definition	Type
std	Detect the start of a tone	Event
etd	Detects the end of a tone	Event
ltd	Detects a long tone	Event

4.2.4.5 DTMF Generator Package - DG (Extends ToneGen)

Table 4-6: DTMF Generator Package - DG (continued on page 45)

Symbol	Definition	Type	S	Duration
d0	DTMF 0	Signal	BR	
d1	DTMF 1	Signal	BR	
d2	DTMF 2	Signal	BR	
d3	DTMF 3	Signal	BR	
d4	DTMF 4	Signal	BR	
d5	DTMF 5	Signal	BR	
d6	DTMF 6	Signal	BR	
d7	DTMF 7	Signal	BR	
d8	DTMF 8	Signal	BR	
d9	DTMF 9	Signal	BR	
ds	DTMF *	Signal	BR	
do	DTMF #	Signal	BR	
da	DTMF A	Signal	BR	
db	DTMF B	Signal	BR	
dc	DTMF C	Signal	BR	
dd	DTMF D	Signal	BR	

4.2.4.6 DTMF Detection Package - DD (Extends ToneDet)

Table 4-7: DTMF Detection Package - DD (continued on page 46)

Symbol	Definition	Type
ce	DigitMap Completion Event	Event
d0	DTMF 0	Event
d1	DTMF 1	Event
d2	DTMF 2	Event
d3	DTMF 3	Event
d4	DTMF 4	Event
d5	DTMF 5	Event
d6	DTMF 6	Event
d7	DTMF 7	Event
d8	DTMF 8	Event
d9	DTMF 9	Event
ds	DTMF *	Event
do	DTMF #	Event
da	DTMF A	Event
db	DTMF B	Event
dc	DTMF C	Event
dd	DTMF D	Event

4.2.4.7 Call Progress Tones Generator Package - CG (Extends ToneGen)

Table 4-8: Call Progress Tones Generator Package - CG

Symbol	Definition	Type	S	Duration	Map to CPT File
dt	Dial tone	Signal	TO	180 sec	1
rt	Ringing tone	Signal	TO	180 sec	2
bt	Busy tone	Signal	TO	180 sec	3
ct	Congestion tone	Signal	TO	180 sec	4
sit	Special Information Tone	Signal	BR	2 sec	5
wt	Warning Tone	Signal	TO	180 sec	6
pt	Payphone Recognition Tone	Signal			Not supported
cw	Call Waiting Tone	Signal	BR	1 sec	9
cr	Caller Waiting Tone	Signal	TO	180 sec	15

4.2.4.8 Call Progress Tones Detection Package – CD (Extends ToneDet)

Table 4-9: Call Progress Tones Detection Package - CD

Symbol	Definition	Type
dt	Dial tone	Event
rt	Ringing tone	Event
bt	Busy tone	Event
ct	Congestion tone	Event
sit	Special Information Tone	Event
wt	Warning Tone	Event
pt	Payphone Recognition Tone	Event
cw	Call Waiting Tone	Event
cr	Caller Waiting Tone	Event

4.2.4.9 Basic Continuity Package - CT

Table 4-10: Basic Continuity Package - CT

Symbol	Definition	Type	S	Duration	Map to CPT file
cmp	Detects test completion	Event			
ct	Initiates sending the tone	Signal	TO	2 sec	UserDefined CO1
rsp	Responds to continuity test	Signal	TO	2 sec	

4.2.4.10 Network Package - NT

Table 4-11: Network Package - NT

Symbol	Definition	Type
jit	Maximal jitter buffer size	Property
netfail	Network failure	Event
qualert	Quality alert	Event
dur	Termination's InContext duration	Statistics
os	Octets sent	Statistics
or	Octets received	Statistics

4.2.4.11 RTP Package - RTP (Extends - NT)

Table 4-12: RTP Package - RTP

Symbol	Definition	Type
pltran	Payload Transition	Event
ps	Packets sent	Statistics
pr	Packets received	Statistics
pl	Packet loss	Statistics
jit	Current inter-arrival jitter value	Statistics
delay	Current packets propagation delay	Statistics

4.2.4.12 TDM Circuit Package – TDMC (Extends - NT)

Table 4-13: TDM Circuit Package - TDMC

Symbol	Definition	Type
ec	Maximal Jitter Buffer size	Property
gain	Gain control	Property

4.2.4.13 Generic Announcement Package - AN

Table 4-14: Generic Announcement Package

Symbol	Definition	Type	Supported Parameters
apf	Initiates the play of a fixed announcement	Signal	An – Announcement number Av – Used as a barge-in value Noc – Number of cycles
apv	Initiates the play of a variable announcement	Signal	Handled in the manner as apf

4.2.4.14 Expanded Call Progress Tones Generator Package - XCG (Extends - ToneGen)

Table 4-15: Expanded Call Progress Tones Generator Package - XCG

Symbol	Definition	Type	S	Duration	Map to CPT File
cmft	Comfort tone	Signal	TO	180 sec	18
roh	Off-hook warning tone	Signal	TO	180 sec	16
nack	Negative Acknowledgement	Signal	TO	180 sec	19
vac	Vacant Number Tone	Signal	TO	180 sec	20
spec	Special Conditions Dial Tone	Signal	TO	180 sec	21

4.2.4.15 Basic Service Tones Generation Package - SRVTN (Extends - ToneGen)

Table 4-16: Basic Service Tones Generation Package - SRVTN

Symbol	Definition	Type	S	Duration	Map to CPT File
rdt	Recall Dial Tone	Signal	TO	180 sec	22
conf	Confirmation Tone	Signal	BR	1 sec	8
ht	Held Tone	Signal	TO	180 sec	23
mwt	Message Waiting Tone	Signal	TO	180 sec	17

4.2.4.16 Expanded Services Tones Generation Package - XSRVTN (Extends - ToneGen)

Table 4-17: Expanded Services Tones Generation Package - XSRVTN

Symbol	Definition	Type	S	Duration	Map to CPT File
xferdt	Call Transfer Dial Tone	Signal	TO	180 sec	24
cft	Call Forward Tone	Signal	BR	1 sec	25
ccst	Credit Card Service Tone	Signal	BR	1 sec	26
srdt	Special Recall Dial Tone	Signal	TO	180 sec	27

4.2.4.17 Basic CAS Package - CAS

Table 4-18: Basic CAS Signal/Events

Symbol	Definition	Type	S	Duration	Map to CPT File	Note
sz	Seizure	Signal/ Event	BR		None	
ans	Answer	Signal/ Event	BR		None	
cf	Clear forward	Signal/ Event	BR		None	
cb	Clear back	Signal/ Event	BR		None	
addr	Address	Signal	BR		None	
casf	CAS failure	Event	-		None	

4.2.4.18 R2 CAS Package – R2

Table 4-19: R2 CAS Signal/Events Table

Symbol	Definition	Type	S	Duration	Map to CPT file	Note
ublk	Unblock	Signal/ Event	BR		None	
r2addr	R2 address	Signal/ Event	BR		None	
sls	Line status	Signal/ Event	BR		None	
r2f	R2 failure	Event	BR		None	
blk	Block	Signal	BR		None	

4.2.4.19 MF Generator Package - MFG (Extends - ToneGen)

Table 4-20: MF Generator Package - MFG (continues on page 51)

Symbol	Definition	Type	S	Duration
mf0	MF 0	Signal	BR	
mf1	MF 1	Signal	BR	
mf2	MF 2	Signal	BR	
mf3	MF 3	Signal	BR	
mf4	MF 4	Signal	BR	
mf5	MF 5	Signal	BR	
mf6	MF 6	Signal	BR	
mf7	MF 7	Signal	BR	
mf8	MF 8	Signal	BR	
mf9	MF 9	Signal	BR	

Table 4-20: MF Generator Package - MFG (continues on page 51)

Symbol	Definition	Type	S	Duration
mfa	MF A	Signal	BR	
mfb	MF B	Signal	BR	
mfc	MF C	Signal	BR	
mfd	MF D	Signal	BR	
mfe	MF E	Signal	BR	
mff	MF F	Signal	BR	
mfg	MF G	Signal	BR	
mfh	MF H	Signal	BR	

4.2.4.20 MF Detection Package - MFD (Extends - ToneDet)

Table 4-21: MF Generator Package - MFG

Symbol	Definition	Type
mf0	MF 0	Event
mf 1	MF 1	Event
mf 2	MF 2	Event
mf 3	MF 3	Event
mf 4	MF 4	Event
mf 5	MF 5	Event
mf 6	MF 6	Event
mf 7	MF 7	Event
mf 8	MF 8	Event
mf 9	MF 9	Event
mfa	MF A	Event
mfb	MF B	Event
mfc	MF C	Event
mfd	MF D	Event
mfe	MF E	Event
mff	MF F	Event
mfg	MF G	Event
mfh	MF H	Event

4.2.5 MEGACO Terminations Naming

The basic entities controlled by MEGACO protocol are called Terminations. Physical Terminations represent a physical entity and ephemeral Terminations represent the stream. Ephemeral Terminations exist only during a connection.

Physical Termination names have up to three components: The gateway (in this case, the TrunkPack board) name, the trunk name and the Endpoint name (for non-trunking gateways, the trunk field doesn't exist).

Ephemeral Termination names have two components: The gateway name and a constant string – 'RTP/' for RTP terminations and 'ATM/' for ATM terminations. So assuming that the gateway name is 'gw', if the first ephemeral Termination is of RTP type, it is called 'gwRTP/1', and if it is of ATM type, it is called 'gwATM/1'.

Set the name parts using the following INI file parameters (the last two are used only for physical Terminations):

'GATEWAYNAME', 'TRUNKNAME', 'ENDPOINTNAME'.

Note that the '/' (the forward slash) should be part of the name used. It is not added automatically.

The default values for the Termination name parts depends on the board type:

- For trunking boards, the default gateway name is 'tgw/', the default trunk name is 's' and the default Endpoint name is '/c'. So the Termination that represents bchannel 1 of trunk 0 will be 'tgw/s0/c1'.

PSTN Interface - mapping Trunk/Bchannel pairs to Endpoints is hardware-specific (refer to Table 4-22.) Note that the number of supported terminations per board is equal to the channel density of the board.

Table 4-22 describes the mapping between Endpoints and channels for a PSTN interface, according to the interface type used. Table 4-22 assumes the following initial values:

Channel density – 60

Gateway name = 'Acgw'

Trunk name = 'T'

Endpoint name = '/C',

Table 4-22 lists only the names for a two trunks unit.

Table 4-22: MEGACO Endpoint Names (continues on page 53 to page 54)

Endpoint Name	E1 - PRI/CAS E1 - Transparent	E1 - Transpar- ent 62	T1 - PRI	T1 – CAS T1- Transparent
Acgw/T0/C1	Trunk#0/TS1	Trunk#0/TS1	Trunk#0/TS1	Trunk#0/TS1
Acgw/T0/C2	Trunk#0/TS2	Trunk#0/TS2	Trunk#0/TS2	Trunk#0/TS2
Acgw/T0/C3	Trunk#0/TS3	Trunk#0/TS3	Trunk#0/TS3	Trunk#0/TS3
Acgw/T0/C4	Trunk#0/TS4	Trunk#0/TS4	Trunk#0/TS4	Trunk#0/TS4
Acgw/T0/C5	Trunk#0/TS5	Trunk#0/TS5	Trunk#0/TS5	Trunk#0/TS5
Acgw/T0/C6	Trunk#0/TS6	Trunk#0/TS6	Trunk#0/TS6	Trunk#0/TS6

Table 4-22: MEGACO Endpoint Names (continues on page 53 to page 54)

Endpoint Name	E1 - PRI/CAS E1 - Transparent	E1 - Transpar- ent 62	T1 - PRI	T1 – CAS T1- Transparent
Acgw/T0/C7	Trunk#0/TS7	Trunk#0/TS7	Trunk#0/TS7	Trunk#0/TS7
Acgw/T0/C8	Trunk#0/TS8	Trunk#0/TS8	Trunk#0/TS8	Trunk#0/TS8
Acgw/T0/C9	Trunk#0/TS9	Trunk#0/TS9	Trunk#0/TS9	Trunk#0/TS9
Acgw/T0/C10	Trunk#0/TS10	Trunk#0/TS10	Trunk#0/TS10	Trunk#0/TS10
Acgw/T0/C11	Trunk#0/TS11	Trunk#0/TS11	Trunk#0/TS11	Trunk#0/TS11
Acgw/T0/C12	Trunk#0/TS12	Trunk#0/TS12	Trunk#0/TS12	Trunk#0/TS12
Acgw/T0/C13	Trunk#0/TS13	Trunk#0/TS13	Trunk#0/TS13	Trunk#0/TS13
Acgw/T0/C14	Trunk#0/TS14	Trunk#0/TS14	Trunk#0/TS14	Trunk#0/TS14
Acgw/T0/C15	Trunk#0/TS15	Trunk#0/TS15	Trunk#0/TS15	Trunk#0/TS15
Acgw/T0/C16	N/A	Trunk#0/TS16	Trunk#0/TS16	Trunk#0/TS16
Acgw/T0/C17	Trunk#0/TS17	Trunk#0/TS17	Trunk#0/TS17	Trunk#0/TS17
Acgw/T0/C18	Trunk#0/TS18	Trunk#0/TS18	Trunk#0/TS18	Trunk#0/TS18
Acgw/T0/C19	Trunk#0/TS19	Trunk#0/TS19	Trunk#0/TS19	Trunk#0/TS19
Acgw/T0/C20	Trunk#0/TS20	Trunk#0/TS20	Trunk#0/TS20	Trunk#0/TS20
Acgw/T0/C21	Trunk#0/TS21	Trunk#0/TS21	Trunk#0/TS21	Trunk#0/TS21
Acgw/T0/C22	Trunk#0/TS22	Trunk#0/TS22	Trunk#0/TS22	Trunk#0/TS22
Acgw/T0/C23	Trunk#0/TS23	Trunk#0/TS23	Trunk#0/TS23	Trunk#0/TS23
Acgw/T0/C24	Trunk#0/TS24	Trunk#0/TS24	N/A (D-channel)	Trunk#0/TS24
Acgw/T0/C25	Trunk#0/TS25	Trunk#0/TS25		
Acgw/T0/C26	Trunk#0/TS26	Trunk#0/TS26		
Acgw/T0/C27	Trunk#0/TS27	Trunk#0/TS27		
Acgw/T0/C28	Trunk#0/TS28	Trunk#0/TS28		
Acgw/T0/C29	Trunk#0/TS29	Trunk#0/TS29		
Acgw/T0/C30	Trunk#0/TS30	Trunk#0/TS30		
Acgw/T0/C31	Trunk#0/TS31	Trunk#0/TS31		
Acgw/T1/C1	Trunk#1/TS1	Trunk#1/TS1	Trunk#1/TS1	Trunk#1/TS1
Acgw/T1/C3	Trunk#1/Trunk#1/ TS2	Trunk#1/Trunk#1/ TS2	Trunk#1/Trunk#1/ TS2	Trunk#1/Trunk#1/ TS2
Acgw/T1/C3	Trunk#1/TS3	Trunk#1/TS3	Trunk#1/TS3	Trunk#1/TS3
Acgw/T1/C4	Trunk#1/TS4	Trunk#1/TS4	Trunk#1/TS4	Trunk#1/TS4
Acgw/T1/C5	Trunk#1/TS5	Trunk#1/TS5	Trunk#1/TS5	Trunk#1/TS5
Acgw/T1/C6	Trunk#1/TS6	Trunk#1/TS6	Trunk#1/TS6	Trunk#1/TS6
Acgw/T1/C7	Trunk#1/TS7	Trunk#1/TS7	Trunk#1/TS7	Trunk#1/TS7
Acgw/T1/C8	Trunk#1/TS8	Trunk#1/TS8	Trunk#1/TS8	Trunk#1/TS8
Acgw/T1/C9	Trunk#1/TS9	Trunk#1/TS9	Trunk#1/TS9	Trunk#1/TS9

Table 4-22: MEGACO Endpoint Names (continues on page 53 to page 54)

Endpoint Name	E1 - PRI/CAS E1 - Transparent	E1 - Transpar- ent 62	T1 - PRI	T1 – CAS T1- Transparent
Acgw/T1/C10	Trunk#1/TS10	Trunk#1/TS10	Trunk#1/TS10	Trunk#1/TS10
Acgw/T1/C11	Trunk#1/TS11	Trunk#1/TS11	Trunk#1/TS11	Trunk#1/TS11
Acgw/T1/C12	Trunk#1/TS12	Trunk#1/TS12	Trunk#1/TS12	Trunk#1/TS12
Acgw/T1/C13	Trunk#1/TS13	Trunk#1/TS13	Trunk#1/TS13	Trunk#1/TS13
Acgw/T1/C14	Trunk#1/TS14	Trunk#1/TS14	Trunk#1/TS14	Trunk#1/TS14
Acgw/T1/C15	Trunk#1/TS15	Trunk#1/TS15	Trunk#1/TS15	Trunk#1/TS15
Acgw/T1/C16	N/A	Trunk#1/TS16	Trunk#1/TS16	Trunk#1/TS16
Acgw/T1/C17	Trunk#1/TS17	Trunk#1/TS17	Trunk#1/TS17	Trunk#1/TS17
Acgw/T1/C18	Trunk#1/TS18	Trunk#1/TS18	Trunk#1/TS18	Trunk#1/TS18
Acgw/T1/C19	Trunk#1/TS19	Trunk#1/TS19	Trunk#1/TS19	Trunk#1/TS19
Acgw/T1/C20	Trunk#1/TS20	Trunk#1/TS20	Trunk#1/TS20	Trunk#1/TS20
Acgw/T1/C21	Trunk#1/TS21	Trunk#1/TS21	Trunk#1/TS21	Trunk#1/TS21
Acgw/T1/C22	Trunk#1/TS22	Trunk#1/TS22	Trunk#1/TS22	Trunk#1/TS22
Acgw/T1/C23	Trunk#1/TS23	Trunk#1/TS23	Trunk#1/TS23	Trunk#1/TS23
Acgw/T1/C24	Trunk#1/TS24	Trunk#1/TS24	N/A (D-channel)	Trunk#1/TS24
Acgw/T1/C25	Trunk#1/TS25	Trunk#1/TS25		
Acgw/T1/C26	Trunk#1/TS26	Trunk#1/TS26		
Acgw/T1/C27	Trunk#1/TS27	Trunk#1/TS27		
Acgw/T1/C28	Trunk#1/TS28	Trunk#1/TS28		
Acgw/T1/C29	Trunk#1/TS29	Trunk#1/TS29		
Acgw/T1/C30	Trunk#1/TS30			
Acgw/T1/C31	Trunk#1/TS31			

5 Device Management

5.1 SNMP Management

5.1.1 SNMP Overview

SNMP (Simple Network Control Protocol) is a standard network-based client/server based control protocol to manage devices in the Network. The client program (called the network manager) makes connections to a server program, called the SNMP agent. The SNMP agent, embedded on a remote network device, serves information to the manager regarding the device's status. The database used by the Agent to retrieve information, is referred to as the SNMP Management Information Base (MIB), and is a standard set of statistical and control values. Apart from the standard MIBs documented in IETF's RFC documents, SNMP additionally allows the usage of private MIBs, containing non-standard information set.

Directives, issued by the network manager client 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 the identifier, or set the identifier to a new value.

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 network management client programs so that they can become aware of MIB variables and their usage.

The TrunkPack-VoP Series contains an embedded SNMP agent supporting both general network MIBs (such as the IP MIB), VoP-specific MIBs (such as RTP, MGCP, Trunk, etc.) and a proprietary MIB (known also as ACL MIB) enabling a deeper probe into the interworking of the board/module. All the supported MIBs files are supplied as part of the release.

5.1.2 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 an operant device supports. 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 until an error is encountered (indicating that all MIB object names have been "walked").

- **Set Request:** The SNMP standard provides a method of effecting an action associated with 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 by which devices can "reach out" to a network manager on their own (via the "trap" message) to notify 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.

5.1.3 SNMP MIB Objects

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

- The "mgmt" SNMP branch contains the standard SNMP objects usually supported (at least in part) by all network devices.
- The "private" SNMP branch contains those "extended" SNMP objects defined by network equipment vendors.
- The "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. In the case of "Discrete" objects, this instance number is zero. In the case of "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, in 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).

5.1.4 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.

5.1.5 TrunkPack-VoP Series Supported MIBs

The TrunkPack-VoP Series contains an embedded SNMP agent supporting the following MIBs:

- The Standard MIB (MIB-II) - 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.
- RTP MIB - The RTP MIB is supported per the RFC-2959. It contains objects relevant to the RTP streams generated and terminated by the board and to the RTCP information related to these streams.
- Trunk MIB- The Trunk MIB is supported per the RFC-2495. It contains objects relevant to E1/T1 Trunk interfaces.

In addition to the standard MIBs, the TrunkPack-VoP Series contains several AudioCodes proprietary MIBs:

- ACBoard MIB - This proprietary MIB contains objects related both to the configuration of the board and channels as well as to run-time information. Through this MIB, the User can set up the board configuration parameters, reset the board, and monitor the board's operational robustness and quality of service during run-time and receive TRAP events.
- ACH323 MIB - This proprietary MIB contains an objects related H.323 protocol engine embedded in the TrunkPack. Through this MIB, Users can set up AudioCodes' specific H.323 configuration parameters.
- The ACBoard MIB has the following Groups:
 - boardConfiguration
 - boardInformation
 - channelConfiguration
 - channelStatus
 - notifications
 - reset

The **ACBoard MIB** groups are detailed below:

- BoardConfiguration:

The User can "Set" values to the board Configuration parameters but the new values take effect after the next board remote Reset. There is an exception for the above rule, regarding the "Setting" of the DefaultChannelSettings sub-group parameters. For this sub-group, the updated values are in effect upon the next channel opening (i.e., no need to perform remote Reset). The User can retrieve a Board Configuration Parameter by using "Get" from boardConfiguration group. If the retrieved parameter was modified and a Reset was not performed, the User gets the updated value, although it may still not be in-effect.

- BoardInformation:

Contains Read-only parameters such as: board name, serial number, number of channels, Number of DSPs, CPU speed, *ini* file version number, software version and date.

- **ChannelConfiguration:**
Read-only Parameters to see the configuration of each channel: voice settings, fax settings, etc.
- **ChannelStatus:**
Read-only parameters about the status of channel, such as: is the channel active, is it in a Silence Period, is it in Fax mode, ON/OFF hook, etc.

This group also contains counters: packets to DSP, errors, lost packets, underruns, etc.
- **Notifications:**
For the ACBoard MIB, the following notifications are available:
acEvBoardStarted
acEvResettingBoard

The notifications are sent to the manager that is defined in the SNMPManagerIP parameter in the *ini* file .
- **Reset:**
This “group” contains the remoteReset parameter:

Setting an integer value to this parameter uses a remote board reset. On remote reset invocation, the board sends the acEvResettingBoard Trap.

Note: The current (updated) BoardConfiguration parameters are programmed into the board, provided that the User does not use an *ini* file after reset, that overrides the updated parameters.

Additional MIBs are to be supported in future releases.

5.2 Embedded Web Server

The **MediaPack Series** contains an Embedded Web Server to be used both for device configuration and for run-time monitoring.

AudioCodes' Embedded Web Server enables Users equipped with no more than a standard web browser to:

- 1 Provision all parameters (refer to Figure 5-4: “Advanced Configuration Parameters” on page 63).
- 2 Verify configuration changes in the Status screens (refer to Figure 5-13: “Trunk and Channel Status Screen” on page 73).
- 3 Download the CMP, INI, Voice Prompt, CPT and CAS Files (refer to Section 5.7.2 on page 76)

Users can employ this facility to set up the **TP-1610** configuration parameters needed to configure the device. Users also have the option to reset the device in order to apply the new set of parameters.

Access to the embedded Web server is controlled by protection and security mechanisms described below.

5.2.1 Password Control

The Embedded Web Server is protected by a unique username-password combination. The first time that Users access the Web interface, they're prompted to provide their username-password to obtain access. Subsequent requests are negotiated by the browser on behalf of the User, so that the User doesn't have to re-enter the username-password for each request, but the request is still authenticated.

5.2.1.1 The Embedded Web Server Username Password

The default username-password for all devices is:

- Username = "Admin"
- Password = "Admin"

The User can change the Web password and Web User name from the "Advanced Configuration -> Change Password" page.

The password and Web User name can be 7 characters long at most. The new password is active only after restarting the device using the reset button of the Embedded Web Server. Otherwise, the "old" password is still active.

The User can reset the Web password (to the default values) using an *ini* file parameter called "RESETWEBPASSWORD" using a BootP TFTP configuration. The Web password will automatically return to the default password.

5.2.1.2 Authentication Method

The Embedded Web Server uses the MD5 authentication method that the HTTP 1.1 protocol supports. MD5 authentication takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. It cannot produce two messages having the same message digest, or produce any message having a given pre-specified target message digest.

Most browsers support this method, which provides some protection in that another browser User cannot access the secured pages unless the User provides the username-password. The username-password is encoded so that it cannot be monitored using a packet sniffer.

5.2.2 Web Configuration

Initially, the Embedded Web Server displays the default parameters that are pre-installed in the board. Users can modify parameters using the Embedded Web Server. In addition, the Embedded Web Server can be used to download *ini* file parameters.

5.2.2.1 Read-only Mode

The Embedded Web server can be initialized to "read-only mode" by setting the "DISABLEWEBCONFIG" *ini* file parameter (the default state is read-write mode).

In this mode, all the Web pages are presented in read-only mode. By selecting this mode, the User disables the capability to modify the configuration data. In addition, the User does not have access to the "Change Password" page and to the reset page. When the device is controlled through PCI, the embedded Web server is always in read-only mode.

5.2.2.2 Disable/Enable Embedded Web Server

If Users wants to deny access to the device through HTTP protocol, the embedded Web server task can be disabled. To disable the Web task, use the *ini* file parameter called "DISABLEWEBTASK". The default sets the Web task as enabled.

When the device is controlled through PCI, the Embedded Web Server is always activated. The User cannot disable the task in PCI mode.

5.3 Accessing the Embedded Web Server

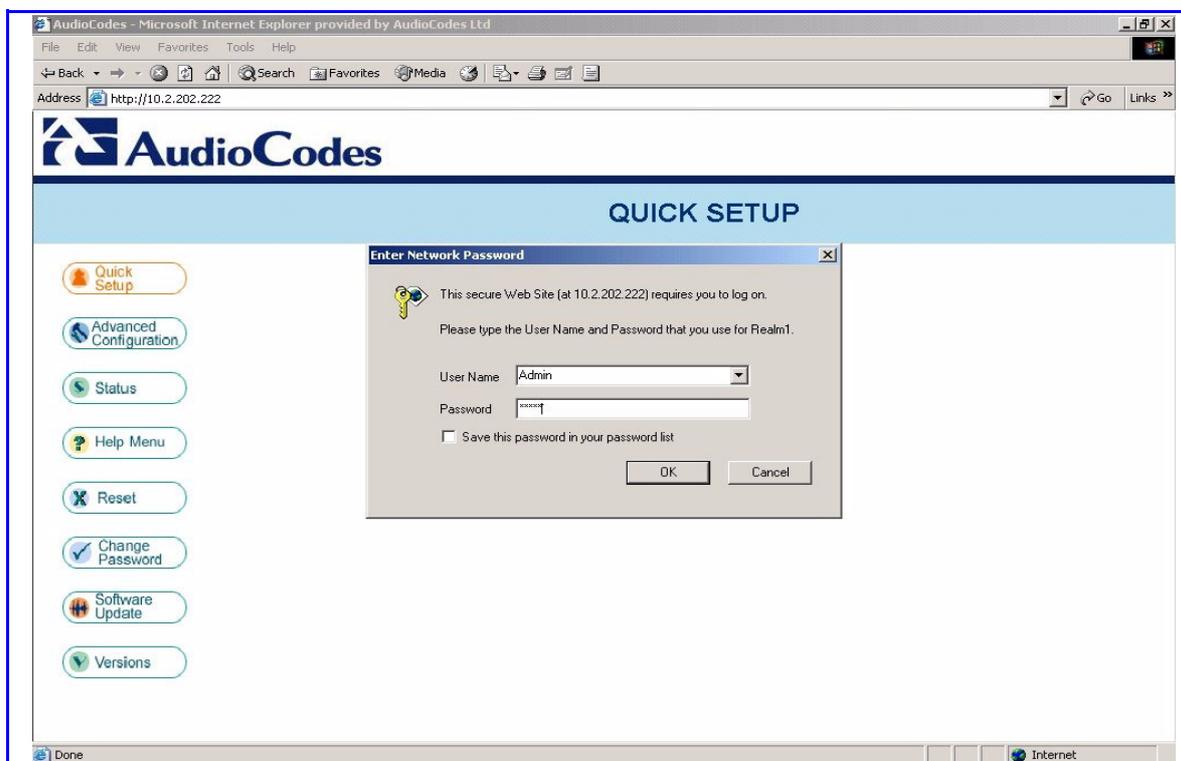
➤ To access the Embedded Web Server:

- Step 1.** Open any standard Web-browsing application such as Microsoft™ Internet Explorer™ (Ver. 5.0 and higher) or Netscape™ Navigator™ (Ver. 7.0 and higher).

Note: Some java-script applications are not supported in Netscape.

- Step 2.** Specify the IP address of the device in the 'Address' field (e.g., http://10.1.229.17); the Embedded Web Server 'Enter Network Password' screen appears.

Figure 5-1: Enter Network Password screen



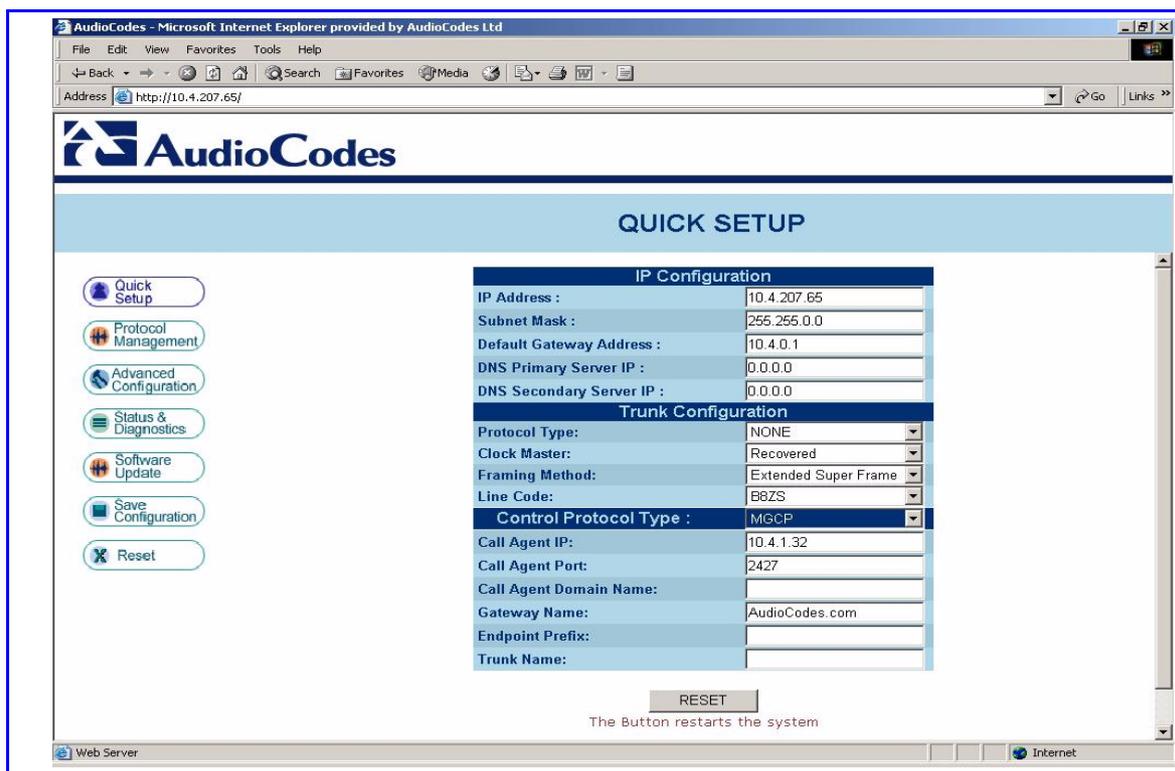
➤ To do a Quick Setup, take the following 4 steps:

- Step 1.** Enter the User Name and Password (default: Admin, Admin).

Note: The User Name and Password fields are case-sensitive.

- Step 2.** Click **OK**. The 'Quick Setup' screen is activated.

Figure 5-2: Quick Setup Screen



- Step 3.** In the Quick Setup screen, you can enter or modify appropriate information for the IP Configuration, Trunk Configuration and Control Protocol (per type).

For IP Configuration, enter appropriate addresses for IP Address, Subnet Mask, Default Gateway Address, DNS Primary Server IP and DNS Secondary Server IP.

For Trunk Configuration, select the appropriate option from dropdown menus for Protocol Type, Clock Master, Framing Method and Line Code.

For Control Protocol Type, select the appropriate option for the Protocol type and enter appropriate information for the Call Agent IP, Call Agent Port, Call Agent Domain Name, Gateway Name, Endpoint Prefix and Trunk Name.

- Step 4.** At the bottom of the screen, click the **Reset** button. A dialog box appears in which you confirm the reset action. The new information is added to the system configuration while the system is restarted. A message informing you of the waiting period appears.

5.4 Protocol Management

The Protocol Management screen offers access to the Protocol Definition screen in which User define the control protocol type.

➤ To access the Protocol Management menu, take this step:

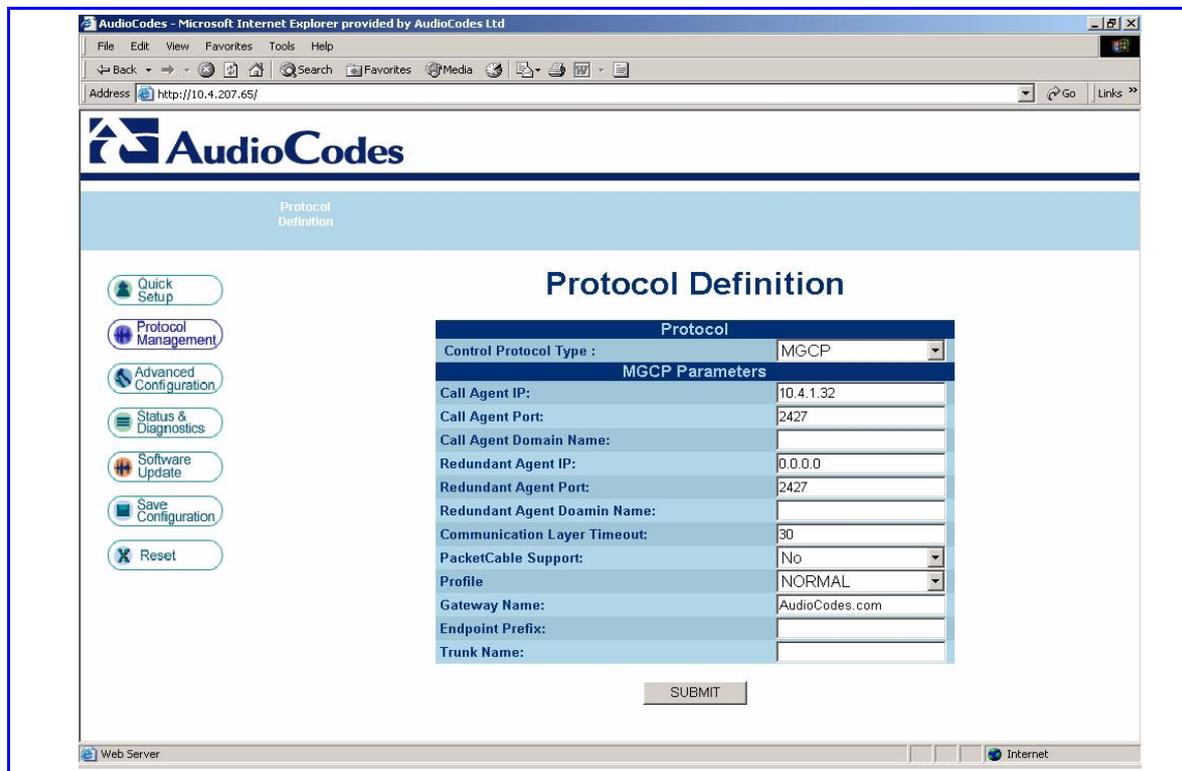
- From the main menu list on the left, click on the Protocol Management button. The Protocol Management screen with the sub-menu bar on the top is displayed.

5.4.1 Protocol Definition

➤ **To define the control protocol type, take these 3 steps:**

Step 1. On the sub-menu bar on the top, click the Protocol Definition option. The Protocol Definition screen appears.

Figure 5-3: Protocol Definition Screen



Step 2. From the Control Protocol Type dropdown list, select the appropriate option. The parameter fields are modified accordingly.

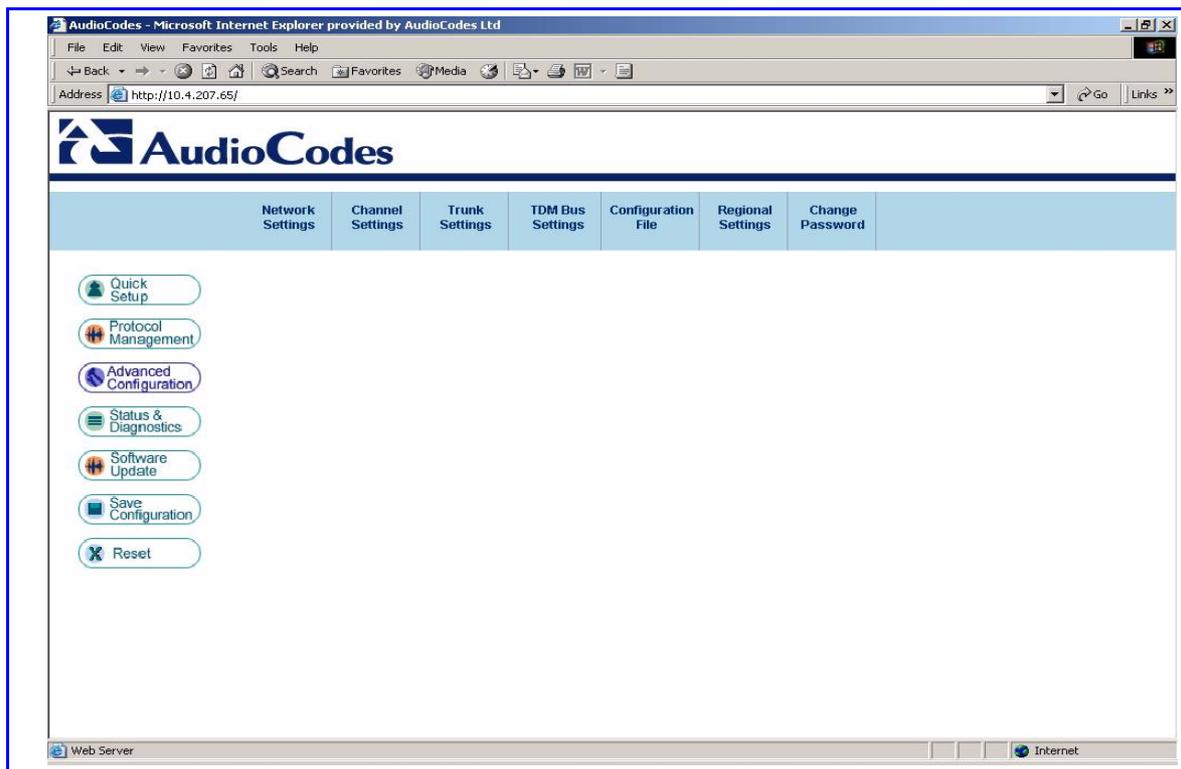
Step 3. Enter the appropriate values to the parameter fields and click the **Submit** button at the bottom. The changes are entered into the system and the screen is refreshed.

5.5 Advanced Configuration Menu

➤ **To access the Advanced Configuration menu, take the following 2 steps:**

Step 1. To access the device's Advanced Configuration parameters, from the main menu list on the left, click on the Advanced Configuration button. The Advanced Configuration Parameters screen with the sub-menu bar on the top is displayed.

Figure 5-4: Advanced Configuration Parameters

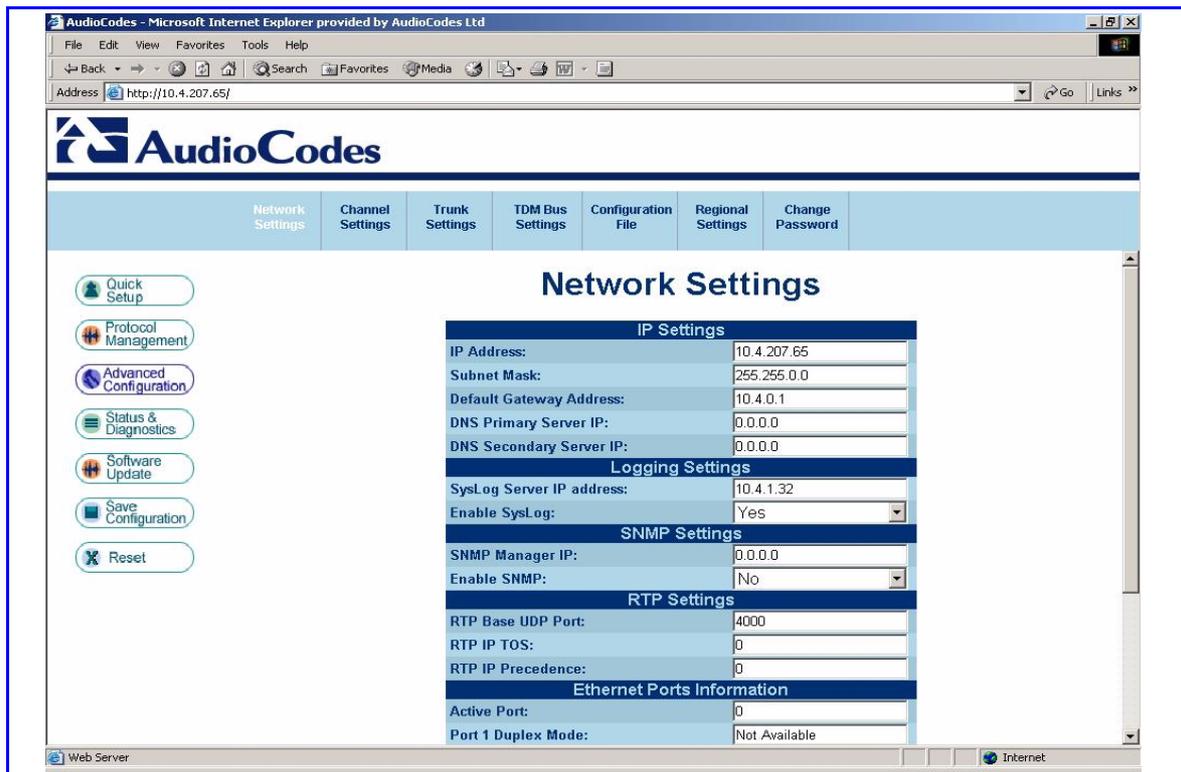


- Step 2.** To display the active configuration of a section and its relevant parameter values, click on an option in the sub-menu bar on the top (Network Settings, Channel Settings, etc.).

5.5.1 Network Settings

➤ **To set Network Settings, take the following 3 steps:**

- Step 1.** On the sub-menu bar on the top, click the Network Settings option. The Network Settings screen appears.

Figure 5-5: Network Settings Screen


Step 2. In the Network Settings screen, you can enter or modify appropriate information for the IP Settings, Logging Settings, SNMP Settings, RTP Settings and Ethernet Ports Information.

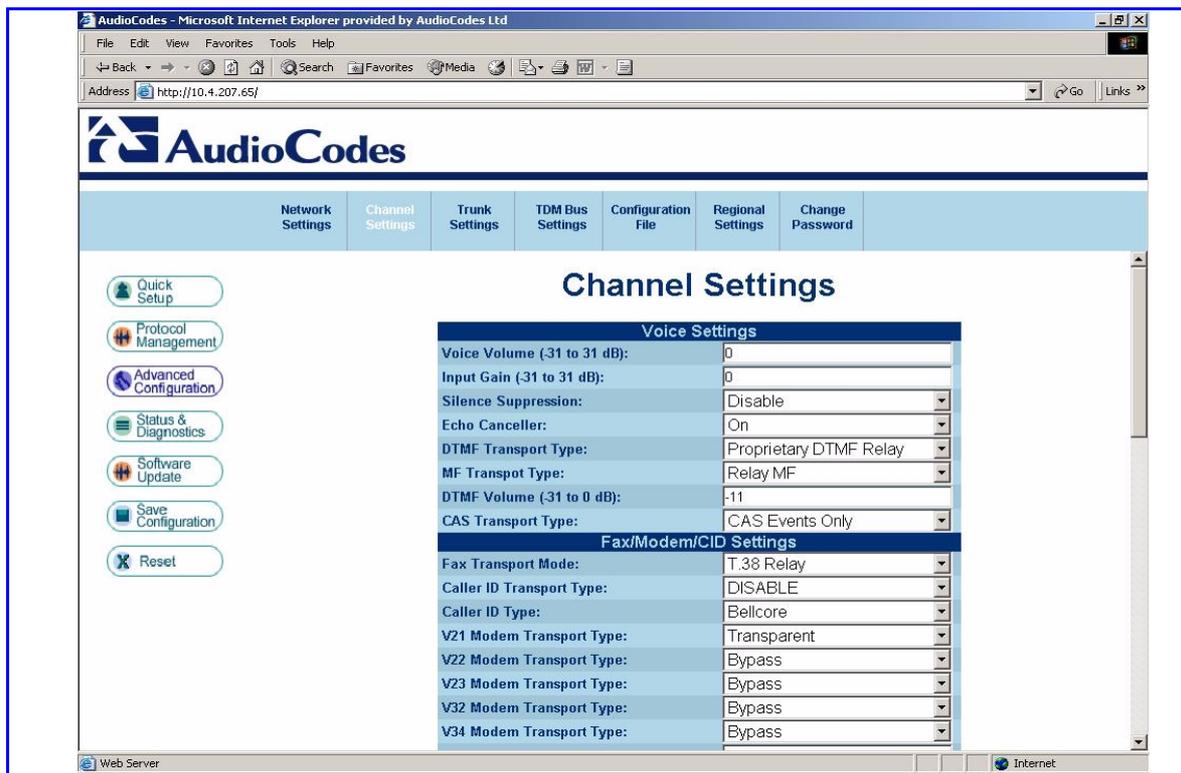
Step 3. At the bottom of the screen, click the **Submit** button. The screen is refreshed.

5.5.2 Channel Settings

➤ **To set Channel Settings, take the following 3 steps:**

Step 1. On the sub-menu bar on the top, click the Channel Settings option. The Channel Settings screen appears.

Figure 5-6: Channel Settings Screen



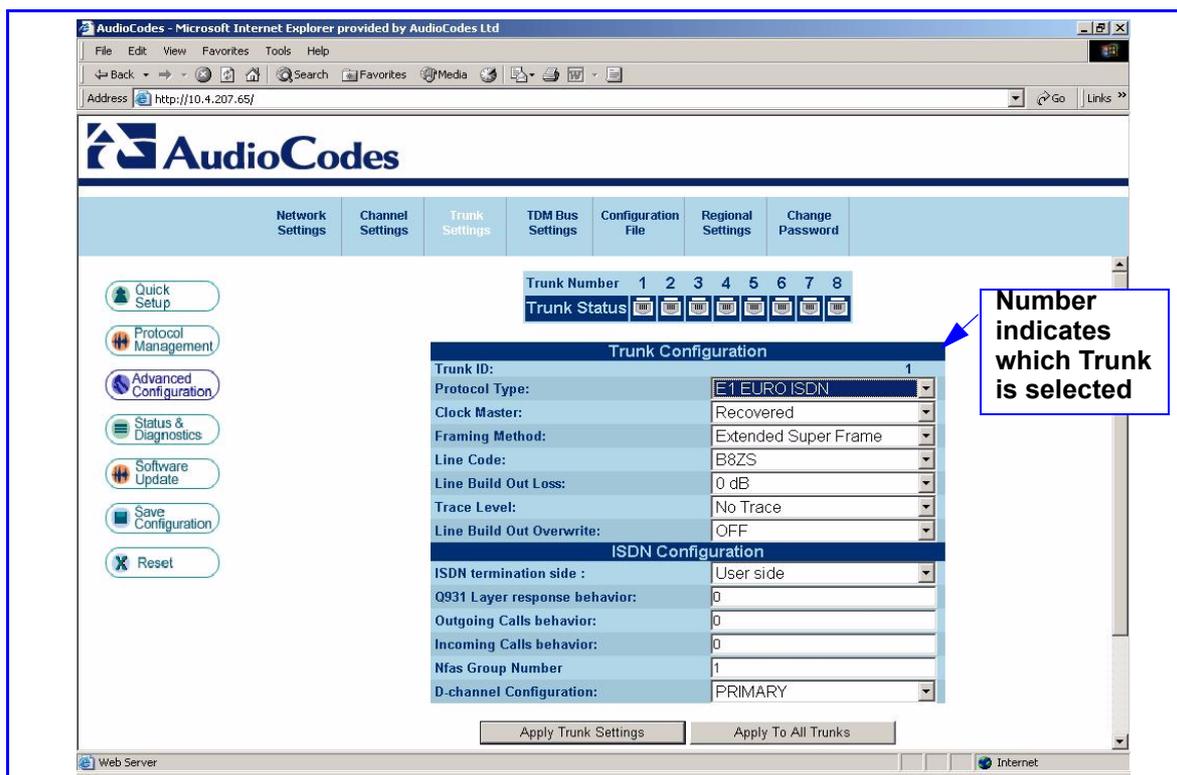
- Step 2.** In the Channel Settings screen, you can enter or modify appropriate information for the Voice Settings and Fax/Modem/CID Settings.
- Step 3.** At the bottom of the screen, click the **Submit** button. The screen is refreshed.

5.5.3 Trunk Settings

➤ **To set Trunk Settings, take the following 5 steps:**

- Step 1.** On the sub-menu bar on the top, click the Trunk Settings option. The Trunk Settings screen appears.
- Initially, the screen appears with the parameters read-only (indicated by being greyed) . The **Stop Trunk** button appears at the bottom of the screen.
- Step 2.** To make a change to any of the parameters, click the **Stop Trunk** button to return the screen to a modifiable state. The parameters are no longer greyed.

Figure 5-7: Trunk Settings Screen



Step 3. From the Trunks displayed on the top, select a one by clicking a Trunk Status

indicator  . The number of the Trunk is displayed in the upper-right-hand corner of the Trunk Configuration display. The parameters displayed are for the selected Trunk only.

The Trunk Status indicators can appear colored. Table 5-1 shows the possible indicators and their descriptions.

Table 5-1: Trunk Status Color Indicator Key

Indicator	Color	Description
	White	Disabled
	Green	Active - OK
	Yellow	RAI Alarm
	Red	LOS Alarm
	Blue	AIS Alarm

When modifying the Protocol Type, there are three menu types:

- Transparent
- ISDN
- CAS

The menu is modified to include additional parameters appropriate to the type selected.

Step 4. After modifying any parameters, do one of the following:

- To apply the changes to the selected trunk only, click the **Apply Trunk Settings** button.
- To apply the changes to the all trunks, click the **Apply To All Trunks** button.

The screen is refreshed. Parameters become read-only (indicated by being greyed) . The **Stop Trunk** button appears at the bottom of the screen.

To make a change to any of the parameters, you must click the **Stop Trunk** button to return the screen to a modifiable state.

Step 5. To commit the changes to the Flash memory, in the main menu on the left, click the **Reset** button to go to the **Reset** screen and reset the system.

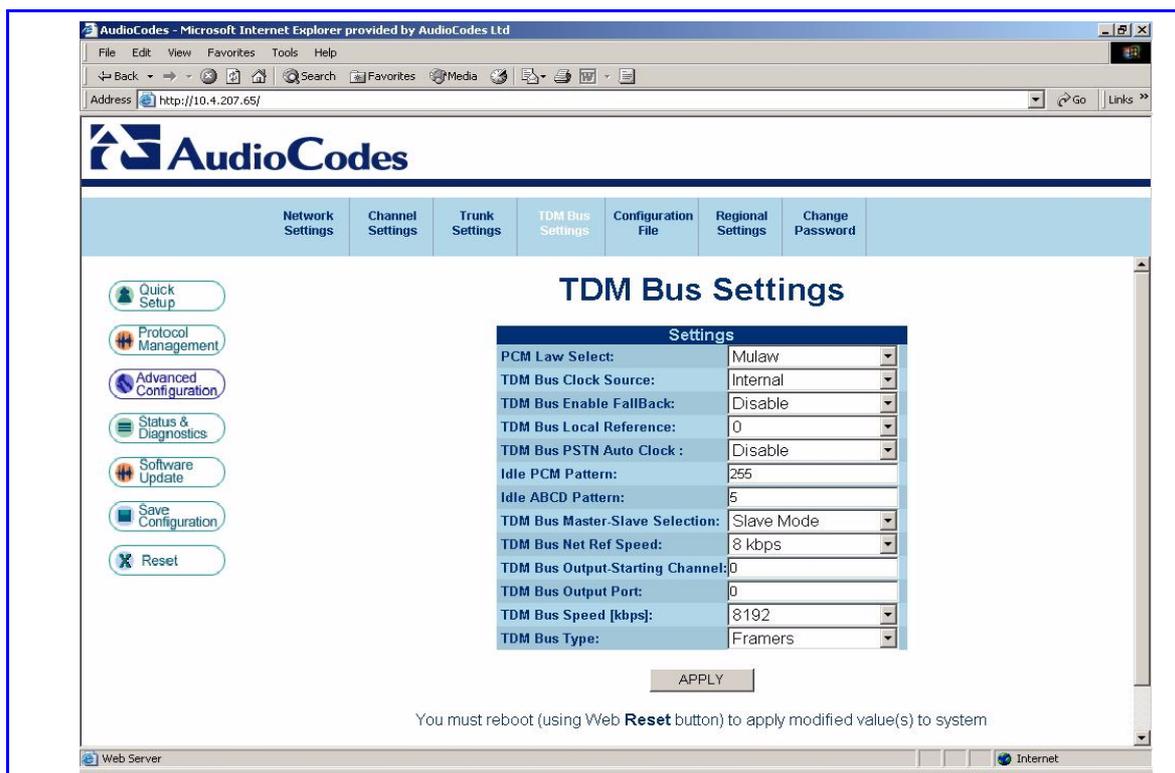
Note 1: Reset can be scheduled for a later time period when call traffic is at a minimum. If you choose to schedule the Reset for a later time, be sure to use the Save Configuration screen (described on page 79) to retain the changes to the configuration on the local PC.

Note 2: If you are modifying multiple screens, perform the reset after you are finished modifying all of the screen you intended and NOT after each screen.

5.5.4 TDM Bus Settings

➤ **To set TDM Bus settings, take the following 3 steps:**

Step 1. On the sub-menu bar on the top, click the TDM Bus Settings option. The TDM Bus Settings screen appears.

Figure 5-8: TDM Bus Settings Screen


Step 2. After modifying any parameters, click the **Apply** button.

The screen is refreshed.

Step 3. To commit the changes to the Flash memory, in the main menu on the left, click the **Reset** button to go to the **Reset** screen and reset the system.

Note 1: Reset can be scheduled for a later time period when call traffic is at a minimum. If you choose to schedule the Reset for a later time, be sure to use the Save Configuration screen (described on page 79) to retain the changes to the configuration on the local PC.

Note 2: If you are modifying multiple screens, perform the reset after you are finished modifying all of the screen you intended and NOT after each screen.

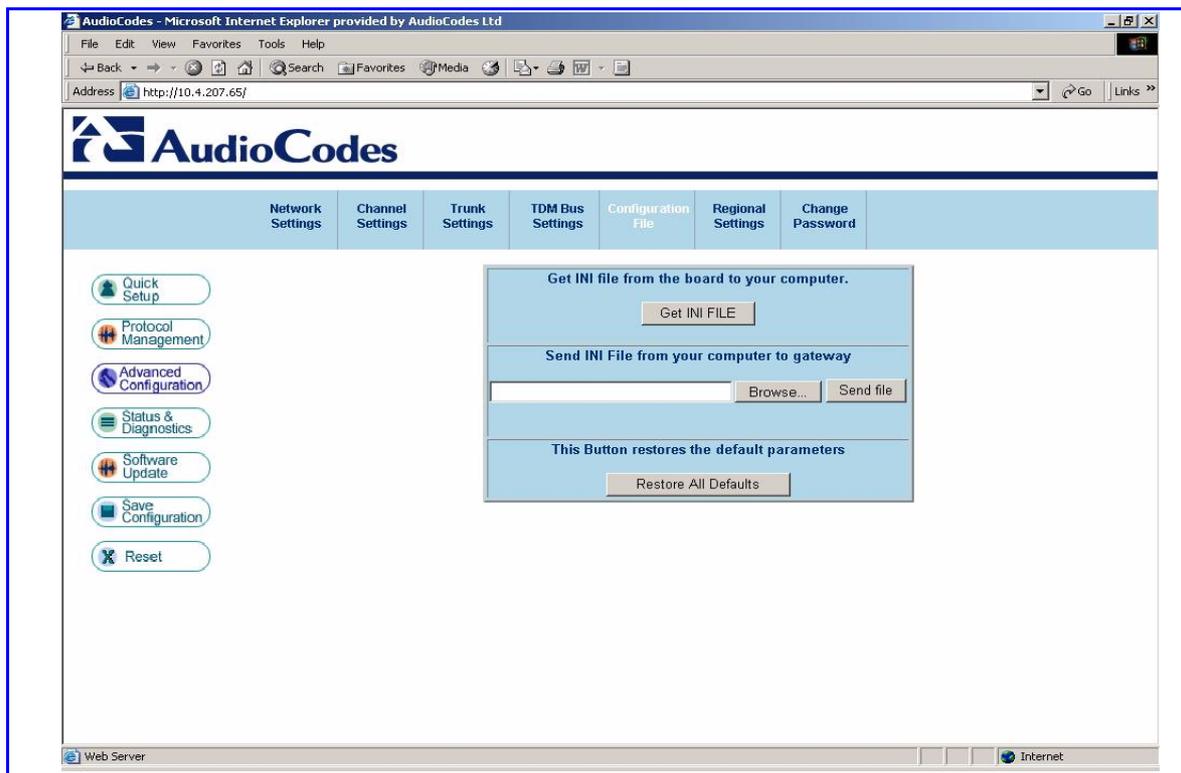
5.5.5 Configuration File

Users can get the *ini* file from the board and modify it using the *ini* parameters listed in Section 5.7.2 on page 76. Users can also send a predefined *ini* file either one supplied by AudioCodes or one you defined using this screen.

➤ **To access the Configuration File screen, take this step:**

- On the sub-menu bar on the top, click the Configuration File option. The Configuration File screen appears.

Figure 5-9: Configuration File Screen

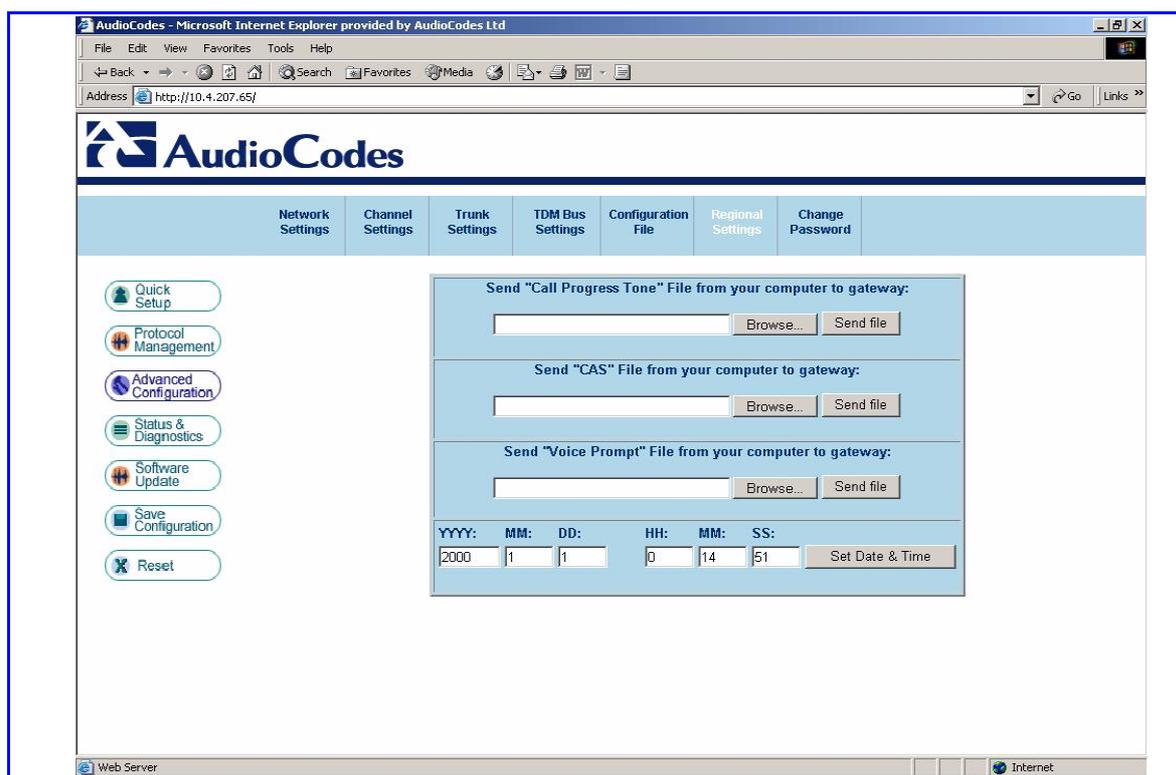


- **To get the board's *ini* file, take this step:**
 - Click on the **Get INI File** button. You are prompted to select a location to which to save it.
- **To send an *ini* file to the board, take this step:**
 - Step 1.** Use the **Browse** button to locate the predefined *ini* file.
 - Step 2.** Click the **Send File** button. The file is sent to the board, overwriting the previous one.

Users can restore default parameters by clicking the **Restore All Defaults** button.

5.5.6 Regional Settings

- **To access the Regional Settings screen, take this step:**
 - On the sub-menu bar on the top, click the Regional Settings option. The Regional Settings screen appears.

Figure 5-10: Regional Settings Screen


➤ **To send a Call Progress Tone, CAS, or Voice Prompt file to the board, take these 2 steps:**

- Step 1.** Use the **Browse** button to locate the predefined **Call Progress Tone**, **CAS**, or **Voice Prompt** file as appropriate.
- Step 2.** Click the **Send File** button. The file is sent to the board, overwriting the previous one. The screen is refreshed and a message informs you about the waiting period.

➤ **To set the date and time, take these 2 steps:**

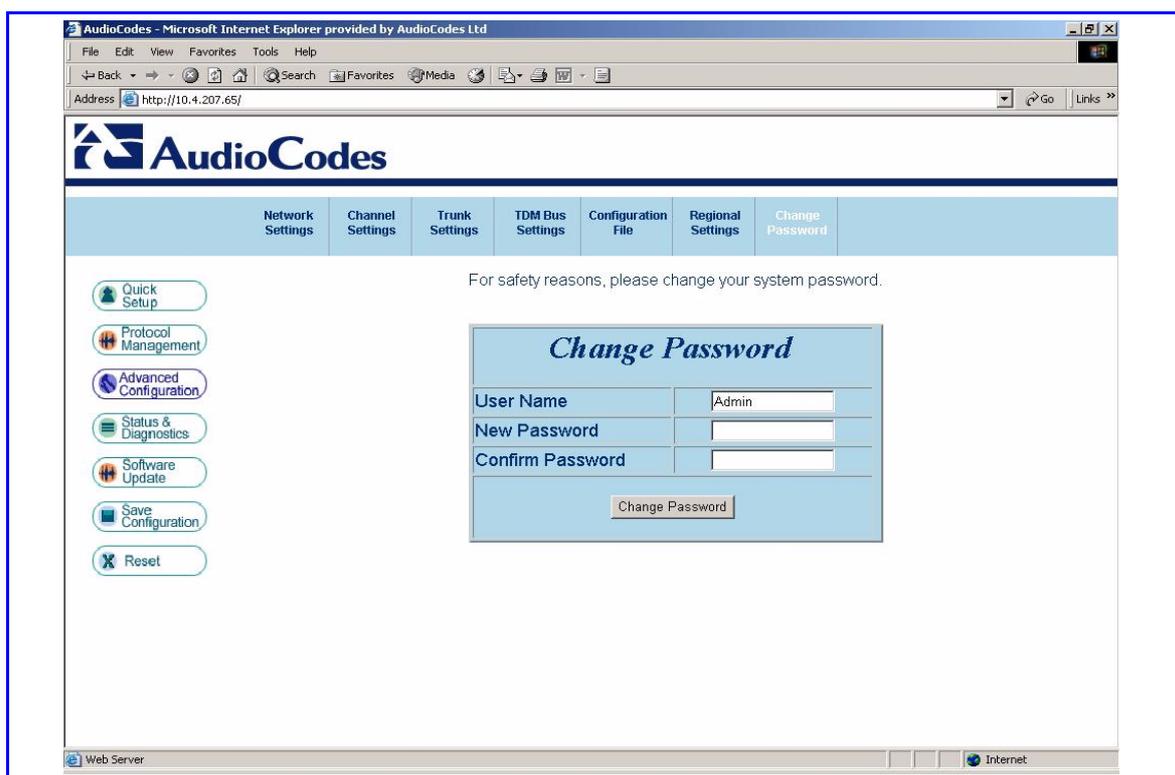
- Step 1.** Enter the date and/or time using the YYYY, MM, DD field for Year, Month and Day and HH, MM, SS fields for Hour, Minutes and Seconds, as appropriate.
- Step 2.** Click the **Set Date & Time** button. The date and time is set on the board, accordingly. The screen is refreshed and a message informs you about the waiting period.

5.5.7 Change Password

➤ **To change the User name and/or password, take these 2 steps:**

- Step 1.** On the sub-menu bar on the top, click the Change Password option. The Change Password screen appears.

Figure 5-11: Change Password Screen

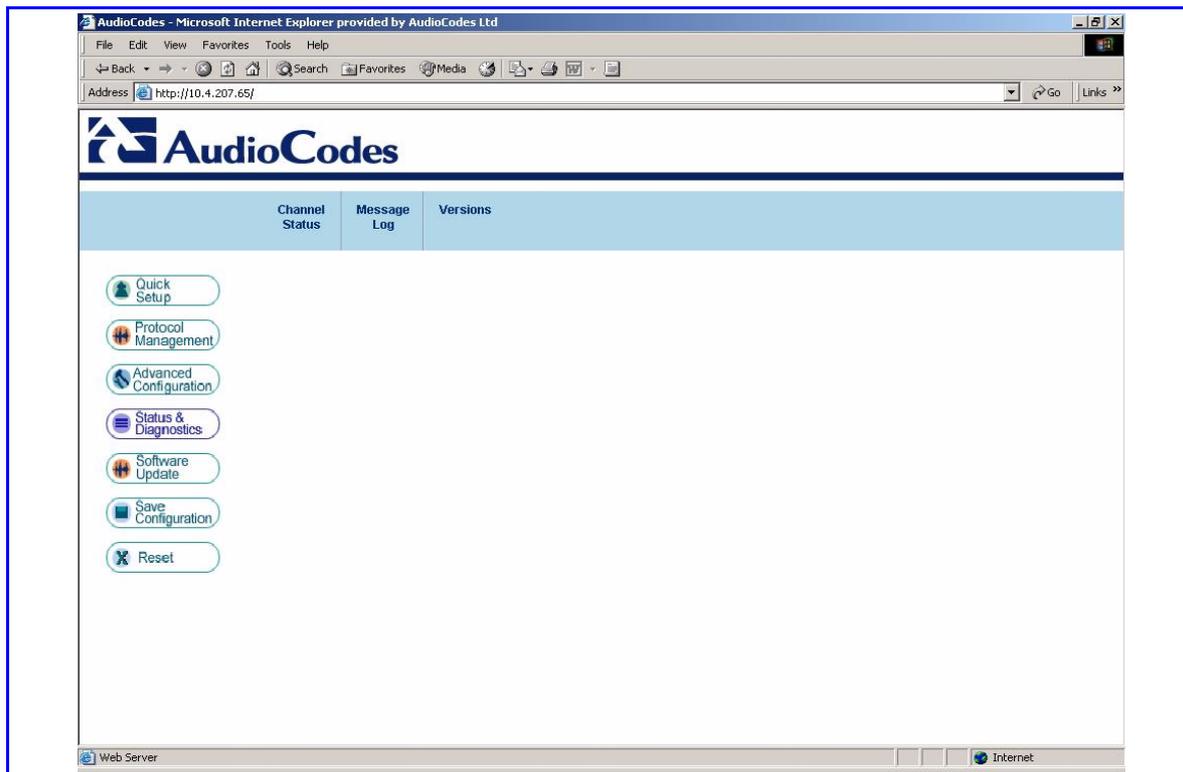


- Step 2.** Enter user name and new password information into the fields and click the **Change Password** button.

5.6 Status and Diagnostic Menu

➤ **To access the Status and Diagnostics menu, Take this step:**

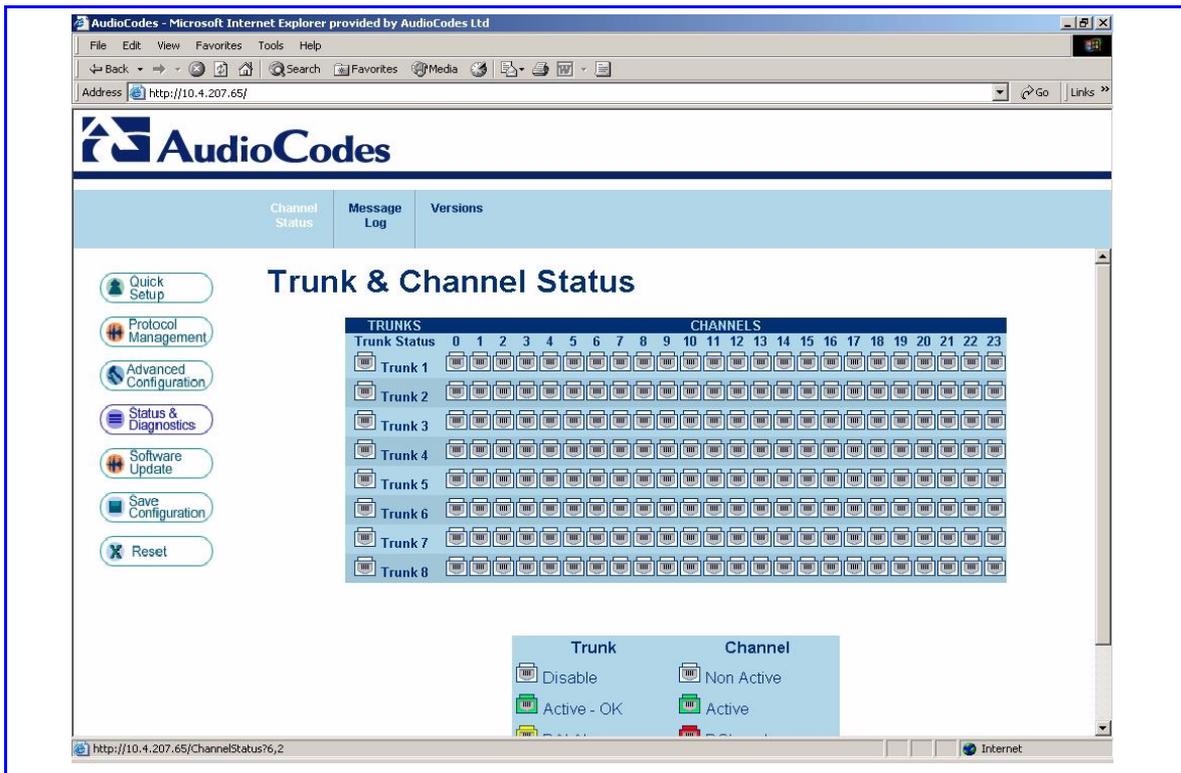
- To access the Status and Diagnostics menu, from the main menu list on the left, click on the Status and Diagnostics button. The Status and Diagnostics screen with the sub-menu bar on the top is displayed.

Figure 5-12: Status and Diagnostic Menu Screen


5.6.1 Trunk and Channel Status

- **To access the Trunk and Channel Status screen, take this step:**
 - On the sub-menu bar on the top, click the Channel Status option. The Trunk and Channel Status screen appears. The screen is Read-only.

Figure 5-13: Trunk and Channel Status Screen



The number of trunks and channels depends of the system configuration. The example in Figure 5-13 depicts a system with 8 spans and 24 T1 links (0 to 23).

The Trunk Status indicators can appear colored. Table 5-2 shows the possible indicators and their descriptions.

Table 5-2: Trunk and Channel Status Color Indicator Key

Trunk			Channel		
Indicator	Color	Description	Indicator	Color	Description
	White	Disabled		White	Non-Active
	Green	Active - OK		Green	Active
	Yellow	RAI Alarm		Red	PCI mode
	Red	LOS Alarm			
	Blue	AIS Alarm			

Step 3. To display a screen with a summary of parameter information relevant to a channel, click on the channel (see Section 5.6.1 on page 72.)

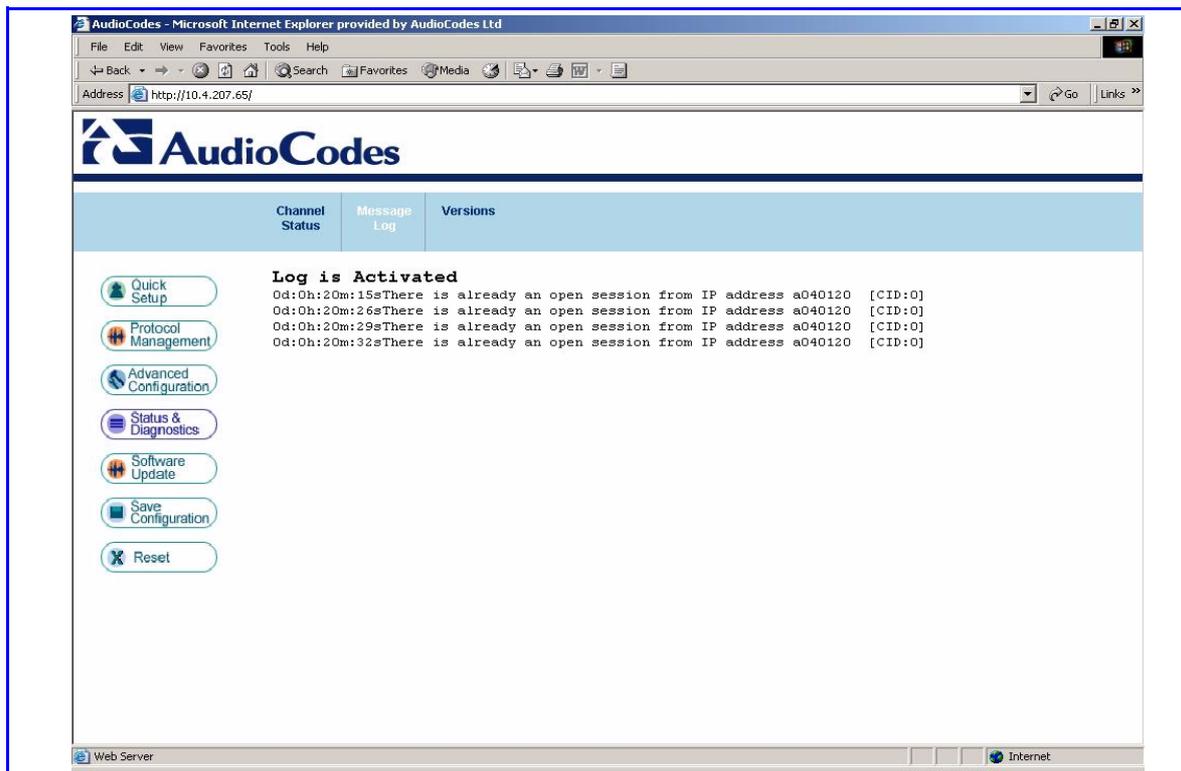
5.6.2 Message Log

The Message Log is and similar to a syslog. It provides debug messages useful in pursuing troubleshooting issues.

➤ To display the Message Log, take this step:

- On the sub-menu bar on the top, click the Message Log option. The Message Log screen appears.

Figure 5-14: Message Log Screen



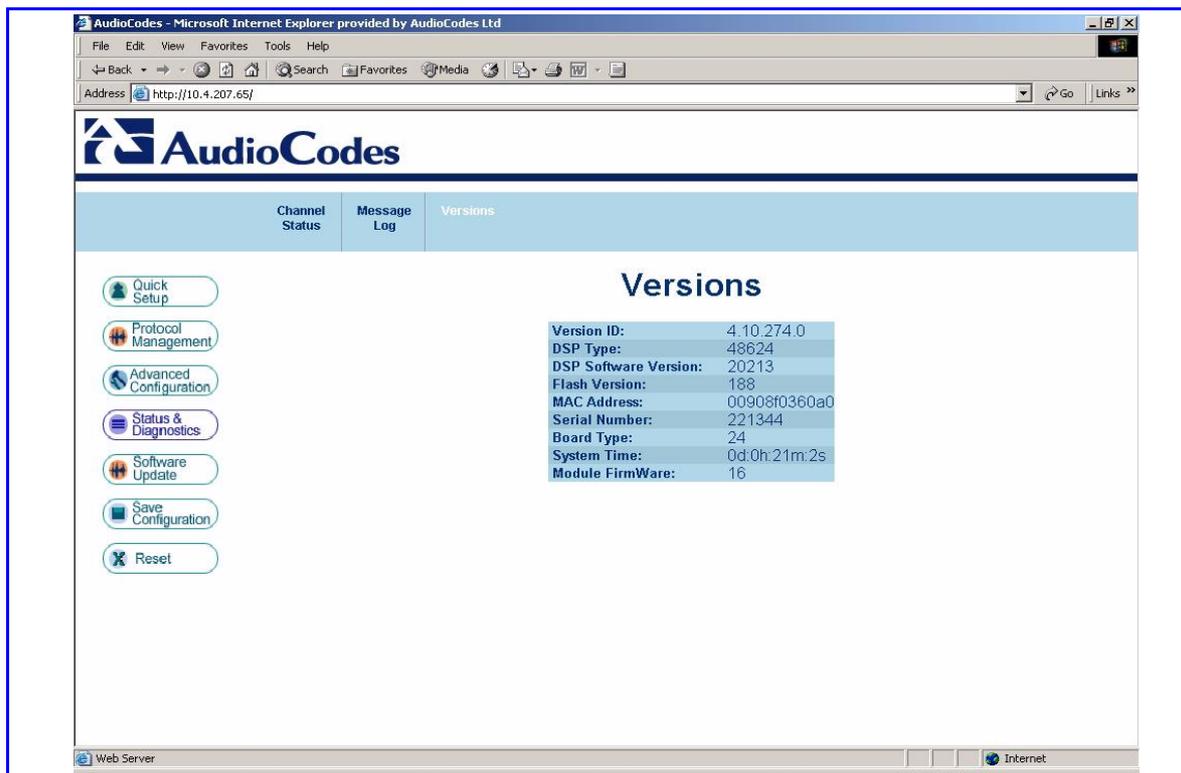
5.6.3 Versions

The Versions screen displays hardware and software product information.

➤ To display the Versions screen, take this step:

- On the sub-menu bar on the top, click the Versions option. The Versions screen appears.

Figure 5-15: Versions Screen



5.7 Software Update

The Software Update screen offers two options for downloading current software update files, TFTP and HTTP for which screens are provided for this purpose. In addition the License screen is provided for Users to enter their updated license keys.

➤ To display the Software Update screen, take this step:

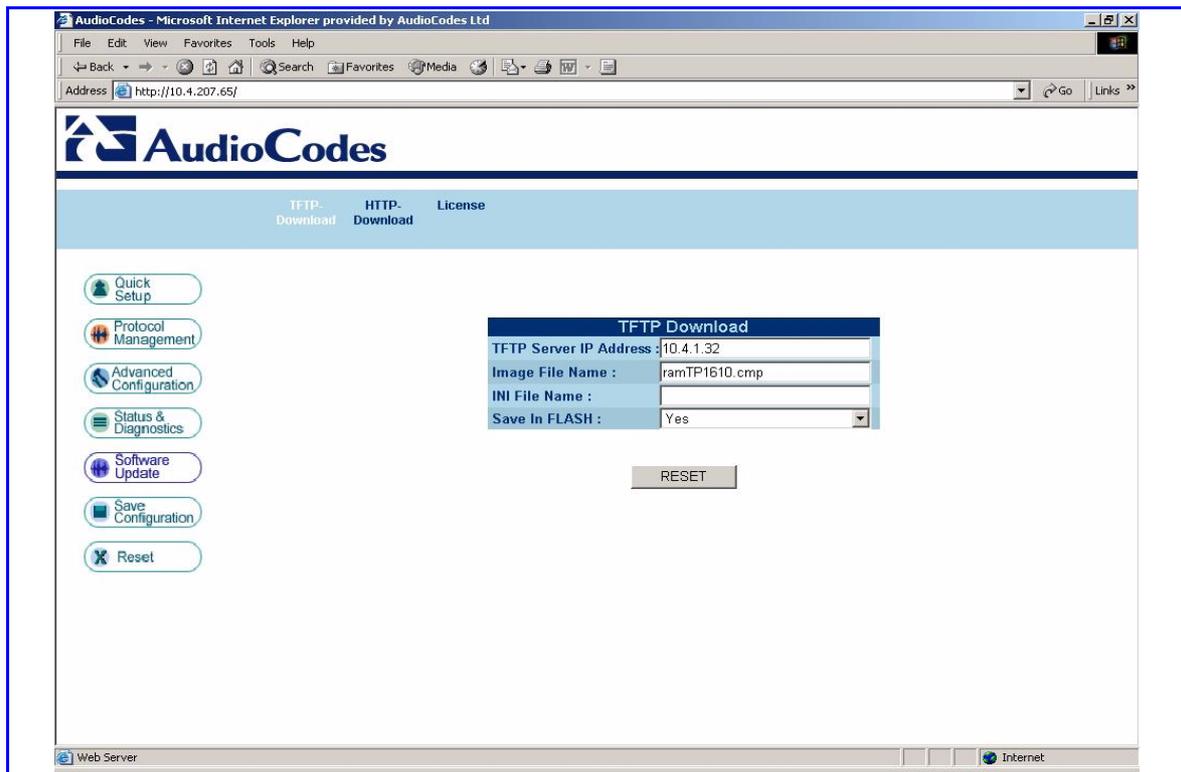
- To access the Software Update menu, from the main menu list on the left, click on the Software Update button. The Software Update screen with the sub-menu bar on the top is displayed.

5.7.1 TFTP Download

The TFTP Download screen facilitates the download of *ini* file software updates using this protocol.

➤ To use the TFTP Download screen, take these 2 steps:

- Step 1.** On the sub-menu bar on the top, click the TFTP Download option. The TFTP Download screen appears.

Figure 5-16: TFTP Download Screen


Step 2. Enter the relevant values in the fields and click the **Reset** button. A dialog box appears in which you confirm the reset action. The new information is added to the system configuration while the system is restarted. A message informing you of the waiting period appears.

5.7.2 HTTP Download

The HTTP Download screen facilitates the download of software updates using this protocol. Types of software file updates include:

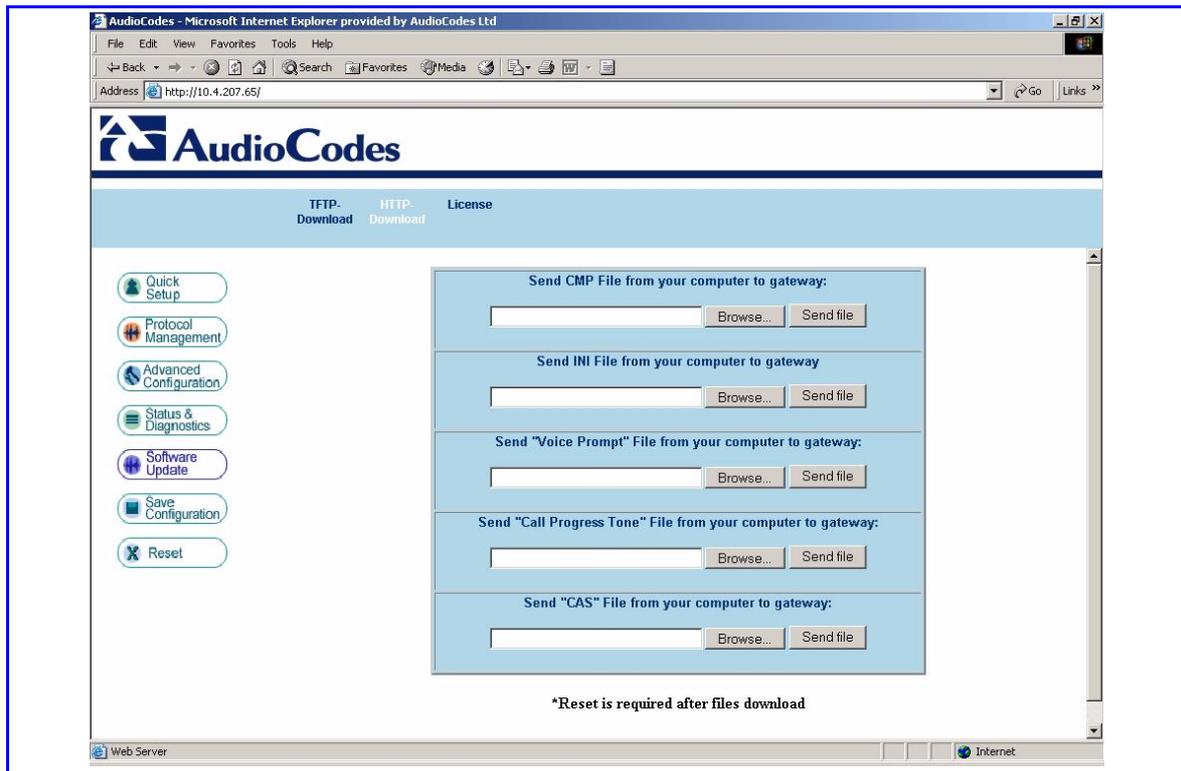
- CMP
- *ini*
- Voice Prompt
- Call Progress Tone
- CAS



To use the HTTP Download screen, take these 4 steps:

Step 1. On the sub-menu bar on the top, click the HTTP Download option. The HTTP Download screen appears.

Figure 5-17: HTTP Download Screen



- Step 2.** Use the **Browse** button to locate the appropriate file on your computer.
- Step 3.** Click the **Send File** button. The file is sent to the Media Gateway.
- Step 4.** To commit the changes to the Flash memory, in the main menu on the left, click the **Reset** button to go to the **Reset** screen and reset the system.

- Note 1:** Reset can be scheduled for a later time period when call traffic is at a minimum. If you choose to schedule the Reset for a later time, be sure to use the Save Configuration screen (described on page 79) to retain the changes to the configuration on the local PC.
- Note 2:** If you are modifying multiple screens, perform the reset after you are finished modifying all of the screen you intended and NOT after each screen.

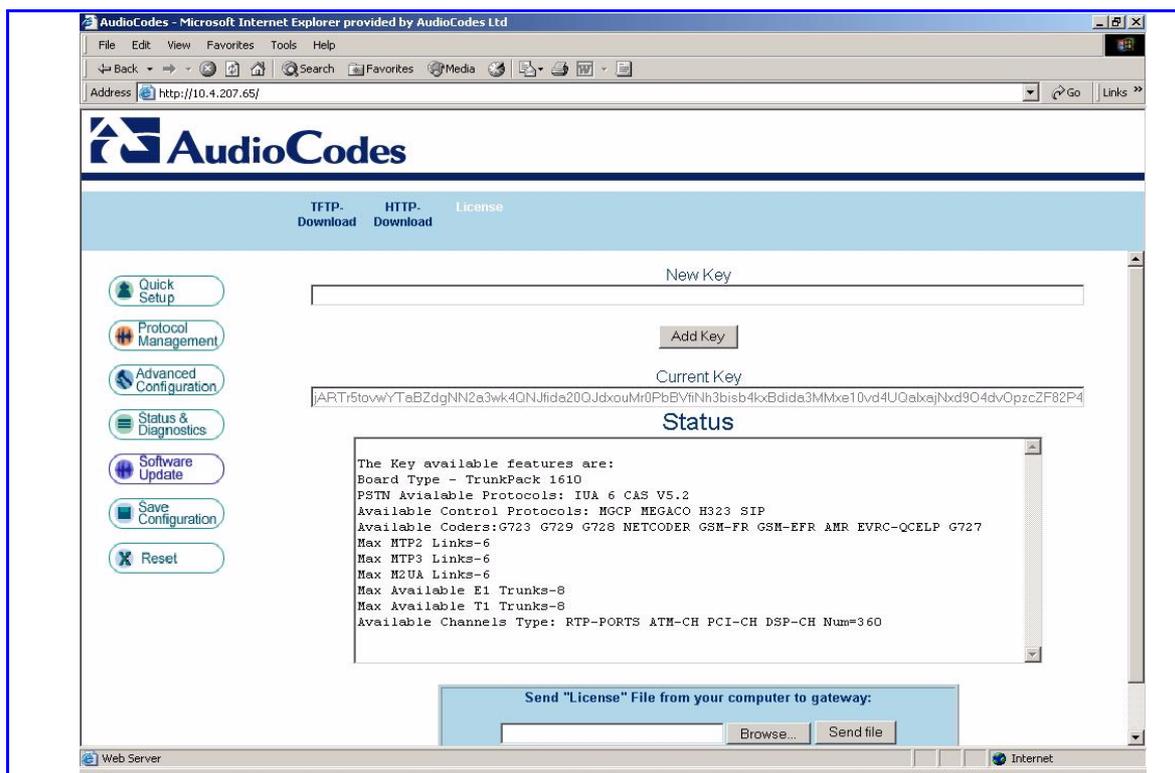
5.7.3 License

The License screen allows the User to enter an updated license key.

➤ **To use the License screen, take these 5 steps:**

- Step 1.** On the sub-menu bar on the top, click the License option. The License screen appears.

Figure 5-18: License Screen



- Step 2.** In the New Key field, enter the updated license key supplied by AudioCodes.
- Step 3.** Click the **Add Key** button. The new key information overwrites the previous key installed on the computer.
- The current key information is displayed in the Current Key field. Information about the key and software limitations are displayed in the Status field.
- Step 4.** To send a license key installed on your computer to the Media Gateway, access the license key file using the **Browse** button and click the **Send File** button.
- Step 5.** To commit the changes to the Flash memory, in the main menu on the left, click the **Reset** button to go to the **Reset** screen and reset the system.

Note 1: Reset can be scheduled for a later time period when call traffic is at a minimum. If you choose to schedule the Reset for a later time, be sure to use the Save Configuration screen (described on page 79) to retain the changes to the configuration on the local PC.

Note 2: If you are modifying multiple screens, perform the reset after you are finished modifying all of the screen you intended and NOT after each screen.

5.8 Save Configuration

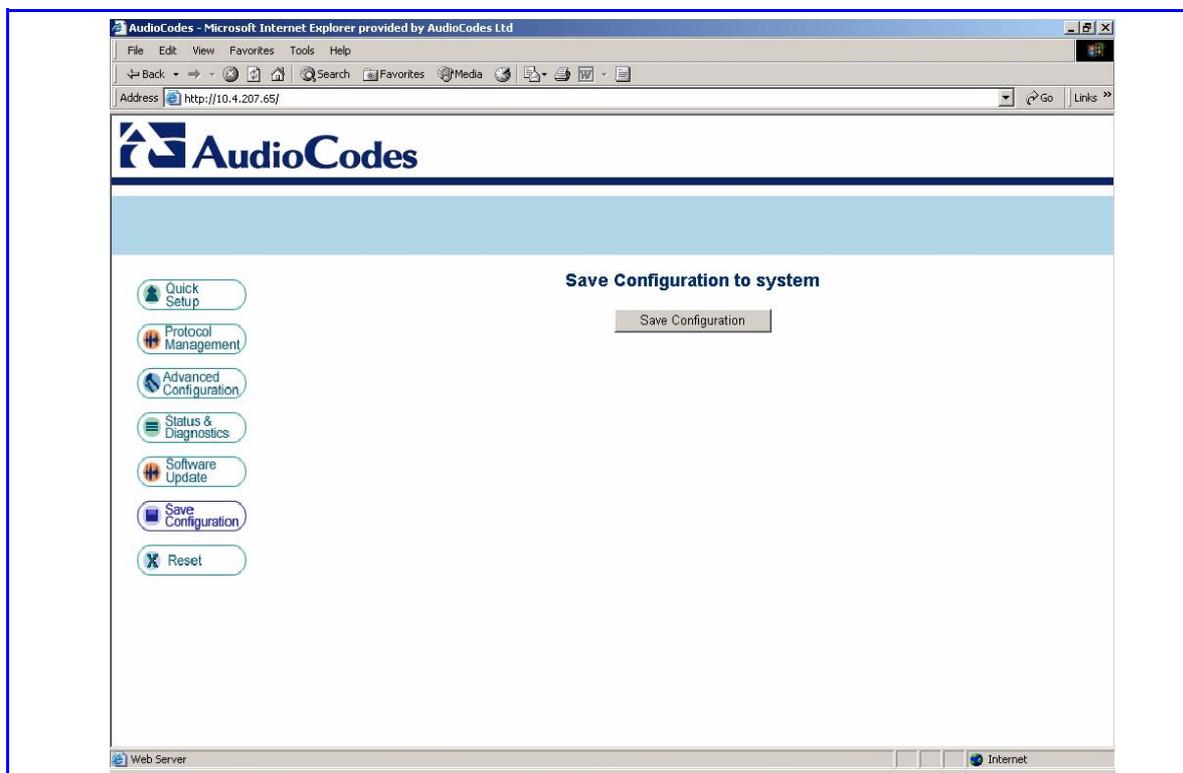
The Save Configuration screen allows the User to save changes to the configuration locally until a system Reset is performed.

Note: If you perform the reset immediately after making the changes to the configuration there is no need to use the Save Configuration function prior to the reset.

➤ **To use the Save Configuration screen, take these 2 steps:**

Step 1. To access the Save Configuration menu, from the main menu list on the left, click on the Save Configuration button. The Save Configuration screen is displayed.

Figure 5-19: Save Configuration Screen



Step 2. Click the Save Configuration button. The new information is saved to the Flash memory on the board. A message informing you that it has been saved appears.

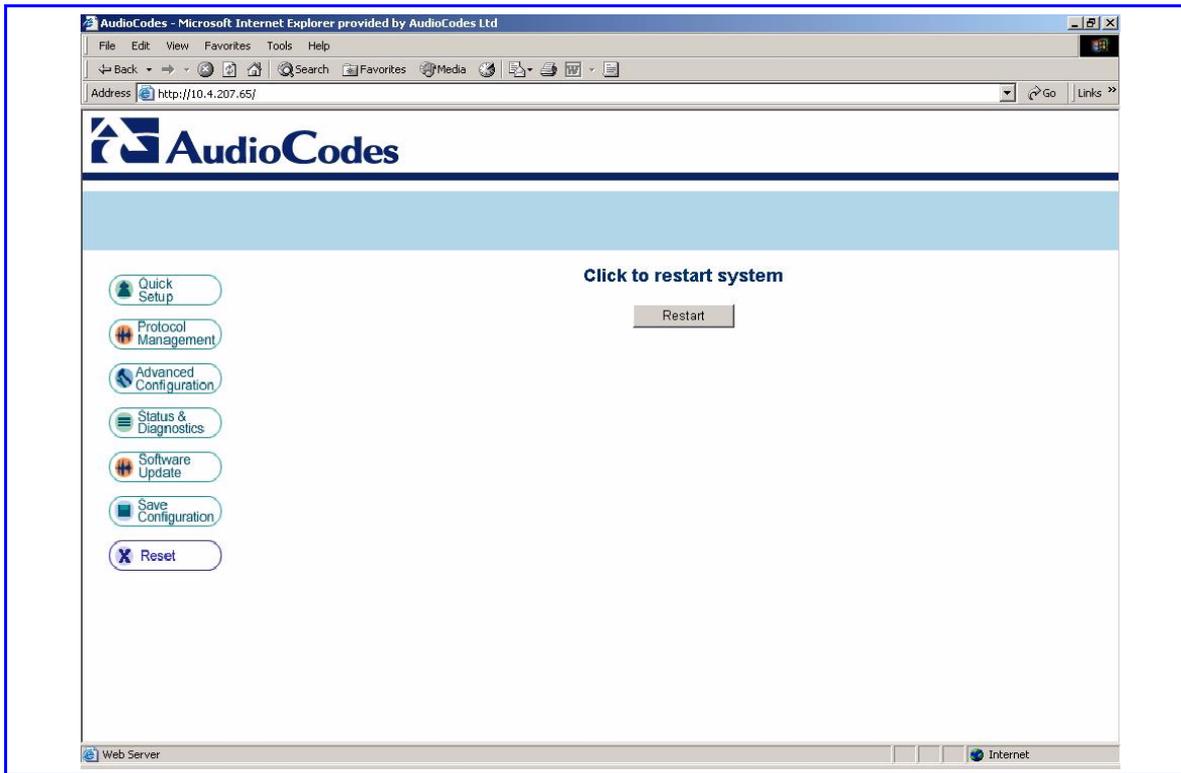
5.9 Reset Button

The Reset button ensures that all changes to the configuration are permanently saved in the Flash memory of the Media Gateway board.

➤ **To use the Reset Button screen, take these 2 steps:**

Step 1. To access the Reset menu, from the main menu list on the left, click on the Reset button. The Reset screen is displayed.

Figure 5-20: Reset Screen



- Step 2.** Click the **Restart** Button. All of the changes made to the configuration are saved to the Flash memory of the Media Gateway board. The board is shut down and re-activated. A message about the waiting period is displayed. The screen is refreshed.

6 Mediant 2000 Provisioning

AudioCodes provides a wide range of diagnostic tools to enable the User to easily identify an error condition and to provide a solution or workaround.

6.1 Provisioning of the Mediant 2000 for Call Control Protocol

Initial configuration of the **Mediant 2000** is provided by a configuration *ini* file. If required, the *ini* file is downloaded using BootP and TFTP procedures. The *ini* file name is provided in the Boot File Name field in BootP TFTP Server, refer to the section, “*ini* File Field and Sections” in AudioCodes’ “VoPLib Reference Library User’s Manual”, Document #: LTRT-00744.

The *ini* file contains:

- **Mediant 2000** basic configuration parameters including E1/T1 interface configuration (refer to Table 6-1 on page 82)
- MGCP specific parameters (refer to Table 6-2 on page 87)
- MEGACO specific parameters (refer to Table 6-3 on page 89)
- MEGACO/MGCP common parameters (refer to Table 6-4 on page 91)
- MEGACO/MGCP Default Channel parameters (refer to Table 6-5 on page 92)
- Names for optional configuration files (CAS signaling, Call Progress Tones and Voice Prompt files). For detailed information, refer to the section, “Configuration Files Description” in AudioCodes’ “VoPLib Reference Library User’s Manual”, Document #: LTRT-00744.

All *ini* file data is downloaded at startup and stored in non-volatile memory. The provisioning procedure should be used again only to modify the **Mediant 2000** parameters, otherwise BootP TFTP is not required.

Default Channel Parameters are only required when using the **Mediant 2000** with MGCP or MEGACO protocols. These parameters are set for all **Mediant 2000** channels.

Users do not have to specify all (or any) of the fields as each field not specified is set to its default value. These default values are used even if the current configuration stored in the board’s non-volatile memory is different from the default; that is, using the *ini* file resets all unspecified board configuration parameters to their default values. Note that some of the fields are shared for both protocols.

The Default Channel Parameters also define the DTMF/MF, Fax and Modem transfer modes. For a detailed description of these modes, refer to Appendix G in AudioCodes’ “VoPLib Reference Library User’s Manual”, Document #: LTRT-00744.

Table 6-1 describes the **Mediant 2000** basic configuration fields that are common to all control protocols.

Table 6-1: Mediant 2000 Basic Configuration Parameters (continued to page 87)

<i>ini</i> File Field Name	Valid Range and Description
BaseUDPPort	Lower boundary of UDP ports to be used by the board. The upper boundary is BoardBaseUDPPort+10*(Number of Channels). Must be a multiple of 10. Range 0 to 55000 (default = 4000).
BootPDelay	1 = 1 second (default) 2 = 10 seconds 3 = 30 seconds 4 = 60 seconds 5- 120 seconds Delay before start BootP
BootPRetries	1 = 1 BootP retries =1 sec 2 = 2 BootP retries =3 sec 3 = 3 BootP retries =6 sec 4 = 10 BootP retries=30 sec 5 = 20 BootP retries =60 sec 6 = 40 BootP retries=120 sec 7 = 100 BootP retries = 300 sec 15 = BootP retries indefinitely. Number of BootP Retries. Default value is 3.
CASFileName_0 to CASFileName_7	These are the names (and path names) of the files containing the CAS protocol configuration. Refer to Section 3.4.3 "CAS Protocol Configuration Files" in the VoPLib Reference Library User's Manual, Document #: LTRT-00740 for additional information on how to create these files. It is possible to use up to 8 files, but if less than 8 files are used they should be consecutively numbered from zero.
CASTableIndex_N	This is a CAS protocol file index (where N is the trunk number). It indicates the CAS protocol file to use at the specific Trunk. The index value is according the number at the field "CASFileName_X".
CASTablesNum	Indicates how many CAS protocol configuration files are loaded, its range being 1 to 8, and it should match the number of "CASFileName_X" fields.
ClockMaster	0/1 (default = 0); Selects the trunk clock source. When choosing acCLOCK_MASTER_OFF (= 0), the clock is recovered from the line. When choosing acCLOCK_MASTER_ON (= 1) the trunk clock source is provided by the internal/TDM bus clock source depending on the TDMBusClockSource parameter.
DchConfig	Defines D-channel configuration. This setting is only applicable to ISDN PRI protocols that support NFAS and/or D-channel backup procedures. 0 = Primary (default) 1 = Backup 2 = NFAS
DisableSNMP	Null If the field name exists, then disable SNMP access.

Table 6-1: Mediant 2000 Basic Configuration Parameters (continued to page 87)

<i>ini</i> File Field Name	Valid Range and Description
DisableTPNCPEvent	According to enumacTEvent This parameter disables Events Reporting for the selected event (refer to: enumacTEvent)
DisableTrunkAfterReset_X	1 = Disable 0 = Enable (default) Disables a Trunk. Use the _X form to disable a specific trunk(s) [X= the trunk number] or the general form(DisableTrunkAfterReset) for disabling all the board trunks.
DisableWebConfig	0 = Set to Read/Write mode = default; 1 = Set to Read-only mode
DisableWebTask	0 = Enable Web Task= default; 1 = Disable Web Task
DSPVersionTemplate Number	Selects the DSP load number. Each load has different Coder list, Channel capacity and features supported
EnableContinuityTones	Enables/Disables Continuity Test tone detection and generation according to the ITU-T Q.724 recommendation
EnableDiagnostics	0 = No diagnostics (default) 1 = Perform diagnostics = Full test of DSPs, PSTN, PCM Switch, LAN Phy and Flash. 2 = Perform diagnostics = Full test of DSPs, PSTN, PCM Switch, LAN Phy. Partial test of Flash (quicker mode).
EnableSyslog	1 = Enable 0 = Disable (default) Enable syslog protocol log
ForceExceptionDump	1 = Enable 0 = Disable (default) Enable - send last software exception dump each time board restarts. Refer to "Software Exception Dump" in AudioCodes' "VoPLib Reference Library User's Manual", Document #: LTRT-00744.
FramingMethod	Selects the Framing method to be used for this trunk. 0 = default according to protocol type (E1 or T1) E1 default = E1 CRC4 MultiFrame Format extended G.706B (as c) T1 default = T1 Extended SuperFrame with CRC6 0 = E1 Extended SuperFrame Format (E1 default) 1 = T1 SuperFrame Format (T1 default). a = E1 DoubleFrame Format b = E1 CRC4 MultiFrame Format c = E1 CRC4 MultiFrame Format extended G.706B C = T1 Extended SuperFrame without CRC6 F = J1 Extended SuperFrame with CRC6 (Japan)
HeartBeatDestIP	IP address in dotted notation format: xxx.xxx.xxx.xxx Sets the destination IP to which the heartbeat packets are sent

Table 6-1: Mediant 2000 Basic Configuration Parameters (continued to page 87)

<i>ini</i> File Field Name	Valid Range and Description
HeartBeatDestPort	Destination UDP port Sets the destination UDP port to which the heartbeat packets are sent
HeartBeatIntervalmSec	Time in milliseconds Sets the time delay between consecutive heartbeat packets
IdleABCDPattern	Range 0x0 to 0xF; ABCD (CAS) Pattern to be applied to CAS signaling bus when the channel is idle. This is only relevant when using PSTN interface with CAS protocols. Set to -1 for default.
IdlePCMPattern	PCM Pattern applied to E1/T1 timeslot (B-channel) when channel is idle. Range 0x00 to 0xFF; -1 = default pattern, 0xFF for ?-Law, 0x55 for A-law.
IniFileVersion	Long integer value This optional field contains the ini file version number. This ini version number is reported in the acEV_BOARD_STARTED event (refer to "struct.acTBoardStartupInfo" in AudioCodes' "VoPLib Reference Library User's Manual", Document #: LTRT-00744).
ISDNGeneralCC Behavior	This is the bit-field used to determine several general ISDN behavior options. Refer to the Appendix A.8 "ISDN Flexible Behavior" in the VoPLib Reference Library User's Manual, Document #: LTRT-00740.
ISDNICallsBehavior	This is the bit-field used to determine several behavior options which influence how the Q.931 protocol behaves. Refer to the Appendix A.8 "ISDN Flexible Behavior" in the VoPLib Reference Library User's Manual, Document #: LTRT-00740.
ISDNInCallsBehavior	This is the bit-field used to determine several behavior options which influence how the ISDN Stack INCOMING calls behave. Refer to the Appendix A.8 "ISDN Flexible Behavior" in the VoPLib Reference Library User's Manual, Document #: LTRT-00740.
ISDNNFASInterfaceID	0 to 255 (default = 0). Defines the Interface ID; works with NS_EXPLICIT_INTERFACE_ID bit. Refer to the Appendix A.8 "ISDN Flexible Behavior" in the VoPLib Reference Library User's Manual, Document #: LTRT-00740. Also used for IUA protocols to determine the Interface ID
ISDNOutCallsBehavior	This is the bit-field used to determine several behavior options which influence how the ISDN Stack OUTGOING calls behave. Refer to Appendix A.8 in the VoPLib Reference library User's Manual.
LineBuildOutLoss	Selects the line build out loss to be used for T1 trunks (not for E1 trunks). 0 = 0 dB (default) 1 = -7.5 dB 2 = -15 dB 3 = -22.5 dB
LineCode	Use to select line code. B8ZS or AMI for T1 spans and HDB3 or AMI for E1 spans. 0 = use B8ZS line code (for T1 trunks only) (default). 1 = use AMI line code. 2 = use HDB3 line code (for E1 trunks only).
LineStatusChange TrapEnableDefault	Null Enables the Trap <i>Line Status Change</i> (as defined in DS1 MIB) for all the trunks.
LowDSPResources EventHyst	Number of DSPs between Low and High watermarks of Low DSP Resource Notifications

Table 6-1: Mediant 2000 Basic Configuration Parameters (continued to page 87)

<i>ini</i> File Field Name	Valid Range and Description
LowDSPResourcesEventThreshold	This value determines when Low number of DSP Resource Notification is issued.
MaxEchoCancellerLength	The maximum Echo Canceller Length capability. This parameter may reduce the channel capacity of the product.
NFASGroupNumber	0 to 4 (default = 0 = not NFAS trunk). Relevant only for T1 ISDN NFAS trunks, indicates the group number of the NFAS group. Valid NFAS group numbers are only 1 to 4, 0 indicating that this trunk is not NFAS (in this case the ISDNNFASInterfaceID and DchConfig parameters are ignored).
PCMLawSelect	Selects the type of PCM companding law in input/output TDM bus (TDM bus is defined using the TDMBusType parameter). 1 = A-law 3 = μ -Law
ProtocolType	Sets the PSTN protocol to be used for this trunk. NONE = 0 E1_EURO_ISDN = 1 T1_CAS = 2 T1_RAW_CAS = 3 T1_TRANSPARENT = 4 E1_TRANSPARENT_62 = 5 E1_TRANSPARENT_60 = 6 E1_MFCR2 = 7 E1_CAS_R2 = 8 E1_RAW_CAS = 9 T1_NI2_ISDN = 10 T1_4ESS_ISDN = 11 T1_5ESS_9_ISDN = 12 T1_5ESS_10_ISDN = 13 T1_DMS100_ISDN = 14 J1_TRANSPARENT = 15 T1_NTT_ISDN = 16 Japan - Nippon Telegraph Telephone E1_AUSTEL_ISDN = 17 Australian Telecom E1_HKT_ISDN = 18 European Hong Kong - HKT E1_KOR_ISDN = 19 Korean Operator T1_HKT_ISDN = 20 European Hong Kong - HKT over T1 E1_QSIG = 21 QSIG over E1 E1_TNZ_ISDN = 22 Telecom New Zealand Flag T1_QSIG = 23 QSIG over T1 V5_2_AN = 26 T1_IUA = 28 E1_IUA = 29 Relevant only when TDMBusType=acFRAMERS.
ResetWebPassword	Allows resetting of Web password to default value (no values).
SITDetectorEnable	Enables/Disables SIT detection according to the ITU-T recommendation E.180/Q.35.

Table 6-1: Mediant 2000 Basic Configuration Parameters (continued to page 87)

<i>ini</i> File Field Name	Valid Range and Description
SNMPManagerIP	IP address in dotted notation format. xxx.xxx.xxx.xxx IP address of default SNMP manager. SNMP traps are sent to this manager.
SNMPPort	Valid UDP port number; default = 161 This parameter specifies the port number for SNMP requests and responses. Usually should not be specified, use default.
SNMPTrapPort	Valid UDP port number; default = 162 This parameter specifies the port number for SNMP Traps. Usually, it should not be specified. Use the default.
SysLogServerIP	IP address in dotted format notation, e.g., 192.10.1.255.
TDMBusClockSource	Selects the clock source on which the board synchronizes. 1 = local oscillator 3 = MVIP 4 = PSTN Network 8 = H.110A 9 = H.110B 10 = Net reference1 11 = Netreference2 12 = SC2M 13 = SC4M 14 = SC8M
TDMBusLocal Reference	0 to (MAX_TRUNK_NUM-1) When ClockSource is set to acH110BusClockSource_Network, this parameter selects the TrunkID to be used as the clock synchronization source of the board. When using H.110/H.100 bus this parameter also selects the trunk which is used as the clock source for the NETREF clock generation (in this case ClockSource might not be set to acH110BusClockSource_Network).
TDMBusMasterSlave Selection	0 = Slave mode 1 = H110A (or SC, MVIP) Master 2 = H.110B Master Set SC/MVIP/H.110 to either Master mode (the board is the clock source for the TDM bus) or slave mode (some other board in the system must supply the clock to the TDM bus).
TDMBusType	Selects the TDM bus interface to be used (only one TDM bus interface can be enabled at one time, although more than one can physically exist on the board). 0 = acMVIP_BUS; 1 = acSC_BUS; 2 = acFRAMERS; 4 = acH100_BUS
TerminationSide	For ISDN only, to select the ISDN Termination side. User side = 0 = default Network side = 1

Table 6-1: Mediant 2000 Basic Configuration Parameters (continued to page 87)

<i>ini</i> File Field Name	Valid Range and Description
TraceLevel	Defines the Protocol Trace level 0 = No trace 1 = Full ISDN Trace (all ISDN layers) 3 = ISDN Q.931 Message Trace 4 = ISDN Trace without Duplication 7 = ISDN Q.921 Message Trace (raw data) 20 = V5 Full Trace without Duplication (all layers and protocol entities) 22 = V5 Full Trace Layer 3 and above without Duplication

6.2 Provisioning the Mediant 2000 for MGCP & MEGACO

The initial MGCP and MEGACO control protocols configuration for the **Mediant 2000** includes the following information:

- MGCP specific parameters (refer to Table 6-2)
- MEGACO specific parameters (refer to Table 6-3 on page 89)

Table 6-2: MGCP Specific Parameters (continued on page 88 to page 89)

<i>ini</i> File Field Name	Valid Range	Description
CallAgentIP	IP address in dotted notation.	MGCP Call Agent IP address to be used for the initial RestartInProgress MGCP message (RSIP). e.g. 192.10.1.255. Set to 0.0.0.0 to avoid sending RSIP. This parameter, if in use, overrides the Call Agent IP address provided in BootP.
CallAgentPort	0 to 65535 (default= 2427)	UDP Port of MGCP Call Agent.
DePopulatedChannels Number	0 to max number of channels in board. Set to -1 = to use all.	This parameter can be used to enable MGCP call agent to access only subset of the on board channel bank. Set to -1 (or don't specify) to use all channels.
DialToneDuration	1 to 65535 seconds (default=180 sec)	This parameter overrides default 180 sec time out defined for the dial tone signal.
DigitMapTimeoutTimer	1 to 65535 seconds (default = 4 sec)	Defines timeout value (T symbol) in digit map.
DTMFDigitLength		Time to play DTMF in msec.
DTMFInterDigitInterval		Time between DTMFs played in msec.
EndpointPrefix	String[19] Default=""	A string of up to 19 characters representing the B-channel number in the range 0 to 31 for E1, or 1 to 24 for T1. The 'TrunkName'/'EndpointPrefix' fields, together generate the local endpoint name on trunk-enabled gateways.

Table 6-2: MGCP Specific Parameters (continued on page 88 to page 89)

<i>ini</i> File Field Name	Valid Range	Description
GatewayMGCPPort	0 to 65535 (default= 2427)	If User wants to force Media Gateway to listen to another UDP port in addition to the original 2427 as defined in RFC 2705.
GateWayName	String[64] (default = Audiocodes.com)	Gateway's identification name towards the MGCP Call Agent. If empty, Gateway name holds the IP address of the board.
MGCPActiveEndpoints	String[20] default: all endpoints are active.	List of active endpoints separated by commas. Use hyphen to define range of endpoints. For example: "1,3,5-7" means that endpoints 1,3,5,6,7 are active. Functions only with End point Naming configuration. With Trunk Naming configuration the results are unexpected.
MGCPCompatibility Profile	1 = Normal (= default) 8 = SpecificEndPointOld 32 = NCSDigitNotification	SpecificEndPointOld, for Users that get the AUEP with asterisk response with the old format EP0@AC.com, EP1@AC.com, EP2@AC.com..... NCSDigitNotification, e.g. every digit notified separately on each Notification message.
MGCPDebugMode	0 = Off (= default) 1 = On	When 'On' dumps the incoming and outgoing MGCP commands.
MGCPDTMFDetection Point	0 = at the beginning 1 = at the end of DTMF (default = 1)	Defines if the detection of DTMF events is notified at the end of DTMF or at the beginning.
MGCPEndPoint NumberingOffset	> 0 (Default = 0)	User can add offset to endpoints. This functions only with End point naming configuration. With Trunk Naming configuration the results are unexpected
MGCPPersistentEvents	String [128] (Default = 'L/hd,L/hu,L/hf,L/oc')	User can select new persistent events as needed. For example: MGCPERSISTENTEVENTS = 'L/dl(N),D/x(N),L/oc(N)'. Persistently detects dial tones, report on any DTMFs and expired Call Progress tones
MGCPSendMACWith RSIP	No Value (default = Don't send)	When this parameter exists in the ini file, generated RSIPs have the MG's MAC address in addition to the usual parameters. This parameter is sent as an MGCP extension parameter.
MGCPVersion	String[10]	MGCP version number
ProvisionedCallAgents	Up to 10 IP addresses in dotted format notation	User can specify up to ten different IP addresses to be provisioned by the Gateway. MGCP call agent must be specified as one of the call agents in the list. If command is received from a non-provisioned IP address, command is ignored. If this parameter is omitted in the ini file, all IP addresses are provisioned

Table 6-2: MGCP Specific Parameters (continued on page 88 to page 89)

<i>ini</i> File Field Name	Valid Range	Description
QuarantineModeState	0 = Disable = default 1 = Enable	When enabled, the Quarantine events are handled according to RFC 2705. In non-quarantine modes, a Notification is sent immediately on event detection.
RedundantAgentIP	IP address in dotted format notation	Redundant MGCP call agent IP address for initial RestartInProgress MGCP message (RSIP), e.g., 192.10.1.255. Set to 0 to avoid sending RSIP.
RedundantAgentPort	0 to 65535 (default = 2427)	UDP Port of Redundant MGCP call agent.
RedundantCallAgent Name	String[64] (default = "")	Redundant MGCP Call Agent domain name
RSIPOnNetwork Disconnection	0 = Don't send 1 = Send RSIP = default	This flag specifies whether to send or don't send RSIP when LAN is re-connected.
RTCPInterval	0 to 65535 msec default= 5000 (5 sec)	Time interval between the adjacent RTCP reports. Set in milliseconds.
TrunkName	String[19] Default=""	A string of up to 19 characters followed by the Trunk number representing the Trunk in the range 0 to 8. The 'TrunkName'/'EndpointPrefix' fields, together generate the local endpoint name on trunk-enabled gateways.
UseBracketsWith GatewayName	0 = Off 1 = On (default)	When using setting 'GateWayName' as empty string, and setting the parameter to '1', Gateway name is board IP address with added brackets. i.e., [10.2.211.11]
UseWildcardWithRSIP	0 = Don't use 1 = Use = default	When wildcard is used, RSIPs turn in a single message on EndPoint Naming configuration, and Single message for each trunk in Trunk Naming configuration. If Off and number of channels is less that 64, RSIP message sent for each Endpoint.

- MEGACO specific parameters (refer to Table 6-3)
- MEGACO/MGCP common parameters (refer to Table 6-4 on page 91)

Table 6-3: MEGACO Specific Parameters (continued on page 90)

<i>ini</i> File Field Name	Valid Range	Description
DigitMapName	String[10] (default = Null)	Name of the provisioned digit map.
DigitMapping	String[152] (default = Null)	The digit map patterns separated by a vertical bar ()
EndPointName	String[19] (default = "line/" for analog board and "/c" for Trunking boards)	Prefix of the endpoint part of the termination name

Table 6-3: MEGACO Specific Parameters (continued on page 90)

<i>ini</i> File Field Name	Valid Range	Description
GatewayName	String[63] (default = NULL for analog board and "tgw/" for Trunking boards)	Gateway prefix in the termination name
KeepAliveEnabled	0 = disable >0 = enable	This parameter can be used to enable a KeepAlive message (NOP ServiceChange).
KeepAliveInterval	1 to 300 seconds (default = 12 seconds)	This parameter is used to define the interval in seconds of a keepalive message
MEGACOASN1Profile	Integer. default = 0	Used for profiling of the binary ASN.1 parsing. Current profile supported number is 0 (default), supports examples from Web site of Ericsson™: http://www.erlang.org/project/megaco/encoding_comparison/encoded_messages/ber/
MEGACOEncoding	0 = Text mode = default 1 = binary ASN.1	This parameter sets the MEGACO Coding method. If set to 1, then it supports the binary ASN.1 format of the MEGACO protocol.
MegacoTrunkIDOffset	0 or Positive value Default = 0	Sets the offset to the Trunk numbering. For example, MEGACOTRUNKIDOFFSET=2 causes the first trunk number to be 2.
MGCPCompatibility Profile	1 = V12 2 = V1 3 = V1 (with adaptations) (default = 2)	Profile 1 is compatible with the old v12 MEGACO version. Profile 2 is MEGACO V1 compatible. Profile 3 is the same as 2, but value of 'init' parameter in on-hook off-hook reports is 'True' 'False' and not 'On' 'Off'.
MGCExecutionTime	0 to 2000 milliseconds (default = 100 msec)	Estimated execution time of MGC (in millisec)
MGExecutionTime	0 to 2000 milliseconds (default = 100 msec)	Estimated execution time of Media Gateway (in millisec)
MGProvisionalResponseTime	0 to 20000 milliseconds (default = 100 msec)	Provisional response timer for Media Gateway (in millisec)
ProvisionedCall Agents	Legal IP Address (default = 0)	List of up to 5 legal IP addresses separated by ',' for the ServiceChange command. Can use either dotted format 192.10.1.X
ProvisionedCall AgentsPorts	0 to 65535 (default= 2944)	List of up to 5 UDP Ports of MEGACO Call Agent, separated by ','.
TrunkName	String[19] (default = "s" for Trunking boards)	Trunk prefix (used only for trunking gateways).

Note 1: The combination of GatewayName, TrunkName and EndPointName generates the correct physical termination name.

Note 2: For analog Gateways, combination of GatewayName and EndPointName generates the correct physical termination name.

- MEGACO/MGCP common parameters (refer to Table 6-4)

Table 6-4: MGCP/MEGACO Common Parameters (continued to page 92)

<i>ini</i> File Field Name	Valid Range	Description
CallAgentDomainName	String[64] (default= "")	MGCP Call Agent domain name.
ConnectionIDBase	> 0 (default = 20)	Defines the lowest number for the Connection ID values assigned by the gateway.
ConnectionIDRange	> 0 (default = 999999999)	Defines the range for the Connection ID values assigned by the gateway.
DefaultPacketization Period	5 to 80 msec default = 20 msec (for G.723, default = 30)	Default packetization period (Frame Size)
DNSPriServerIP	LegalIP address (default = 0)	IP of primary DNS server IP
DNSSecServerIP	LegalIP address (default = 0)	IP of secondary DNS server IP.
ECNonLinearProcessing Mode	0 = ECAdaptiveNlp = default 1 = ECDisableNlp 2 = ECSilenceOutputNlp	Defines echo cancelation non-linear processing mode.
FlashHookPeriod	>0 msec (default = 400 msec)	Flash hook period.
IPDIFFSERV	of 0-63 (default 63)	Sets the DiffServ field of the IP header for all packets generated from this channel. This parameter can be used instead of the IPPRECEDENCE and IPTOS parameters.
IPPrecedence	0 to 7 (default =0)	Sets value of the IP precedence field in the IP header for all packets generated from this channel. Follow the actIPPrecedence enumeration.
IPTOS	0 to 15 (default =0)	Sets value of IP Type Of Service field in IP header for all packets generated from this channel.
MGControlProtocolType	0 = None 1 = MGCP = default 2 = MEGACO	This parameter defines the control protocol type.
MGCPCommunication LayerTimeout	>0 sec (default = 30 sec)	Assumed delay of the communication layer. It is used in retransmission.
MGCPDefaultCoder	Valid coder names (default = G.711 μ -law)	The User can set a coder (according to the known coder name)
MGCPDefault PacketizationPeriod	See 'DefaultPacketization Period'	
MGCPRetransmission Timeout	0 to 10000 msec (default = 200 msec)	Sets initial time for first retransmission. Then Retransmission intervals thereafter increase exponentially.

Table 6-4: MGCP/MEGACO Common Parameters (continued to page 92)

<i>ini</i> File Field Name	Valid Range	Description
RandomizeTransactionId	0 to 1 (default = 1)	Defines if the transactions produced by the board start with a fixed number or random number. Refer also to 'TransactionIdRange' and 'TransactionIdBase'
RingOffPeriod	>0 msec	Default ringing OFF period on analog line
RingOnPeriod	>0 msec	Default ringing ON period on analog line
RTPCName	Up to 255 characters string (default ="ChX" X=CID)	RTP Canonical Name. Should be unique among all participants in one RTP session. Up to 255 char string.
SingleSIDPacketWithSCEG 729	SingleSIDPacketWithSCEG729 = 1, option is "ON". SingleSIDPacketWithSCEG729 = 0, option is "Off". (default = 0)	If parameter is "ON" and the channel was opened or modified to work with G.729 coder with Silence Suppression, while Silence is detected only a single SID packet is sent. Otherwise, SID packets are sent frequently according to energy changes that require an SID packet for each change.
TransactionIdBase	> 0 (Default = 2000)	Defines the minimum number for transaction ID.
TransactionIdRange	> 0 (Default = 999999999)	Defines the range for the transaction ID.

- MEGACO/MGCP Default Channel parameters (refer to Table 6-5)

Table 6-5: MGCP/MEGACO Default Channel Parameters (continued on page 94 to page 96)

<i>ini</i> File Field Name	Valid Range	Description
CallerIDTransportType	0 = Disable 1 = Relay 3 = Mute = default	DisableCallerID: Caller ID detectors are not activated. The CallerID signal flows in the regular RTP audio stream. RelayCallerID: Currently same as Mute. MuteCallerID: CallerID signals detected and reported but muted from the RTP voice stream.
CallerIDType	0 = BellCor 1 = ETSI 2 = NTT	The supported caller ID standard types
CallProgressDetector Enable	0 = Disable 1 = Enable = default	Enables or disables detection of Call Progress Tones.
CasTransportType	0 = No Relay over the network. 1 = Enable CAS relay RFC 2833	Controls the ABCD Signaling transport over IP type
CNGDetectorMode	0 = Disable, 1 = Relay, 2 = Events Only.	FAX CNG tone Detector mode
DisableNAT	0 = Disable = default, 1 = Enable	The User can choose to disable the NAT feature by setting this parameter

Table 6-5: MGCP/MEGACO Default Channel Parameters
(continued on page 94 to page 96)

<i>ini</i> File Field Name	Valid Range	Description
DisableRTCPRandomize		Controls whether RTCP reports intervals are randomized, or each exactly RTCPMeanTxInterval msec
DJBufMinDelay	0 to 150 (default = 150)	Dynamic Jitter Buffer Minimum Delay (msec)
DJBufOptFactor	0 to 12 (default = 7)	Dynamic Jitter Buffer frame error/delay optimization.
DTMFDetectorEnable	0 = Disable 1 = Enable = default	Enables or disables detection of DTMF signaling.
DTMFTransportType	0 = Mute 1 = Proprietary relay 2 = Transparent 3 = RFC 2833 relay (default =3)	Mute - erases digit from voice stream, does not relay to remote. Proprietary relay - erases digit from voice stream, relay to remote using proprietary syntax. Transparent - digits remains in voice stream. RFC 2833 relay - erase digit from voice stream, relay to remote using RFC 2833 syntax.
DTMFVolume	31 to 0 dBm	DTMF generation Gain Control
ECHybridLoss	0 = 6 dBm = default 1 = 9 dBm 2 = 0 dBm 3 = 3 dBm	Sets worst case ratio between signal level transmitted to hybrid & echo level returning from hybrid. Set this per worst hybrid in the system in terms of echo return loss. Refer to acTECHybridLoss enumeration:
EnableEchoCanceller	0 = Off 1 = On = default	Enables or disables the Echo Canceller (refer also to 'ForceEchoOff')
EnableEnergyDetector	0 = Disable = default, 1 = Enable	Activates the Energy Detector
EnablePatternDetector	0 = Disable = default, 1 = Enable	Activates the Pattern Detector
EnableRFC2658 Interleaving	0 = Disable = default 1 = Enable	When enabled, RTP packets include interleaving byte for VBR coders.
EnableSilenceCompression	0 = Disable. 1 = Enable. 2 = Enable without adaptation	Silence Suppression mode
EnableStandardSIDPayload Type	0 = Disable = default, 1 = Enable	If set to 1, the SID packets are sent with the RTP SID type (RFC 3389)
EnergyDetectorQualityFactor	0 to 10	Determines the Energy Detector's sensitivity (0 -10). 0 means Insensitive; 10 means Most Sensitive.
EnergyDetectorThreshold	0 to 7	A signal that crosses the threshold, invokes a suitable 'Above' or 'Below' Threshold event. ActualThreshold = -44 dBm + (EDThreshold * 6).

Table 6-5: MGCP/MEGACO Default Channel Parameters
(continued on page 94 to page 96)

<i>ini</i> File Field Name	Valid Range	Description
FaxBypassPayloadType	0 = 127 default = 102	User can modify the Fax Bypass mode RTP packets Payload type by setting this parameter. In case of congestion (the selected payload type already is used for other coders/modes) - a TP_SETUP_PARAMETER_INVALID_ERROR is issued and the Payload type is set to the default value (102). It's the User's responsibility avoid congestion with other payload types.
FaxModemBypassCoder Type	Legal coders (default = G.711 A-law)	The User can set the bypass coder (according to the known coders)
FaxModemBypassM	1, 2 (default =1)	Number of 20 msec payloads for generating one RTP fax/modem bypass packet.
FaxModemRelayVolume	0 to 15 (default = 6)	Fax gain control. The range 0-15 relates to -18.5 dBm to -3.5 dBm in 1 dBm steps.
FaxRelayECMEnable	0 = Disable 1 = Enable = default	Enables or disables the using of ECM mode during Fax Relay.
FaxRelayEnhanced RedundancyDepth	0 to 4 (default =0)	Number of repetitions to be applied to control packets when using T.38 standard.
FaxRelayMaxRate	0 = 2400 bps 1 = 4800 2 = 7200 3 = 9600 4 = 12000 5 = 14400 = default	Limits the maximum rate at which fax messages are transmitted.
FaxRelayRedundancy Depth	0 to 2 (default = 0)	Amount of redundancy for fax relay packets
FaxTransportMode	0 =Transparent 1 = relay = default 2 = bypass 3 = Transparent with Events	Sets the Fax transport method.
ForceEchoOff	Refer to the 'EnableEchoCanceller' field name	
ForceSilence Suppression	Refer to the 'EnableSilenceCompression' field name	
IBSDetectionRedirection	0 = PCM 1 = Network	Determines the IBS Detection Direction
InputGain	-32 to 31 (default = 0 = 0 dB).	PCM input gain. The range is 31 dB to +31 dB in 1 dB steps.
MFTransportType	0 = Mute 1 = -Proprietary relay = default 2 = Transparent 3 = RFC 2833 relay	0 = erase MFs from voice transport, not relayed to remote. 1 = erase MFs, relay to remote using proprietary relay method. 2 = MFs not erased, not relayed to remote. 3 = MFs are muted from the voice stream & relayed according to RFC 2833.
ModemRelayMaxRate	-	Not used

Table 6-5: MGCP/MEGACO Default Channel Parameters
(continued on page 94 to page 96)

<i>ini</i> File Field Name	Valid Range	Description
ModemRelayRedundancy Depth	0 to 2 (default = 0)	Not used – should be set to 0.
RFC2198PayloadType	96 to 101, 106 to 127	This parameter sets the RFC 2198 (RTP Redundancy) packet's RTP Payload Type field
RFC2833PayloadType	96 to 101, 106 to 127 (default = 96)	The range of the RFC 2833 Payload Type is {96 to 101, 106 to 127}. Any other values than this range, results in an error, and setting the Payload Type to default. The range (96 to 127) is the Dynamic RTP Payload type range for all kinds of non-hard-coded RTP Payload types (refer to RFC 1890).
RTPRedundancyDepth	0 = Disable = default 1 = Enable	Enables or disables generation of RFC 2198 redundancy packets.
TestMode	0 = acCoder Loopback 1 = PCM Loopback 2 = ToneInjection 3 = NoLoopback = default	Testing mode applied. Coder Loopback performs an encoder-decoder loopback inside DSP device. PCMLoopback loops back incoming PCM to the outgoing PCM. ToneInjection generates a 1000 Hz tone to outgoing PCM. NoLoopback sets the channel to work in a normal mode.
UserDefinedToneDetector Enable	0 = Disable = default 1 = Enable	0 = Disable User Defined Tones Detector 1 = Enable User Defined Tones Detector
UseT38orFRF11	0 = T.38; 1 = FRF.11 = default	Selects protocol to send/ receive fax relay.
V21ModemTransportType	0 = Transparent = default 2 = bypass 3 = Transparent with Events	Sets the V.21 modem transport method (must be set to 0 - disable)
V22ModemTransportType	0 = Transparent 2 = bypass = default 3 = Transparent with Events	Sets the V.22 modem transport method
V23ModemTransportType	0 = Transparent 2 = bypass = default 3 = Transparent with Events	Sets the V.23 modem transport method
V32ModemTransportType	0 = Transparent 2 = bypass = default 3 = Transparent with Events	Sets the V.32 modem transport method
V34ModemTransportType	0 = Transparent 2 = bypass = default 3 = Transparent with Events	Sets the V.34 modem transport method

**Table 6-5: MGCP/MEGACO Default Channel Parameters
(continued on page 94 to page 96)**

<i>ini</i> File Field Name	Valid Range	Description
VoicePayloadFormat	0/1/2 (default = 0)	Sets RTP Payload Format. When = 1, the RTP Payload Format is ATM. Enables working with Vendors that use G.726 ATM Payload Format over RTP. Use enum acTVoicePayloadFormat.
VoiceVolume	-32 to 31 (default = 0 = 0 dB)	Voice gain control; the range is -31 dB to +31 dB in 1 dB steps, -32 = mute.

6.3 *ini* File Structure

The *ini* file can contain any number of parameters, that are divided into groups by their functionality. The general form of the *ini* file is shown in Figure 6-1.

Figure 6-1: *ini* File Structure

```
[Sub Section Name]

Parameter_Name = Parameter_Value
Parameter_Name = Parameter_Value
.
..

; REMARK

[Sub Section Name]
.
```

6.3.1 *ini* File Structure Rules

- Lines beginning with a semi-colon (;) (as the first character) are ignored.
- Carriage Return must be the final character of each line.
- Number of spaces before and after "=" is not relevant.
- If there is a syntax error in the parameter name the value is ignored.
- Syntax errors in the parameter value field can cause unexpected errors (because parameters may be set to the wrong values).
- Sub-section names are optional.
- The values of String parameters, for example VoicePromptsFileName, should be placed between two inverted commas ('...').
- The parameter field is NOT case sensitive.
- Parameter values should be entered only in decimal format, except for the Call Agent IP address.
- The *ini* file should be ended with one or more carriage returns.

6.3.2 *ini* File Example

An example of an *ini* file is shown in Figure 6-2.

Figure 6-2: Example of an *ini* File

```
[TDM BUS configuration]
; 1=aLaw 3=ulaw
PCMLawSelect = 1
BaseUDPPort = 4000
[Trunk Configuration]
; E1_euro_isdn
ProtocolType = 1
; USER_TERMINATION_SIDE
TerminationSide = 0
; EXTENDED_SUPER_FRAME
FramingMethod = 0
;HDB3
LineCode = 2
[MGCP]
EndpointName = 'ACgw'
CallAgentIP = 10.1.2.34
[Channel Params]
DJBufferMinDelay = 75
RTPRedundancyDepth = 1
[Files]
CallProgressTonesFilename = 'CPUSA.dat'
BurnCallProgressTonesFile = 1
VoicePromptsFilename = 'tpdemo_723.dat'
BurnVoicePromptsFile = 0
CasFilename = 'E_M_WinkTable.con'
BurnCasFile = 1
Trace Level = 0
```

An example of an *ini* file for MEGACO is shown in Figure 6-3.

Figure 6-3: ini File Example for MEGACO

```

[MEGACO]

; List of Call agents, separated by ','. The default is the load ing
computer.
PROVISIONEDCALLAGENTS = 10.2.1.254

; List of ports for the above Call Agents, separated by ','. The
default is 2944.
PROVISIONEDCALLAGENTSSPORTS = 2944

; IP of the LOG server
LOGSERVERIP = 10.2.1.254

; Disables the watchdog. If set to one, the board will reset when
stuck.
WATCHDOGSTATUS=0

; The next 3 field are the parts of the termination name. For exam
ple, if you wish the
; name to look like: 'gw3/tr0/ep1', you shall define: ;GATEWAYNAME =
'gw3/',
; TRUNKNAME = 'tr', and ENDPOINTNAME = 'ep' . The RTP is built from
the GATEWAYNAME,

; the string 'RTP' and a number. In this example it shall be: gw3/
RTP0.;
GATEWAYNAME = 'c4'
TRUNKNAME = 's'
ENDPOINTNAME = 'c'; This parameter activates the MEGACO!!!! If omit-
ted, MGCP will be active
MGCONTROLPROTOCOLTYPE = 2;; The following disables the keep-alive
mechanism if set to 0, else it is enabled
KEEPALIVEENABLED = 1

;

; This parameter defines the profile used. 1 - is for version 12, 2 -
for version 1
MGCPCOMPATIBILITYPROFILE = 1
    
```

Note 1: Using Windows Properties Display, verify that the MS-DOS name of the *ini* file is *mediant.ini*, and NOT erroneously *mediant.ini.ini* or *mediant~.ini*.

Note 2: To restore **Mediant 2000** default configuration parameters, use the *mediant.ini* file without any valid parameters or with a semicolon (;) character preceding all lines in the file.

6.4 Using Call Progress Tones

The Call Progress Tones configuration file contains the definitions of the Call Progress Tones to be detected/generated by the **Mediant 2000**. Users can use either one of the configuration files supplied by AudioCodes or construct their own file.

The Call Progress Tones configuration file used by the **Mediant 2000** is a binary file (with the extension `.dat`). Users can construct their own configuration file by starting from `tone.ini` file format, then modifying the file, and finally converting it into binary format using the “Download conversion utility” supplied with the **Mediant 2000** package.

Click the “Convert dBm values” checkbox in the Download conversion utility.

To download the Call Progress Tones File to the **Mediant 2000**, a correct definition should be used in the *ini* file. Refer to Section 6.4.3 on page 104 for the description of the procedure on how to generate and download the Call Progress Tones file.

6.4.1 Format of the Call Progress *ini* File

The Call Progress Tones section of the *ini* file starts from the following string:

- **[NUMBER OF CALL PROGRESS TONES]** – containing the following key only:
 - Number of Call Progress Tones – defines the number of Call Progress Tones to be 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 type
 - 1 Dial Tone
 - 2 Ringback Tone
 - 3 Busy Tone
 - 4 Congestion Tone
 - 5 Special Information Tone
 - 6 Warning Tone
 - 7 Reorder Tone
 - 8 Confirmation Tone
 - 9 Call Waiting Tone
 - Low Freq [Hz] – Frequency in hertz of the lower tone component in the case of a dual frequency tone, or the frequency of the tone in the case of a single tone.
 - High Freq [Hz] – Frequency in hertz of the higher tone component in the case of a dual frequency tone, or zero (0) in the case of a single tone.
 - Low Freq Level [-dBm] – Generation level 0 to – 31 dBm
 - High Freq Level – Generation level 0 dBm to – 31 dBm. The value should be set to ‘32’ (mute) in the case of a single tone.
 - First Signal On Time [10 msec] – “Signal On” period (in 10 msec units) for the first cadence on-off cycle.
 - First Signal Off Time [10 msec] – “Signal Off” period (in 10 msec units) for the first cadence on-off cycle.
 - Second Signal On Time [10 msec] – “Signal On” period (in 10 msec units) for the second cadence on-off cycle.
 - Second Signal Off Time [10 msec] – “Signal Off” period (in 10 msec units) for the second cadence on-off cycle.

Using this configuration file, Users can create up to 16 different Call Progress Tones using up to 15 different frequencies (in the range of 300 Hz to 2000 Hz). Each of the Call Progress Tones is specified by the following two parameters: the Tone Frequency (either single or dual frequencies are supported) and Tone Cadence. This is specified by 2 sets of ON/OFF periods, but Users can discard the use of the first On/Off cycle by setting the relevant parameters to zero. When the tone is made up of a single frequency, the second frequency field should be set to zero.

For a continuous tone (such as dial tone), only the “First Signal On time” should be specified. In this case, the parameter specifies the detection period. For example, if it equals 300, then the tone is detected after 3 seconds (300 x 10 msec).

Note 1: When defining several continuous tones, the “First Signal On Time” parameter should have the same value for all tones.

Note 2: The tones frequency should differ by at least 40 Hz from one tone to other defined tones.

6.4.2 Default Template for Call Progress Tones

The **Mediant 2000** is initialized with the Default Call Progress Tones Configuration Template shown in Figure 6-4. If you need to change one of the tones, edit the default call progress.txt file.

For example: to change the dial tone to 440 Hz only, replace the #Dial tone section in Figure 6-4 with the following text:

#Dial tone

[CALL PROGRESS TONE #1]

Tone Type=1

Low Freq [Hz]=440

High Freq [Hz]=0

Low Freq Level [-dBm]=10 (-10dBm)

High Freq Level [-dBm]=32 (use 32 if only 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

Users can specify several tones of the same type using Tone Type definition. These additional tones are used only for tone detection. Generation of a specific tone is according to the first definition of the specific tone. For example, Users can define an additional dial tone by appending the second dial tone definition lines to the tone *ini* file. The **Mediant 2000** reports dial tone detection if either of the two tones is detected.

Figure 6-4: Default Call Progress Tone Template (continued to page 104)

```
[NUMBER OF CALL PROGRESS TONES]
Number of Call Progress Tones=9

#Dial tone
[CALL PROGRESS TONE #0]
Tone Type=1
Low Freq [Hz]=350
High Freq [Hz]=440
Low Freq Level [-dBm]=13 (-13dBm)
High Freq Level [-dBm]=13
First Signal On Time [10msec]=300
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=0
Second Signal Off Time [10msec]=0

#Dial tone
[CALL PROGRESS TONE #1]
Tone Type=1
Low Freq [Hz]=440
High Freq [Hz]=0
Low Freq Level [-dBm]=10 (-10dBm)
High Freq Level [-dBm]=0
First Signal On Time [10msec]=300
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=0
Second Signal Off Time [10msec]=0

#Ringback
[CALL PROGRESS TONE #2]
Tone Type=2
```

Figure 6-4: Default Call Progress Tone Template (continued to page 104)

```

Low Freq [Hz]=440
High Freq [Hz]=480
Low Freq Level [-dBm]=19 (-19dBm)
High Freq Level [-dBm]=19
First Signal On Time [10msec]=0
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=200
Second Signal Off Time [10msec]=400

#Ringback
[CALL PROGRESS TONE #3]
Tone Type=2
Low Freq [Hz]=440
High Freq [Hz]=0
Low Freq Level [-dBm]=16 (-16dBm)
High Freq Level [-dBm]=0
First Signal On Time [10msec]=0
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=100
Second Signal Off Time [10msec]=300

#Busy
[CALL PROGRESS TONE #4]
Tone Type=3
Low Freq [Hz]=480
High Freq [Hz]=620
Low Freq Level [-dBm]=24 (-24dBm)
High Freq Level [-dBm]=24
First Signal On Time [10msec]=0
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=50
Second Signal Off Time [10msec]=50

#Busy
[CALL PROGRESS TONE #5]
Tone Type=3
    
```

Figure 6-4: Default Call Progress Tone Template (continued to page 104)

```
Low Freq [Hz]=440
High Freq [Hz]=0
Low Freq Level [-dBm]=20 (-20dBm)
High Freq Level [-dBm]=0
First Signal On Time [10msec]=0
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=50
Second Signal Off Time [10msec]=50

#Reorder tone
[CALL PROGRESS TONE #6]
Tone Type=7
Low Freq [Hz]=480
High Freq [Hz]=620
Low Freq Level [-dBm]=24 (-24dBm)
High Freq Level [-dBm]=24
First Signal On Time [10msec]=0
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=25
Second Signal Off Time [10msec]=25

#Confirmation tone
[CALL PROGRESS TONE #7]
Tone Type=8
Low Freq [Hz]=350
High Freq [Hz]=440
Low Freq Level [-dBm]=20 (-20dBm)
High Freq Level [-dBm]=20
First Signal On Time [10msec]=0
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=10
Second Signal Off Time [10msec]=10

#Call Waiting Tone
[CALL PROGRESS TONE #8]
Tone Type=9
```

Figure 6-4: Default Call Progress Tone Template (continued to page 104)

```

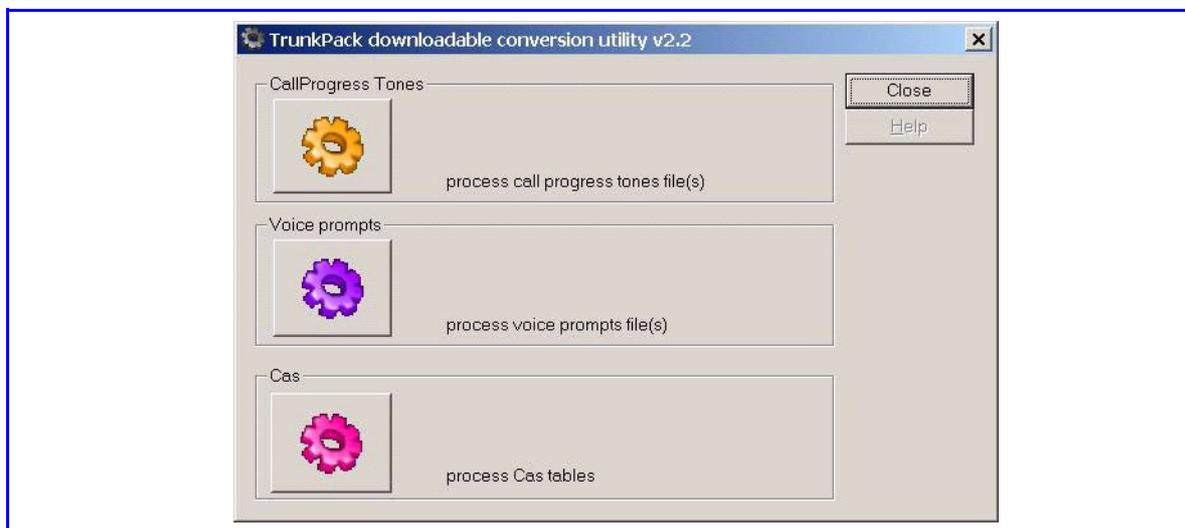
Low Freq [Hz]=440
High Freq [Hz]=0
Low Freq Level [-dBm]=20 (-20dBm)
High Freq Level [-dBm]=0
First Signal On Time [10msec]=0
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=30
Second Signal Off Time [10msec]=900
    
```

6.4.3 Call Progress Tone Generation and Download Procedure

Follow the directions below to generate and download the Call Progress Tone file.

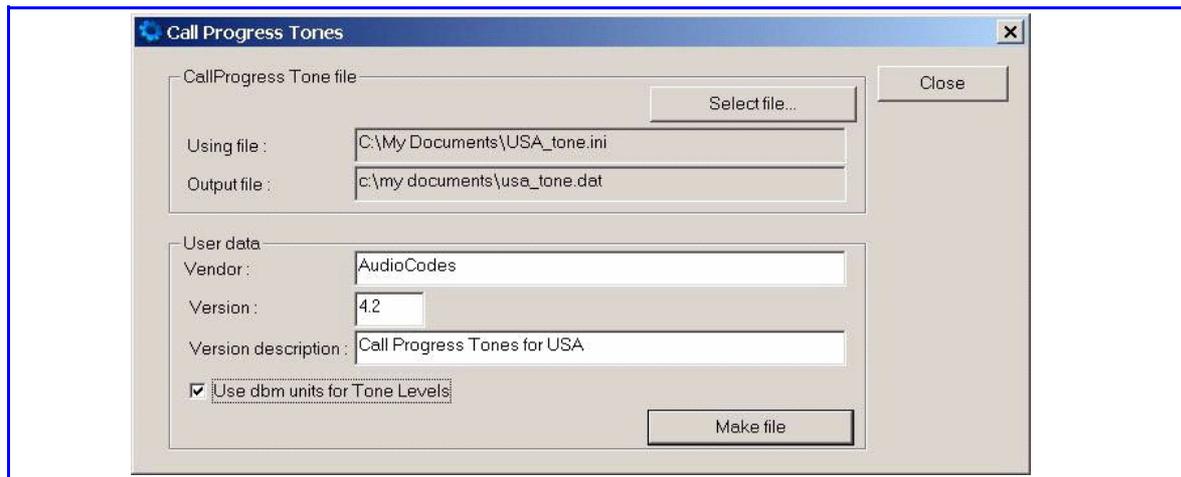
➤ **To generate the Call Progress Tone file, take the following 8 steps:**

- Step 1.** Prepare the tone.ini file.
- Step 2.** Use the “Download conversion utility” to generate a binary tone.dat file.

Figure 6-5: Download Selection Screen


- Step 3.** Click “Process a new file...”
- Step 4.** Select an input file such as usa_tone.ini and fill the Vendor and Version fields.

Figure 6-6: File Selection Screen



Step 5. Click the 'Convert Code Values into dBm' checkbox.

Step 6. Click the "Make File" button and close the application.

Step 7. Edit the *Mediant.ini* file and add the following two lines:

```
CallProgressTonesFilename = 'usa_tone.dat'
```

```
BurnCallProgressTonesFile = 1
```

Step 8. Save the "*usa_tone.dat*" and "*Mediant.ini*" files in TFTP folder

The files can now be loaded to the board using the TFTP or BootP servers, or the Web browser.

Reader's Notes

7 Diagnostics

AudioCodes provides a wide range of diagnostic tools to enable Users to easily identify an error condition and to provide a solution or work around.

Troubleshooting the Mediant 2000 system centers on the TP-1610board.

The following sections describe alternative AudioCodes diagnostics:

- **Troubleshooting**
- **Alarms**
- **Control Protocol Reports**
- **SysLog Support**
- **Solutions to Problems**

7.2 TP-1610 Troubleshooting

To troubleshoot the TP-1610, simply reinitialize it using the AudioCodes BootP/TFTP Server. For a full description of the AudioCodes BootP/TFTP Server, refer to Appendix C, 'AudioCodes BootP/TFTP Server Utility' on page 137.

7.2.1 Reinitializing the TP-1610

➤ **Before reinitializing the TP-1610, take the next 5 steps:**

- Step 1.** Ensure the board is correctly physically installed (see Section 2, 'Hardware Installation' on page 21).
- Step 2.** Ensure the board is connected to the network if the re-initialization process is network based. If the re-initialization process is based on the PCI interface and not on the network interface, no network connection is necessary.
- Step 3.** If the re-initialization process is based on the PCIinterface, ensure the board's download driver file is correctly placed in the correct path (refer to Section 3.1, 'Software Installation' on page 33).
- Step 4.** If the re-initialization process is network-based, ensure the AudioCodes BootP/TFTP Server was installed on your PC; if not, install it from the AudioCodes Software CD supplied with the TP-1610. (Refer to Appendix C, 'AudioCodes BootP/TFTP Server Utility' on page 137.)
- Step 5.** Make sure you've copied the correct *.cmp and *.ini configuration files to your c:/ drive. (Refer to Appendix C, 'AudioCodes BootP/TFTP Server Utility' on page 137.)

➤ **To reinitialize the board via the network, take the next 7 steps:**

- Step 1.** Open the AudioCodes BootP/TFTP Server from Start>Programs (refer to Section 3, 'Software Installation' on page 33.)
- Step 2.** In the **Preferences** screen, navigate to the directory under which you located the *.cmp and *.ini files.
- Step 3.** In the **Client Configuration** screen (Services menu>Clients), click the **Apply & Reset** button. The Client MAC address is detected and automatically displayed.
- Step 4.** Configure the parameters IP address, Subnet, Gateway.
- Step 5.** In the Boot File field, select the *.cmp file.

Step 6. In the *ini* File field, select the *.ini file.

Step 7. Click **OK**. You're returned to the AudioCodes BootP/TFTP Server main screen. View the status message pane and verify that "Client Enabled" is displayed.

To reinitialize the board via the PCI, refer to the GUI Demo Application in Appendix H of AudioCodes' "Software Utilities Manual", Document #: LTRT-00702.

7.3 Alarms

7.3.1 LED Visual Indicator Alarms

Table 7-1: Chassis Indicators

Placement	Color	Function
Right side of front panel	Green	Power is on
Right side of front panel	Red	Fan failure - indicates that any of the internal fans has significantly reduces its speed or has gotten stuck
Left side of front panel	Red	Power supply failure - indicates that one of the two AC redundant power supplies is faulty

Table 7-2: Board Status LED Indicators

Label	Color	Function
FAIL	Red	Normally OFF; Red shows board failure (fatal error)
ACT	Green	Board initialization sequence terminated OK
	Yellow	N/A

The FAIL LED is normally OFF, and turns to Red to indicate board failure.

The ACT LED is lit Green as soon as download is completed successfully.

Table 7-3: Trunks Status LED Indicators

Label	Color	Signal Description
(Currently not in use)	Green	Trunk is synchronized (normal operation)
	Red	Loss due to one of the following 4 signals:
	LOS	Loss of Signal
	LFA	Loss of Frame Alignment
	AIS	Alarm Indication Signal (The blue alarm)
	RAI	Remote Alarm Indication (The yellow alarm)

Table 7-4: Ethernet LED Indicators

Label	Color	Function
LINK	Green	Link all OK
ACT	Yellow	Transmit/Receive Activity

During correct **TP-1610** operation, the Ready LED is lit Green; changing to Red indicates a failure.

DATA LED indicates RTP activity; it will flash if RTP voice packets are transmitted or received over the network. The CTRL LED indicates activity of MGCP messages received from the Call Agent/GW controller.

In the correct **TP-1610** operation the E1/T1 Sync LED is lit Green and the E1/T1 alarm LED is turned off. Any other condition indicates failure, either in the E1/T1 cable, in the **TP-1610** or at the remote side.

Table 7-5: Auxiliary LED Indicators

Label	Color	Function
PWR	Green	Power is supplied to the board
SWAP READY	Blue	The board can be removed or inserted

Before removing the board, wait for the Blue LED to flash ON and then stay OFF

7.3.2 TP-1610 Self-Testing

The **TP-1610** features two self-testing modes: **rapid** and **detailed**.

Rapid self-test mode is used each time the **TP-1610** completes the initialization process. (Refer to Section Section 7.2.1 on page 107 “Reinitializing the **TP-1610**”.) This is a short test phase in which the only error detected and reported is failure in initializing hardware components. All Status and Error reports in this self-test phase are reported through the PCI Interface and Network Interface ports, as well as indicated by the LED Status Indicators.

Detailed self-test mode is used when initialization of the **TP-1610** is completed and if the configuration parameter EnableDiagnostics is set to 1 (this parameter can be configured through the *ini* file mechanism). In this mode, the **TP-1610** tests all the hardware components (memory, DSP, etc., the difference being, when EnableDiagnostics is set to 1, flash is tested thoroughly, and when EnableDiagnostics is set to 2, flash is partially tested), outputs the status of the test results, and ends the test. To continue operational running, reset the board but this time configure the EnableDiagnostics parameter to 0.

7.4 Control Protocol Reports

7.4.1 Error Indications through the Network Interface

7.5 SysLog Support

7.5.1 SysLog Overview

SysLog protocol is an event notification protocol that allows a device to send event notification messages across IP networks to event message collectors - also known as SysLog servers.

Since each process, application and operating system was written somewhat independently, there is little uniformity to SysLog messages. For this reason, no assumption is made upon the contents of the messages other than the minimum requirements of its priority.

SysLog uses UDP as its underlying transport layer mechanism. The UDP port that has been assigned to SysLog is 514.

The SysLog message is transmitted as an ASCII 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.

Example:

```
02/10 16:39:56.907 : 10.4.229.18 : INFO :PhysTerms#: 192, Ephem-
Term#: 360 TotalPoolsSize: 4216132 Retransmission size: 544768
Average perchannel: 17249 [File:mcMain.cpp Line:2118]
```

```
02/10 16:39:56.891 : 10.4.229.18 : INFO :PhysChn#=192 EPName: c
GWName: gw TRName: s NofTr=8 ChnPreTrnk=24 [File:mcMain.cpp
Line:2109]
```

```
02/10 16:39:53.157 : 10.4.229.18 : WARNING :DSP Version Info:
acNumberOfDSPs=60, acNumberOfDSPChannels=240 [File:Main.cpp
Line:725]
```

7.5.2 SysLog Operation

7.5.2.1 Sending the SysLog Messages

The AudioCodes application performs as a SysLog client. Messages (currently Error Reports events) generated by AudioCodes application are sent via UDP and IP to a SysLog server application. AudioCodes does **NOT** provide a SysLog server application.

A typical SysLog server application enables filtering of the messages according to priority, IP sender address, time, date, etc.

Such applications can easily be found on the web and most of them are free. Examples of SysLog Servers downloadable from the Internet are:

- 1 Kiwi Enterprises: http://www.kiwi-enterprises.com/software_downloads.htm

- 2 The US CMS Server: http://uscms.fnal.gov/hanlon/uscms_server/
- 3 TriAction Software: <http://www.triaction.nl/Products/SyslogDaemon.asp>
- 4 Netal SL4NT 2.1 Syslog Daemon: <http://www.netal.com>

7.5.2.2 Setting the SysLog Server IP Address

A SysLogServerIP Address parameter is supplied via an *ini* file in order to determine the address of the SysLog server.

7.5.2.3 Controlling the Activation of the SysLog Client

The activation of the SysLog client is controlled by an EnableSyslog *ini* file parameter. Setting it to 1 enables SysLog protocol log.

7.5.2.4 *ini* File Example for SysLog

Figure 7-1: The *ini* File Example for SysLog

```
[Syslog]
SyslogServerIP=10.2.0.136
EnableSyslog =1
```

7.6 Solutions to Problems

7.6.1 Possible Common Problems

Solutions to possible common problems are described in Table 7-6.

Table 7-6: Solutions to Common Problems (continued on page 112)

Problem	Possible Cause	Solutions
No communication	Software does not function in the device	Try to “ping” to the device. If ping fails, check for network problems/definitions and try to reset the device
	Network problem	Check cables.
	Network definitions	Check if default gateway can reach IP of the board.
		Check if the board got the correct IP.
		Check the validity of IP address, subnet and default gateway. If default gateway is not used, enter 0.0.0.0
	BootP didn't reply to board	Check if BootP server replied to the board at restart. See it in the log of BootP server.
		Try to restart BootP server.
Check the MAC address of the board in BootP server.		

Table 7-6: Solutions to Common Problems (continued on page 112)

Problem	Possible Cause	Solutions
<i>ini</i> file was not loaded	TFTP server down	Check if TFTP server working.
	TFTP server didn't get the request	Check this in its log.
	TP-1610 didn't request the file from your TFTP	Check that the TFTP server IP address is that the TP-1610 is trying to use.
	TFTP server bug	Try to restart TFTP server.
	BootP sent to a board with the wrong TFTP server address	Check the address of TFTP being used.
	<i>ini</i> file does not exist in default directory of TFTP	Check default directory of TFTP server and check that <i>ini</i> file exists there.
	Wrong <i>ini</i> file name	Verify in windows explorer that file extensions are displayed and the <i>ini</i> file isn't by mistake " <i>XXX.ini.ini</i> ". Verify that extension <i>ini</i> is in lowercase letters.
	TFTP's timeout setting is too short	Verify that: Timeout = 5 sec, # of retransmission = 10
Wrong <i>ini</i> file loaded	<i>ini</i> file is not in the correct position	Old <i>ini</i> file was probably loaded. Check which <i>ini</i> file was loaded. This can be done using the SysLog server. The Gateway displays contents of <i>ini</i> file before it began.
	<i>ini</i> file corrupted	check <i>ini</i> file syntax
BootP reply from wrong BootP server	Other BootP servers contain MAC address of board	Check that only your BootP server contains TP-1610 MAC addresses.

7.6.2 Possible Voice Problems

Solutions to possible voice problems are described in Table 7-7

Table 7-7: Solutions to Possible Voice Problems

Problem	Possible Cause	Solutions
G.711 voice quality is bad (clicks)	Silence compression not compatible (when working with different Gateway other than AudioCodes Gateway)	Disable it and check if quality is better.
	Packet size not compatible (with G.711)	Check that packet period in remote side is 20 msec. Check that the correct μ -law or A-law compression is in use
No voice	There is no match in codecs	Change codec definition.

8 Functional Specifications

8.1 Mediant 2000 Selected Specifications

Table 8-1: Selected Specifications (continues until page 117)

Item	Characteristic
Capacity (Max)	
Network Ports/DSP Calls (independent digital voice, fax or data ports)	480 All media processing ports can be tied to IP-RTP, PSTN-DS0 or H.110 Time Slots independently
Configurations	
DSP Channel Configuration	240, 480 ports
Media Processing	
Voice Messaging	Host-based record/play, *.wav format (G.711Alaw, G.711MuLaw, G.711Linear, MS-GSM, GSM-6.10, G.723Low, G.726-16, G.726-24, G.726-32, G.726-40, OKI-ADPCM) Depended on the supported coder in the board. Up to full board channel capacity ports of record/playback - PCI bus based (Coder depended) Playback to/Record from PSTN side PSTN / TDM to *.wav (per coders listed above) PSTN / TDM to LBR raw data
IP Transport	VoIP (RTP/ RTCP) per IETF RFC 1889 and RFC 1890
Packet interface	Integral NIC (IP) or PCI; Selectable per port
DTMF/MF Relay	DTMF/MF Relay over RTP per RFC 2833
DTMF/MF Modes	DTMF/MF Relay, Mute, transfer in coder as voice.

Table 8-1: Selected Specifications (continues until page 117)

Item	Characteristic
Voice Processing	Supports all ports
	Prompts and Announcements playback (10 MB integral memory for 20 min of G.711 or 200 min for G.723 recorded prompts)
	Dynamic Network Jitter Buffer
	Call Progress Tones generation and detection
	RTP recorded packet correction
	Transcoding of G.711 RTP to any Low Bit-Rate Coder RTP stream using one DSP ch resource
	IP to/from IP mediation using same coder without using DSP channel resource
	Media duplication using the same coder without using additional DSP channel resources
Gain Control	
Output Gain Control	Programmable: -31 dB to +31 dB in steps of 1 dB
Input Gain Control	Programmable: -31 dB to +31 dB in steps of 1 dB
Voice Compression (Independent dynamic vocoder selection per channel)	G.711 PCM, 64 kbps (μ-law/A-law)
	G.726/G.727 ADPCM/E-ADPCM (16 to 40 kbps)
	OKI-ADPCM
	G.723.1 MP-MLQ, 6.3 kbps ACELP, 5.3 kbps
	G.729A CS-ACELP, 8.0 kbps
	GSM 6.10, 13 kbps
	NetCoder at 6.4 to 9.6 kbps, 800 bps steps
Silence Suppression Voice Activity Detection (VAD) Comfort Noise Generation (CNG)	G.723.1 Annex A
	G.729 Annex B
	PCM and ADPCM – Per RFC 3389 or Proprietary
	NetCoder
	GSM 6.10
Echo Cancelation	G.165 & G.168-compliant 32, 64, 128 msec echo tail (64 and 128 may reduce channel capacity)
Fax and Modem Messaging	
Fax Relay / ByPass	Supports all ports per board (per board capacity)
	Group 3 real-time Fax Relay to 14.4 kbps with auto fallback
	Tolerant of delays of up to 9 seconds
	T.30 (PSTN) and T. 38 (IP) compliant (real-time fax)
	CNG tone detection & Relay per T.38
	Auto Fax ByPass to G.711 or ADPCM

Table 8-1: Selected Specifications (continues until page 117)

Item	Characteristic
Modem	Auto-switch to PCM or ADPCM for modem signals (V.34 or V.90 modem detection)
Signaling	
In-band/Out-of-band Signaling (DTMF & Tone Detection/ Generation)	DTMF per TIA 464B
	DTMF over RTP per RFC 2833
	MFC-R2, MF-R1, MF-R1 (US) including FG-A/B/D
	Packet side or PSTN side generation/detection of DTMF and User Defined Call Progress Tones (PSTN, IP) & Country Test Tones (per ITU-t Q.724)
CAS Relay	ABCD signaling over RTP per RFC 2833
SIGTRAN	IUA over SCTP per RFC 3057/2960 SS-7 MTP-2, MTP-3 link termination M2UA, M3UA over SCTP
PSTN Protocols	CAS - T1 robbed bit: WinkStart, delay dial, immediate start, FGB, FGD, etc. MFC/R2 numerous country variants Unique script for each county variant, enabling maximum flexibility of the entire state machine of each CAS protocol.
	CCS - ISDN PRI: ETSI EURO ISDN, ANSI NI2, DMS, 5ESS, Japan INS1500, QSIG Basic Call, Australian Telecom, New Zealand Telecom, Hong Kong Variant, Korean MIC
Management Interfaces	
SNMP V2	Standard MIB-2, RTP MIB, Trunk MIB, AudioCodes' proprietary MIBs
Embedded Web Server	Enabling device configuration and run-time monitoring with an Internet browser
Firmware Download Options	
Firmware download	Remote TFTP or Web or via PCI
Boot option (for network control)	Locally from Flash or Remote Software download (TFTP & BootP support)
Control Protocols	
MGCP (RFC 2705)	Call control, CAS package, Basic announcements package, Conferencing
MEGACO (H.248)	Call control, CAS2 and Rpackage, Basic announcements package, Conferencing
TPNCP	AudioCodes proprietary TPNCP (Via Ethernet or PCI/cPCI interface)
Processor	
Control Processor	Motorola PowerQUICC 8260
Control Processor Memory	SDRAM – 64 MB

Table 8-1: Selected Specifications (continues until page 117)

Item	Characteristic
Signal Processors	AudioCodes AC486 VoIP DSP based on: TI DSP TMS5541 – each core at 133 MHz
Interfaces	
Ethernet	Dual redundant 10/100 Base-T ports RJ-45 connectors off rear I/O, PICMG 2.16 cPSB backplane for media streaming and call control. Half or Full duplex with auto-negotiation
TDM	H.110 CT bus interface PICMG 2.5 Rev 1.0 compliance, double loaded(The board includes two H.110 loads)
TDM Bus Speed	H.110 - 4096 Time Slots using 8 MHz
PSTN	Up to 16 E1 or 16 T1 spans Up to 16 E1 or 16 T1 spans via RTM rear panel I/O module two 50-pin Telco connectors (DDK 57AE-40500-21D), each handling up to 8 E1/T1/J1 ports. Optionally 2, 4 or 8 spans with RJ-48 shielded connectors
PCI Bus	33 MHz, 32 bit, slave mode (PICMG 2.0 revision 2.1)
Physical	
Physical	6U single cPCI slot. PICMG 2.0, R2.1 and R2.16 and R.3.0 CompactPCI™ card
Supply Voltages and Power Consumption (typical)	<u>Universal 90 - 260 VAC, 2A, 47-63 Hz</u> <u>(optional: -48 VDC nominal)</u> <u>(Optional dual AC power supply)</u> DC Power Connector- MSTB2.5/2-STF of Phoenix Contact
Environmental	Operational: 0° to 45° C 32° to 113° F Storage: -10° to 70° C 14° to 158° F Humidity: 10 to 90% non-condensing
Hot Swap	Full hot swap supported
Host Interface	Via cPCI bus, using AudioCodes' libraries Via Packet interface using AudioCodes' proprietary TPNCIP or standard control protocols
Enclosure Dimensions	1U, 19-inch rack mount, shelf or desk top, 2-slot cPCI chassis 44.5 x 445 x 300 mm; 1.75 x 17.5 x 12 inch (h x w x d)
Type Approvals	
Telecommunication Standards	Canadian Telecom IC CS03 (approved DC only) FCC part 68 European ISDN PRI, CTR4, CTR12 & CTR13. (Designed to meet, approval pending.)
Safety and EMC Standards	UL 60950, FCC part 15 Class B, CE Mark (EN55022, EN60950, EN55024, EN300 386)
Environmental	NEBS Level 3: GR-63-Core, GR-1089-Core, Type 1&3, ETS300 019 (approved) For DC powered version
LED Indicators	

Table 8-1: Selected Specifications (continues until page 117)

Item	Characteristic
LED Indications on Front Panel	Power, Ready/Fail T1/E1 status, LAN status Swap ready indication
Connectors and Switches	
Rear Panel	Power: (model depended) - Standard AC power socket or - DC power - MSTB2.5/2-STF of Phoenix Contact E1/T1 Trunk and Ethernet connectors
Front Panel	HW Reset button
Diagnostics	
BootP/TFTP Server Configuration Tool	Used to upload firmware
Front panel LEDs	Provide visual status indications and alarms
SysLog events	Supported by SysLog servers
Operating Systems	
O/S Support	Windows™ NT, Windows™ 2000, Windows™ XP
	Unix™ Solaris™ (Sparc™ or Intel™ machines), Unix™
	Linux™ (Intel™ machines)
	OS wrapper to support other operating systems

Reader's Notes

Appendix A Mediant 2000 Configuration Guide for Nortel H.248 Application

A.1 Purpose

The purpose of this document is to provide information about the configuration settings for the AudioCodes Mediant 2000 as used in the Nortel Networks CS2K and SL100 solutions. This document discusses the default settings that are set for each Mediant 2000 and the basic configuration steps and parameters to use in setting up the various control protocols and TDM side configurations. This documentation details the provisioning method to be used by Nortel Networks to reduce or eliminate the need for manual changes to the initialization file.

A.2 Introduction

The AudioCodes product lines are used in a wide variety of applications. In order to have the flexibility to handle this, configuration parameters are used to setup the system for its role in the different applications. For the Nortel Networks applications, AudioCodes has worked closely with Nortel Networks designers to define values for a set of parameters that allow interworking between the Mediant 2000 and Nortel Networks equipment. These default parameters are downloaded to the system through a configuration file referred to as the "ini" (initialization) file. The parameters are set in the runtime memory through loading immediately following the loading of the executable (.cmp) file. In most cases, the parameters found in the *ini* file can also be changed via the Embedded Web Server for the product. For these parameters, inclusion in the *ini* file is a matter of convenience and enables the user to set up the Mediant 2000 in a shorter time since it is not necessary to configure these parameters through the Embedded Web Server each time the system is loaded. Some of the parameters in the *ini* file are not configurable through any other means than the *ini* file. These parameters typically determine functionality at a low level and are not often, if ever, changed for the Nortel Networks application.

Other parameters that require configuration tend to be location-specific parameters, such as network settings, trunk configurations, and MGC to Gateway settings that must be matched for the two to interwork properly. These parameters can only be set with specific site knowledge and are set through the Embedded Web Server.

This document assumes knowledge of the BootP/TFTP loading process for the Mediant 2000, familiarity with the concepts and settings for the protocols used in the Nortel Networks solution, and a general knowledge of the AudioCodes Mediant 2000 system and Embedded Web Server.

A.3 Overview

The following topics are discussed in this document:

- **Default *ini* File Parameters**
- **Locality-Specific Parameters**
- **Setting up Trunk Configurations**
- **T.38 & Bypass Fax Support**

A.4 Nortel Default Initialization File Parameters

The default initialization (*ini*) file contains parameters necessary for the inter-working of the AudioCodes Mediant 2000 with the Nortel Networks solution. Descriptions of each parameter are included in the *ini* file. Any system parameter not included in the file assumes the default setting for the .cmp file loaded on the system. Appendix B on page 133 contains an example of the *ini* file for the AudioCodes Mediant 2000 in the Nortel Networks solution.

As delivered from AudioCodes, the Mediant 2000 has in memory the latest 4.2 load approved by Nortel at the time of assembly. Once on site, the system requires upgrade with the latest version of the software, as well as the default *ini* file, tones file, and any site specific configurations as needed. There are two methods for loading the software and accompanying files – through the BootP/TFTP application or through http using the Embedded Web Server provided with the system. Refer to Section A on page 119. Refer to Nortel documentation for configuring the BootP/TFTP server on the SDM. The new load should be burned to flash when possible.

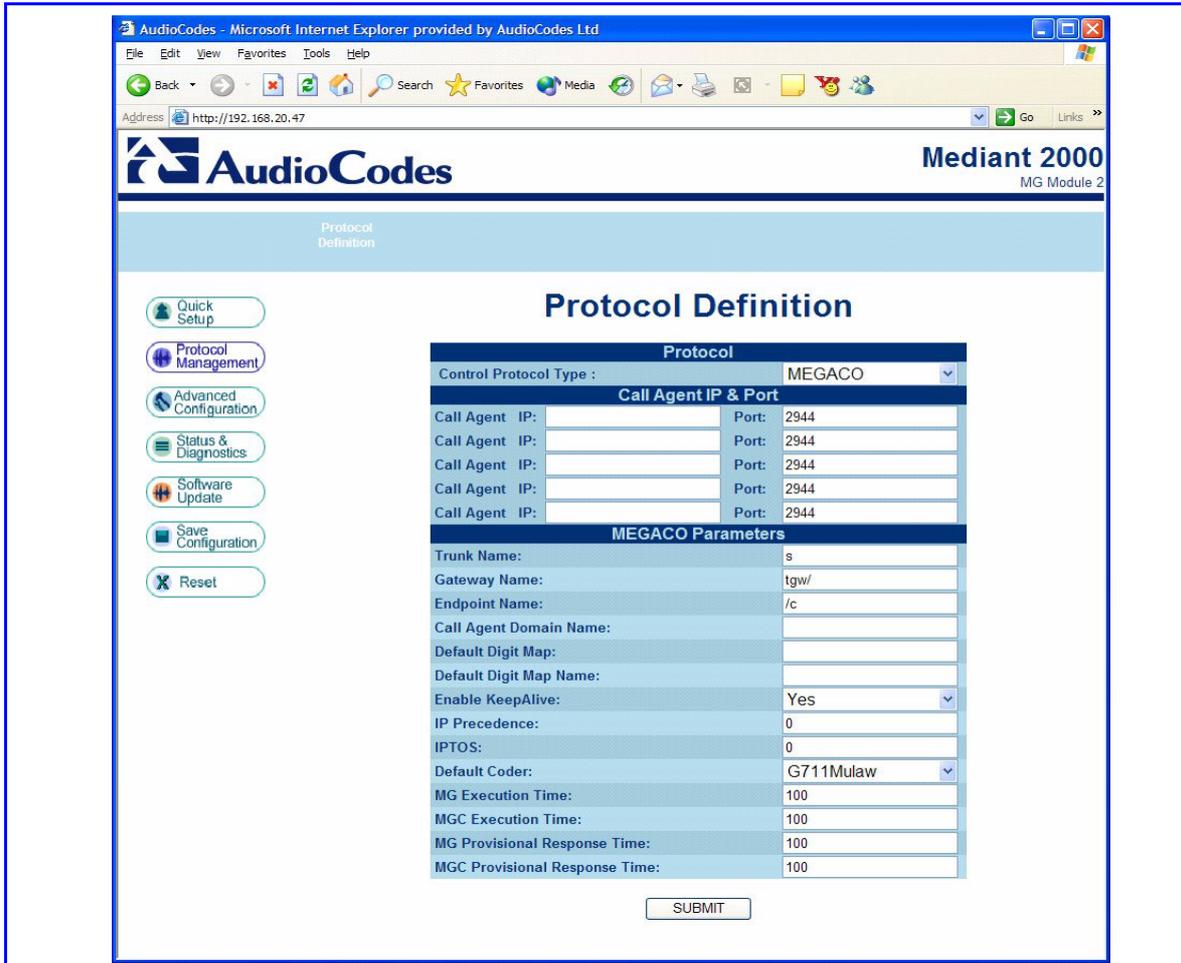
Instructions for setting up the system are provided in this document using the Embedded Web Server and the http download. These instructions assume that the reader has the latest update for the Mediant 2000 load and the *ini* file shown in Appendix B on page 133.

➤ To configure the Mediant 2000 to work within the Nortel configuration, take these 13 steps:

For details about accessing the screen mentioned in this section, refer to Section 5 on page 55.

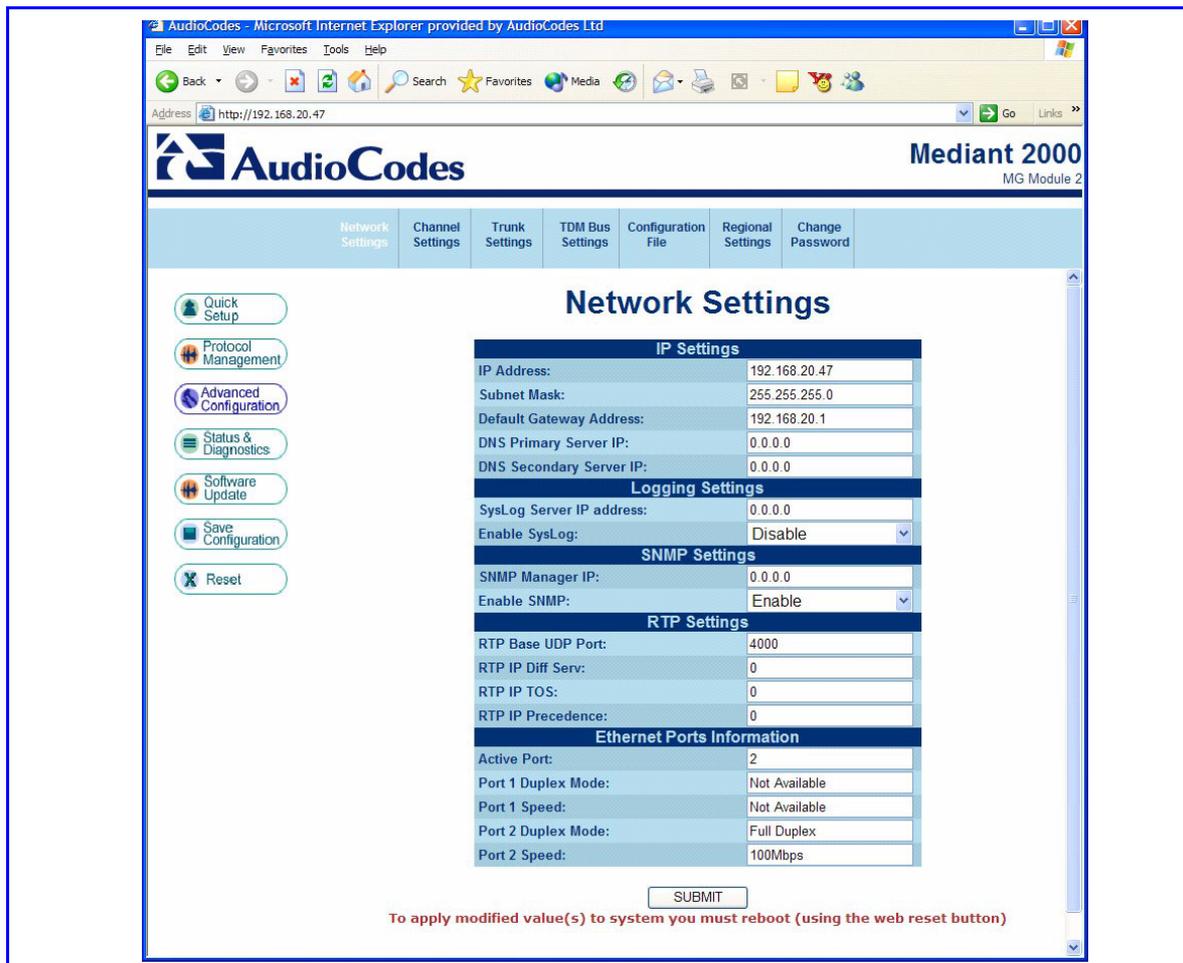
- Step 1.** Boot up the system and access the Embedded Web Server.
- Step 2.** From the Advanced Configuration menu, access the Configuration File screen.
- Step 3.** Click on the **Restore All Defaults** button.
- Step 4.** From the Reset menu, click the **Restart** button.
- Step 5.** After the system resets, again access the Embedded Web Server.
- Step 6.** From the Software Update menu, access HTTP Download screen.
- Step 7.** Browse your desktop for the latest Mediant 2000 load. Click **Send file** to send the file to the Gateway.
- Step 8.** After the system resets, again access the Embedded Web Server and again from the Software Update menu, access the HTTP Download screen
- Step 9.** Browse your desktop for the generic default *ini* file. Click **Send File** to send the file to the Gateway.
- Step 10.** After the successful download message appears at the bottom of the screen, click on the **Reset** button. The system restarts.
- Step 11.** After the system resets, again access the Embedded Web Server and check that the Mediant successfully loaded the default *ini* file and parameters.
- Step 12.** From the Protocol Management menu, access the Protocol Definition screen. It should look similar to the Figure A-1, on page 121.

Figure A-1: Protocol Definition



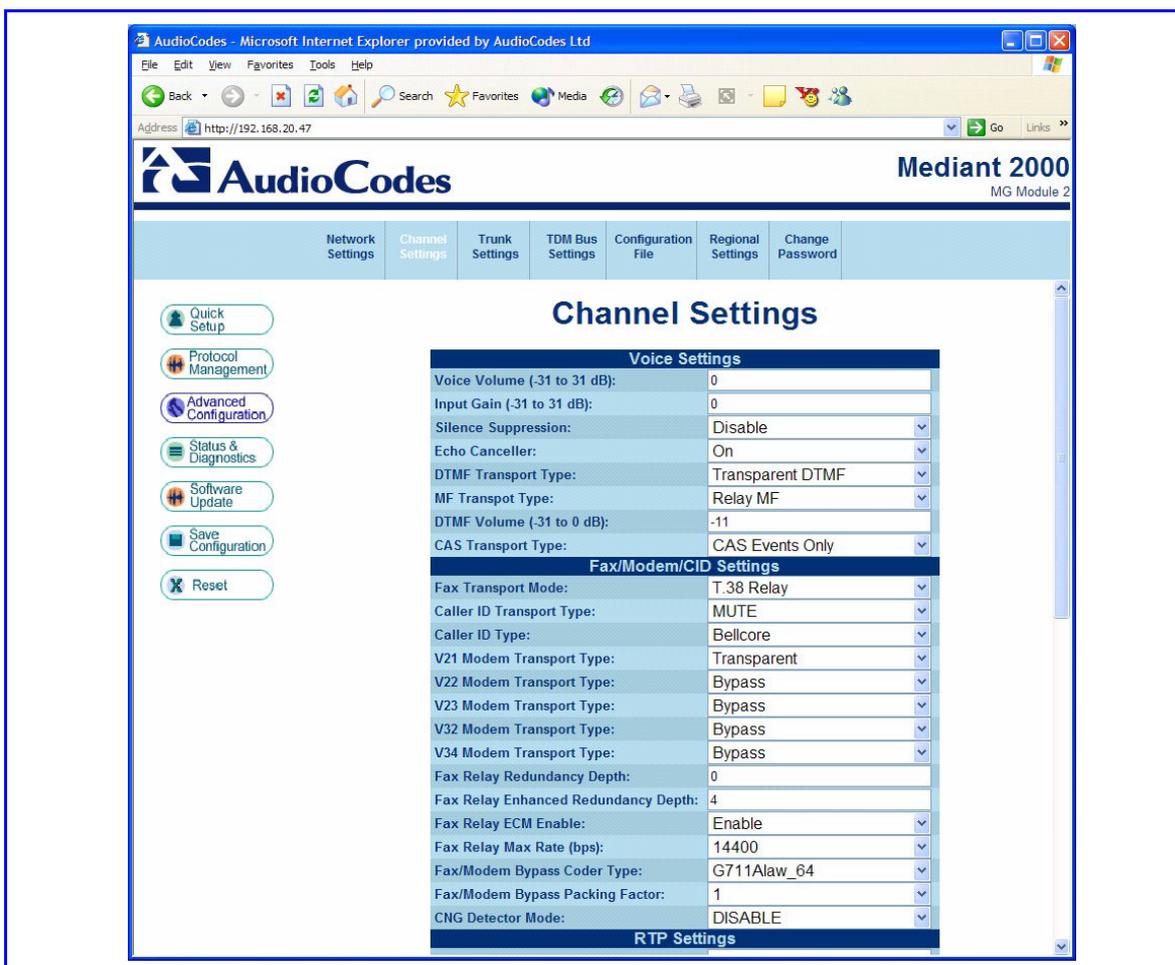
Step 13. From the Advanced Configuration menu, access the Network Settings screen. This window should look similar to Figure A-2, on page 122 (with the exception of the IP settings).

Figure A-2: Network Settings



- Step 14.** From the Advanced Configuration menu access the Channel Settings screen. Check that the DTMF Transport Type is set to “Transparent DTMF”. (Refer to Figure A-3, on page 123.)

Figure A-3: Channel Settings



Step 15. From the Advanced Configuration menu, access the TDM Settings screen. The parameters should be set as shown below. Specifically, the following parameters are set by the default *ini* file provided: TDM Bus Clock Source (Network), TDM Bus PSTN Auto Clock (Enable), and TDM Bus Type (Framers)

Note: In the example, PCM Law Select is set up for PCM companding law of type μ -law (default) for input and output. This must be matched to the same type of companding being used on the remote end of the TDM trunks (μ -law or A-law).

A.5 Locality-Specific Parameters

Locality-Specific parameters are parameters that are specific to a particular Mediant 2000 in the network. These parameters include any IP addresses for other nodes that the Mediant comes in contact with or settings that are specific to the logical Mediant in its function within the system. These parameters also include trunk settings which are detailed in Section A.6 on page 126.



To configure locality-specific parameters, take these 11 steps:

- Step 1.** Access the Mediant through the Embedded Web Server.
- Step 2.** From the Protocol Management menu, access the Protocol Definition screen.
- Step 3.** On the first Call Agent IP line, enter the IP address of the MGC (GWC) to which this GW should be communicating.
- Step 4.** Trunk Name, Gateway Name, and Endpoint name should be set to correspond to the values set within the MGC (GWC) to allow for proper interworking between the Nortel GWC and the AudioCodes Mediant 2000.

An example is:

Gateway Name = "DS1" (this is the trunk type on the GWC),

Trunk Name = "/0" (this is the span name on the GWC),

Endpoint Name = "/"

where the representation is DS1/0X/Y which represents timeslot Y on framer X. Note – a parameter "MEGACOTRUNKIDOFFSET" is set to a value of "1" in the default *ini* file to align the AudioCodes logical mapping (begins with 0) with the Nortel architecture (begins with 1).

- Step 5.** Click the **Submit** button to save changes to the Protocol Definition screen, which should look similar to Figure A-4, on page 124.

Figure A-4: Protocol Definition

Protocol	
Control Protocol Type :	MEGACO
Call Agent IP & Port	
Call Agent IP:	192.168.10.54
Port:	2944
Call Agent IP:	
Port:	2944
Call Agent IP:	
Port:	2944
Call Agent IP:	
Port:	2944
MEGACO Parameters	
Trunk Name:	/0
Gateway Name:	DS1
Endpoint Name:	/
Call Agent Domain Name:	
Default Digit Map:	
Default Digit Map Name:	
Enable KeepAlive:	Yes
IP Precedence:	0
IPTOS:	0
Default Coder:	G711Mulaw
MG Execution Time:	100
MGC Execution Time:	100
MG Provisional Response Time:	100
MGC Provisional Response Time:	100

- Step 6.** From the Advance Configuration menu, access the Network Settings screen.
- Step 7.** Enter the Syslog Server IP address and enable the Syslogs by setting “Enable Syslog” to Enable.
- Step 8.** Click the **Submit** button to apply the values. The Network Settings screen should look similar to Figure A-5.

Figure A-5: Network Settings

The screenshot displays the 'Network Settings' configuration page for Mediant 2000. The interface is organized into several sections:

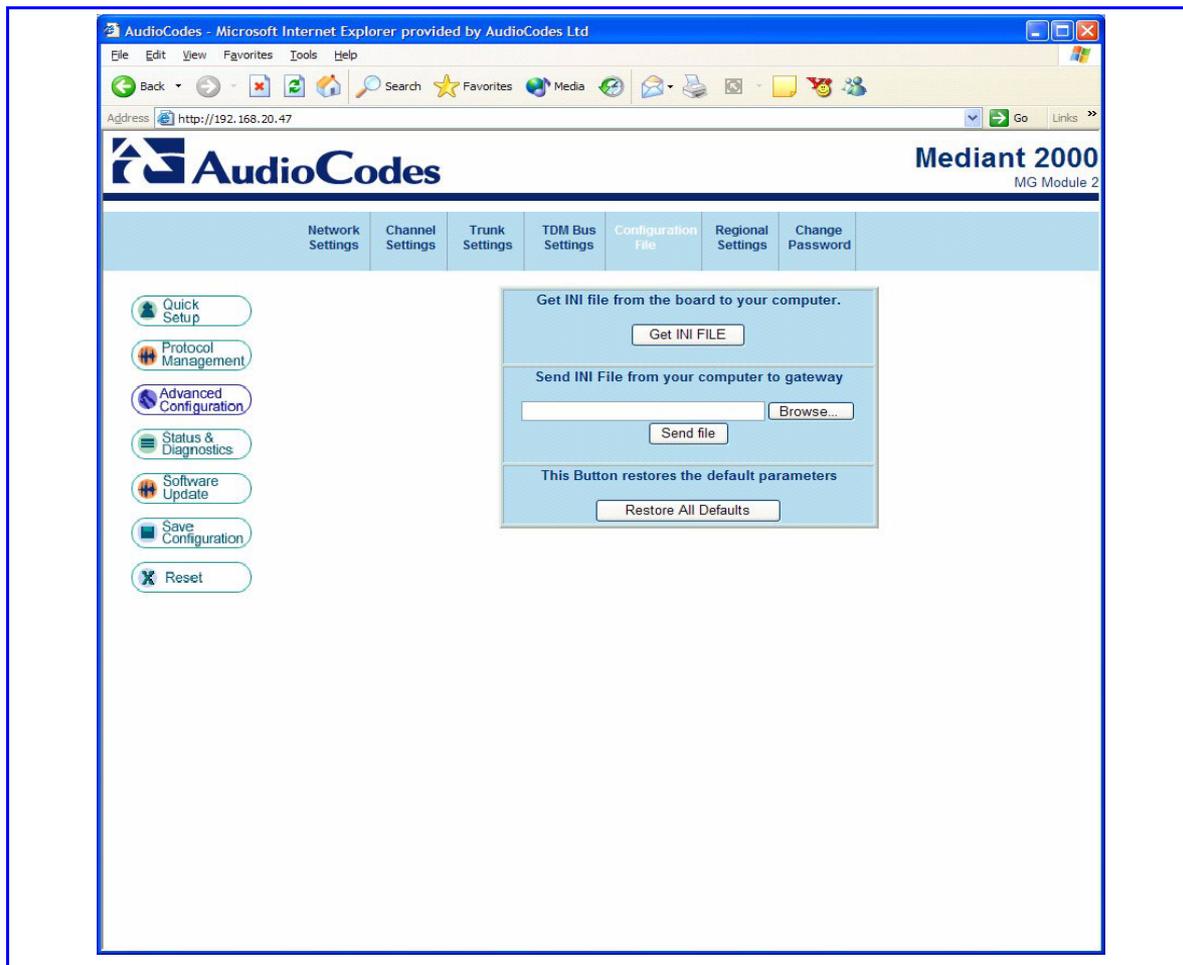
- IP Settings:**
 - IP Address: 192.168.20.47
 - Subnet Mask: 255.255.255.0
 - Default Gateway Address: 192.168.20.1
 - DNS Primary Server IP: 0.0.0.0
 - DNS Secondary Server IP: 0.0.0.0
- Logging Settings:**
 - SysLog Server IP address: 0.0.0.0
 - Enable SysLog: Disable (dropdown menu)
- SNMP Settings:**
 - SNMP Manager IP: 0.0.0.0
 - Enable SNMP: Enable (dropdown menu)
- RTP Settings:**
 - RTP Base UDP Port: 4000
 - RTP IP Diff Serv: 0
 - RTP IP TOS: 0
 - RTP IP Precedence: 0
- Ethernet Ports Information:**
 - Active Port: 2
 - Port 1 Duplex Mode: Not Available
 - Port 1 Speed: Not Available
 - Port 2 Duplex Mode: Full Duplex
 - Port 2 Speed: 100Mbps

A 'SUBMIT' button is located at the bottom of the form. Below the button, a red warning message states: "To apply modified value(s) to system you must reboot (using the web reset button)".

At this point, the Mediant is ready to have its TDM configuration setup (described in A.6). If this is to be the stopping point for configuration of the system, the user can reset the system and download the board configuration file from the Mediant to save as a backup that can be loaded in later as needed. The next steps in this section describe how to do this.

- Step 9.** From the Reset menu, click the **Restart** button.
- Step 10.** After the system resets, it includes all of the default settings from the default *ini* file, plus the additional settings that were configured through the Embedded Web Server. (Verify as needed).
- Step 11.** From the Advanced Configuration menu, access the Configuration File screen as shown in Figure A-6, on page 126.

Figure A-6: Configuration File



Step 12. To download the boards new configuration file to your desktop, click the **Get INI file** button. This file can be loaded back into the system at a later time should it become necessary to restore the system’s settings to those in this configuration file. Be sure to label the file appropriately to identify its contents.

A.6 Setting up Trunk Configurations

This section provides guidelines and instructions for provisioning the trunk configurations that have been selected for use on the Mediant by Nortel Networks for this release. These include the following protocols on the Mediant 2000:

- **T1_IUA – relays ISDN signaling through SIGTRAN IUA and SCTP protocols**
- **E1_IUA – relays ISDN signaling through SIGTRAN IUA and SCTP protocols**
- **T1_Transparent – channels 1-24 mapped to DSP channels**
- **E1_Transparent_60 - channels 1-31, except 16 mapped to DSP channels**
- **E1_Transparent_62 – channels 1-31 mapped to DSP channels**

A.6.1 ISDN Relay

The Mediant 2000 supports the relay of ISDN signaling messages using the SIGTRAN IUA and SCTP protocols. A signaling message enters the Mediant from an ISDN connection going through the data link layer and the Q.931 protocol data unit is relayed to the MGC (GWC) using IUA over SCTP over IP.

The setup for ISDN Relay involves selecting an E1 or T1, network or user termination, and datafilling the IUA interface identifier with the correct D-channel mapping. IUA trunks are configured per D-channel. This configuration is done on the Advanced Configuration ... Trunk Settings screen.

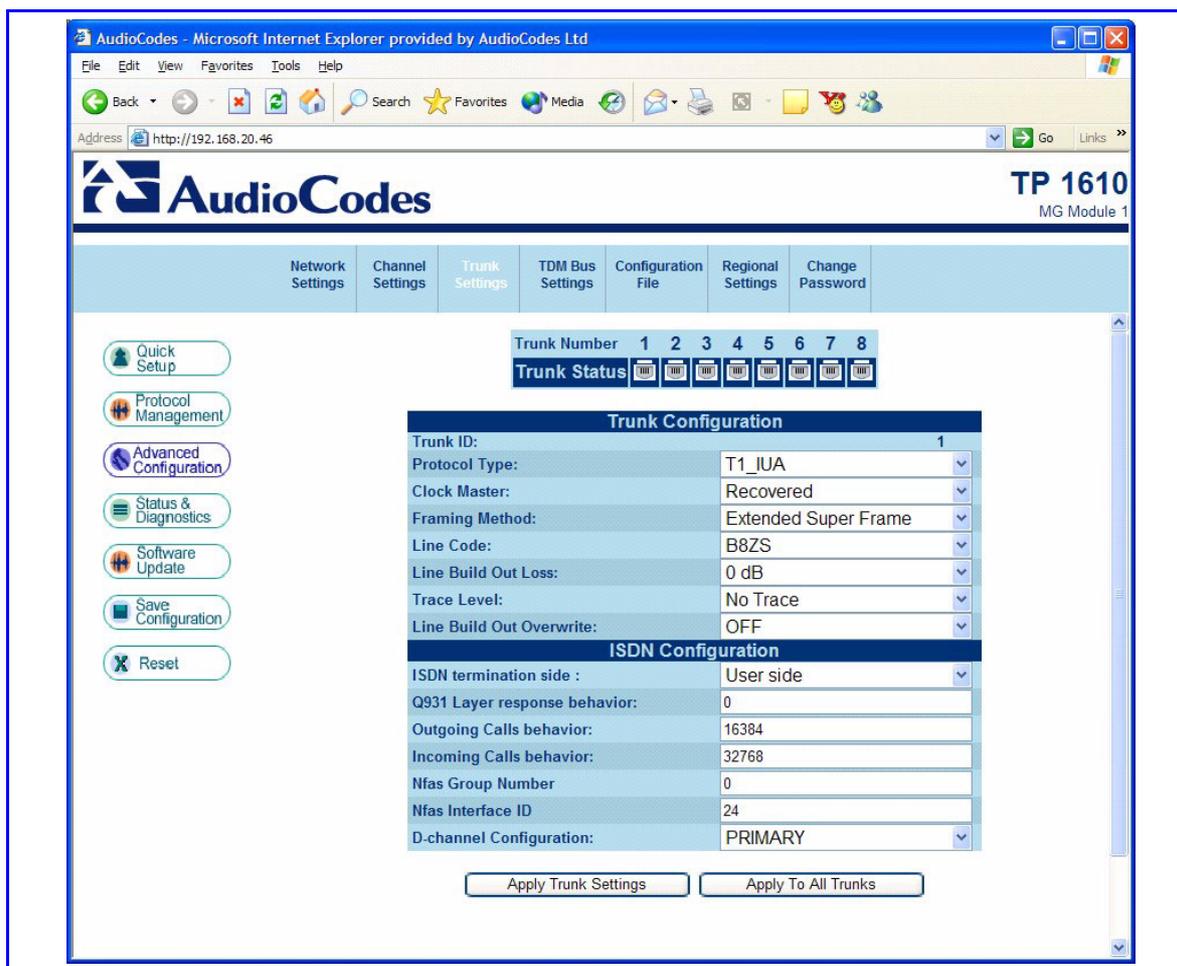
A.6.1.1 Example 1: Configuring ISDN Relay for a Single T1 (User-side).



To configure ISDN Relay for a Single T1, take these 8 steps:

- Step 1.** Access the Mediant 2000 via the Embedded Web Server.
- Step 2.** From the Advanced Configuration menu, access the Trunk Settings screen.
- Step 3.** Acknowledge notice to restart board after changing the trunk configuration.
- Step 4.** For TrunkID 1, select T1_IUA for the protocol type.
- Step 5.** Set the ISDN Termination Side accordingly (user/network). This should be the compliment of the setting in the Nortel equipment for the trunk.
- Step 6.** Verify that "Outgoing Calls Behavior" = 16384. (This was set in the default *ini* file).
- Step 7.** Set the "NFAS Interface Id" = X, where X is the logical mapping for IUA interface to the D-channel between the MGC and the Mediant 2000. In this example, 24 is used. The "NFAS Interface Id" field is used to accept the value for the IUA interface ID. The screen should appear as shown in Figure A-7.

Figure A-7: Trunk Settings



Step 8. Click the **Apply Trunk Settings** button.

Note that the AudioCodes 4.2 release does not support run time provisioning of the IUA and transparent trunks. Once trunk settings are applied, the board must be restarted.

Thereafter, datafill changes to IUA trunks after the IUA protocol has been started for a particular trunk will require that the board be taken back to the configuration at the end of the previous section (Section A.5 on page 123) and all trunk interfaces be re-datafilled.

Once the board has been restarted, if configured properly, the IUA Trunk Status becomes green.

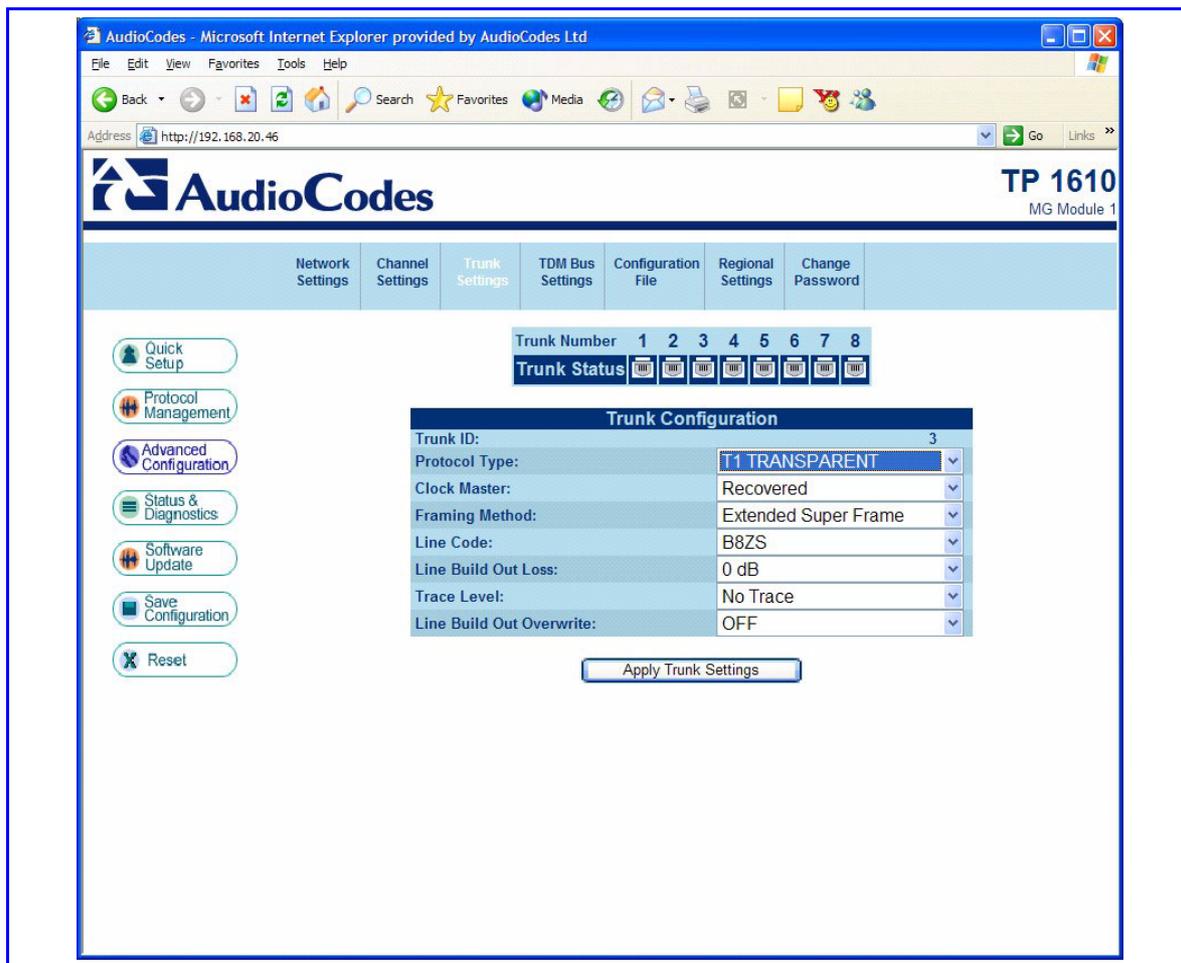
A.6.1.2 Example 2: Configuring ISDN Relay for Multiple Trunks (2 T1s-User-side)

This configuration is the same as the example 1, except 2 T1_IUA trunks are configured each with their own IUA interface ID.

A.6.1.3 Example 3: Configuring ISDN Relay for NFAS Trunks (4 T1s, Single D-channel, User-side)

This configuration is similar to Example 1 in that 1 T1_IUA trunk is configured to handle the D-channel, but additionally, 3 T1_Transparent trunks are configured for the 3 remaining B-channel only trunks. Section A-8 on page 129 shows the T1_Transparent Trunk screen.

Figure A-8: Transparent Trunk

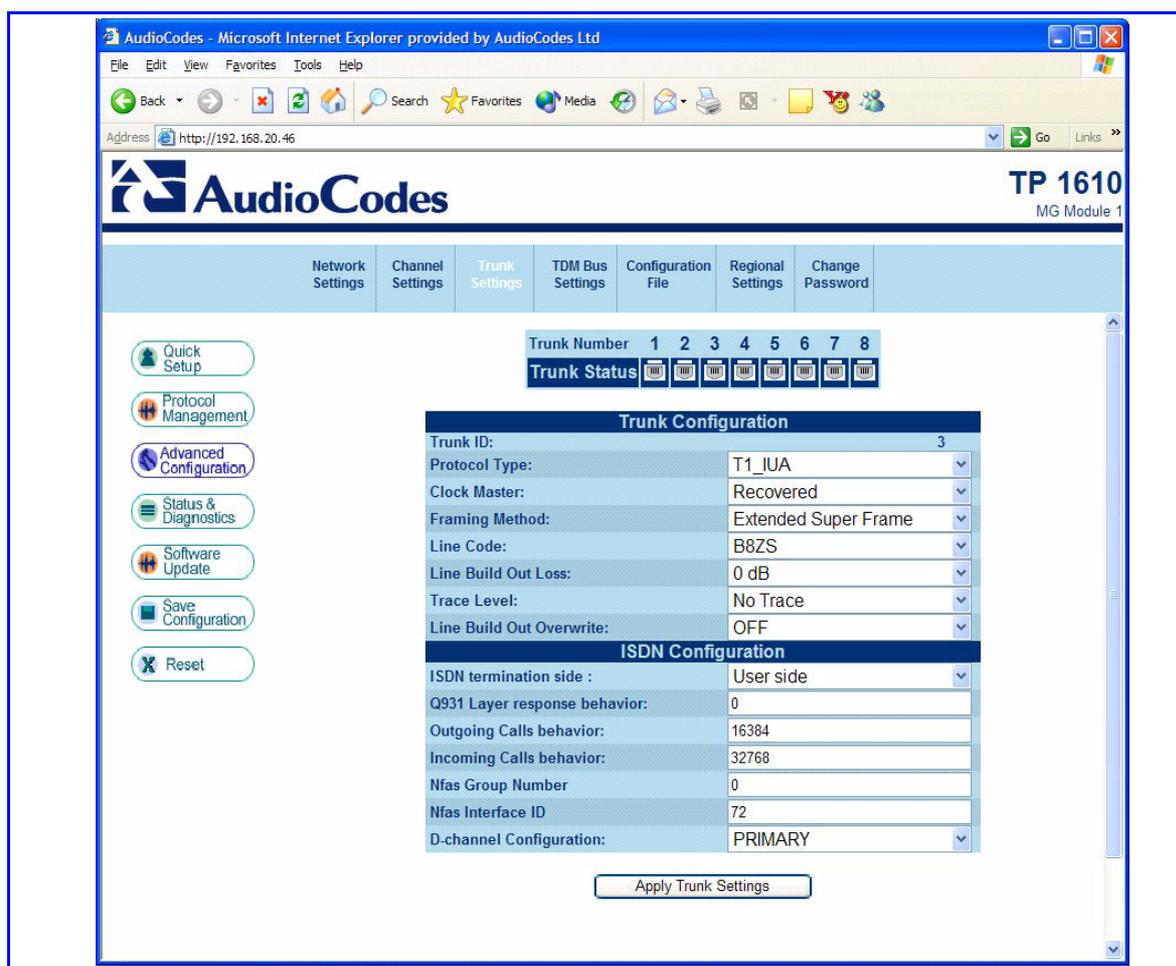


A.6.1.4 Example 4: Configuring ISDN relay for NFAS trunks with Backup D-channel (4-T1s, User-side)

This example is the same as Example 3, except the designated backup span should also be made into a T1_IUA trunk. Figure A-9, on page 130 shows a second T1_IUA trunk screen set up to handle the signaling for a backup D-channel.

Note that this configuration (except for the IUA interface ID – in the NFAS interface ID below) is the same as that used for the primary channel. The screens of the Embedded Web Server are used for many protocols. The IUA protocols do not terminate on the Mediant board, therefore, not all fields are used. (Note that the “D-channel Configuration” field has no meaning in this context).

Figure A-9: T1_IUA trunk for a Backup D-channel



Note: The Mediant 2000 is the Signaling Gateway and the GWC is the MGC. The Mediant is handling layer 2 (Q.921) and not layer 3 (Q.931). The Mediant passes the Q.931 through SIGTRAN (IUA) to the MGC. NFAS intelligence is in the Q.931 layer, so the Mediant is really unaware of whether the signaling is over a primary or back-up channel. The Mediant only knows that it is receiving messaging in the IUA pipe.

A.7 T.38 & Bypass Fax Support

This section discusses support for T.38 and Fax Bypass. The Mediant 2000 supports two levels of negotiation for T.38 Fax. The first is positive negotiation and is not used by Nortel. Instead, the second level, "Full level" is configured on the system through setting bit 2 of the MGCPCompatibilityProfile parameter. This parameter is set in the Nortel default initialization file, therefore the Full level is always the negotiation method used.

Support of T.38 was added to the SDP according to the following rules:

- **If the call manager wants this call to support T.38, it should send an additional line in the local SDP to the board, as in the following example:**

```
v=0
c= IN IP4 $
m=audio $ RTP/AVP 0
```

```
m=image $ udptl t38
```

The first three lines describe the voice stream, and can vary according to the user's needs. Attributes to the voice ('a' lines) are added after the first 'm' line. The 'm=image' line, however, is required, and should appear in the exact format above. The board returns a fully specified line, with the local port used for the T.38.

- **The fax redundancy can be requested by including the following attribute line after the 'm=image' line:**

```
a=T38FaxUdpEC:T38UdpRedundancy
```

The negotiation rules are as follows:

- **If the 'm=image' line was not received both in local AND remote descriptors, T.38 will NOT be used.**
In this case, if the "m=audio" line contained the G.711 coder as the **SECOND** coder option, the fax mode will be bypass, and the G.711 coder is used for it. If the second coder was not G.711, the Fax transport type is transparent.
- **If the fax redundancy attribute line did not appear both in local and remote descriptors, redundancy is NOT used.**

Reader's Notes

Appendix B Example of the *ini* File

```
.*****  
;  
;** Ini File **  
.*****  
;  
  
.*****  
;  
;** The parameters set up in this ini file allow the M2K to be used within  
;** the Nortel SL100 and CS2K solutions. This file should not be manually  
;** edited and the values of the following parms should not be changed.  
.*****  
;  
  
[BSP Params]  
  
; TDMBUSCLOCKSOURCE selects the clock source on which the board synchronizes  
; A value of 4 = PSTN Network  
TDMBUSCLOCKSOURCE = 4  
  
; INIFILEVERSION is the version of this default ini file  
INIFILEVERSION = 6  
  
; CallProgressTonesFilename is the name (and path) of the file containing the Call  
; Progress Tones definition. The tones.dat file should be present in the same directory  
; as the ini file in which it is defined when using bootp/tftp to download files to  
; the M2K.  
CALLPROGRESSTONESFILENAME = 'tones.dat'  
  
[Analog Params]  
  
; The BurnCallProgressTonesFile indicates to the system whether or not the call  
; progress tones file should be stored in non-volatile memory.  
BURNCALLPROGRESSTONESFILE = 1  
  
[ControlProtocols Params]  
  
; MGCPCompatibilityProfile controls profiling of various MEGACO features including
```

; T.38 and Bypass support and characteristics of servicechange request, ptime, and
; notify transaction requests.

MGPCCOMPATIBILITYPROFILE = 20

; ProvisionedCallAgentsPorts is the list of ports for the CallAgents.

PROVISIONEDCALLAGENTSPORTS = 2944, 2944, 2944, 2944, 2944

;MGControlProtocolType determines the protocol type to be used for the MG.

;MGControlProtocolType = 2 is the setting for MEGACO

MGCONTROLPROTOCOLTYPE = 2

[MEGACO Params]

; KeepAliveEnabled enables a keep alive message (NOP ServiceChange). For CS2K M2K
; solution, enabling this parm will allow the M2K to detect a LAN Failure and issue
; a SC(DISC) on root until received by CS2K, which will use that msg to set all the
; trunks back to idle and the MGC will follow up with a re-registration.

KEEPALIVEENABLED = 1

; MegacoTrunkIDOffset is a logical integer that is added to the internal integer
; value for a respective trunk element. This value helps to map the internal logical
; number of the framer to the physical numbering. Internally, the M2K starts at 0
; logically. Nortel framers start numbering at 1.

MEGACOTRUNKIDOFFSET = 1

[PSTN Params]

;TDMBusPSTNAutoClockEnable enables or disables the auto fallback PSTN clock feature.
; When set to 1, fallback is enabled.

TDMBUSPSTNAUTCLOCKENABLE = 1

; ISDNOutCallsBehavior is a bit field that determines behavior options, influencing
; how the ISDN Stack OUTGOING calls behave. The value 16384 (0x4000) in this param-

eter

; enables reversed DLCI TEI support like in the RFC (see RFC3057 p.21 Figure 6).

; For Nortel interworking this parameter must be set to 16384 (15th bit in binary).

; If not the DLCI TEI representation will not be presented for interworking and the

; IUA interfaces will not align.

ISDNOUTCALLSBEHAVIOR = 16384

; Q931RelayMode set to 3 is used to port all Q931 (layer 3) messages via IUA handling
; up to the signaling element of the Gateway Controller rather than terminating the Q931
; protocol on the board itself

Q931RELAYMODE = 3

[Voice Engine Params]

; These parameters support the detection of COT tones within the DSP associated to the
; physical element within a context and report the detection back to the H248
; interworking control protocol stack. This is used for SS7 interworking support of
; continuity tones on the physical paths. COT support is supported for Transparent
; framer definitions when applied by use of the protocoltype definitions per framer.
; EnableContinuityTones enables (1) or disables (0) Continuity Test tone detection and
; generation according to the IUT-T Q.724 recommendation.

ENABLECONTINUITYTONES = 1

ENABLECONTINUITYTEST = 1

; DTMFTransportType set to transparent enables the M2K to negotiate for RFC2833
; dynamically (via parsing of the SDP data provided by the GWC embedded within
; the H248 messaging). If the two sides *do not* agree, the unit will leave the
; DTMF digits within the bearer path and it will be passed to the distant RTP
; device within the codec that was established for the bearer path. If the SDP
; information *is negotiated*, then the dtmf detected digits will be removed from
; the bearer path and relayed to the remote RTP device by use of RFC2833 DTMF relay.

DTMFTRANSPORTTYPE = 2

Reader's Notes

Appendix C AudioCodes BootP/TFTP Server Utility

C.1 Introduction

AudioCodes **BootP/TFTP Server Utility** enables easy configuration and provisioning of AudioCodes boards and Media Gateways. It contains BootP and TFTP servers with specific adaptations to AudioCodes' requirements.

C.1.1 Key Features

- Internal BootP server supporting hundreds of entities.
- Internal TFTP server.
- Contains all required data for AudioCodes products in predefined format.
- Provides a TFTP server address, enabling network separation of TFTP and BootP servers.
- 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 (for this version and above).
- Unicast BootP respond.
- User-initiated BootP respond, for remote provisioning over WAN.
- Filtered display of BootP requests.
- Location of other BootP servers that contain the same MAC entity.
- Common log screen for both BootP and TFTP sessions.
- Works with Windows™ 98, Windows™ NT, Windows™ 2000.

C.1.2 Specifications

BootP standards: RFC 951 and RFC 1542

TFTP standards: RFC 1350 and RFC 906

Operating System: Windows™ 98, Windows™ NT, Windows™ 2000,
and Windows™ XP

Maximum number of MAC entries: 200

BootP Fields:

- Hardware address (MAC): 12 hex digits
- IP address
- Subnet
- Default Gateway
- TFTP server IP; (Using the TFTP server IP field enables download of firmware from a different Host)
- Boot File
- *ini* File
- New MAC (optional)

Screens:

- File Upload and Message screen
- Preferences screen
- Client Configuration screen
- Template definition screen

C.1.3 BootP/TFTP Server Utility Installation

The BootP/TFTP Server Utility can be installed on a PC from the AudioCodes Software & Documentation CD:

TrunkPack Boards (TP) Compact disk Catalog Number: LSTC-00001

IPmedia Platforms (IPM) Compact disk Catalog Number: LSTC-00002

TrunkPack Modules (TPM) Compact disk Catalog Number: LSTC-00003

MediaPack Series (MP) Compact disk Catalog Number: LSTC-00005

To install the BootP/TFTP Server Utility, unzip the TP3.4.2.exe file and navigate to the BootP *.exe file. The installation procedure is facilitated by prompts. After completing the procedure, open Start>Programs>BootP; the BootP/TFTP Server main screen is displayed.

C.1.4 Logging Screen

The AudioCodes BootP/TFTP Server Utility main screen (Figure C-1, on page 140) includes the Log line, printed per BootP request with the following parameters:

- Hardware (MAC) address
- Status (found or not found in cache)
- Date and Time
- Assigned IP address (if found)
- Client name

Clicking on Log line shows all BootP reply parameters or enables entry to a new entity.

C.1.5 Preferences Screen

The Preferences screen (Figure C-2, on page 141) is used to define BootP and TFTP configuration parameters:

- TFTP directory
- *ini* File Mask
- Boot File Mask
- TFTP timeout and number of retransmissions
- BootP replay type (broadcast or unicast)
- BootP ARP mode (dynamic or static)
- Number of initiated BootP replies (send after remote reset), optionally used when the Media Gateway (MP-102, MP-104, MP-108 and MP-124, or TP board/module) is installed behind the firewall that blocks BootP broadcast requests

C.1.6 Client Configuration Screen

The Client Configuration screen (Figure C-3, on page 142) shows:

- All client entities
- MAC
- Name
- IP per entity

Using this screen, Users can:

- Add a new entry
- Delete an existing entry
- Modify an existing entry
- Test a selected client for finding all BootP servers that respond to a BootP request with a specific MAC address

If a template is selected, any parameter can be entered manually or copied from the selected template, by marking the checkbox to the right of the parameter. Usually, only an IP address is entered manually while other parameters are copied from the template.

C.1.7 Template Screen

The Template screen (Figure C-1, on page 140) enables the User to add, modify, or delete templates.

The template includes:

- Subnet
- Gateway, TFTP server
- BootFile
- *ini* file
- Server IP

C.2 Screen Details

C.2.1 Main Screen

Figure C-1: Main Screen

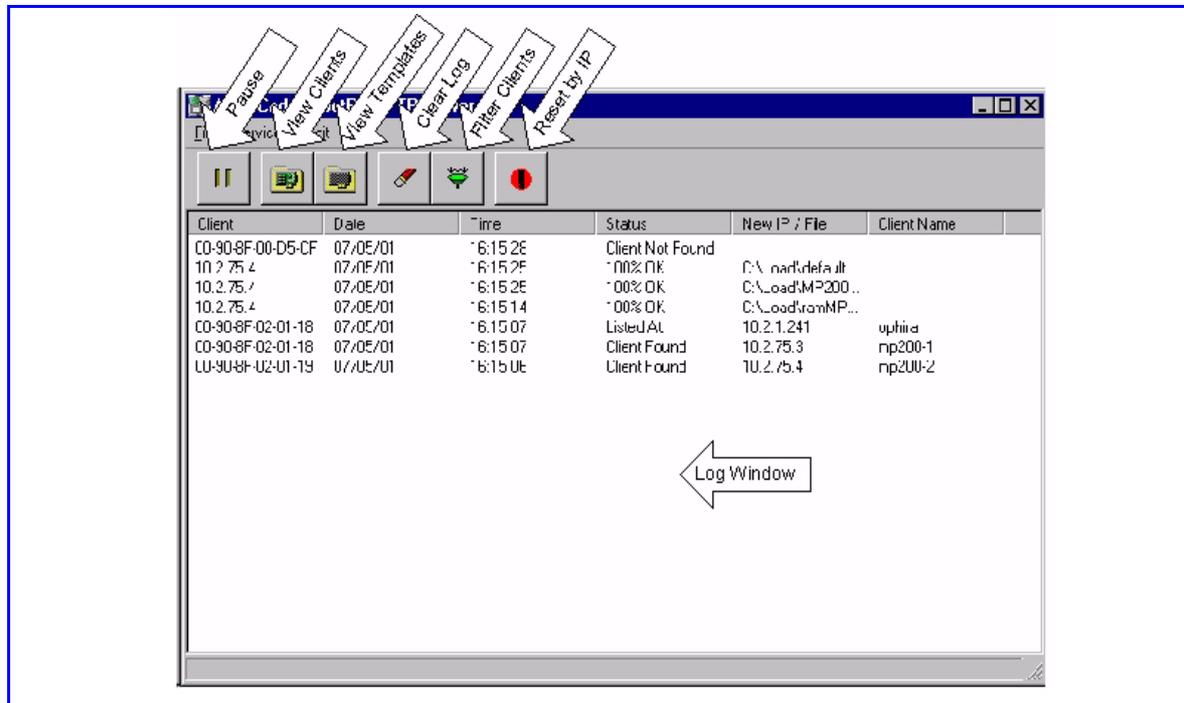


Figure C-1 shows the main screen of the AudioCodes **BootP/TFTP Server Utility**, featuring:

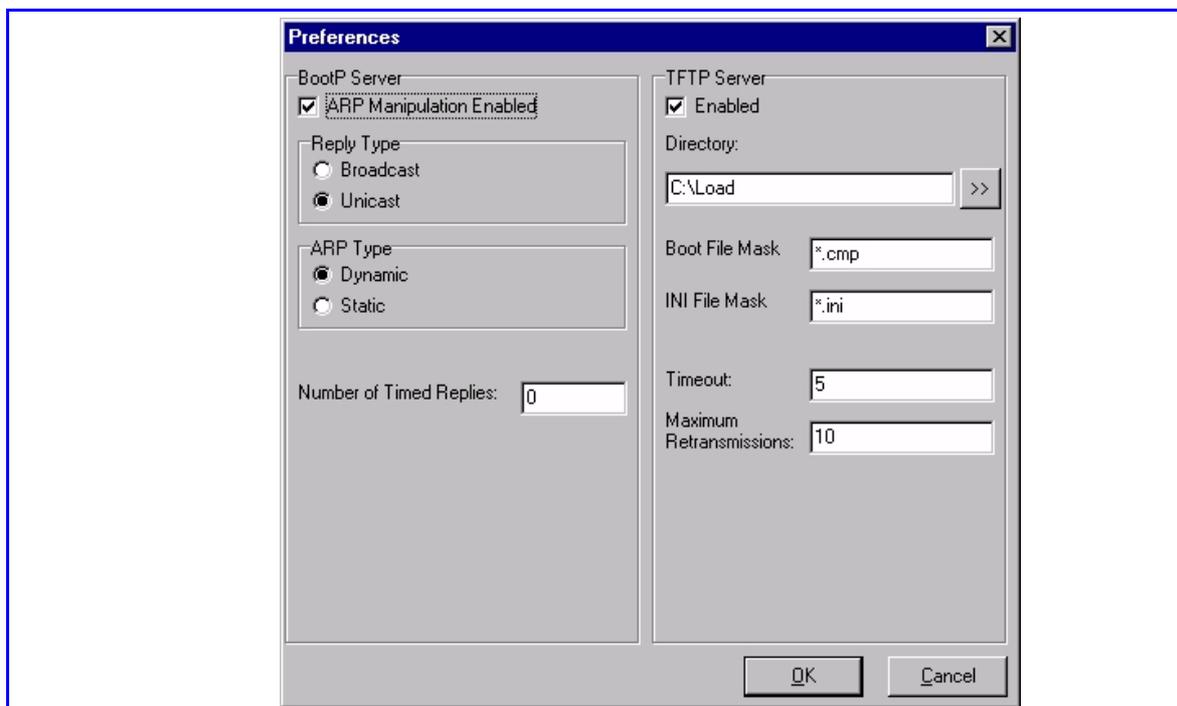
- **Program State** - With this button, the User can pause the program. When the program has paused, no replies to BootP requests are sent.
- **View Clients button** - This button opens up the Clients Configuration screen.
- **View Templates button** - This button opens up the Templates Configuration screen.
- **Clear Log button** - This button clears the log.
- **Filter Unknown Clients button** - This button filters all BootP requests that are not listed in the client configuration screen.
- **Reset button** - This button opens a dialog, in which Users can enter an IP of a client. The program sends a reset command to that client.
- **Edit/Preferences** - Selecting “Preferences” in the Edit menu opens the Preferences screen for defining BootP and TFTP parameters.
- **Log Screen** - All BootP requests and TFTP sessions are displayed, including the time and date of the request. In addition, the response type is also displayed:
 - “Client Not Found”
 - “Client Found”
 - “Client’s Mac Changed”
 - “Client Disabled”
- “Listed at” (when using the “test selected clients” button).

- For a TFTP session, file name and download status are displayed.
- **Pop-Up Menu** - When the User right-clicks on a line in the log screen, the pop-up menu opens. In this menu there two options:
 - **Reset:** When this option is selected, the program searches the database for the selected MAC. When the client is found, the program adds the client's MAC to the ARP table, and then sends a reset command to the client. Note that by performing the remote reset this way, the User doesn't have to know the current IP of the client. To perform this, the User must have administrator privileges, or else an error message appears.
 - **View Client:** This option is the same as double-clicking on a line. When selected, the Clients Screen opens. If the Client's MAC is found in the database, it is focused. If not, a new client is added, with the MAC filled out. The User must only fill in the remaining fields.

C.2.2 Preferences Screen

In the Preferences screen, Figure C-2, BootP and TFTP configuration parameters are defined.

Figure C-2: Preferences Screen



For the TFTP server, the User can configure a TFTP directory and a value for TFTP Timeout and Maximum Retransmissions. Set these values to 5 and 10 as shown above.

The TFTP server can be disabled by clearing the Enable checkbox.

In the BootP section, the User can select ARP mode: Dynamic or Static, and reply type: Broadcast or Unicast. For a typical application, use Dynamic ARP mode and Unicast, as shown above.

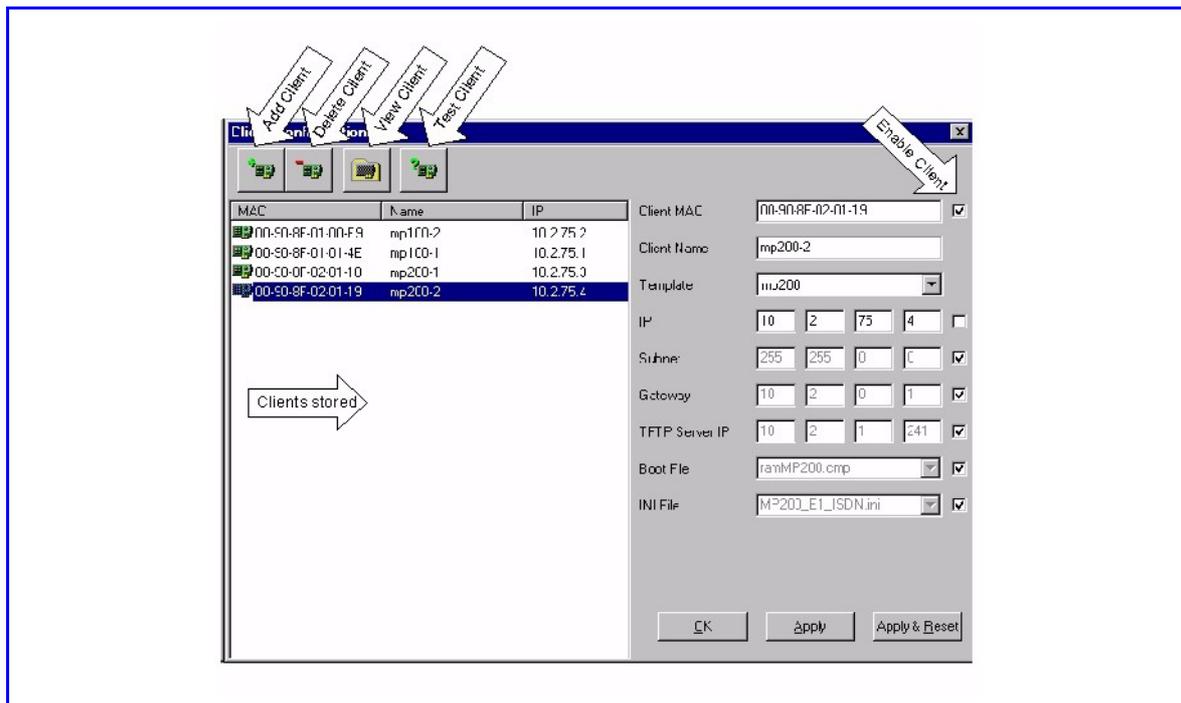
This option requires the **User to have administrator privileges** otherwise an error message appears. If you don't have administrator privileges, **uncheck** the ARP Manipulation Enabled checkbox in the Preferences Screen.

The “Number of Timed Replies” (the number of initiated timed BootP replies) can be used when an MP-1xx (or TP board/module) is installed behind a Firewall that blocks BootP broadcast requests. In a typical application, this feature can be disabled by entering “0” in this field. When selected, several BootP replies are sent to the MP-1xx (or to the TP board/module) immediately after the remote reset command.

C.2.3 Client Configuration Screen

Figure C-3 is the Client Configuration Screen in which clients are added and defined.

Figure C-3: Client Configuration Screen



In the left pane of the screen is the client list. By clicking on a client in this list, the following parameters for this client are displayed on the right side of the screen:

- **Client MAC:** This is the MAC address of the client. When the User edits the MAC, a new client is added, with the same parameters as the previous client.
- The client can be disabled by un-checking the check box on the right side of the Client MAC, causing the BootP server not to reply to the BootP request. The client can be enabled by checking the check box. Click on the 'Apply' button each time the client enable check box is checked or unchecked.
- **Client Name:** Free text for client description.
- **Template:** The template to be used for this client. When a template is selected, its parameters override all of the previous parameters.
- **IP, Subnet, Gateway:** Normal IP parameters.
- **TFTP Server IP:** The IP of the TFTP Server.
- **Boot File, ini File:** The files to request from the TFTP server.
- **Call Agent:** The IP of the MGCP Call Agent that will be controlling the gateway.

Note the seven check boxes to the right of the parameters. These enable the User to assign only the selected fields from the template. The rest can be unique for each client. When the field is assigned a value from the selected template, the field is grayed out.

After performing changes, press “Apply” to save them. By clicking on “Apply & Reset”, the program saves the changes to the database, performs a remote reset to the client by adding the client’s MAC to the ARP table, and then sends out a reset command. This option works **only if “ARP Manipulation Enabled”** checkbox in the “Preferences” screen is **checked** (in Figure C-2, on page 141) otherwise an error message appears. It requires the User to have **administrator privileges**. The remote reset is supported for software in this version and up.

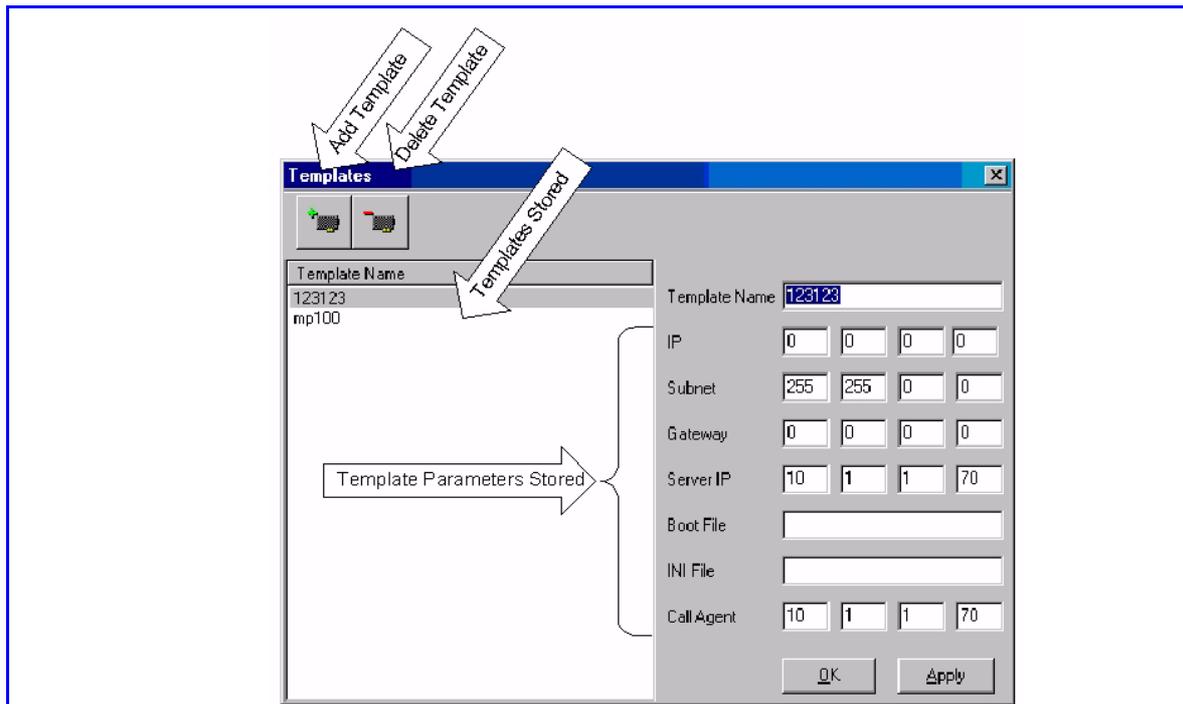
When adding a new client, click on the “Add Client” button; a client with blank parameters is displayed. After filling out the parameters, click on “Apply”; the client is added.

To find out if there is another BootP server on the net that contains a client with the same MAC address, click on “Test Selected Clients”; in the log screen, view the IP addresses of all BootP servers that contain the same MAC address in the status ‘Listed At’. In normal operation, BootP client MAC address should be listed only on a single BootP server. If the MAC address is listed in multiple BootP servers, it must be removed from other BootP servers.

C.2.4 Templates Screen

Figure C-4 shows the Templates screen, which provides a fast way to configure a number of clients that have the same parameters (except for the IP address). To use the Templates screen, create a template, and then apply the template to the client by selecting it.

Figure C-4: AudioCodes BootP/TFTP Templates Screen



Reader's Notes

Appendix D RTP/RTCP Payload Types

RTP Payload Types are defined in RFC 1889/1890. AudioCodes has added new payload types, to enable advanced use of other coder types. These types are reportedly not used by other applications.

Note: Refer to the Release Notes for the supported coders.

D.1 Packet Types Defined in RFC 1890

Table D-1: Packet Types Defined in RFC 1890

Payload Type	Description	Basic Packet Rate [msec]
0	G.711 μ -Law	20
2	G.726-32	20
4	G.723 (6.3/5.3 kbps)	30
8	G.711 A-Law	20
18	G.729	20
200	RTCP Sender Report	Randomly, approximately every 5 sec (when packets are sent by channel)
201	RTCP Receiver Report	Randomly, approximately every 5 sec (when channel is only receiving)
202	RTCP SDES packet	
203	RTCP BYE packet	
204	RTCP APP packet	

D.2 AudioCodes Defined Payload Types

Table D-2: AudioCodes Defined Payload Types

Payload Type	Description	Basic Packet Rate [msec]
35	G.726 16 kbps	20
36	G.726 24 kbps	20
38	G.726 40 kbps	20
39	G.727 16 kbps	20
40	G.727 24-16 kbps	20
41	G.727 24 kbps	20
42	G.727 32-16 kbps	20
43	G.727 32-24 kbps	20
44	G.727-32 kbps	20
45	G.727 40-16 kbps	20
46	G.727 40-24 kbps	20
47	G.727 40-32 kbps	20
49	NetCoder 4.8 kbps	20
50	NetCoder 5.6 kbps	20
51	NetCoder 6.4 kbps	20
52	NetCoder 7.2 kbps	20
53	NetCoder 8.0 kbps	20
54	NetCoder 8.8 kbps	20
55	NetCoder 9.6 kbps	20
56	Transparent PCM	20
100	DTMF relay	20
101	Fax Relay	Different packet rates
102	Fax Bypass	20
103	Modem Bypass	20
104	RFC 2198 (Redundancy)	Same as channel's voice coder.

D.3 Mediant 2000 Default RTP/RTCP/T.38 Port Allocation

The following table shows the **Mediant 2000** Default RTP/RTCP/T.38 Port Allocation.

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

Channel Number	RTP Port	RTCP Port	T.38 Port
1	4000	4001	4002
2	4010	4011	4012
3	4020	4021	4022
4	4030	4031	4032
5	4040	4041	4042
6	4050	4051	4052
7	4060	4061	4062
8	4070	4071	4072
:	:	:	:
n	4000 + 10(n-1)	4001 + 10(n-1)	4002 + 10(n-1)
:	:	:	:
120	5190	5191	5192
:	:	:	:
192	5910	5911	5912
:	:	:	:
384	7830	7831	7832
:	:	:	:
480	8790	8791	8792

Reader's Notes

Appendix E DTMF, Fax and Modem Transport Modes

E.1 DTMF/MF Relay Settings

Users can control the way DTMF/MF digits are transported to the remote Endpoint, using the `DTMFtransport/MFtransport` configuration parameters. The following five modes are supported:

- **DTMF/MFtransportType= 0 (MuteDTMF/MF).** In this mode DTMF/MF digits are erased from the audio stream and are not relayed to the remote side. Instead silence is sent in the RTP stream.
- **DTMF/MFtransportType= 1 (RelayDTMF/MF).** In this mode, DTMF/MF digits are erased from the audio stream and are relayed to the remote side using a proprietary RTP syntax.
- **DTMF/MFtransportType= 2 (TransparentDTMF/MF).** In this mode DTMF/MF digits are left in the audio stream and the DTMF/MF relay is disabled.
- **DTMF/MFtransportType= 3 (acRelayDTMFOverRTP/ acRFC2833RelayMF).** In this mode, DTMF/MF digits are relayed to the remote side using the RFC 2833 Relay syntax.
- **DTMFtransportType = 7 (acRFC2833RelayDecoderMute).** In this mode, DTMF digits are being relayed to the remote side using the RFC 2833 Relay syntax. RFC 2833 digit packets that are received from the remote side are being muted on the audio stream.

E.2 Fax/Modem Settings

Users may choose to use one of the following transport methods for Fax and for each modem type (V.21/V.22/V.23/Bell/V.32/V.34):

- **fax relaydemodulation / remodulation**
- **bypass using a high bit rate coder to pass the signal**
- **transparent passing the signal in the current voice coder**
- **transparent with events transparent + issues fax/modem events**

When the fax relay mode is enabled, distinction between fax and modem is not immediately possible at the beginning of a session. The channel is therefore in "Answer Tone" mode until a distinction is determined. The packets being sent to the network at this stage are fax relay packets. (The packets can be either T.38-complaint, or FRF.11-based proprietary syntax, selected by setting the channel's configuration parameter `UseT38orFRF11`.)

E.3 Configuring Fax Relay Mode

When `FaxTransportType= 1` (relay mode), then on detection of fax the channel automatically switches from the current voice coder to answer tone mode, and then to fax relay mode. The `UseT38orFRF11` configuration parameter defines either T.38-compliant network packets or proprietary FRF.11-based packets (the last mode should be used mostly for backward-compatibility with previous software versions).

When Fax transmission has ended, the reverse switching from fax relay to voice is performed. This mode switching automatically occurs at both the local and remote Endpoints.

Users can limit the fax rate using the FaxRelayMaxRate parameter and can enable/disable ECM fax mode using the FaxRelayECMEnable parameter.

When using T.38 mode, the User can select between two protection strategies – redundancy packets or forward error correction (FEC). This selection is made using the T38FaxRelayProtectionMode configuration parameter. The User can also control a special (proprietary) redundancy mode that was specially designed to improve protection against packet loss using the EnhancedFaxRelayRedundancyDepth parameter. Although this is a proprietary redundancy scheme, it is compatible with other T.38 decoders. When using FRF.11 mode, only redundancy packets are supported. The depth of the redundancy in both protocols (that is, the number of repetitions) is defined by the FaxRelayRedundancyDepth configuration parameter.

Note: T.38 mode currently supports only the T.38 UDP syntax.

E.4 Configuring Fax/Modem Bypass Mode

When VxxTransportType= 2 (FaxModemBypass, Vxx can be one of the following: V32/V22/V21/Bell/V34/Fax), then on detection of Fax/Modem, the channel automatically switches from the current voice coder to a high bit-rate coder, as defined by the User, with the FaxModemBypassCoderType configuration parameter.

If relay is enabled for one of the modes (FAX/Modem), then the Answer Tone mode packets are relayed as fax relay packets.

During the bypass period, the coder uses the packing factor (by which a number of basic coder frames are combined together in the outgoing WAN packet) set by the User in the FaxModemBypassM configuration parameter. The network packets generated and received during the bypass period are regular voice RTP packets (per the selected bypass coder) but with a different RTP Payload type.

When Fax/Modem transmission ends, the reverse switching, from bypass coder to regular voice coder, is carried out.

Note: When Fax relay is enabled, V21TransportType must be set to disable (Transparent) mode.

E.5 Supporting V.34 Faxes

Unlike the T.30 fax machines, the V.34 fax machines has no relay standard to transmit the data over IP to the remote side. Therefore ACL gives the following operation modes to transport the V.34 fax data over the IP.

(For all the setups described below, the CNG detector is disabled.)

E.5.1 Using Bypass Mechanism for V.34 Fax Transmission:

Configuration:

- Fax transport mode – Relay/Bypass
- Vxx modem mode – Bypass

Expected events for V.34Fax → V.34Fax – Bypass Mode are shown in Table E-1.

Table E-1: V.34Fax → V.34Fax – Bypass Mode

Calling	Answering
	EV_DETECT_MODEM (2100 AM + Reversal)
EV_DETECT_MODEM (Refer to Note 1 below)	
	EV_DETECT_FAX (V.21 at V.8 format)
EV_DETECT_FAX (Refer to Note 1 below)	
EV_END_FAX	EV_END_FAX

Note 1: The board is changing its status to bypass mode upon receiving fax bypass packet from the remote side.

Note that if the fax transport type is set to relay, the User can implement fax relay benefits for the T.30 fax machines and, in parallel, use V.34 fax with its full rate. Therefore, this setup is most recommended by ACL. Also note that if CNG relay is used, in some cases, such as for manual answering machine, the fax may revert to T.30 fax with a speed of 14400 bps.

E.5.2 Using Events Only Mechanism for V.34 Fax Transmission

Use events only mode to transmit V.34 fax with its maximum capabilities:

Configuration:

- Fax transport mode – Events only mode
- Vxx modem mode – Events only mode

Expected events for V.34Fax → V.34Fax – Events Only Mode are shown in Table E-2.

Table E-2: V.34Fax → V.34Fax – Events Only Mode

Calling	Answering
	EV_DETECT_MODEM (2100 AM + Reversal)

E.5.3 Using Relay mode for all kind of fax machines (T.30 and V.34).

The User can force the V.34 fax machines to fallback to T.30 and work at relay mode.

Configuration:

- Fax transport mode – Relay
- Vxx modem mode – Disable
- CNG detectors mode – Disable

In this mode, the fax events are identical to the regular T.30 fax session over T.38 protocol.

Expected events for V.34Fax → V.34Fax – Relay Mode are shown in Table E-3.

Table E-3: **V.34Fax → V.34Fax – Relay Mode**

Calling	Answering
	EV_DETECT_ANSWER_TONE
	EV_DETECT_FAX
EV_DETECT_FAX	
EV_END_FAX	EV_END_FAX

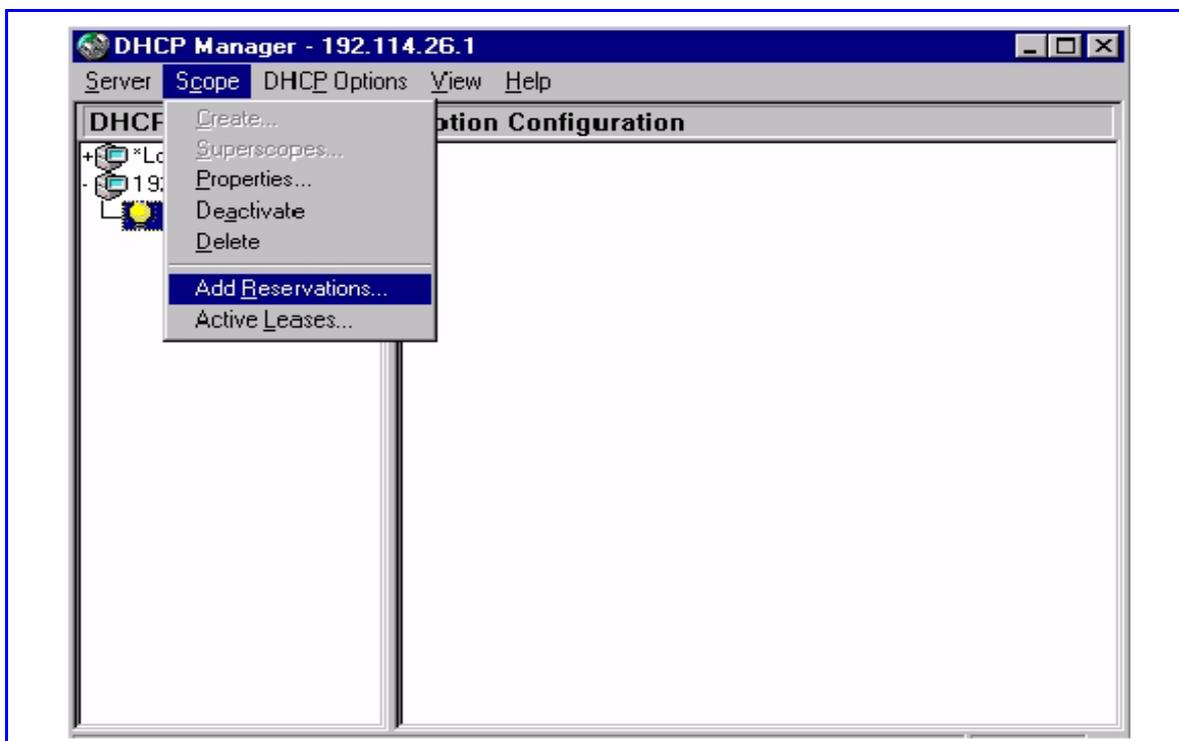
Appendix F DHCP Server Configuration

Note: For correct operation with BootP clients, install Windows™ NT4 service pack 4 **after** enabling the DHCP server service on the NT server.

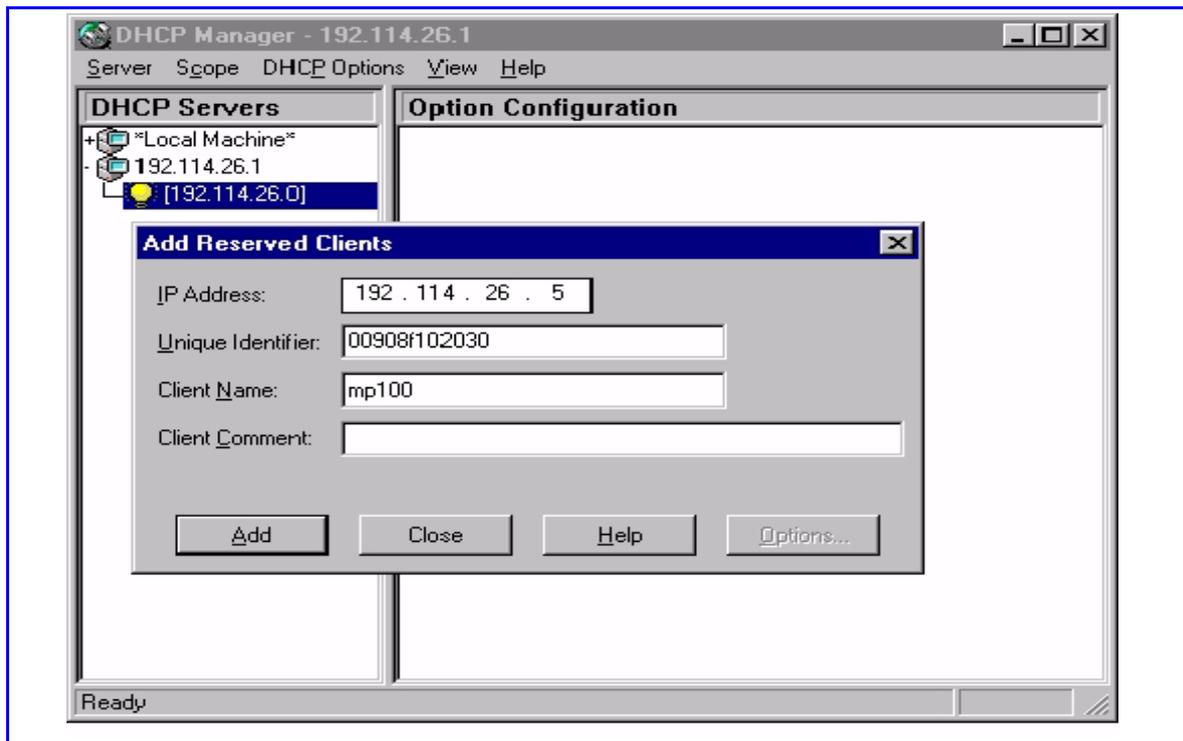
➤ **To configure the DHCP Server, take the next 16 steps:**

Step 1. Start the system. DHCP Manager displays the following screens:

Figure F-1: Scope Menu

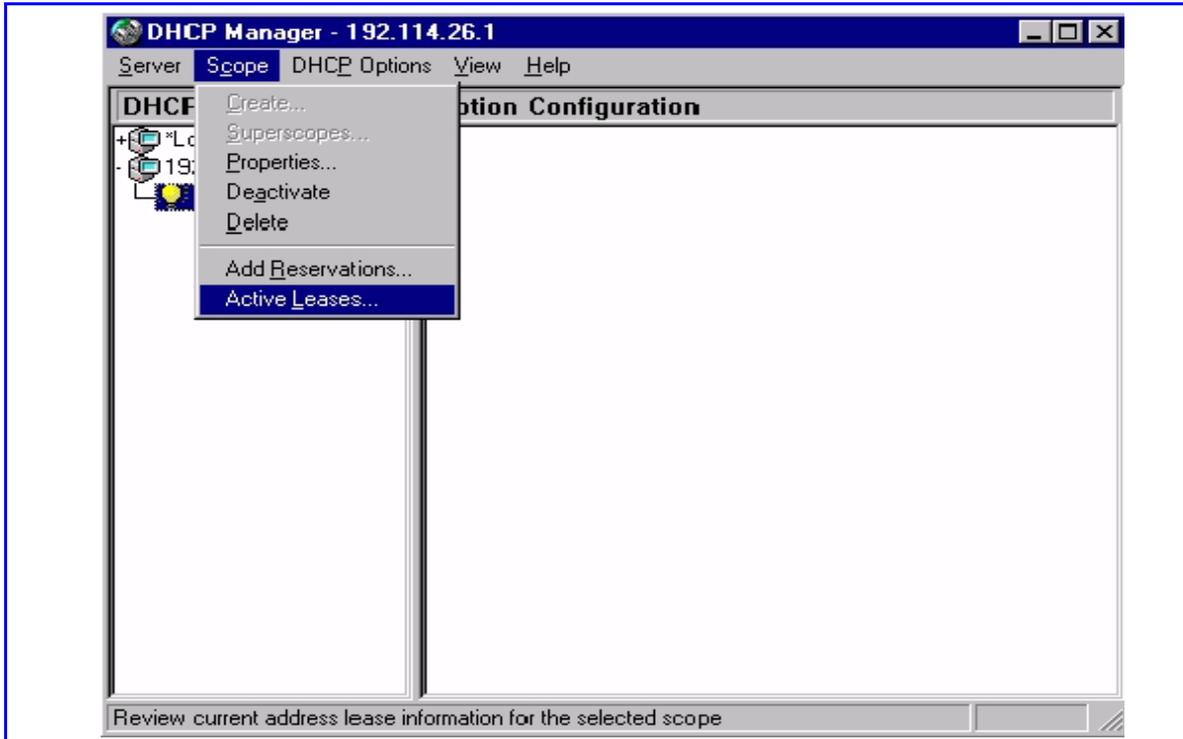


Step 2. From the **Scope** menu, select **Add Reservations**. The Add Reserved Clients screen appears.

Figure F-2: Reserved Clients Menu


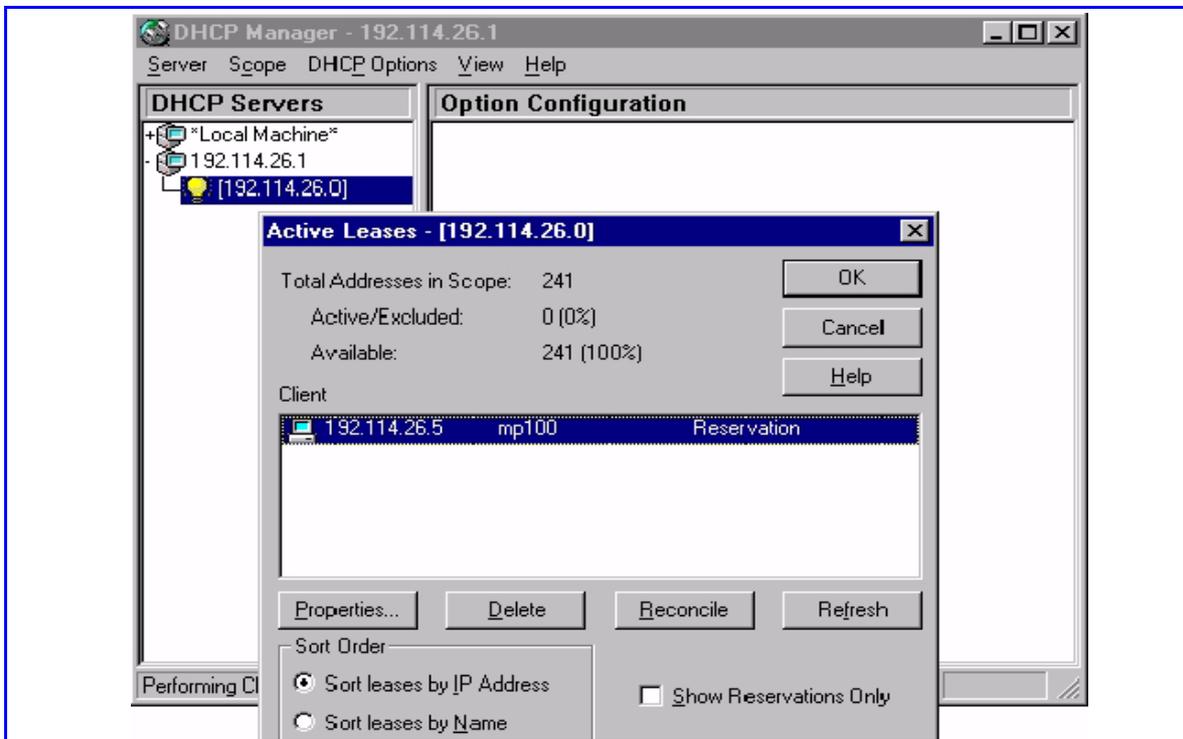
- Step 3.** Enter the **IP address** you want to allocate to the **Mediant 2000** Media Gateway. The IP address reservation should be inside your DHCP Scope.
- Step 4.** In the Unique Identifier field, enter the hardware MAC address, 12 digits, from your **Mediant 2000** Gateway (provided in **Mediant 2000** documentation).
- Step 5.** In the **Client name** field, **enter** any free text.
- Step 6.** Press the **Add** button and then press the **Close** button.
- Step 7.** From **Scope** menu, select **Active Leases**. Active Leases Select Screen

Figure F-3: Scope Menu, Active Leases Option

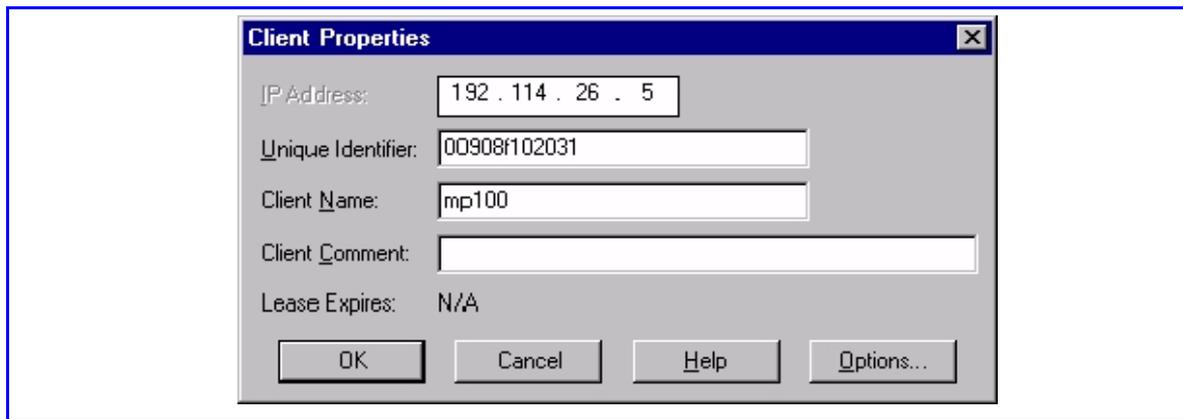


The **Active Leases** screen displays the following selection box:

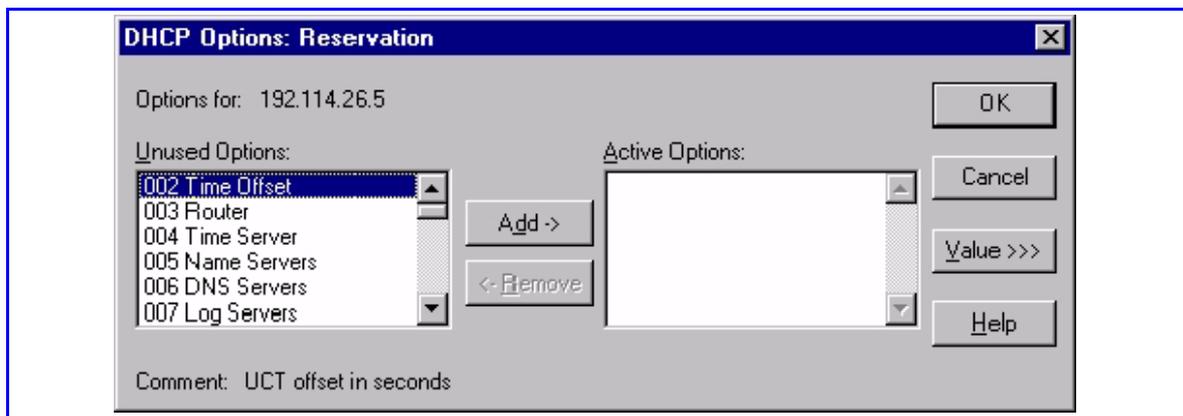
Figure F-4: Active Leases Selection Box



Step 8. Click the **Properties** button. The Client Properties screen appears.

Figure F-5: Client Properties Screen


Step 9. Click the **Options** button to display the DHCP Options: Reservation screen.

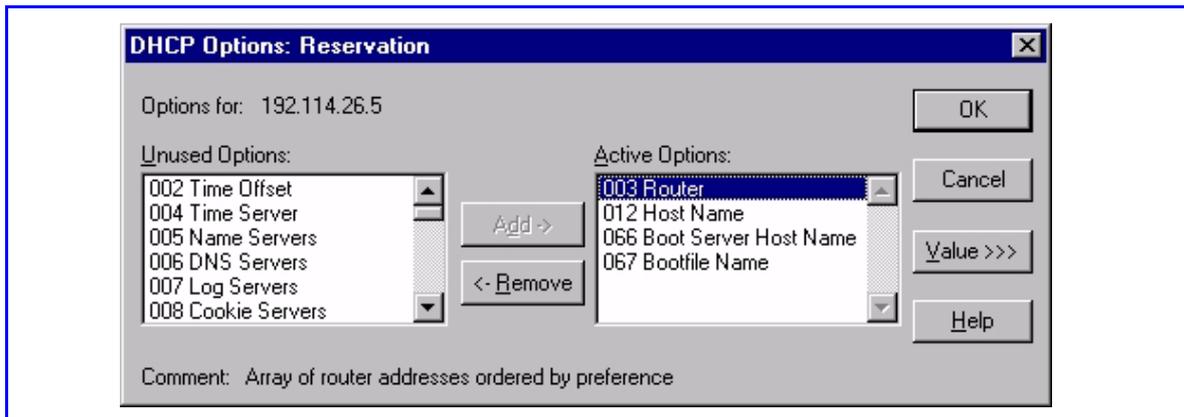
Figure F-6: DHCP Options - Reservation Screen


Step 10. From the left-hand table of the Reservation screen, add the following extension fields:

- **003 Router** (Default router/Gateway)
- **012 Host Name** (BootP client name such as **Mediant 2000** or other customer selected name)
- **066 Boot Server Host Name** (Domain name or IP address of the TFTP server)
- **067 Bootfile Name** (Such as "ram.cmp")

The selected extension fields are displayed in the right-hand table of the Reservation screen as shown below:

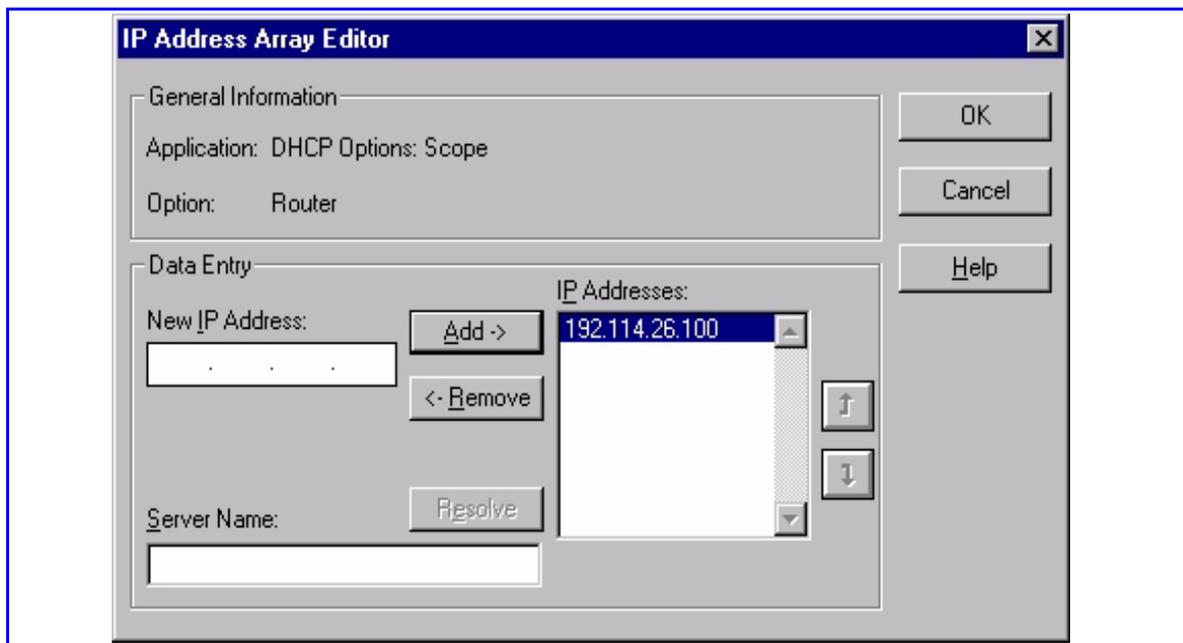
Figure F-7: DHCP Options - Active Options Displayed



Step 11. Select the **003 Router** field as shown above, and click on the **Value** button.

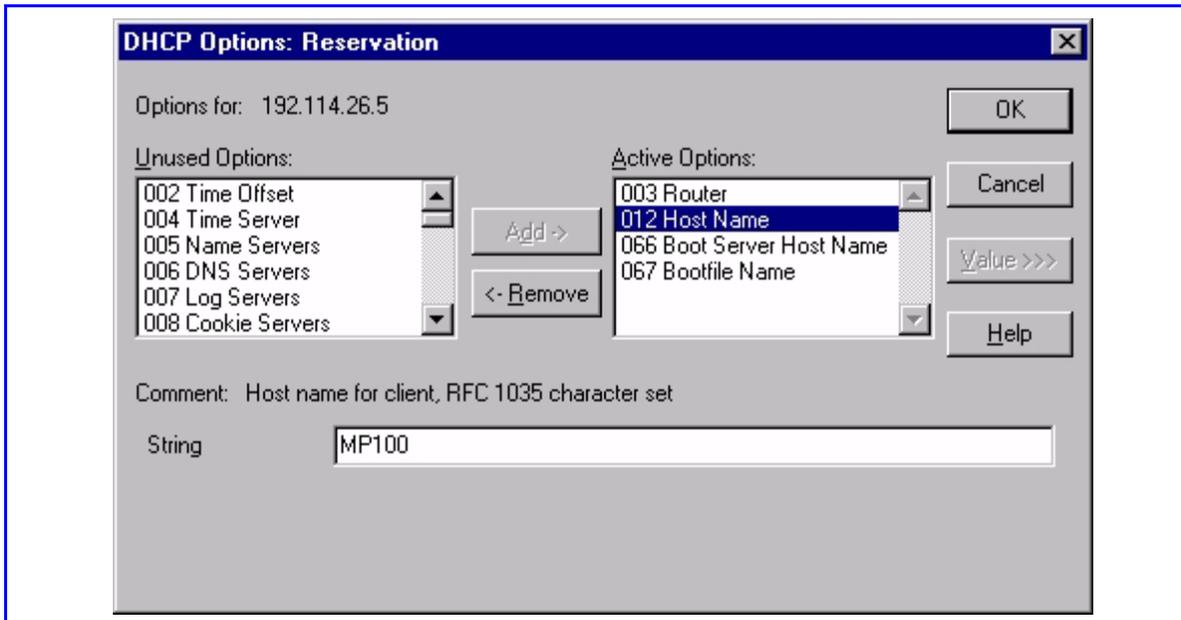
Step 12. Edit the **IP Address Array** in the IP Address Array Editor screen.

Figure F-8: IP Address Array Editor Screen



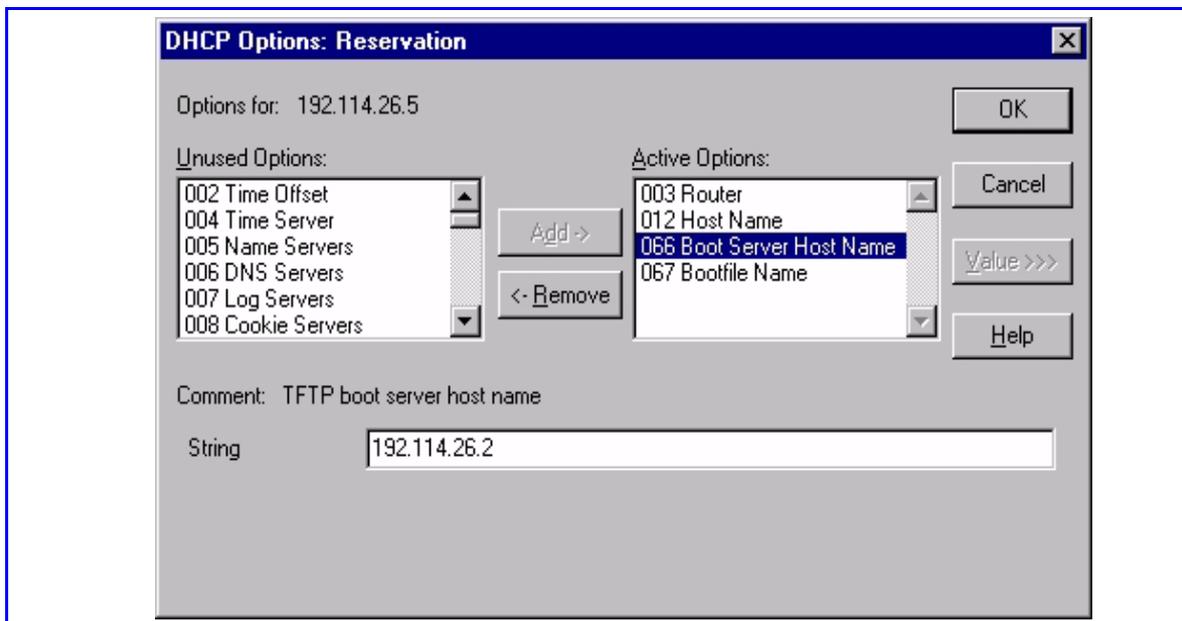
Step 13. Click **OK**, and set the value for the Host name, which can be any character string.

Figure F-9: Host Name Screen



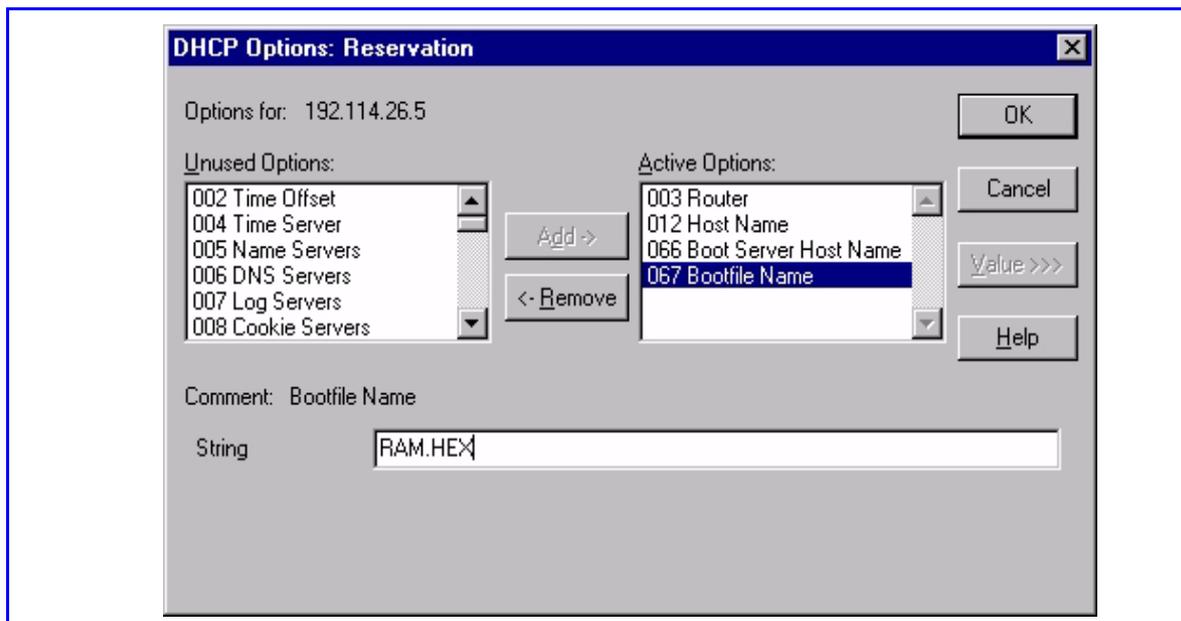
Step 14. Click **OK**, and set the value of the IP address for the Boot Server Host Name. (That is, the server where the TFTP server is installed.)

Figure F-10: Boot Server Host Name Screen



Step 15. Click **OK** and set the value of the Boot file name for the TFTP downloaded file.

Figure F-11: Bootfile Name Screen



Step 16. Click **OK**, and exit from the DHCP server installation.

Reader's Notes

Appendix G BootP Server Configuration

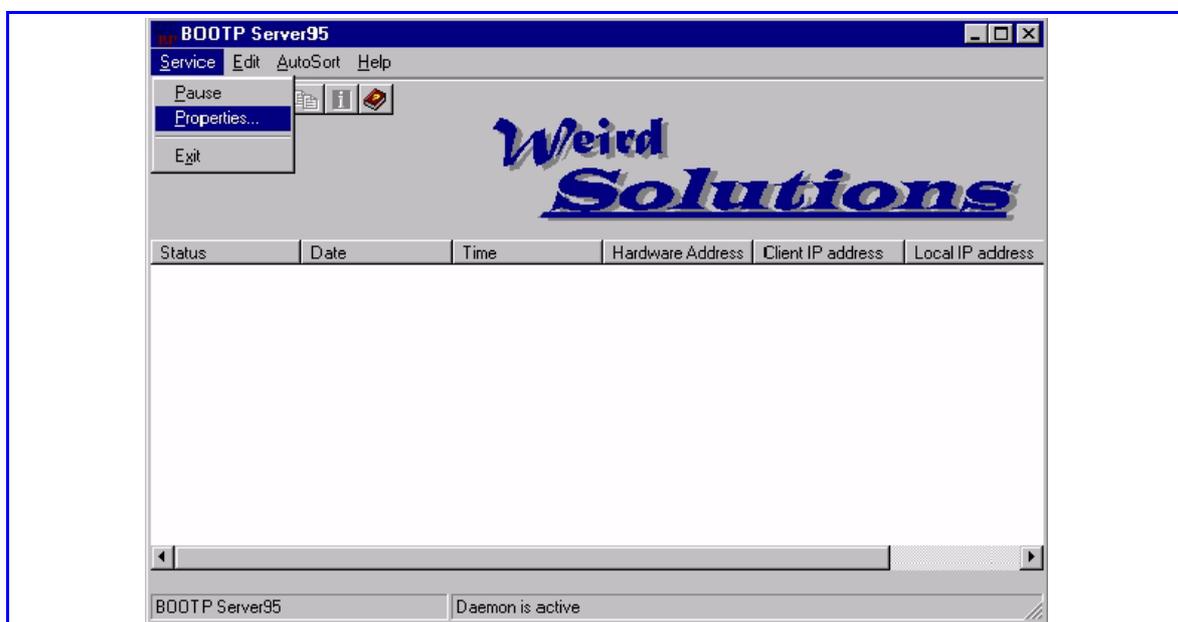
G.1 BootP Server Configuration

The BootP Server 95 can be downloaded from www.weird-solutions.com; it can be installed on Windows™ 95/98 or Windows™ NT.

➤ **To configure the BootP Server, take the next 17 steps:**

Step 1. From the Programs menu on Windows™ 98 PC, start the BootP Server 95 application . The following screen appears:

Figure G-1: Properties Screen



Step 2. Open the **Service** menu and select **Properties**.

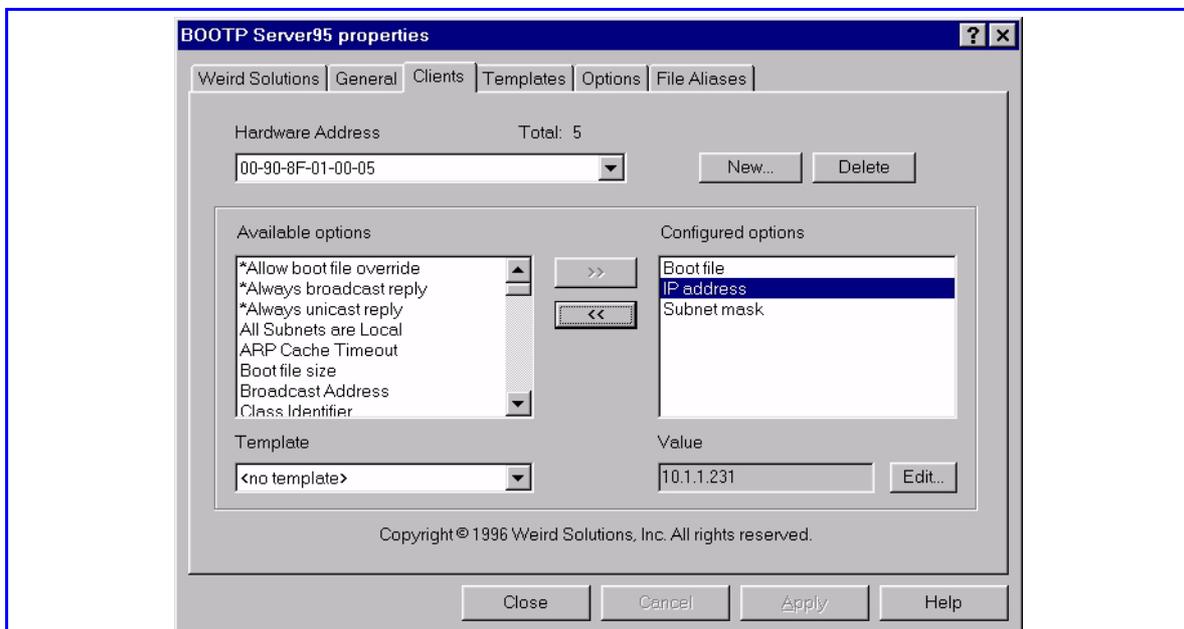
Figure G-2: Clients Screen



Step 3. Select the **Clients** tab.

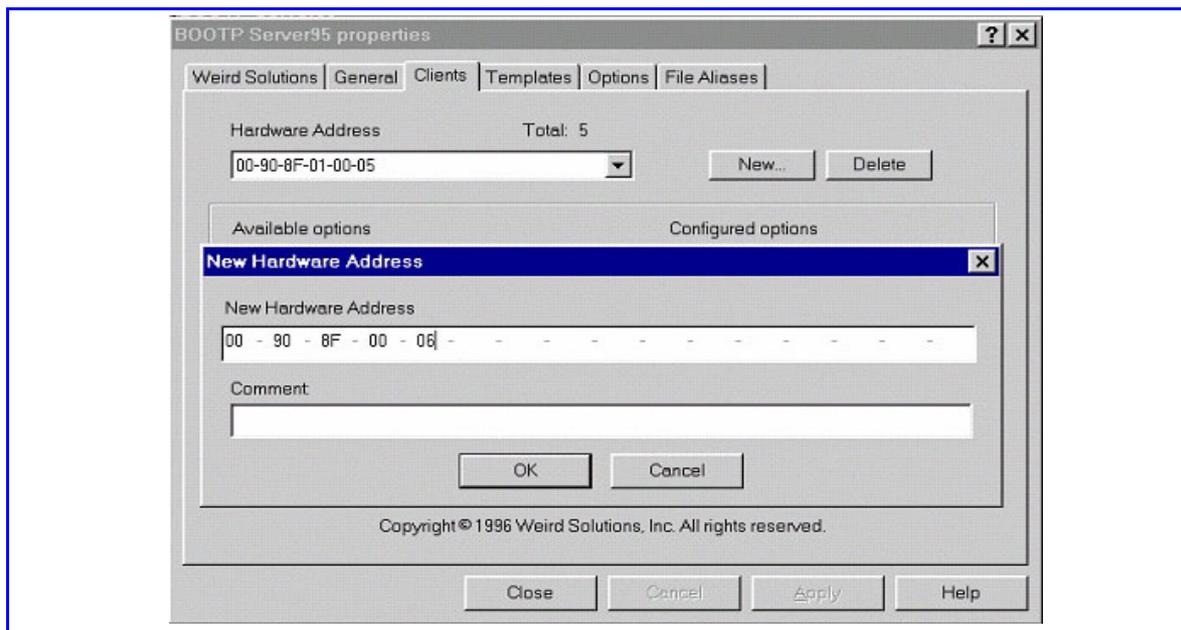
Step 4. In the **Options** screen (below), click the **New** button to define a new Hardware Address (MAC address), or select an already defined Hardware Address if you would like to modify some of its previously Configured options.

Figure G-3: Options Screen



Step 5. Enter the new Hardware Address of 12 digits and press **OK**. The Hardware Address can be found on a Label attached to the bottom of the **Mediant 2000** unit, (it is also called the MAC address).

Figure G-4: New Hardware Address

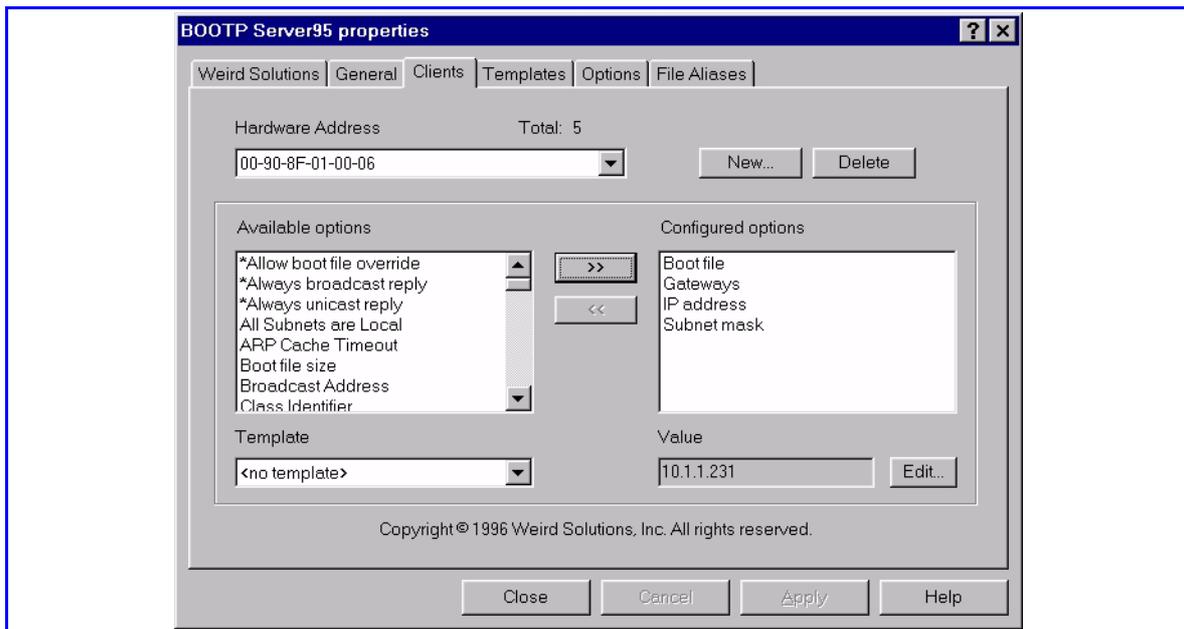


Note: Do not fill the Comment field.

Refer to Section G.3 on page 168 if you need to change the given MAC address.

The new hardware address appears in the Hardware Address drop-down list.

- Step 6.** In the Available options list, select an option. Use the scroll-down arrows to assess the options on the list. Click the  button to move the option to the Configured options list on the right. You can move back from any option by using the  button.

Figure G-5: Configured Options Screen


Step 7. From the Available Options list, select the following options and move them to the Configured options list:

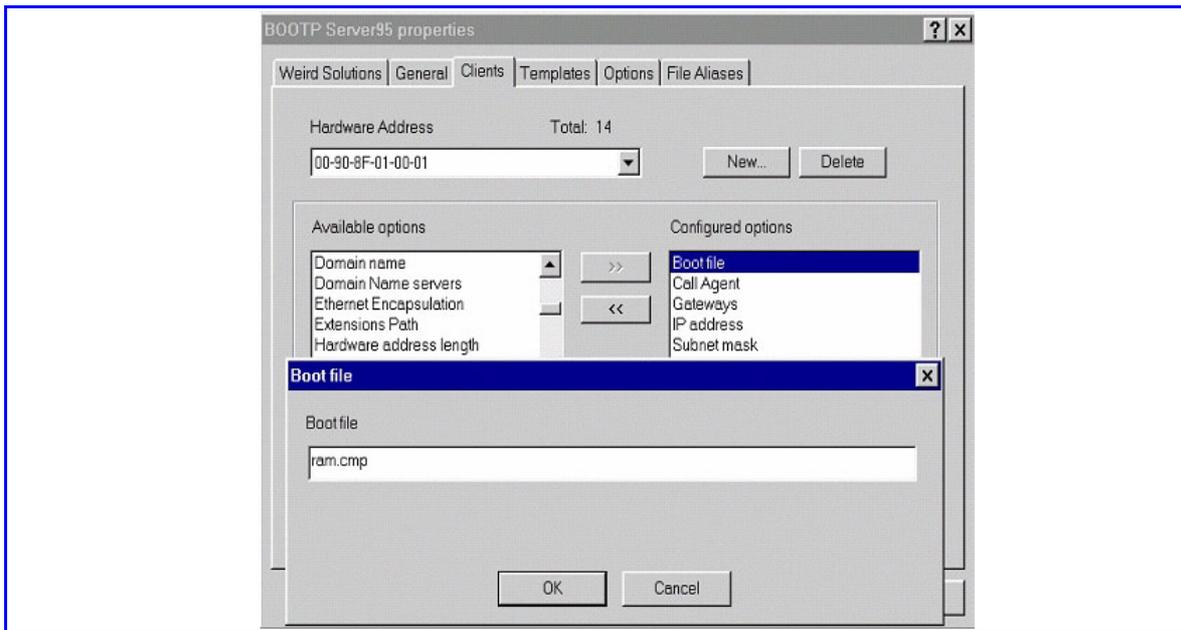
- Boot file
- Gateways
- IP address
- Subnet mask
- Call Agent - (see Section E-2 for setting the Call Agent option).

Step 8. Set a value to each of the following options:

- Boot file - such as ram.cmp
- Gateways - IP address of default Gateway, can be any address within the subnet
- IP address - such as 10.1.1.33
- Subnet Mask - such as 255.255.255.0 (see the restrictions in step 14.)
- Call Agent - IP address of the Call Agent.

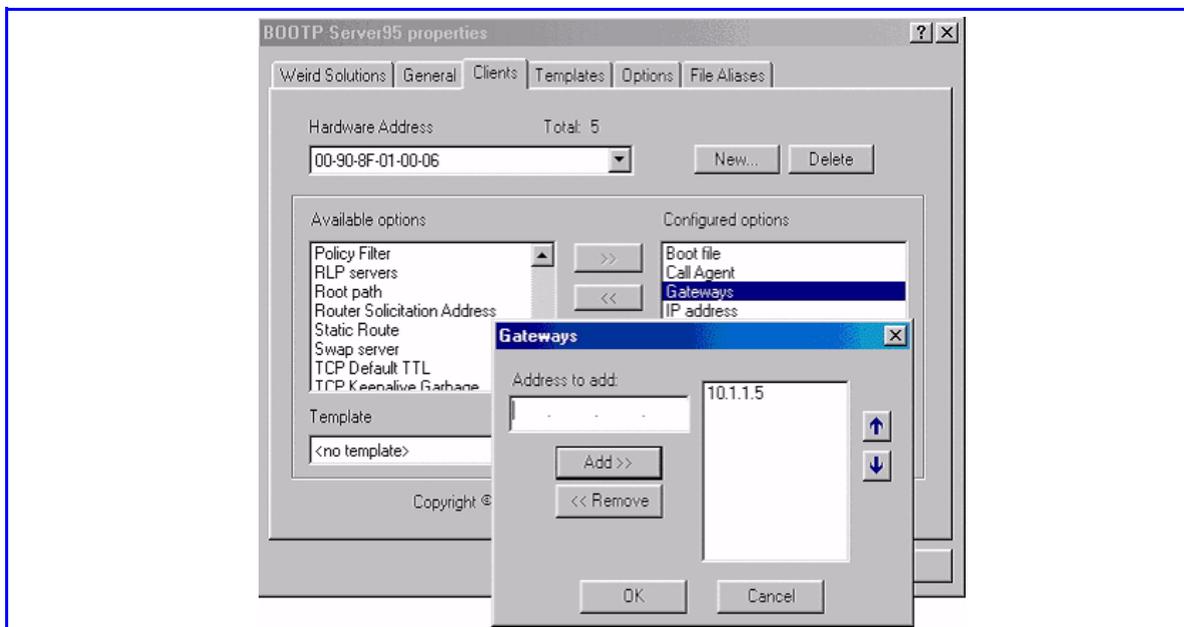
Step 9. To enter the Boot File name, from the Configured Options list, select **Boot file** and press the **Edit** button.

Figure G-6: Boot File Screen



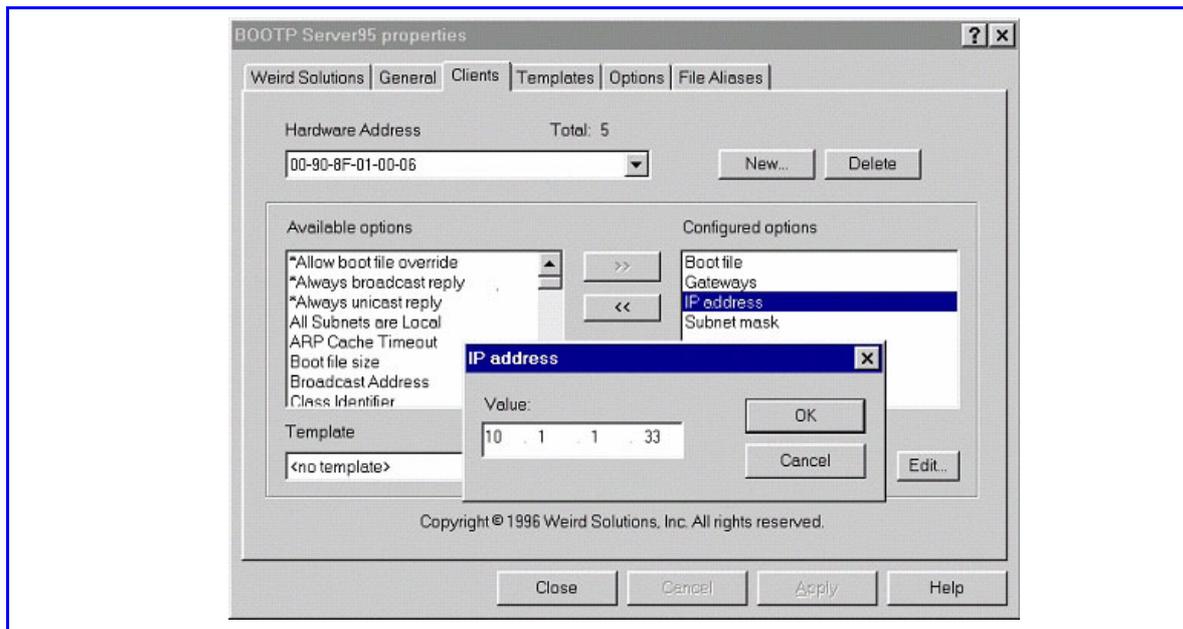
- Step 10.** Enter the boot file name and optional *ini* file name such as *ram.cmp;tp1610.ini*. Click **OK**.
- Step 11.** Enter the **Default Gateway IP address**, by selecting Gateways and pressing the Edit button. The default Gateway address must be within your subnet.

Figure G-7: Default Gateway Screen



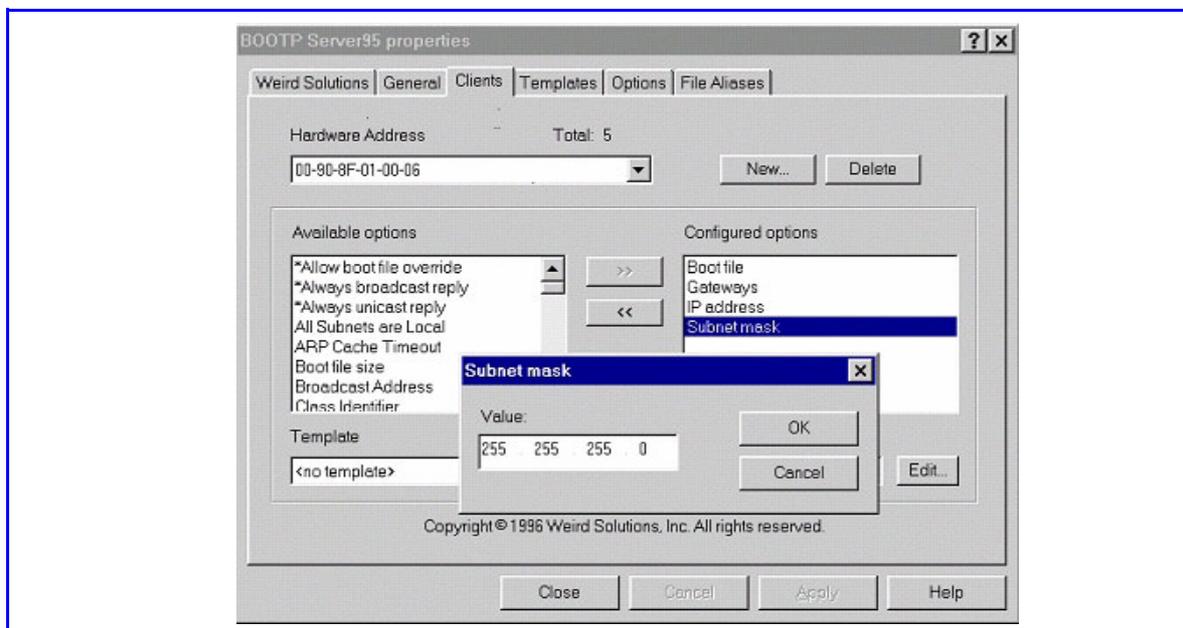
- Step 12.** Enter the **Address to add**, then press **Add** button and **OK**. **Only a single default Gateway can be used.**
- Step 13.** Enter the **Mediant 2000 IP address**.

Figure G-8: Mediant 2000 IP Address Screen



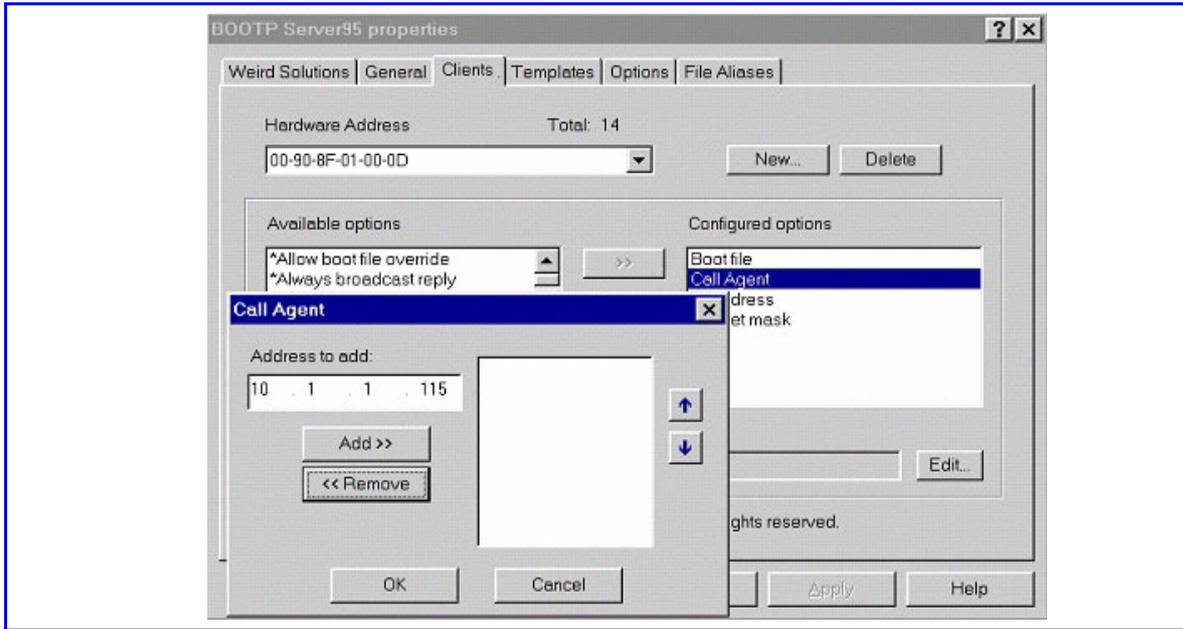
- Step 14.** Enter the Subnet Mask. Set the Subnet Mask to a valid value in accordance with the IP address. (That is, class C IP addresses can only have subnet mask starting with 255.255.255.X, while class B IP addresses can only have subnet mask starting with 255.255.X.X, and class A IP addresses can only have subnet mask starting with 255.X.X.X).

Figure D-9: Subnet Mask Screen



- Step 15.** Enter the **Call Agent IP Address**, by selecting Call Agent and pressing the Edit button.

Figure G-9: Call Agent IP Screen

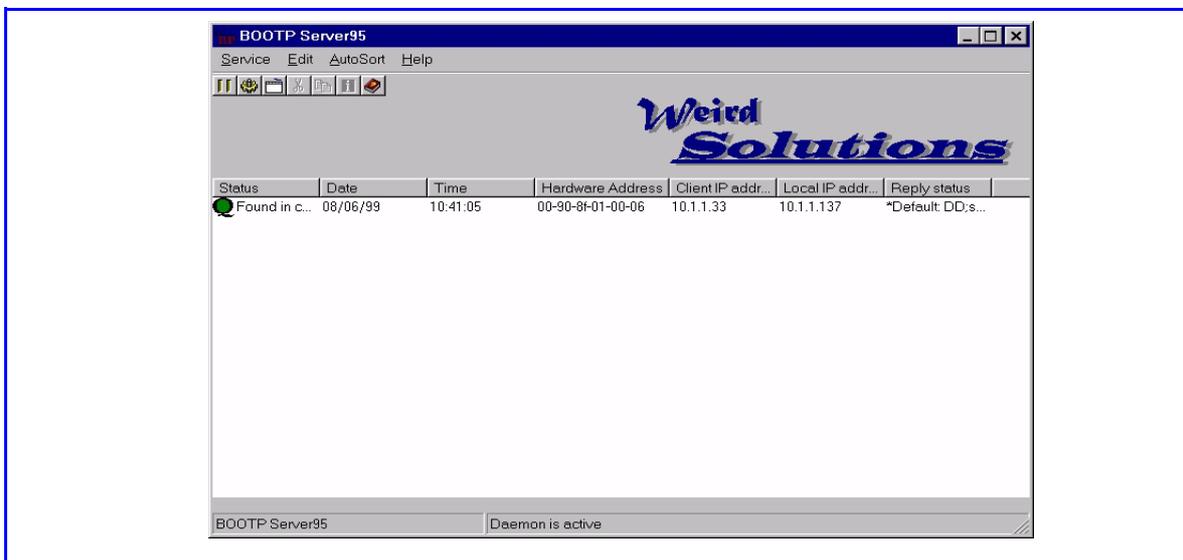


Step 16. Enter the **Address to add**, then press **Add** button and **OK**.

Step 17. Click **Close** to finish the BootP configuration.

After a Power up or Reset operation, the following Status screen is displayed:

Figure G-10: Status Screen



******* After the BootP process has been successfully completed, the **Mediant 2000** receives the name of the Image file (ram.cmp), IP address, Subnet mask, default Gateway. The next step is to download the Image file from TFTP server.

Note: The TFTP server and BootP servers should be installed on the same Host.

G.2 Setting Call Agent Address Option

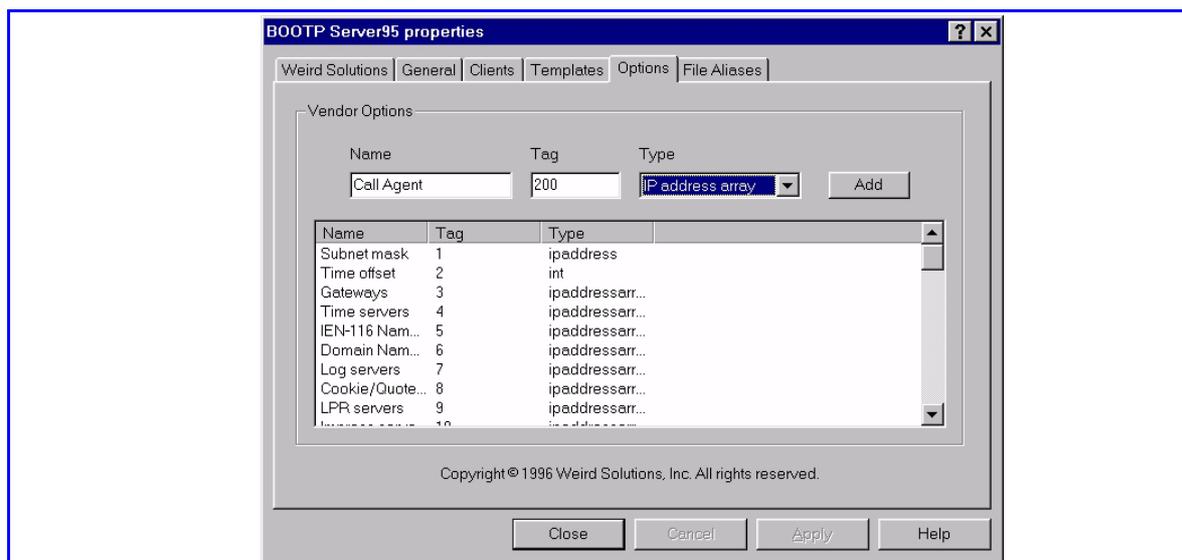
The Call Agent Address is required especially for MGCP RSIP (restart in progress) messages.

The Call Agent address can be set if it has not yet been defined in the list of the Available options.

➤ **To set the Call Agent Address, follow the next 2 steps:**

Step 1. Select Options menu and enter Name, Tag and Type fields in the 'Vendor options' as shown on the next screen. (Tag = 200, Type = IP address array).

Figure G-11: Call Agent (IP Address Array) Screen



Step 2. Press **Add** and **OK**.

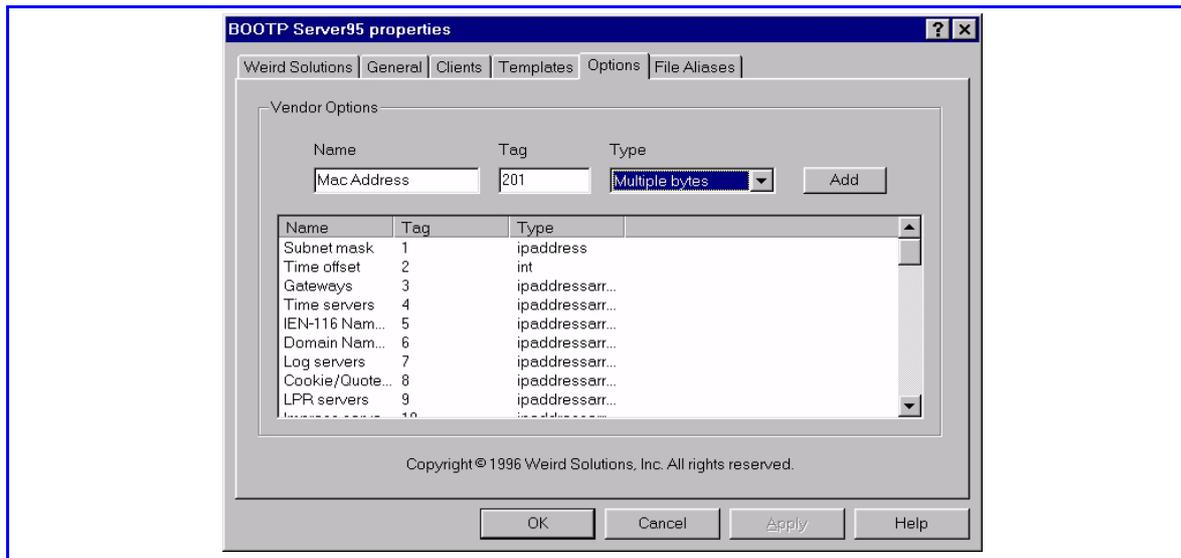
G.3 Changing MAC Hardware Address

If the MAC address has not yet been defined in the list of the Available options, follow the next procedure.

➤ **To change the MAC Hardware address, take the next 7 steps:**

Step 1. Select Options menu and enter Name, Tag and Type fields in the 'Vendor options' as shown on the next screen. (Tag = 201, Type = Multiple bytes).

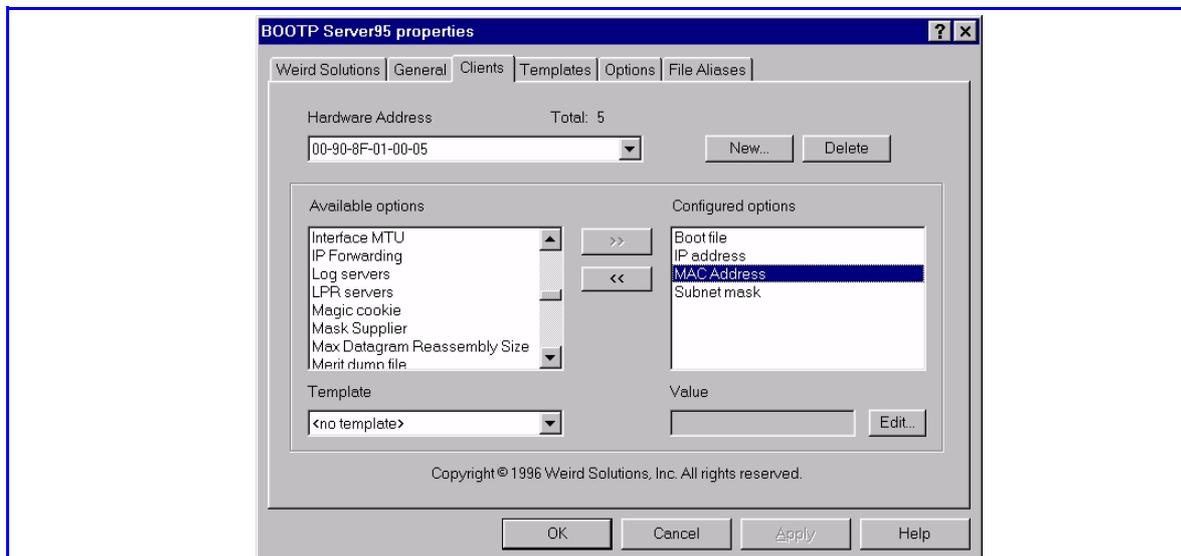
Figure G-12: MAC (Multiple Bytes) Screen



Step 2. Press **Add** and **OK**.

Step 3. From the **Configured options** list, select MAC address. In addition to MAC address, configure and set a value to the IP address, but other parameters are not required. (For these procedures, the IP address can be any valid IP address.)

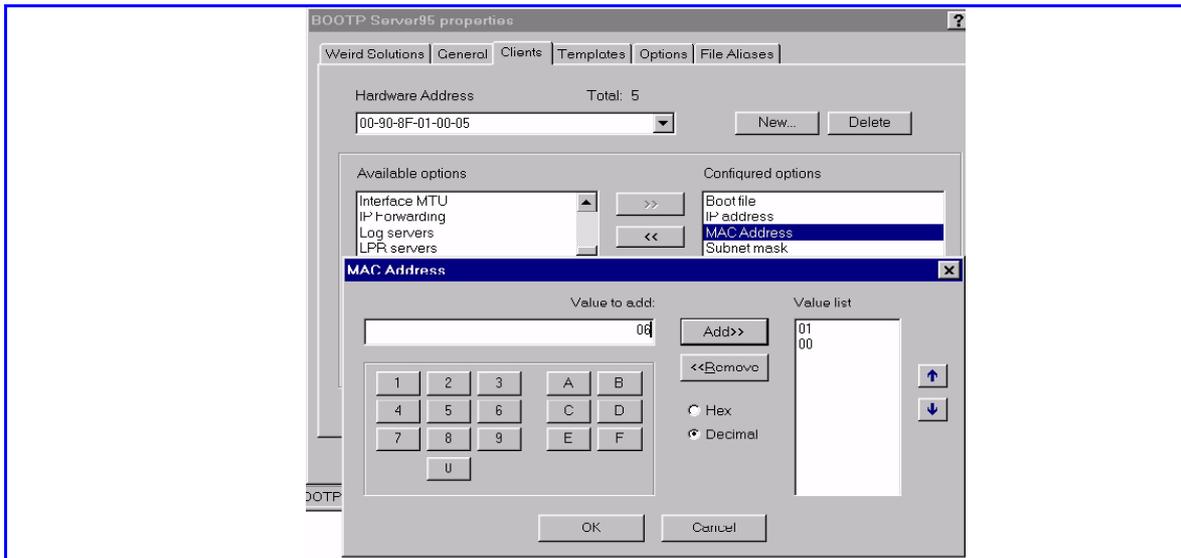
Figure G-13: MAC (Options) Screen



Step 4. Click on Edit button and enter value for new MAC address. Enter six LSB digits such as 01 00 06. The MSB digits of the MAC address are fixed, and set in the **Mediant 2000** software to 00 90 8F.

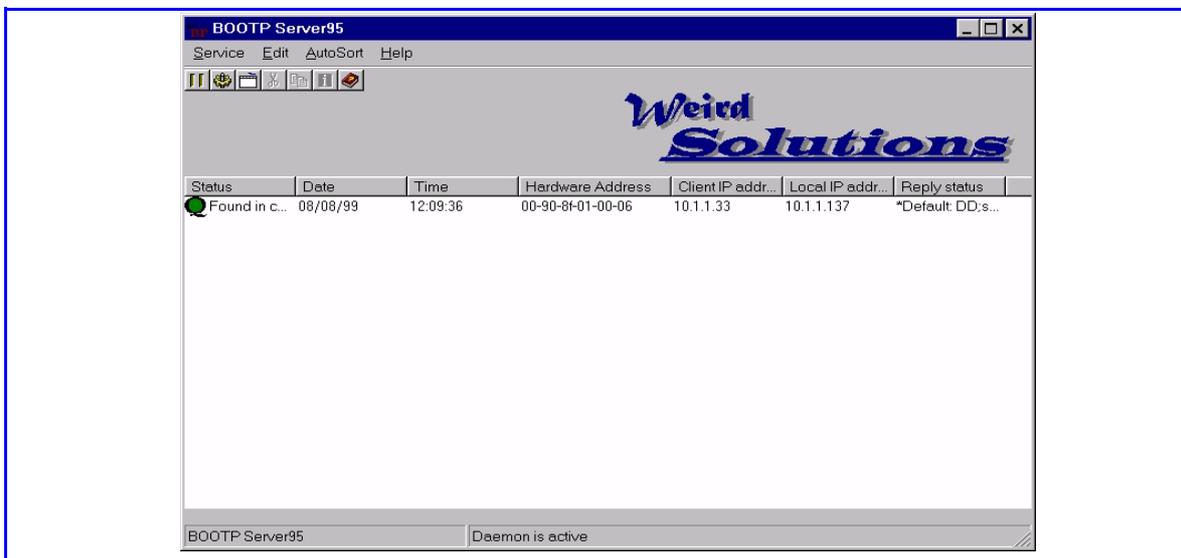
After entering the LSB digits you will receive the MAC address: **00 90 8F 01 00 06**

Figure G-14: MAC Address Screen



- Step 5.** Press **OK** and then **Close**.
- Step 6.** To program the new MAC address in the **Mediant 2000** unit, connect the **Mediant 2000** to power or press its Reset button.
- Step 7.** To validate the programming of new MAC address, press again on the Reset button and view the printed line on the RS-232 terminal screen, and on the BootP GUI screen.

Figure G-15: Status Validation Screen



Appendix H TFTP Server Configuration

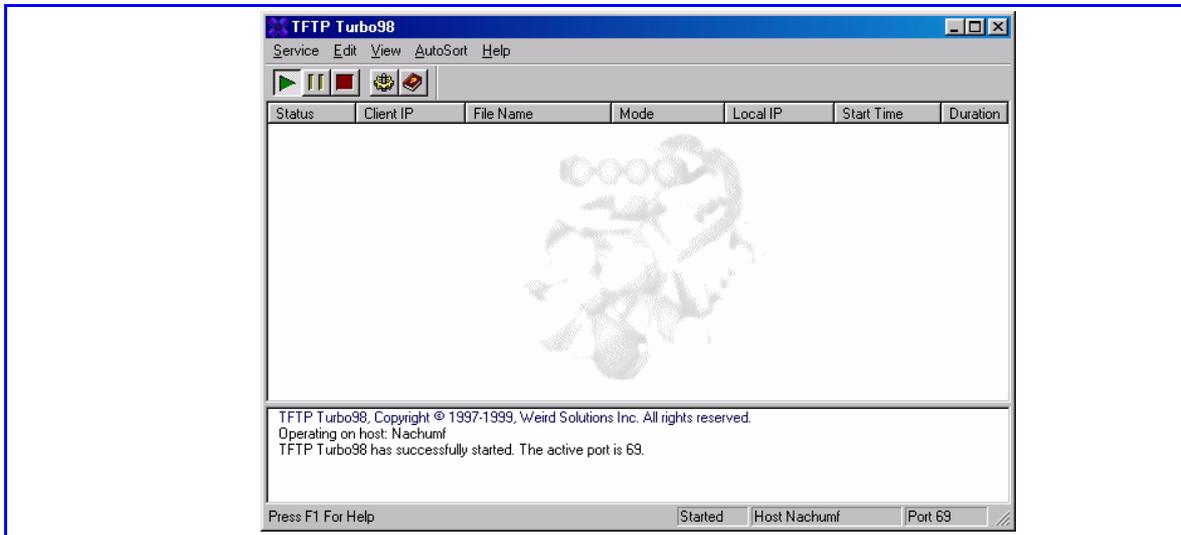
The TFTP server (“TFTP Turbo 98”) can be downloaded from www.weird-solutions.com; it can be installed on Windows™ 95/98 or Windows™ NT.

Note: The TFTP server and BootP servers should be installed on the same Host

➤ **To configure the TFTP Server, take the next 5 steps:**

Step 1. After entering the TFTP server application, choose the **Service** menu and then **Properties**.

Figure H-1: TFTP Initial Screen



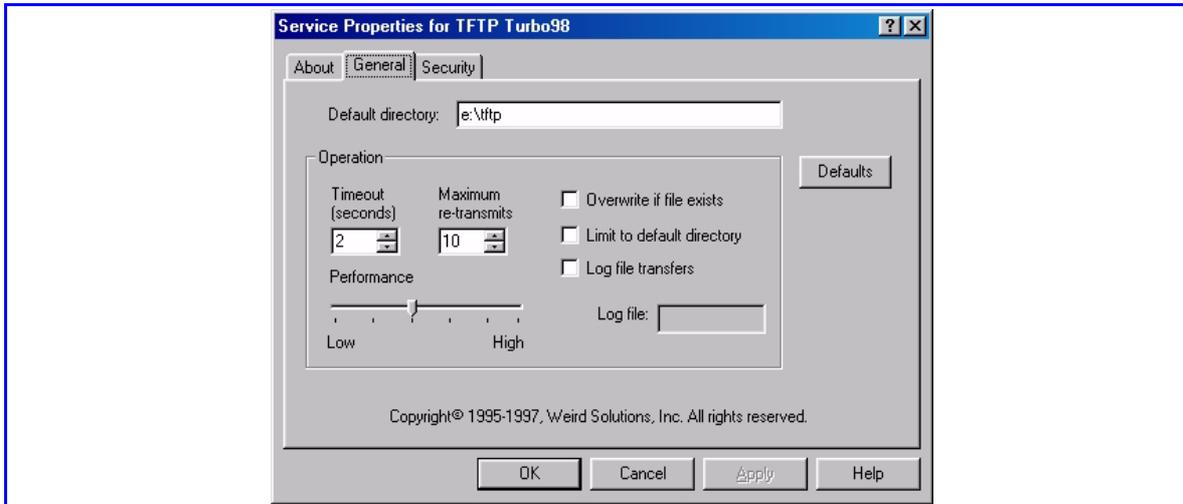
The Service Properties Screen is displayed.

Figure H-2: Service Properties Screen



Step 2. Select the **General** button to display the General screen.

Figure H-3: TFTP General Screen



Step 3. Enter the **Default directory** in which the **ram.cmp** file is located. Do not forget to copy the file to that directory, before starting to use the TFTP server.

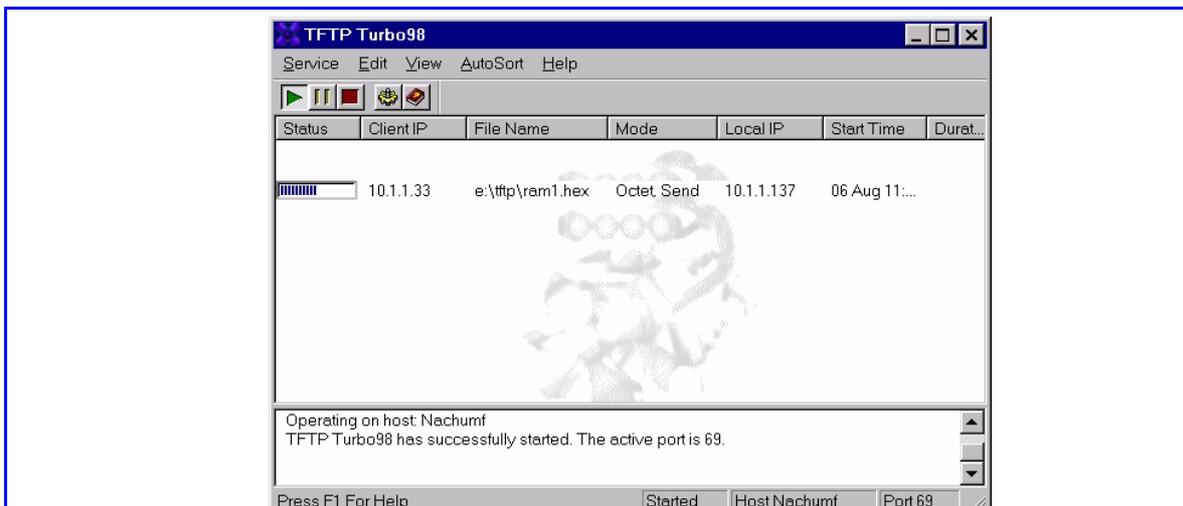
Step 4. Set **Timeout** to 2 seconds, and **Maximum re-transmits** to 10.

Step 5. Select **Close** button to finish the TFTP server installation.

During the **Mediant 2000** starting procedure, its TFTP client requests the TFTP server to download the **ram1.cmp**, and an optional **tp1610.ini** file.

The TFTP Final Status screen is displayed.

Figure H-4: TFTP Final Status Screen



Appendix I CAS Protocol Table

I.1 Constructing a CAS Protocol Table

Constructing or Modifying a CAS Protocol Table for CAS-Terminated Protocols

The protocol table file is a textual file containing the protocol's state machine that defines the whole protocol process. It is constructed of States, pre-defined Actions/Events, and pre-defined Functions. Thus, the User has full control of the CAS protocol by a relatively simple textual tool. The User is capable of defining or changing any CAS protocol by writing the protocol state machine in a text file with a few simple rules defined by AudioCodes.

The User Procedure to generate the protocol file is shown below:

- **Learn the protocol text file rules (rules detailed in this manual and syntax are based on C pre-processor commands).**
- **Get the AudioCodes example.**
- **Build the specific protocol/script text (xxx.txt) file and its related numerical value h file (xxx.h).**
- **Compile the xxx.txt with the “TrunkPack Downloadable conversion utility” to produce the xxx.dat file. Refer to Section D.1 (“API Demonstration Utilities”) for a detailed description of the utility usage.**
- **Download User_protocol.dat file to the board via acOpenBoard() command at initialization phase.**

I.2 Table Elements

CASSetup.h file includes all the pre-defined tools needed to build a new protocol text file or modifying an existing one. The protocol table file is composed of the following bricks:

I.3 INIT variables

INIT variables that the User can change their numeric values in UserProt_defines_xxx.h. For example, INIT_RC_IDLE_CAS defines the ABCD bits expected to be received in the IDLE state, INIT_DTMF_DIAL defines the on-time and off-time for the DTMF digits generated towards the PSTN. See the detailed list in CASSetup.h and in the sample protocol text file. Refer to the following ST_INIT detailed explanation.

I.4 Actions

Actions (i.e., protocol table events) - Actions are protocol table events activated either by the DSP (e.g., EV_CAS_01) or by the User (e.g., EV_PLACE_CALL, EV_TIMER_EXPIRED1). The full list of the possible pre-defined events list can be found in the CASSetup.h file.

I.5 Functions

Functions - The 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 ms.) SEND_CAS(AB value, CD value). A full list of the possible pre-defined functions list can be found in the CASSetup.h file.

I.6 States

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 colons. The body of the state is composed of up to 4 unconditional performed functions and list of actions that may trigger this state.

As an example, the following Table I-1, "ST_DIAL: Table Elements," on page 174 was taken from an E&M wink start table protocol file:

Table I-1: ST_DIAL: Table Elements

Action	Function	Parameter			Next State
		#1	#2	#3	
FUNCTION0	SET_TIMER	2	Extra Delay Before Dial	None	DO
EV_TIMER_EXPIRED2	SEND_DEST_NUM	None	None	None	NO_STATE
EV_DIAL_ENDED	SET_TIMER	4	No Answer Time	None	ST_DIAL_ENDED

When the state machine reaches the dial state, it sets timer number 2 and then waits for one of the two possible actions to trigger: either timer 2 expiration or end of dial event. When timer 2 expires, the protocol table executes SEND_DEST_NUM function and stays 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 NEXT_STATE field.

Although Users can define their own states, there are two states defined in the CASSetup.h file and must appear in every protocol table created.

Those two states are ST_INIT and ST_IDLE.

- **ST_INIT** - When channels initialization is selected, the table enters Init state. This state contains functions that initialize the following global parameters:
 - RC_IDLE_CAS - Defines the ABCD bits expected to be received in the IDLE state in the specific protocol.
 - TX_IDLE_CAS - Defines the ABCD bits transmitted on IDLE state in the specific protocol.
 - DIAL_PLAN - A change regarding the issue of an incoming call dialed number is implemented in revision 3.21 as opposed to revision 3.2 and earlier. In revision 3.2 and earlier, Users had to pre-define the expected number of digit to receive an incoming call. If a lower number of digits than expected were received, the call setup would have failed.

Revisions 3.21 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 max value defined as DIAL_PLAN Parameter #1 and #2 for destination and ANI numbers respectively.
- A pre-defined time-out value defined as DIAL_PLAN Parameter #3 elapses.

Note: This method is not used when working with MFC/R2 protocols. MFC/R2 uses expected number of digits defined in ProtUser_defines_xxx.h.

- DTMF_DIAL - Defines the on-time and off-time for the DTMF digits generated towards the PSTN.
- COMMA_PAUSE_TIME - Defines the delay between each digit when a comma is used as part of the dialed number string. (See acPSTNPlaceCall for further explanation).
- DTMF_DETECTION - Defines the min/max On time for DTMF digit dialing detection.
- PULSE_DIAL_TIME - Not supported by current stack version. Defines the Break and Make time for pulse dialing.
- PULSE_DIAL - Not supported by current stack version. Defines the Break and Make ABCD bits for pulse dialing.
- DEBOUNCE - Defines the interval time of CAS to be considered as a hit.
- COLLECT_ANI - Enable or Disable reception of ANI in a specific protocol.
- DIGIT_TYPE - Defines the dialing method used (DTMF, MF). On MFC/R2 protocols this parameter is not applicable (digits are assumed to be R2 digits).
- **ST_IDLE** - When no active call is established or 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.

I.7 Reserved Words

Reserved words - like DO, NO_STATE, etc. see the detailed list in CASSetup.h.

I.8 State's Line Structure

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

- 1 action/event
- 2 function
- 3 parameter #1
- 4 parameter #2
- 5 parameter #3
- 6 next state

I.9 Action/Event

Action/event - The name of the table's events which are the possible triggers for the whole protocol state machine. Those can be selected from the list of events in the CASSetup.h file (e.g., EV_DISCONNECT_INCOMING).

At the beginning of the state, there can be up to 4 special unconditional action/events called FUNCTION. These events are functions that are unconditionally performed when the table reaches the state. These actions are labeled FUNCTION0 to FUNCTION3.

Following is the list of available protocols table actions (events to the state machine):

1 User Command Oriented:

- EV_PLACE_CALL - When using **acPSTNPlaceCall()**.
- EV_ANSWER - When using **acPSTNAnswerCall()**.
- EV_DISCONNECT_OUTGOING - When using the function **acPSTNDisconnectCall()** and the call is outgoing.
- EV_DISCONNECT_INCOMING - When using the function **acPSTNDisconnectCall()** and the call is incoming.
- EV_RELEASE_CALL - When using **acPSTNReleaseCall()**

2 CAS Change Oriented:

- 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).

3 Timer Oriented:

- EV_TIMER_EXPIRED1 - timer 1 that was previously set by table had expired.
- EV_TIMER_EXPIRED2 - timer 2 that was previously set by table had expired.
- EV_TIMER_EXPIRED3 - timer 3 that was previously set by table had expired.
- EV_TIMER_EXPIRED4 - timer 4 that was previously set by table had expired.
- EV_TIMER_EXPIRED5 - timer 5 that was previously set by table had expired.
- EV_TIMER_EXPIRED6 - timer 6 that was previously set by table had expired.
- EV_TIMER_EXPIRED7 - timer 7 that was previously set by table had expired.
- EV_TIMER_EXPIRED8 - timer 8 that was previously set by table had expired.

4 Counter Oriented:

- EV_COUNTER1_EXPIRED - counter 1 value reached 0.
- EV_COUNTER2_EXPIRED - counter 2 value reached 0.

5 IBS oriented:

- EV_RB_TONE_STARTED - ring back tone according to its definition in the call progress *ini* file (type and index) was detected.
- EV_RB_TONE_STOPPED - ring back tone according to its definition in the call progress *ini* file (type and index) was stopped after it was previously detected.

6 MF Oriented (MFCR2 protocol related):

- EV_MFRn_1 - MF digit 1 detected.
- EV_MFRn_2 - MF digit 2 detected.
- EV_MFRn_3 - MF digit 3 detected.
- EV_MFRn_4 - MF digit 4 detected.
- EV_MFRn_5 - MF digit 5 detected.
- EV_MFRn_6 - MF digit 6 detected.
- EV_MFRn_7 - MF digit 7 detected.
- EV_MFRn_8 - MF digit 8 detected.
- EV_MFRn_9 - MF digit 9 detected.
- EV_MFRn_10 - MF digit 10 detected.
- EV_MFRn_11 - MF digit 11 detected.
- EV_MFRn_12 - MF digit 12 detected.
- EV_MFRn_13 - MF digit 13 detected.
- EV_MFRn_14 - MF digit 14 detected.
- EV_MFRn_15 - MF digit 15 detected.
- EV_MFRn_1_STOPPED - MF digit 1 previously detected, now stopped.
- EV_MFRn_2_STOPPED - MF digit 2 previously detected, now stopped.
- EV_MFRn_3_STOPPED - MF digit 3 previously detected, now stopped.
- EV_MFRn_4_STOPPED - MF digit 4 previously detected, now stopped.
- EV_MFRn_5_STOPPED - MF digit 5 previously detected, now stopped.
- EV_MFRn_6_STOPPED - MF digit 6 previously detected, now stopped.
- EV_MFRn_7_STOPPED - MF digit 7 previously detected, now stopped.
- EV_MFRn_8_STOPPED - MF digit 8 previously detected, now stopped.
- EV_MFRn_9_STOPPED - MF digit 9 previously detected, now stopped.
- EV_MFRn_10_STOPPED - MF digit 10 previously detected, now stopped.
- EV_MFRn_11_STOPPED - MF digit 11 previously detected, now stopped.
- EV_MFRn_12_STOPPED - MF digit 12 previously detected, now stopped.
- EV_MFRn_13_STOPPED - MF digit 13 previously detected, now stopped.
- EV_MFRn_14_STOPPED - MF digit 14 previously detected, now stopped.
- EV_MFRn_15_STOPPED - MF digit 15 previously detected, now stopped.
- EV_END_OF_MF_DIGIT - When using **DialMF()** and no more dialed number digits are available. (They have already been sent. For example, the far side requests the next ANI digit, but all digits have been already sent). This event usually appears in MFR2 tables.
- EV_NO_ANI - When using **DialMF()** and no ANI was specified by the outgoing User in the **acPSTNPlaceCall()** function. (MFCR2 protocols specifications should define what to do when no ANI digits are available. Usually I-12 is sent).

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

- EV_ACCEPT - When using **acCASAcceptCall** (used only in MFC/R2) with CALLED_IDLE as its reason parameter.
- EV_REJECT_BUSY - When using **acCASAcceptCall** with CALLED_BUSY as its reason parameter.
- EV_REJECT_CONGESTION - When using **acCASAcceptCall** with CALLED_CONGESTION as its reason parameter.
- EV_REJECT_UNALLOCATED - When using **acCASAcceptCall** with CALLED_UNALLOCATED as its reason parameter.
- EV_REJECT_RESERVE1 - When using **acCASAcceptCall** with CALLED_RESERVE1 as its reason parameter.
- EV_REJECT_RESERVE2 - When using **acCASAcceptCall** with CALLED_RESERVE2 as its reason parameter.

7 Miscellaneous:

- EV_DIALED_NUM_DETECTED - (Incoming call) dialed destination number has been collected after START_COLLECT was previously activated and the condition for incoming_call_detected event is satisfied (see ST_INIT for conditions details).
- EV_DIAL_ENDED - Dialing initiated by table SEND_DEST_NUM has been completed (last digit has been sent).
- EV_ANI_NUM_DETECTED - This action is used to inform the script file of a successful reception of the ANI digits string, or when timeout of digit waiting occurs. This is reported at the incoming call detected event, when ANI flag is YES.
- EV_FIRST_DIGIT - Reception of first digit out of the incoming digit string. Used in the FXO protocols, where informing the script of receiving of the first digit, enables the script to use SEND_PROG_TON function to stop the dial tone.

I.10 Function

Function - the function rubric 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 rubric, no function is executed.

I.11 Parameters

Table I-2: CAS Parameters

Parameter #1	These columns are used as the function's parameters. NONE should be placed when the parameter is not essential. The list of global parameters can be found in CasSetup.h.
Parameter #2	
Parameter #3	

List of available User-functions and their parameters:

- **SET_TIMER (timer number, timeout)** - Used to set timers that are managed per B-channel, and their expiration triggers the state machine table. Each protocol table/state machine can use up to 8 timers per B-channel/call, (timeout in ms).
- **SEND_CAS (AB value, CD value)** - ABCD bits sent as line signaling for the specific channel the call is setup.

- **SEND_EVENT (event type, cause)** - The specific event type is sent to the host/user and retrieved by the User applying **acGetEvent()**.
- **SEND_DEST_NUM()** - Enbloc dialing when the number is input at the call's PlaceCall function. Three types are possible: (1) ADDRESS (2) InterExchangePrefix (3) ANI
- **DEL_TIMER (timer number)** - Delete specific or all timers (0 for all) for the B-channel.
- **START_COLLECT** - Initiating the collection of address information i.e., the dialed (destination) number for incoming calls where appropriate by the protocol. At the time between START_COLLECT and STOP_COLLECT no digit are reported to the User (EV_DIGIT is blocked), the destination number is reported in the EV_INCOMING_CALL_DETECTED event.
- **STOP_COLLECT** - See START_COLLECT.
- **SET_COUNTER (counter number, counter value or NONE)** - Used to set counters that are managed per B-channel and their expiration triggers the state machine. The counter initialization value should be non-negative number. To delete all timers, perform this function with 0 in the counter number field.
- **DEC_COUNTER (counter number)** - Decrease counter value by 1. When counter value reaches 0 EV_COUNTERx_EXPIRES is sent to the table (The x represent the counter number).
- **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 regarding sending digits.

- 1 **AddressVector and ANIDigitVector** - These parameters are initialized when using a **PlaceCall** function. When the code reaches the dialing section it sends MF digit according to the MF type specified in the MF type rubric (the types are defined in CASSetup.h file):
 - ADDRESS - Send digit from the address vector (destination number) according to the index requested. (See Index explanation).
 - ANI - Send digit from the ANI vector (source number) according to the index requested.
 - SPECIFIC - Send the MF digit specified in the Parameter #2 rubric.
 - SOURCE_CATEGORY - Send pre-defined source category MF digit. The source category digit is set as the SourceNumberingType parameter when using a **PlaceCall** function. The second and third parameters have no use when using this type.
 - TRANSFER_CAPABILITY - Send pre-defined line category MF digit. The line category digit is set as the TransferCapability parameter when using a **PlaceCall** function. The second and third parameters have no use when using this type.
- 2 **Index** - The Index parameter specifies the offset of the next digit to be sent from vector (ADDRESS or ANI types described above):
 - Use index 1 for sending next digit in the vector.
 - Use index -n for sending last but n digit. Underflow can occur if n is greater than the number of digits sent so far.
 - Use index 0 to send last sent digit.
 - Use index **SEND_FIRST_DIGIT** (see CASSetup.h) to start sending the digits vector from the beginning.

- 3 MF Send Time** - The send time parameter specifies the maximum transmission time of the MF.
- STOP_SEND_MF - Stop sending the current MF.
 - SEND_PROG_TON - (Operation, Tone or NONE)
- Two operations are available.
- a Send the call progress tone specified in the Parameter #2 rubric (The second parameter can be taken from CASsetup.h).
 - b Stop sending the last parameter.
- CHANGE_COLLECT_TYPE (Collect Type) - used only in MFCR2 protocol by the incoming User to indicate his waiting for the reception of MF digit of the requested type. The type can be one of the following:
 - ADDRESS - User is waiting for the reception of address digits.
 - ANI - User is waiting for the reception of ANI digits.
 - SOURCE_CATEGORY - User is waiting for the reception of the source category.
 - TRANSFER_CAPABILITY - User is waiting for the reception of the source transfer capability (line category).

I.12 Next State

Next state_ - this column contains the next state the table moves to, after executing the function for that action/event line. When the User selects 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 the User selects to stay in the same state they are currently in, the unconditional actions (FUNCTION0) at the beginning of the state are performed. In contrast, NO_STATE skips these function and just waits for another action to come.

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

Appendix J PSTN Trace Utilities

J.1 About PSTN Trace Utilities

LOCATION:

.\\Utilities\\PSTN Trace Utility

DESCRIPTION:

These utilities are designed to convert PSTN trace binary files to text format. The binary PSTN trace files are generated when the user sets the PSTN interface to trace mode.

OPERATION:**Generating a Trace/audit Text File for Cas Protocols**

- **To generate a readable text file out of the binary trace file when using CAS protocols, take these 3 steps:**

- Step 1.** Rename the PSTN trace binary file to **CASTrace0.dat**.
- Step 2.** Copy it to the same directory in which the translation utility, **CAS_Trace.exe**, is located.
- Step 3.** Run **CAS_Trace.exe** (no arguments are required). The text file, **CASTrace0.txt**, is created.

Generating a Trace/audit Text File for PRI Protocols

- **To generate a readable text file out of the binary trace file when using ISDN protocols, take these 2 steps:**

- Step 1.** Copy the PSTN trace binary file to the same directory in which the translation utility **CONVERT_TRACE.BAT** is located. The following files should reside in the same directory: **Dumpview.exe**, **Dumpview.cfg** and **ReadMe.txt**.
- Read carefully the **ReadMe.txt** in order to understand the usage of the translation utility.
- Step 2.** Run the **CONVERT_TRACE.BAT**. The text file is created.

J.2 Enabling PSTN Trace

This section contains direction on starting and collecting the PSTN trace via the Web. (Refer to Figure J-1 below for a view of the Trunk Traces). Also, note if the PSTN trace is of a PRI or CAS collection based on the framer involved in the trace. This information is needed to properly parse the captured data.

- **To start and collect the PSTN trace via the Web, take these 11 steps:**

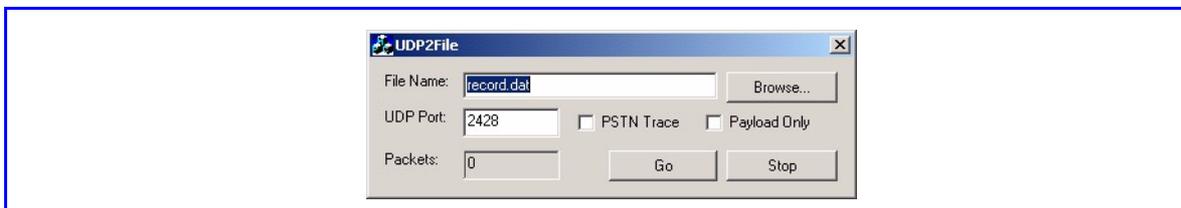
- Step 1.** Run the UDP2File utility.
- Step 2.** Determine the trace file name.
- Step 3.** Determine the UDP port.
- Step 4.** Mark the PSTN Trace check box.
- Step 5.** Click the Run button. The UDP2File utility starts to collect the trace messages.

- Step 6.** Activate the Web page by entering <M2K IP address>/PSTNTraces (e.g., http://10.8.8.101/PSTNTraces). The user and password is the same for the unit.
- Step 7.** In the Web page, set the trace level of each trunk.
- Step 8.** Enable the trace via the Web.
- Step 9.** Determine the UDP port (the same as in step 3).
- Step 10.** Click the SUBMIT button. The board starts to send the trace messages.
- Step 11.** In the UDP2File utility (Refer to Figure J-2 below) you should see the number in the packets counter increasing.

Figure J-1: Trunk Traces



Figure J-2: UDP2File Utility



Appendix K MEGACO Compliance

Table K-1, “MEGACO Compliance Matrix” on page 183 to **page 195**, below summarize the supported MEGACO features respectively. The Reference column in Table K-1 refers to IETF RFC 3015 from September 2002.

K.1 MEGACO Compliance Matrix

Table K-1: MEGACO Compliance Table (continues until page 195)

Section (in RFC3015)	Item	Support	Comments
7	Commnds supported:		
	Add	Yes	
	Modify	Yes	
	Subtract	Yes	
	Move	Yes	
	AuditValue	Yes	
	AuditCapabilities	Yes	
	Notify	Yes	
	ServiceChange	Yes	
7.1	Descriptors		
7.1.1	Specifying Parameters:		
	Fully specified	Yes	
	Under specified	Yes	
	Over specified	Yes	
	Handling unspecified mandatory parameters.	Yes	
	Wildcarded termination ID	Yes	
7.1.2	Modem Descriptor:		
	V.18	No	
	V.22	No	
	V.22bis	No	
	V.32	No	
	V.32bis	No	
	V.34	No	
	V.90	No	

Table K-1: MEGACO Compliance Table (continues until page 195)

Section (in RFC3015)	Item	Support	Comments
	V.91	No	
	Synchronous ISDN	No	
7.1.3	Multiplex Descriptor:		
	H.221	No	
	H.223	No	
	H.226	No	
	V.76	No	
7.1.4	Media Descriptor:		
	Termination State Descriptor	Yes	
	Stream Descriptor	Yes	
	Local Control Descriptor	Yes	
	Local Descriptor	Yes	
	Remote Descriptor	Yes	
7.1.5	Termination State Descriptor:		
	Service State:		
	Test	Yes	
	Out of service	Yes	
	In service	Yes	
	EventBufferControl:	Yes	
7.1.6	Stream Descriptor:		
		Yes	
7.1.7	Local Control Descriptor:		
	Mode:		
	Send-only	Yes	
	Receive-only	Yes	
	Send/receive	Yes	
	Inactive	Yes	

Table K-1: MEGACO Compliance Table (continues until page 195)

Section (in RFC3015)	Item	Support	Comments
	Loop-back	Yes	
	ReserveGroup	No	
	ReserveValue	No	
7.1.8	Local & Remote Descriptors:		
	Unspecified Local Descriptor	Yes	
	Unspecified Remote Descriptor	Yes	
	Empty Local Descriptor	Yes	
	Empty Remote Descriptor	Yes	
	Multiple groups	No	
7.1.9	Event Descriptor		
	EventBufferControl		
	Lockstep	Yes	
	off	Yes	
7.1.10	Event Buffer Descriptor		
		Yes	
7.1.11	Signal Descriptor		
	Signal Types		
	On/off	Yes	
	Timeout	Yes	
	Brief	Yes	
	Sequential signal list	Yes	
	Simultaneous signals	No	
	Keep active	Yes	
7.1.12	Audit Descriptor		
	Modem	No	
	Mux	No	
	Events	Yes	
	Media	Yes	
	Signals	Yes	

Table K-1: MEGACO Compliance Table (continues until page 195)

Section (in RFC3015)	Item	Support	Comments
	Observed events	Yes	
	DigitMap	Yes	
	Statistics	Yes	
	Packages	Yes	
	EventBuffer	Yes	
	Empty descriptor	Yes	
7.1.13	Service change Descriptor		
	ServiceChangeMethod	Yes	
	ServiceChangeReason	Yes	
	ServiceChangeAddress	Yes	
	ServiceChangeDelay	Yes	
	ServiceChangeProfile	Yes	
	ServiceChangeVersion	Yes	
	ServiceChangeMGCIId	Yes	
	TimeStamp	Yes	
7.1.14	Digit Map Descriptor		
	Digit Map Names	Yes	
	StartTimer (T)	Yes	
	ShortTimer (S)	Yes	
	LongTimer (L)	Yes	
	DurationModifier (z)	Yes	
	Any digit 0-9 (x)	Yes	
	Zero or more repetitions (.)	Yes	
7.1.15	Statistics Descriptor		
	Octets sent	Yes	
	Octets received	Yes	
	Empty AuditDescriptor in "Sub"	Yes	

Table K-1: MEGACO Compliance Table (continues until page 195)

Section (in RFC3015)	Item	Support	Comments
7.1.16	Package Descriptor		
		Yes	
7.1.17	ObservedEvents Descriptor		
	Request Identifier	Yes	
	Event	Yes	
	Detection Time	Yes	
7.1.18	Topology Descriptor		
	Isolate	Yes	
	Oneway	Yes	
	Bothway	Yes	
	CHOOSE wildcard	Yes	
	ALL wildcard	Yes	
7.2	Command API		
7.2.1	Add		
	Termination ID	Yes	
	MediaDescriptor	Yes	
	ModemDescriptor	No	
	MuxDescriptor	No	
	EventsDescriptor	Yes	
	SignalsDescriptor	Yes	Only one signal per channel Up to 30 signals in a signal list
	DigitMapDescriptor	Yes	
	AuditDescriptor	Yes	Up to 30 signals
			Only one signal per channel.
7.2.2	Modify		
	Termination ID	Yes	
	MediaDescriptor	Yes	
	ModemDescriptor	No	
	MuxDescriptor	No	

Table K-1: MEGACO Compliance Table (continues until page 195)

Section (in RFC3015)	Item	Support	Comments
	EventsDescriptor	Yes	
	SignalsDescriptor	Yes	Only one signal per channel Up to 30 signals in a signal list
	DigitMapDescriptor	Yes	
	AuditDescriptor	Yes	
7.2.3	Subtract		
	Termination ID	Yes	
	AuditDescriptor	Yes	
	Statistical Parameters return	Yes	
7.2.4	Move		
	Termination ID	Yes	
	MediaDescriptor	Yes	
	ModemDescriptor	No	
	MuxDescriptor	No	
	EventsDescriptor	Yes	
	SignalsDescriptor	Yes	Only one signal per channel Up to 30 signals in a signal list
	DigitMapDescriptor	Yes	
	AuditDescriptor	Yes	
7.2.5	AuditValue		
	TerminationID	Yes	
	Wildcard	Yes	
	AuditDescriptor	Yes	
	Media	Yes	
	Modem	No	
	Mux	No	
	Event	Yes	
	Signal	Yes	
	DigitMap	Yes	
	ObservedEvents	Yes	

Table K-1: MEGACO Compliance Table (continues until page 195)

Section (in RFC3015)	Item	Support	Comments
	EventBuffer	Yes	
	Statistics	Yes	
	Packages	Yes	Topology used only for conference
7.2.6	AuditCapabilities		
	TerminationID	Yes	
	Wildcard	Yes	
	AuditDescriptor	Yes	
	Media	Yes	
	Modem	No	
	Mux	No	
	Event	Yes	
	Signal	Yes	
	DigitMap	Yes	
	ObservedEvents	Yes	
	EventBuffer	Yes	
	Statistics	Yes	
	Packages	Yes	
7.2.7	Notify		
		Yes	
7.2.8	Service Change		
	Termination ID	Yes	
	Wildcard	Yes	
	“Root” Termination	Yes	
	ServiceChangeMethod		
	Graceful	No	
	Forced	Yes	

Table K-1: MEGACO Compliance Table (continues until page 195)

Section (in RFC3015)	Item	Support	Comments
	Resart	Yes	
	Disconnected	Yes	
	Handoff	Yes	
	Failover	Yes	
	Extension	No	
	ServiceChangeReason		
	900 Service Restored	Yes	
	901 Cold Boot	Yes	
	902 Warm Boot	No	
	903 MGC Direct Change	Yes	
	904 Termination Malfunctioning	No	
	905 Term Taken out of Service	No	
	906 Loss of lower layer connectivity	Yes	
	907 Transmission Failure	Yes	
	908 MG Impending Failure	No	
	909 MGC Impending Failure	No	
	910 Media Capability Failure	No	
	911 Modem Capability Failure	No	
	912 Mux Capability Failure	No	
	913 Signal Capability Failure	No	
	914 Event Capability Failure	No	
	915 State Loss	No	
	ServiceChangeDelay	No	
	ServiceChangeAddress	Yes	
	ServiceChangeProfile	Yes	
	ServiceChangeVersion	Yes	
	ServiceChangeMgclid	Yes	
	TimeStamp	Yes	
7.2.9	Manipulating and Auditing Context Attributes		
		Yes	

Table K-1: MEGACO Compliance Table (continues until page 195)

Section (in RFC3015)	Item	Support	Comments
7.2.10	Generic Command Syntax		
	Text Encoding	Yes	
	Binary Encoding	Yes	
7.3	Command Error		
	400 – Bad Request	Yes	
	401 – Protocol Error	Yes	
	402 – Unauthorized	No	
	403 – Syntax Error in Transaction	Yes	
	404 - Syntax Error in TransactionReply	Yes	
	405 - Syntax Error in TransactionPending	Yes	
	406 – Version not Supported	No	
	410 – Incorrect Identifier	Yes	
	411 – Unknown ContextId	Yes	
	412 – No ContextId Available	Yes	
	421 - Unknown Action	Yes	
	422 - Syntax Error In Action	Yes	
	430 - Unknown TerminationId	Yes	
	431 – No TerminationId Matched a Wildcard	Yes	
	432 – Out of Termination Id / No TerminationId Available	Yes	
	433 – TerminationId is already in a context	Yes	
	440 - Unsupported or unknown Package	Yes	
	441 – Missing RemoteDescriptor	Yes	
	442 - Syntax Error in Command	Yes	
	443 - Unsupported or unknown Command	Yes	
	444 - Unsupported or unknown Descriptor	Yes	
	445 - Unsupported or unknown Property	Yes	
	446 - Unsupported or unknown Parameter	Yes	
	447 – Descriptor not legal in this command	Yes	

Table K-1: MEGACO Compliance Table (continues until page 195)

Section (in RFC3015)	Item	Support	Comments
	448 - Descriptor appears twice in a command	Yes	
	450 - No such property in this package	Yes	
	451 - No such event in this package	Yes	
	452 - No such signal in this package	Yes	
	453 - No such statistic in this package	Yes	
	454 - No such parameter value in this package	Yes	
	455 - Parameter illegal in this Descriptor	Yes	
	456 - Parameter or Property appears twice in this Descriptor	Yes	
	471 - Implied Add for Multiplex failure	Yes	
	500 - Internal Gateway Error	Yes	
	501 - Not Implemented	Yes	
	502 - Not ready.	Yes	
	503 - Service Unavailable	No	
	504 - Command Received from unauthorized entity	No	
	505 - Command Received before Restart Response	Yes	
	510 - Insufficient resources	Yes	
	512 - Media Gateway unequipped to detect requested Event	Yes	
	513 - Media Gateway unequipped to generate requested Signals	Yes	
	514 – MG cannot send the specified announcement.	Yes	
	515 – Unsupportes Media Type.	Yes	
	517 – Unsupportes or Invalid Mode.	Yes	
	518 – Event Buffer Full	Yes	
	519 – Out Of Space To Store Digit Map.	Yes	
	520 – Media Gateway does not have a digit map.	Yes	
	521 – Termination is “Service Changing”.	No	
	526 – Insufficient Bandwidth.	No	
	529 – Internal Hardware Failure.	No	
	530 – Temporary Hardware Failure.	No	
	531 – Permanent Network Failure.	No	

Table K-1: MEGACO Compliance Table (continues until page 195)

Section (in RFC3015)	Item	Support	Comments
	540 – Unexpected Initial hook state	No	
	581 – Does not Exist.	Yes	
8.	Transactions		
8.1	Common Parameters		
8.1.1	Transaction Identifiers		
	TransactionId	Yes	
	Use of TransactionId '0'	Yes	
8.1.2	Context Identifiers		
	ContextID	Yes	
	CHOOSE Wildcard	Yes	
	All Wildcard	Yes	
8.2	Transaction API		
8.2.1	Transaction Request		
	Multiple actions per request	Yes	
8.2.2	Transaction Reply		
	Multiple actions per reply	Yes	
8.2.3	Transaction Pending		
	Transaction Pending Support	No	
	normalMGCExecutionTime	Yes	
	normalMGCExecutionTime	Yes	
8.3	Messages		
	Receive Messages	Yes	
	Send Messages	Yes	

Table K-1: MEGACO Compliance Table (continues until page 195)

Section (in RFC3015)	Item	Support	Comments
9	Transport		
	Transport over UDP	Yes	
	Transport over TCP	No	
9.1	Ordering of commands		
		Yes	
9.2	Protection against the restart Avalanche		
	Use of default MWD per platform	No	
	Random restart delay.	No	
	Random seed selection	No	
	Detection of local activity	No	
10	Security Considerations		
		No	
11	MG-MGC Control Interface		
11.1	Multiple Virtual Gateways	No	
11.2	Cold Start		
	Primary Call Agent support	Yes	

Table K-1: MEGACO Compliance Table (continues until page 195)

Section (in RFC3015)	Item	Support	Comments
	Secondary Call Agents support	Yes	
	Cyclic check for Call Agent	Yes	
11.3	Negotiation of Protocol Version		
		No	
11.4	Failure of an MG		
		No	
11.5	Failure of an MGC		
		Yes	
		No	
11.5	Failure of an MGC		
		Yes	

Reader's Notes

Appendix L List of Abbreviations

These are most of the abbreviations and an explanation of some of the terms.

Table L-1: List of Abbreviations (continues on page 197 to page 199)

Abbreviation	Meaning
AAL2	ATM Adaptation Layer 2.
ADPCM	Adaptive Differential PCM - voice compression.
AIS	Alarm Indication Signal.
ASN.1	Abstract Syntax Notation.
ATM	Asynchronous Transfer Mode.
A-law	European Compander Functionality Rule, (see μ -law).
bps	Bits per second.
BLES	Broadband Loop Emulation Service by the DSL Forum.
BRI	Basic Rate Interface in ISDN.
CAS	Channel Associated Signaling.
cPCI	Compact PCI.
DHCP	Dynamic Host Control Protocol
DID	Direct Inward Dial.
DSL	Digital Subscriber Line.
DSP	Digital Signal Processor (or Processing).
DTMF	Dual Tone Multiple Frequency (Touch Tone).
E1	2.048 Mbps European Digital Transmission System, (see T1).
E-ADPCM	Enhanced ADPCM.
ETSI	European Telecommunications Standards Institute.
FR	Frame Relay.
GK	Gatekeeper.
GW	Gateway.
G.xxx	An ITU Standard - see References section for details.
H.323 Entity	Any H.323 Component.
IE	Information Element (ISDN layer 3 protocol, basic building block).
IETF	Internet Engineering Task Force.
IPM-260	AudioCodes IPmedia cPCI VoIP Media Processing Board, to 120 ports.
IPM-1610	AudioCodes IPmedia cPCI VoIP Media Processing Board, to 240 ports.
ISDN	Integrated Services Digital Network.

Table L-1: List of Abbreviations (continues on page 197 to page 199)

Abbreviation	Meaning
ISO	International Standards Organization
ITU	International Telecommunications Union.
ITU-T	Telecommunications section of the ITU.
IVR	Interactive Voice Response.
Jitter	Variation of interpacket timing interval.
kbps	thousand bits per second.
LAPD	Line Access Protocol for the D-channel.
LFA	Loss of Frame Alignment.
LOF	Loss of Frame.
Mbps	Million bits per second.
MCU	Multipoint Control Unit (H.323).
Mediant	
MEGACO	Media Gateway Control (Protocol, H.248)
MGC	Media Gateway Controller
MGCP	Media Gateway Control Protocol
MIB	Management Information Base
MP-102	AudioCodes 2-port MediaPack Media Gateway
MP-104	AudioCodes 4-port MediaPack Media Gateway
MP-108	AudioCodes 8-port MediaPack Media Gateway
MP-124	AudioCodes 24-port MediaPack Media Gateway
ms or msec	millisecond; a thousandth part of a second.
MVIP	Multi Vendor Integration Protocol.
NetCoder	AudioCodes Proprietary High Quality, Speech Coder.
NIC	Network Interface Card.
OSI	Open Systems Interconnection.(Industry Standard).
PCI	Personal Computer Interface (Industry Standard).
PCM	Pulse Code Modulation.
PDU	Protocol Data Unit.
POTS	Plain Old Telephone System or Service.
PRI	Primary Rate Interface in ISDN.
PSTN	Public Switched Telephone Network.
QoS	Quality of Service.
RAI	Remote Alarm Indication.

Table L-1: List of Abbreviations (continues on page 197 to page 199)

Abbreviation	Meaning
RAS	Registration, Admission, and Status (control within H.323).
RDK	Reference Design Kit.
RFC	Request for Comment issued by IETF.
RTCP	Real Time Control Protocol.
RTP	Real Time Protocol.
ScBus	Signal Computing Bus – part of SCSA.
SCSA	Signal Computing System Architecture.
SNMP	Simple Network Management Protocol.
Stretto	
TCP	Transmission Control Protocol.
TCP/IP	Transmission Control Protocol/Internet Protocol.
TFTP	Trivial File Transfer Protocol.
TPNCP	AudioCodes TrunkPack Network Control Protocol.
TP-240	AudioCodes TrunkPack VoIP/240 Voice over IP PCI media streaming board, up to 128 ports
TP-1610	AudioCodes TrunkPack VoIP/1610 cPCI media streaming board, to 480 ports
TP-610	AudioCodes TrunkPack VoIP/610 cPCI media streaming board, to 192 ports,
TPM-1100	AudioCodes TrunkPack Module
TrunkPack	AudioCodes series of voice compression boards.
T1	1.544 Mbps USA Digital Transmission System. (see E1).
UDP	User Datagram Protocol.
VCC	Virtual Channel Connection.
VoAAL2	Voice over AAL2 (see above).
VoATM	Voice over Asynchronous Transfer Mode.
VoDSL	Voice over Digital Subscriber Line.
VoFR	Voice over Frame Relay.
VoIP	Voice over Internet Protocol.
VoP	Voice over Packet(s)
VoPN	Voice over Packet Networks.
VPN	Virtual Private Network.
μ -law	American Compander Functionality Rule, (see A-law).
μ s or μ sec	microsecond; a millionth part of a second.

US: AudioCodes Inc, 2890 Zanker Road, Suite # 200, San Jose, CA 95134.

Tel: 408-577-0488 - Fax: 408-577-0492

International: AudioCodes Ltd, 4 HaHoresh Street, PO Box 14, Yehud 56470, Israel.

Tel: +972-3-539 4000 - Fax: +972-3-539 4040

China: AudioCodes Ltd, Room 2811-12, JingGuang Centre

HuJiaLou, Chaoyang District, Beijing 100020, P.R.China

Tel: 86-10-65974134 - Fax: 86-10-65974260

Japan: S Building B1F, 2-30-2, Chuou, Nakano-ku, Tokyo 164-0011, Japan

Tel: Fax: +81-3-5348-1405 - Fax: +81-3-5348-1405

