AudioCodes Systems for Wireline Networks

Mediant™ 2000 VoP Media Gateway

User's Manual

Version 4.4 Beta

Document Number: LTRT-69801

AudioCodes

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	3
Contents, Figures and Tables	5
1. Overview of the Mediant 2000	15
2. Hardware Equipment	21
3. Hardware Installation	33
4. AudioCodes' Software Package	45
5. Standard Control Protocols	59
6. Device Management	97
7. Configuration Parameters and Files	145
8. Diagnostics & Troubleshooting	199
9. Functional Specifications	205
Appendix A. AudioCodes BootP/TFTP Server	211
Appendix B. RTP/RTCP Payload Types	219
Appendix C. DTMF, Fax and Modem Transport Modes	223
Appendix D. CAS Protocol Table	227
Appendix E. Utilities	237
Appendix F. MGCP Compliance	251
Appendix G. MEGACO Compliance	265
Appendix H. Customizing the AudioCodes Web Interface	277
Appendix I. List of Abbreviations	283
Appendix J. Regulatory Information	287

The following AudioCodes manuals should be used for additional information:

"TP Boards Release Notes", Document #: LTRT-81901, for Release 4.4Beta

"VoPLib User's Manual", Document #: LTRT-84401, for Release 4.4 Beta

"VoPLib API Reference Manual", Document #: LTRT-84001, for Release 4.4 Beta

For Regulatory Information and Declaration of Conformity
Refer to Appendix J on page 287

Safety Notice

Installation and service of AudioCodes' Media Gateways must be performed by authorized, qualified service personnel only.

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 logo, Ardito, AudioCoded, AudioCodes, AudioCodes logo, IPmedia, Mediant, MediaPack, MP-MLQ, NetCoder, Stretto, TrunkPack, VoicePacketizer and VoIPerfect, 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 devices.

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. Updates to this document and other documents can be viewed by registered Technical Support customers at www.audiocodes.com under Support / Product Documentation.

Date Published: June-03-2004 Date Printed: June-20-2004

Table of Contents

1 Overview of the Mediant 2000 15

1.1 General Features 16

1.2 TP-1610 Software Overview 17

1.3 Mediant 2000 Applications 17

1.3.1 Available Configurations 17

1.4 Benefits 18

1.5 Functional Block Diagram 19

2 Hardware Equipment 21

2.1 The Mediant 2000 Chassis 21

2.1.1 Mediant 2000 Diagram 21

2.1.2 Chassis LED Indicators 22

2.2 The TP-1610 Board 22

2.2.1 Board Hot-Swap Support 23

2.2.2 TP-1610 Board Front Panel LED Indicators 30

2.3 Optional CPU Board 31

3 Hardware Installation 33

3.1 Unpacking 33

3.1.1 Package Contents 34

3.2 Mounting the Mediant 2000 34

3.2.1 Mounting a Mediant 2000 in a 19-inch Rack 35

3.2.2 Installing the Mediant 2000 on a Desktop 36

3.3 Cabling the Mediant 2000 36

3.3.1 Power Supply Cabling 37

3.3.2 Connecting the E1/T1 Trunk Interfaces 38

3.4 Board Replacement 41

3.4.1 Preliminaries 41

3.4.2 Performing Graceful Lock 42

3.4.3 Removing Boards 42

3.4.3.1 Inserting Boards 43

3.4.3.2 Configuring and Unlocking the Mediant 2000 43

4 AudioCodes' Software Package 45

4.1 Unzipping the Software Package 45

4.1.1 Unzipping When Using a Windows™ Operating System 45

4.1.2 Unzipping When Using a Linux™/Solaris™ Operating System 45

4.1.3 Software Directory Structure 46

4.2 Call Control Protocols 47

4.2.1 Management Protocols 48

4.2.2 Simple Network Time Protocol Support 48

4.2.3 Boot Software and Firmware 49

4.2.4 Provisioning and Configuring 49

4.2.4.1 Configuration Parameters 49

4.2.4.2 Configuration Files 50

4.2.5 Device Startup 50

4.2.6 BootP/DHCP Server 52

4.2.6.1 BootP/DHCP Server Parameters 53

4.2.6.2 Host Name Support 55

4.2.6.3 Selective BootP 55

4.2.6.4 Vendor Specific Information 55

4.2.7 Microsoft™ DHCP/BootP Server 56

4.3 Upgrading Device Software 56
4.4 License Key 57

5 Standard Control Protocols 59

5.1 General 59
5.2 MGCP Control Protocol 59
5.2.1 MGCP Overview 59
5.2.2 MGCP Operation 59
5.2.2.1 Executing MGCP Commands 59
5.2.2.2 Setting MGCP Call Agent IP Address 60
5.2.3 Using DNS with MGCP 60
5.2.4 MGCP KeepAlive Mechanism 60
5.2.5 SDP Support in MGCP 61
5.2.6 Fax Transport Type Setting with Local Connection Options 61
5.2.7 TGCP Compatibility 62
5.2.7.1 Mapping of Payload Numbers to Coders 62
5.2.8 Supported MGCP Packages 64
5.2.8.1 Generic Media Package - G 64
5.2.8.2 DTMF Package - D 65
5.2.8.3 Line Package - L 65
5.2.8.4 Handset Emulation Package - H 66
5.2.8.5 Trunk Package - T 67
5.2.8.6 PacketCable (NCS) Line Package - L 67
5.2.8.7 Announcement Package - A 68
5.2.8.8 RTP Package - R 68
5.2.8.9 CAS Package - MS 70
5.2.8.10 CAS Package - DT 70
5.2.8.11 ISUP Trunk Package - IT 71
5.2.8.12 Media Format Parameter Package - FM 71
5.2.8.13 Fax Package Definition - FXR 72
5.2.8.14 V5 Package Definition X-v5 73
5.2.9 MGCP Endpoint Map 73
5.2.10 Compression Codecs 73
5.3 MEGACO (Media Gateway Control) Protocol 73
5.3.1 MEGACO Overview 73
5.3.2 Operation 74
5.3.2.1 Executing MEGACO Commands 74
5.3.2.2 KeepAlive Notifications From the Gateway 74
5.3.2.3 Setting MEGACO Call Agent IP Address and Port 75
5.3.2.4 Authorization Check of Call Manager IP Addresses 75
5.3.2.5 Support of DiffServ Capabilities 75
5.3.2.6 Handling Events 75
5.3.2.7 Playing Signals 76
5.3.2.8 Mediation 77
5.3.2.9 CAS/R2 Support in MEGACO 78
5.3.2.10 RFC 2833 Support 79
5.3.2.11 Silence Suppression Support 79
5.3.2.12 Fax T.38 and Voice Band Data Support (Bypass Mode) 80
5.3.2.13 Reporting Fax Events 81
5.3.3 SDP Support in MEGACO 81
5.3.3.1 Mapping Payload Numbers to Coders 82
5.3.4 Supported MEGACO Packages 84
5.3.4.1 Generic Media Package - G 84
5.3.4.2 Base Root Package - ROOT 85
5.3.4.3 Tone Generator Package - ToneGen 85
5.3.4.4 Tone Detection Package - ToneDet 85
5.3.4.5 DTMF Generator Package - DG (Extends ToneGen) 85
5.3.4.6 DTMF Detection Package - DD (Extends ToneDet) 86
5.3.4.7 Call Progress Tones Generator Package - CG (Extends ToneGen) .. 87
5.3.4.8 Call Progress Tones Detection Package - CD (Extends ToneDet)..... 87

5.3.4.9 Basic Continuity Package - CT ... 87
5.3.4.10 Network Package - NT ... 88
5.3.4.11 RTP Package - RTP (Extends - NT) ... 88
5.3.4.12 TDM Circuit Package - TDMC (Extends - NT) ... 88
5.3.4.13 Generic Announcement Package - AN ... 88
5.3.4.14 Expanded Call Progress Tones Generator Package - XCG (Extends - ToneGen)... 89
5.3.4.15 Basic Service Tones Generation Package - SRVTN (Extends - ToneGen)... 89
5.3.4.16 Expanded Services Tones Generation Package - XSRVTN (Extends - ToneGen)... 89
5.3.4.17 Basic CAS Package - CAS ... 90
5.3.4.18 R2 CAS Package - R2 ... 90
5.3.4.19 MF Generator Package - MFG (Extends - ToneGen) ... 90
5.3.4.20 MF Detection Package - MFD (Extends - ToneDet) ... 91
5.3.4.21 Inactivity Timer Package - IT... 91
5.3.4.22 Basic Call Progress Tones Generator with Directionality Package - BCG (Extends ToneGen) ... 92
5.3.4.23 Call Type Discrimination Package - CTYP... 92
5.3.4.24 IP Fax Package - IPFAX ... 92
5.3.5 MEGACO Profiling ... 93
5.3.6 MEGACO Terminations Naming ... 93
5.3.6.1 Termination Name Patterns ... 93
5.3.6.2 Old Termination Naming Method ... 94
5.3.6.3 Backward Compatibility... 94
5.3.6.4 Terminations Mapping to a PSTN Interface ... 94

6 Device Management ... 97

6.1 Developing an SNMP-Based Client Program to Manage Your Device/s ... 97
6.1.1 About SNMP ... 97
6.1.1.1 SNMP Message Standard... 97
6.1.1.2 SNMP MIB Objects ... 98
6.1.1.3 SNMP Extensibility Feature ... 99
6.1.2 Carrier Grade Alarm System... 99
6.1.2.1 Active Alarm Table ... 99
6.1.2.2 Alarm History... 100
6.1.3 Cold Start Trap... 100
6.1.4 Performance Measurements for a Third-Party Performance Monitoring System ... 100
6.1.6 SNMP Interface Details... 103
6.1.6.1 SNMP Community Names ... 104
6.1.6.2 Trusted Managers ... 105
6.1.6.3 SNMP Ports ... 106
6.1.6.4 Multiple SNMP Trap Destinations ... 107
6.1.6.5 SNMP Manager Backward Compatibility ... 109
6.2 Administrative State Control ... 109
6.2.1 Node Maintenance ... 109
6.2.2 Graceful Shutdown... 110
6.3 AudioCodes' Embedded Web Server ... 110
6.3.1 Embedded Web Server Protection & Security Mechanisms ... 111
6.3.1.1 Username and Password... 111
6.3.2 Web Configuration ... 111
6.3.2.1 Limiting the Embedded Web Server to Read-Only Mode ... 111
6.3.2.2 Disabling the Embedded Web Server... 111
6.3.3 Correlating PC /Mediant 2000 IP Address & Subnet Mask... 112
6.3.4 Accessing the Embedded Web Server ... 112
6.3.5 Using Internet Explorer to Access the Embedded Web Server ... 113
6.4 Getting Acquainted with the Web Interface ... 114
6.4.1 About the Web Interface Screen... 114
6.4.2 Saving Changes... 115
6.4.3 Protocol Management... 117

6.4.3.1 Protocol Definition 117
6.4.4 Advanced Configuration Screen 119
6.4.4.1 Network Settings 119
6.4.4.2 Channel Settings 120
6.4.4.3 Trunk Settings 122
6.4.4.4 TDM Bus Settings 125
6.4.4.5 Configuration File 126
6.4.4.6 Regional Settings 127
6.4.4.7 Change Password 129
6.4.5 Status and Diagnostic Menu 130
6.4.5.1 Trunk and Channel Status 130
6.4.5.2 Message Log 134
6.4.5.3 Versions 135
6.4.6 Software Update 136
6.4.6.1 Software Upgrade Wizard 136
6.4.6.2 Auxiliary Files Download 140
6.4.6.3 License 141
6.4.7 Save Configuration 142
6.4.8 Reset Button 143

7 Configuration Parameters and Files 145

7.1 Configuration Files for All Call Control Protocols 145
7.1.1 System Parameters 146
7.1.2 Infrastructure Parameters 147
7.1.3 Media Processing Parameters 152
7.1.4 PSTN Parameters 158
7.1.5 SS7 Parameters 165
7.1.6 Parameters Common to All Control Protocols 170
7.1.7 MGCP-Specific Parameters 173
7.1.8 MEGACO-Specific Parameters 175
7.1.9 Web Interface Parameters 178
7.1.10 SNMP Parameters 179
7.1.11 SCTP Parameters 181
7.2 Initialization File (ini files) 182
7.2.1 ini File Structure 182
7.2.1.1 Parameter = Value Rules in the ini File Structure 182
7.2.1.2 About Tables of Parameter Values 183
7.2.1.3 Tables of Parameter Value Rules in the ini File Structure 184
7.2.1.4 ini File Example 186
7.2.2 Secured Configuration File Downloading 188
7.2.2.1 Encoding Mechanism 188
7.3 Downloading and Formating Configuration Files 188
7.3.1 Downloading Configuration Files 188
7.3.2 Call Progress Tones and User-Defined Tones 189
7.3.2.1 Format of the Call Progress Tones Section 189
7.3.2.2 Format of the User-Defined Tones Section 191
7.3.2.3 Default Template for Call Progress Tones 192
7.4 Modifying the Call Progress Tones & Sending it to the Device 195
7.4.1 Converting a Modified CPT ini File to a dat File with AudioCodes' Download Conversion Utility 195
7.4.2 Playing Prerecorded Tones (PRT) 196
7.4.2.1 PRT File Configuration 196
7.4.2.2 Downloading the PRT dat File 197

8 Diagnostics & Troubleshooting 199

8.1 Syslog 199
8.1.1 Operating the Syslog Server 200
8.1.1.1 Sending the Syslog Messages 200
8.1.1.2 Setting the Syslog Server IP Address 200
8.1.1.3 Activating the Syslog Client 201
8.1.1.4 Setting Syslog Server IP Address, Enabling Syslog, in an ini File

(Example)............ 201
8.2 The Embedded Web Server's 'Message Log' (Integral Syslog)............ 201
8.3 Control Protocol Reports............ 201
8.3.1 TPNCP Error Report............ 201
8.3.2 MGCP/MEGACO Error Conditions............ 202
8.3.3 SNMP Traps............ 202
8.4 TP-1610 Self-test............ 202
8.5 Solutions to Possible Problems............ 203
8.5.1 Possible Common Problems............ 203
8.5.2 Possible Voice Problems............ 204
9 Functional Specifications............ 205
9.1 Mediant 2000 Selected Specifications............ 205
Appendix A AudioCodes BootP/TFTP Server............ 211
A.1 Introduction............ 211
A.1.1 Key Features............ 211
A.1.2 Specifications............ 211
A.1.3 BootP/TFTP Server Installation............ 212
A.1.4 Logging Screen............ 212
A.1.5 Preferences Screen............ 213
A.1.6 Client Configuration Screen............ 213
A.1.7 Template Screen............ 214
A.2 Screen Details............ 214
A.2.1 Main Screen............ 214
A.2.2 Preferences Screen............ 216
A.2.3 Client Configuration Screen............ 217
A.2.4 Templates Screen............ 218
Appendix B RTP/RTCP Payload Types............ 219
B.1 Payload Types Defined in RFC 3551............ 219
B.2 Payload Types Not Defined in RFC 3551............ 220
B.3 Default Dynamic Payload Types Which are Not Voice Coders............ 220
B.4 Mediant 2000 Default RTP/RTCP/T.38 Port Allocation............ 221
Appendix C DTMF, Fax and Modem Transport Modes............ 223
C.1 DTMF/MF Relay Settings............ 223
C.2 Fax/Modem Settings............ 223
C.3 Configuring Fax Relay Mode............ 223
C.4 Configuring Fax/Modem ByPass Mode............ 224
C.5 Configuring Modem Bypass NSE mode............ 224
C.6 Supporting V.34 Faxes............ 224
C.6.1 Using Bypass Mechanism for V.34 Fax Transmission:............ 225
C.6.2 Using Events Only Mechanism for V.34 Fax Transmission............ 225
C.6.3 Using Relay Mode for Various Fax Machines (T.30 and V.34)............ 226
Appendix D CAS Protocol Table............ 227
D.1 Constructing a CAS Protocol Table............ 227
D.2 Table Elements............ 227
D.2.1 INIT variables............ 227
D.2.2 Actions............ 227
D.2.3 Functions............ 227
D.2.4 States............ 228
D.3 Reserved Words............ 229
D.4 State’s Line Structure............ 230
D.5 Action/Event............ 230
D.6 Function............ 233
D.7 Parameters............ 233
D.8 Next State............ 235
D.9 Changing the Script File............ 235
D.9.1 General............ 235

D.9.2 MFC-R2 Protocol 235

Appendix E Utilities 237

E.1 API Demonstration Utilities 237

E.2 TrunkPack Downloadable Conversion Utility 237

E.2.1 Generating the Call Progress Tones Configuration Files 238

E.2.2 Generating Voice Prompts Files 239

E.2.3 Generating CAS Protocol Configuration Files 242

E.2.4 Generating VXML Files 243

E.2.5 Generating Prerecorded Tones Files 243

E.2.6 Generating Encrypted ini File(s)/Decrypting ini files 246

E.3 PSTN Trace Utilities 247

E.4 Enabling PSTN Trace via the Web 247

E.5 MGCP Tester Utility 248

E.6 MEGACO Tester Utility 249

Appendix F MGCP Compliance 251

F.1 MGCP Compliance Matrix 251

Appendix G MEGACO Compliance 265

G.1 MEGACO Compliance Matrix 265

Appendix H Customizing the AudioCodes Web Interface 277

H.1 Company & Product Bar Components 277

H.2 Replacing the Main Corporation Logo 278

H.2.1 Replacing the Main Corporation Logo with an Image File 278

H.2.2 Replacing the Main Corporation Logo with a Text String 279

H.3 Replacing the Background Image File 280

H.4 Customizing the Product Name 281

H.5 Modifying ini File Parameters via the Web Interface's AdminPage 282

Appendix I List of Abbreviations 283

Appendix J Regulatory Information 287

List of Figures

Figure 1-1: TP-1610 Functional Block Diagram 19
Figure 1-2: Typical Mediant 2000 VoIP Application 20
Figure 2-1: Front View of Mediant 2000 Populated with AudioCodes' TP-1610 Board 21
Figure 2-2: AudioCodes' TP-1610 Board 22
Figure 2-3: View of AudioCodes' TP-1610 RTM 23
Figure 2-4: TP-1610 Board, Front Panel View 24
Figure 2-5: RTM Panel with 2 Telco Connectors (For 16-Spans) 25
Figure 2-6: RTM Panel with 8 RJ-48c Trunk Connectors 26
Figure 2-7: RTM Panel with 4 RJ-48c Trunk Connectors 27
Figure 2-8: RTM Panel with 2 RJ-48c Trunk Connectors 28
Figure 2-9: RTM Panel with 1 RJ-48c Trunk Connector 29
Figure 3-1: Plastic Bag Contents 34
Figure 3-2: Mediant 2000 Front Panel 34
Figure 3-3: Mediant 2000 Rear Connectors
(16 Spans Dual AC power) 37
Figure 3-4: Mediant 2000 Rear Connections (8 Spans, DC Power) 37
Figure 3-5: DC Power Connector 38
Figure 3-6: 50-Pin Female Telco Board-Mounted Connector 39
Figure 3-7: RJ-48c Trunk Connectors 39
Figure 3-8: RJ-45 Ethernet/LAN Network Connectors 40
Figure 3-9: Mediant 2000 Rear with Connected Cables
(16 Spans and Dual AC) 41
Figure 3-10: Mediant 2000 Rear with Connected Cables
(8 Spans and DC) 41
Figure 4-1: Device Startup Process 51
Figure 5-1: MEGACO-R2 Call Flow Diagram 78
Figure 6-1: Enter Network Password Screen 113
Figure 6-2: Web Interface Screen - Example 114
Figure 6-3: Quick Setup Screen 116
Figure 6-4: MGCP Protocol Definition Screen 118
Figure 6-5: MEGACO Protocol Definition Screen 118
Figure 6-6: Advanced Configuration Parameters Screen 119
Figure 6-7: Network Settings Screen 120
Figure 6-8: Channel Settings Screen 121
Figure 6-9: Trunk Settings Screen 122
Figure 6-10: Q931 Bit Map Screen 124
Figure 6-11: TDM Bus Settings Screen 125
Figure 6-12: Configuration File Screen 127
Figure 6-13: Regional Settings Screen - Sending CPT, CAS
and/or Voice Prompt File to the Device 128
Figure 6-14: Change Password Screen 129
Figure 6-15: Status and Diagnostic Menu Screen 130
Figure 6-16: Trunk and Channel Status Screen 131
Figure 6-17: Channel Status Screen 132
Figure 6-18: RTP/RTCP Settings Screen 132
Figure 6-19: Fax & Modem Settings Screen 133
Figure 6-20: Transport Settings Screen 133
Figure 6-21: Voice Settings Screen 133
Figure 6-22: IBS Detector Settings Screen 133
Figure 6-23: Jitter Buffer Settings Screen 134
Figure 6-24: Message Log Screen 134

Figure 6-25: Versions Screen 135
Figure 6-26: Start Software Upgrade Screen 136
Figure 6-27: Start Software Upgrade Screen 137
Figure 6-28: Load CMP File Dialog Screen 137
Figure 6-29: File Loading Dialog Screen 138
Figure 6-30: File Loading Dialog Screen - CPT Type Displayed 139
Figure 6-31: Finish Dialog Screen 139
Figure 6-32: End of Process Dialog Screen 140
Figure 6-33: Auxiliary Files Download Screen 141
Figure 6-34: License Screen 142
Figure 6-35: Save Configuration Dialog Screen 143
Figure 6-36: Reset Screen 143
Figure 7-1: ini File Structure 182
Figure 7-2: Structure of a Table in an ini File 185
Figure 7-3: Example of an ini File for MGCP 186
Figure 7-4: ini File Example for MEGACO 187
Figure 8-1: AudioCodes' Syslog Server Main Screen and Settings Screen 199
Figure 8-2: Setting the Syslog Server IP Address 200
Figure 8-3: Setting Syslog Server IP Address, Enabling Syslog,
in an ini File (Example) 201
Figure A-1: Main Screen 214
Figure A-2: Preferences Screen 216
Figure A-3: Client Configuration Screen 217
Figure A-4: AudioCodes BootP/TFTP Templates Screen 218
Figure E-1: Downloadable Conversion Utility Opening Screen 237
Figure E-2: Call Progress Tones Dialog 238
Figure E-3: Voice Prompts Window 239
Figure E-4: Select Files Window 240
Figure E-5: Voice Prompts Window with wav Files 241
Figure E-6: File Data Window 241
Figure E-7: Call Associated Signaling (CAS) Window 242
Figure E-8: VXML Window 243
Figure E-9: Prerecorded Tones file(s Window 244
Figure E-10: Prerecorded Tones file(s) Window with wav Files 245
Figure E-11: File Data Window 245
Figure E-12: Encrypted ini file(s) window 246
Figure E-13: Trunk Traces 248
Figure E-14: UDP2File Utility 248
Figure H-1: Web Interface Title Bar 277
Figure H-2: Customized Web Interface Title Bar 277
Figure H-3: Image Download Screen 278
Figure H-4: ini Parameters Screen 282

List of Tables

Table 2-1: Chassis Indicators 22
Table 2-2: Board Status LED Indicators....30
Table 2-3: Trunk Status LED Indicators....30
Table 2-4: Ethernet LED Indicators 30
Table 2-5: Auxiliary LED Indicators....30
Table 3-1: Connections on Each 50-Pin Telco Connector 39
Table 4-1: Software Package Contents....46
Table 4-2: Software Directory Description (continues on page 46 to page 47)....46
Table 4-3: Command Line Switch Descriptions....54
Table 4-4: Vendor Specific Information Field....55
Table 4-5: Vendor Specific Information Fields....56
Table 5-1: Fax Transport Type....61
Table 5-2: MGCP Mapping of Payload Numbers to Coders (continues on page 62 to page 63)....62
Table 5-3: Generic Media Package - G 64
Table 5-4: DTMF Package - D 65
Table 5-5: Line Package - L (continues on page 65 to page 66)....65
Table 5-6: Handset Emulation Package - H 66
Table 5-7: Trunk Package - T 67
Table 5-8: PacketCable (NCS) Line Package - L (continues on page 67 to page 68)....67
Table 5-9: Generic Media Package - G 68
Table 5-10: RTP Package - R....68
Table 5-11: CAS Package - MS....70
Table 5-12: CAS Package - DT 70
Table 5-13: ISUP Trunk Package - IT....71
Table 5-14: Fax Package Definition - FXR....72
Table 5-15: V5 Package Definition....73
Table 5-16: Silence Suppression Operation....80
Table 5-17: MEGACO Mapping Payload Numbers to Coders (continues on page 82 to page 84)....82
Table 5-18: Generic Media Package - G 84
Table 5-19: Base Root Package - ROOT 85
Table 5-20: Tone Generator Package - ToneGen 85
Table 5-21: Tone Detection Package - ToneDet 85
Table 5-22: DTMF Generator Package - DG (continues on page 85 to page 86)....85
Table 5-23: DTMF Detection Package - DD 86
Table 5-24: Call Progress Tones Generator Package - CG 87
Table 5-25: Call Progress Tones Detection Package - CD 87
Table 5-26: Basic Continuity Package - CT 87
Table 5-27: Network Package - NT....88
Table 5-28: RTP Package - RTP 88
Table 5-29: TDM Circuit Package - TDMC 88
Table 5-30: Generic Announcement Package 88
Table 5-31: Expanded Call Progress Tones Generator Package - XCG....89
Table 5-32: Basic Service Tones Generation Package - SRVTN 89
Table 5-33: Expanded Services Tones Generation Package - XSRVTN 89
Table 5-34: Basic CAS Signal/Events....90
Table 5-35: R2 CAS Signal/Events Table 90
Table 5-36: MF Generator Package - MFG (continues on page 90 to page 91) 90
Table 5-37: MF Generator Package - MFG 91

Table 5-38: Inactivity Timer Package - IT 91
Table 5-39: Basic Call Progress Tones Generator with Directionality Package - BCG 92
Table 5-40: Call Type Discrimination Package - CTYP 92
Table 5-41: IP Fax Package - IPFAX 92
Table 5-42: MEGACO Endpoint Names (continues on page 95 to page 96) 95
Table 6-1: Default IP Address and Subnet Mask 112
Table 6-2: Trunk Status Color Indicator Key 123
Table 6-3: Trunk and Channel Status Color Indicator Key 131
Table 7-1: System Parameters 146
Table 7-2: Infrastructure Parameters (continues on page 147 to page 151) 147
Table 7-3: Media Processing Parameters (continues on page 152 to page 157) 152
Table 7-4: PSTN Parameters (continues on page 158 to page 165) 158
Table 7-5: SS7 Parameters (continues on page 165 to page 169) 165
Table 7-6: Control Protocol Parameters (continues on page 170 to page 172) 170
Table 7-7: MGCP-Specific Parameters (continues on page 173 to page 174) 173
Table 7-8: MEGACO-Specific Parameters (continues on page 175 to page 177) 175
Table 7-9: Web Interface Parameters 178
Table 7-10: SNMP Parameters (continues on page 179 to page 180) 179
Table 7-11: SCTP Parameters 181
Table 7-12: Table of Parameter Values Example - Remote Management Connections 183
Table 7-13: Table of Parameter Values Example - Port-to-Port Connections 183
Table 7-14: Call Progress Tones Output Tone Levels 191
Table 7-15: User-Defined Tones Output Tone Levels 192
Table 7-16: Call Progress Tones (continues on page 191 to page 195) 193
Table 8-1: Solutions to Possible Common Problems 203
Table 8-2: Solutions to Possible Voice Problems 204
Table 9-1: Selected Specifications (continued on page 205 to page 209) 205
Table B-1: Payload Types Defined in RFC 3551 219
Table B-2: Payload Types Not Defined in RFC 3551 220
Table B-3: Payload Types Not Defined in RFC 3551 220
Table B-4: Default RTP/RTCP/T.38 Port Allocation 221
Table C-1: V.34Fax to V.34Fax - Bypass Mode 225
Table C-2: V.34Fax to V.34Fax - Events Only Mode 225
Table C-3: V.34Fax to V.34Fax - Relay Mode 226
Table D-1: ST_DIAL: Table Elements 228
Table D-2: CAS Parameters 233
Table F-1: MGCP Compliance Matrix (continues on page 251 to page 263) 251
Table G-1: MEGACO Compliance Matrix (continues on page 265 to page 275) 265
Table H-1: Customizable Logo ini File Parameters for the Image File 279
Table H-2: Customizable Logo ini File Parameters for the String Text 280
Table H-3: Customizable Background ini File Parameters 281
Table H-4: Customizable Product Name ini File Parameters 281
Table I-1: List of Abbreviations (continues on page 283 to page 286) 283

1 Overview of the Mediant 2000

The Mediant 2000 is the cost-effective, entry-level member of 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 correct solution size 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, is 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, an ideal building block for deploying high-density, high availability Voice over IP (VoIP) and wireless enterprise systems.

The Mediant 2000 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 Mediant 2000 supports all necessary functions for voice and fax streaming over IP networks.

The Mediant 2000 supports a broad selection of voice processing related algorithms, including G.711, G.723.1 and G.729A Vocoders, G.168-2000 compliant echo cancelation, 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 redundant 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. An E1/T1 trunk interface module is provided for 16-T1, or 16-E1, or 16-J1 trunks, allowing for full gateway streaming functions in a single cPCI slot.

AudioCodes’ TP-1610 board complies with industry-standard network control protocols including MGCP, MEGACO (H.248) and, optionally, SIP (RFC 3261) or H.323 (ITU ver. 4). The board may also be remotely controlled by TPNCP (AudioCodes’ proprietary TrunkPack Network Control Protocol). These protocols allow for the implementation of a distributed Media Gateway and media server architecture that separates call processing functions from media processing functions, resulting in better redundancy, scalability and higher system availability.

The Mediant 2000 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 Mediant 2000 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 Mediant 2000 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 leading provider of new voice infrastructure network technologies. 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

AudioCodes' Mediant 2000 has the following features:

- **Vocoder configuration options:**
  - PCM/ADPCM, G.723, G.729A, GSM FR and NetCoder
- **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 VoIP calls and FoIP transmissions**
- **Interchangeable IP/RTP or PSTN Endpoints**
- **VoIP packet streaming (RTP/ RTCP) per RFC 1889/1890**
- **TPNCP (AudioCodes' proprietary TrunkPack Network Control Protocol), MGCP (RFC 2705), MEGACO (H.248) and optional H.323 and SIP standards-based 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 (for redundancy) or cPSB back plane (PICMG 2.16) interface**
- **G.168-2000 compliant Echo Cancelation 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 SS7 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**

- **cPSB (PICMG 2.16) support**
- **Rear Transition Module (RTM)**
- **API control via PCI or IP ('hostless' mode)**
- **Management Interfaces: SNMP V2, Embedded Web Server**
- **Compact, rugged 19-inch rack mount unit, 1U 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**
- **Optional dual redundant AC power supplies**

1.2 TP-1610 Software Overview

The 16-span TP-1610 should be considered as two separate Media Gateway modules. Physically, the TP-1610 occupies one slot of a cPCI chassis, but functions as two independent entities, each features its own MAC address, IP address and board handle (when using the acOpenRemoteBoard API function), etc. In addition, each module has 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/Operating System (OS), VoP-compliant, media streaming driver package, also known as VoP Library (VoPLib) or TPNCP. Users may choose not to work with the VoPLib (see Section 4 on page 45). The VoPLib enables users 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. The API also controls all digital PSTN protocols (ISDN PRI, CAS trunks, MFC-R2 trunks, etc.). It enables users to benefit from the integral 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 API.

1.3 Mediant 2000 Applications

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

Examples include:

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

1.3.1 Available Configurations

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

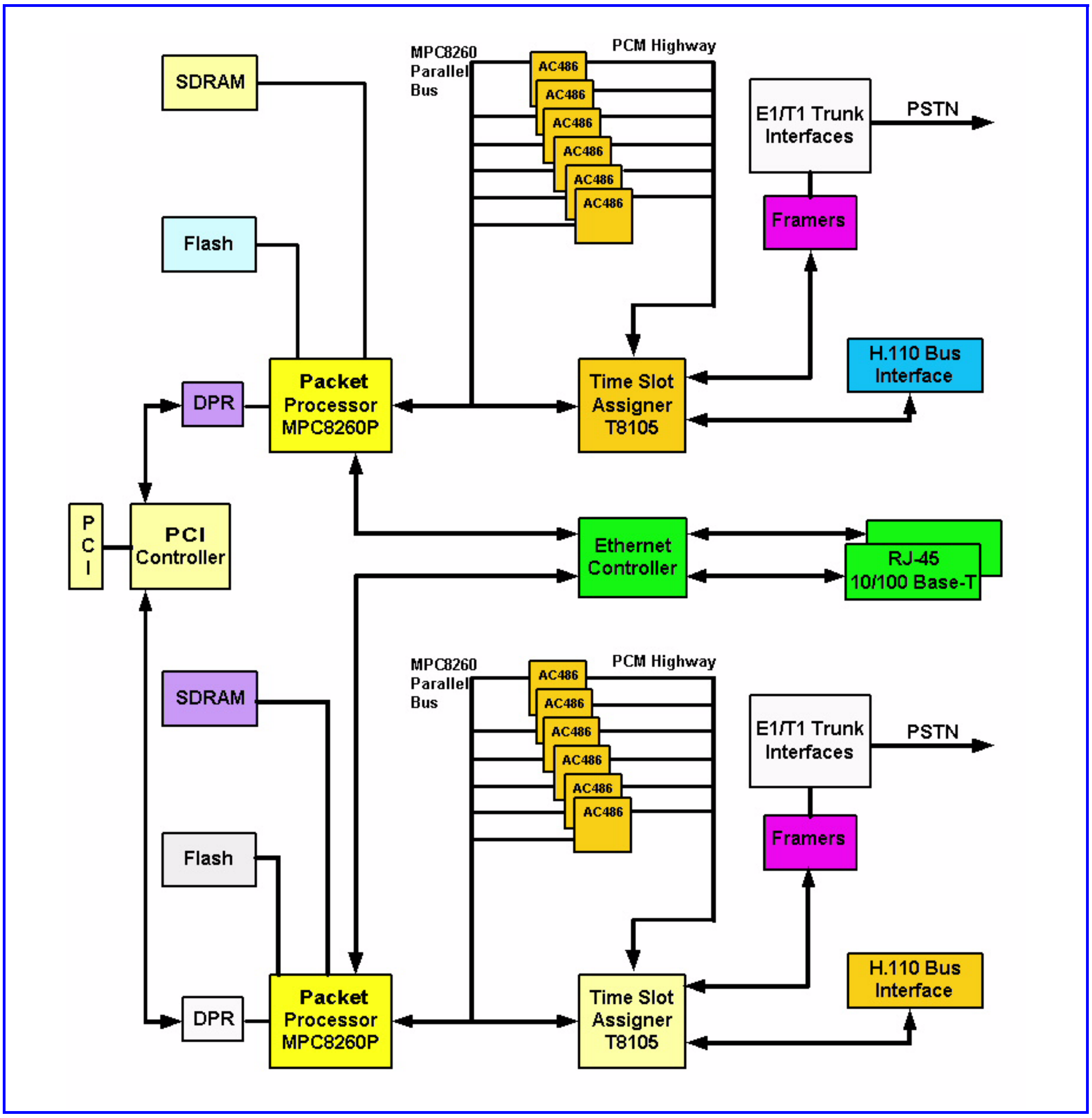
1.4 Benefits

- **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 accumulated by AudioCodes**
- **All-in-one integrated board - Reduced inventory**
- **Scalable distributed architectures**
- **Shorter development cycle**

1.5 Functional Block Diagram

Figure 1-1 illustrates the functionality of AudioCodes' TP-1610 board.

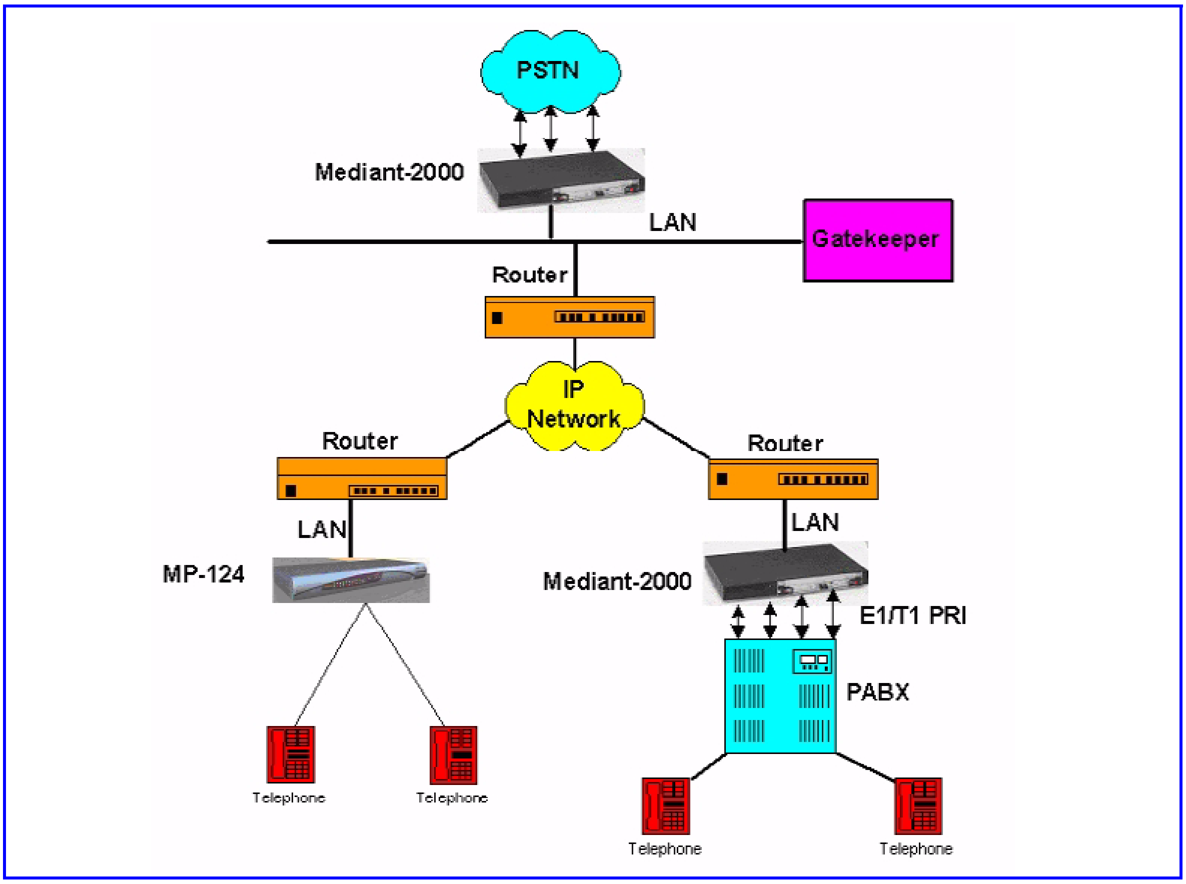
Figure 1-1: TP-1610 Functional Block Diagram



Note: The H.110 Bus Interface in not applicable to the Mediant 2000.

Figure 1-2, illustrates a typical Mediant 2000 VoIP application.

Figure 1-2: Typical Mediant 2000 VoIP Application



2 Hardware Equipment

This section provides details about the hardware equipment for the Mediant 2000 system. This section also describes the system's hardware features and details the various LED functions.

Mediant 2000 includes:

- 1U 19" Chassis
- 1 TP-1610 board
- 1 TP-1610 Rear Transition Module (RTM) board

2.1 The Mediant 2000 Chassis

AudioCodes' Mediant 2000 Media Gateway is comprised of a 19-inch 1U chassis with a single or optional dual 110/220 VAC power supply or a -48 VDC 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.

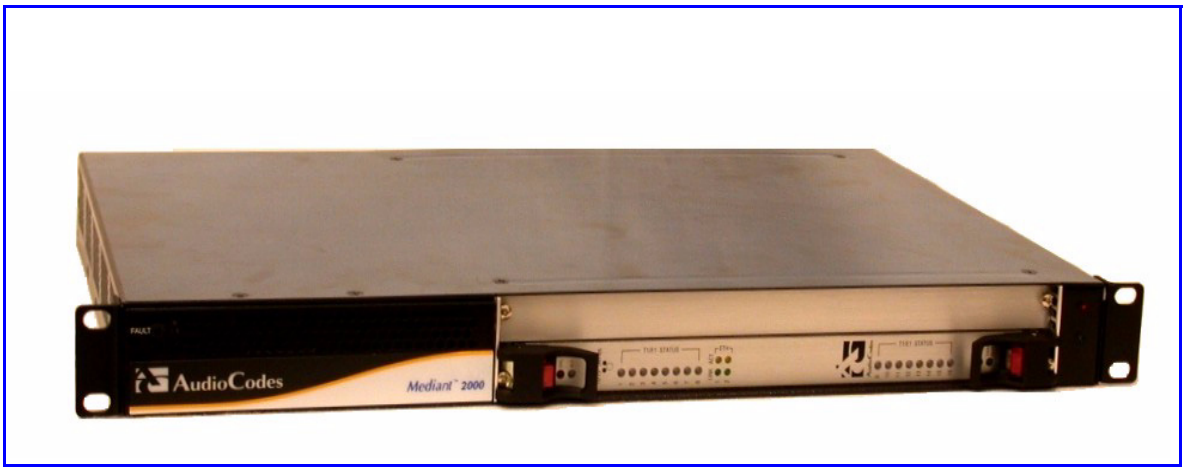
The Mediant 2000 chassis' front cage, slot #1 - the lower slot, houses the TP-1610.

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 optionally can be used by Customers for a CPU board. For more information on CPU board options, refer to Section 2.3 on page 31.

2.1.1 Mediant 2000 Diagram

Figure 2-1: 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 RTM boards

Figure 3-4, on page 37 shows AudioCodes' Mediant 2000 chassis populated with AudioCodes' 8-span RTM. Figure 3-3, on page 39 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 is that the RJ-48c ports are depopulated correspondingly.

2.1.2 Chassis LED Indicators

Table 2-1 details the LED indicators on the front panel of the chassis.

Table 2-1: 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 reduced its speed or has frozen
Left side of front panel	**Red**	Power supply failure - indicates that one of the two AC redundant power supplies is faulty (This LED is not relevant when there is a single AC connection only.)

2.2 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 21.

The front panel of the TP-1610 board is shown in Figure 2-4 on page 24. The panel options for the TP-1610 RTM are shown in Figure 2-5 on page 25 to Figure 2-9 on page 29. Section 2.2.2 on page 30 provides details about the LED indicators and additional information.

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

- RTM rear panel with two 50-pin Telco female connectors, shown in Figure 2-5 on page 25
- RTM rear panel with one to eight RJ-48c connectors, shown in Figure 2-6, on page 26 to Figure 2-8, on page 28.

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

Figure 2-2 displays AudioCodes' TP-1610 board. Figure 2-3 illustrates the board's corresponding Rear Transition Module (RTM).

Figure 2-2: AudioCodes' TP-1610 Board

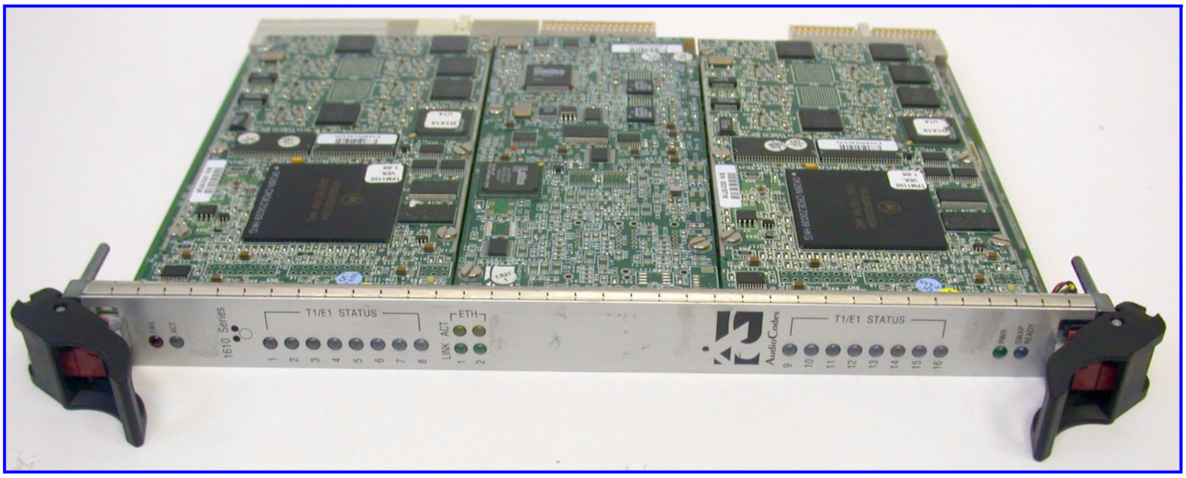
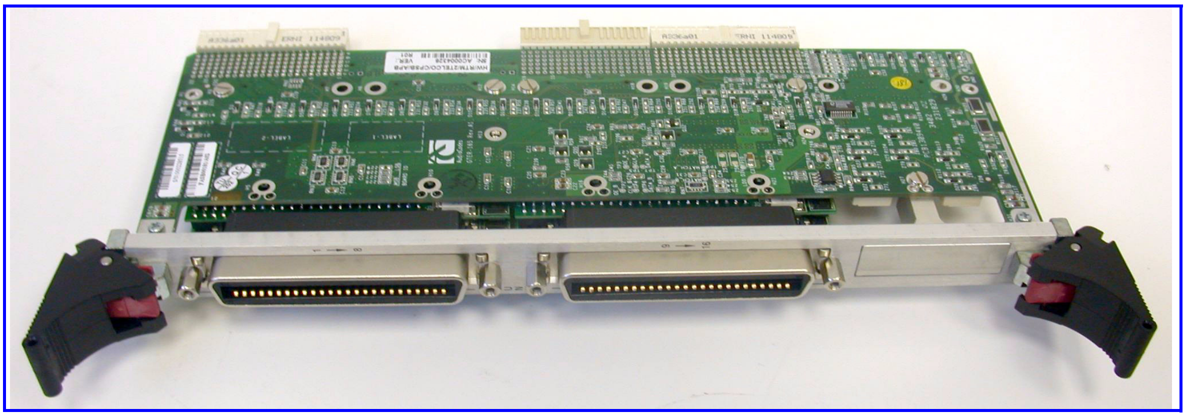


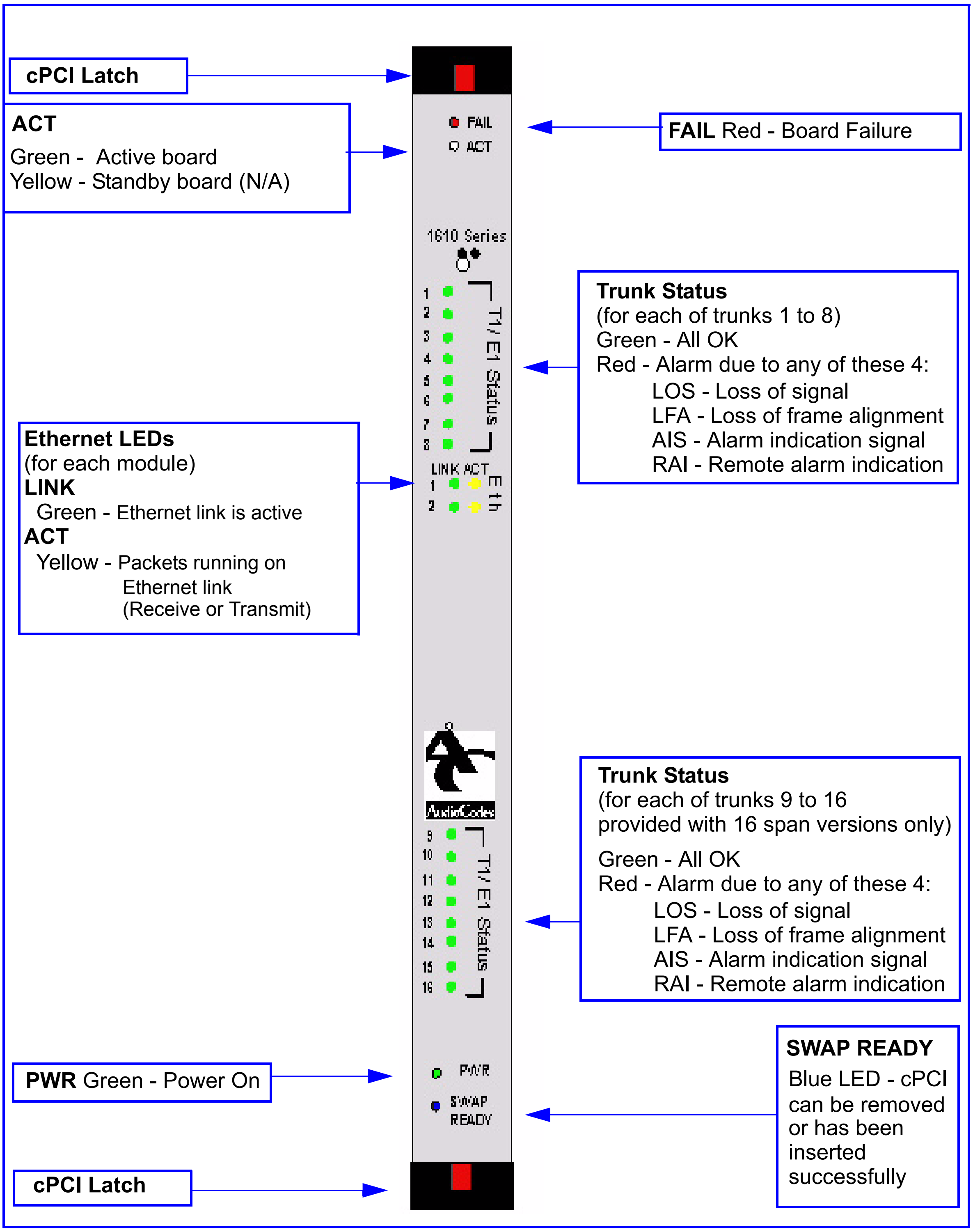
Figure 2-3: View of AudioCodes' TP-1610 RTM



2.2.1 Board Hot-Swap Support

The TP-1610 board is hot swappable and can therefore be removed from a slot (and inserted into a slot) when the cPCI system is under power.

Figure 2-4: TP-1610 Board, Front Panel View



cPCI Latch

ACT
Green - Active board
Yellow - Standby board (N/A)

FAIL Red - Board Failure

Trunk Status
(for each of trunks 1 to 8)
Green - All OK
Red - Alarm due to any of these 4:
LOS - Loss of signal
LFA - Loss of frame alignment
AIS - Alarm indication signal
RAI - Remote alarm indication

Ethernet LEDs
(for each module)
LINK
Green - Ethernet link is active
ACT
Yellow - Packets running on Ethernet link (Receive or Transmit)

Trunk Status
(for each of trunks 9 to 16 provided with 16 span versions only)
Green - All OK
Red - Alarm due to any of these 4:
LOS - Loss of signal
LFA - Loss of frame alignment
AIS - Alarm indication signal
RAI - Remote alarm indication

SWAP READY
Blue LED - cPCI can be removed or has been inserted successfully

PWR Green - Power On

cPCI Latch

Figure 2-5: RTM Panel with 2 Telco Connectors (For 16-Spans)

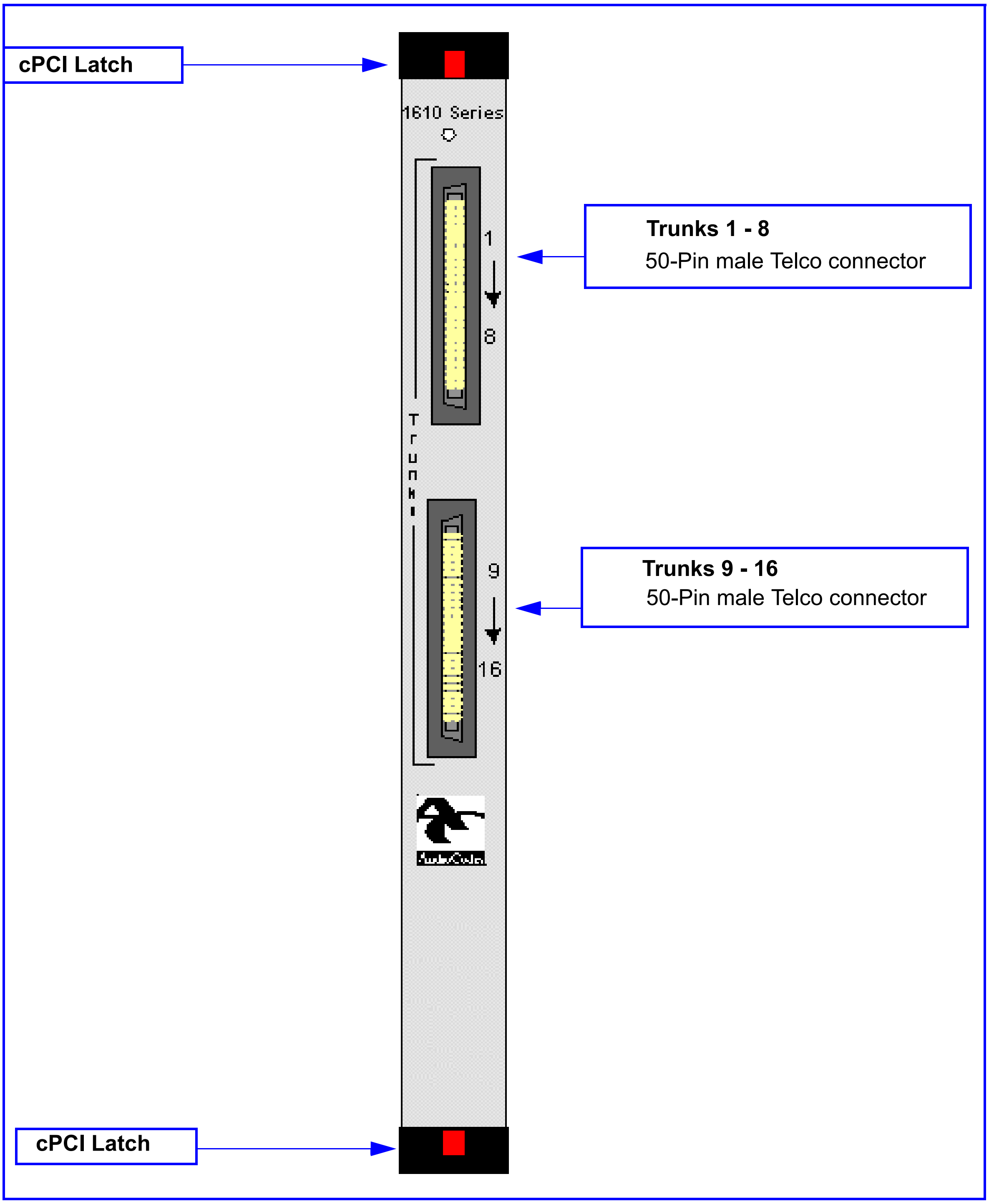


Figure 2-6: RTM Panel with 8 RJ-48c Trunk Connectors

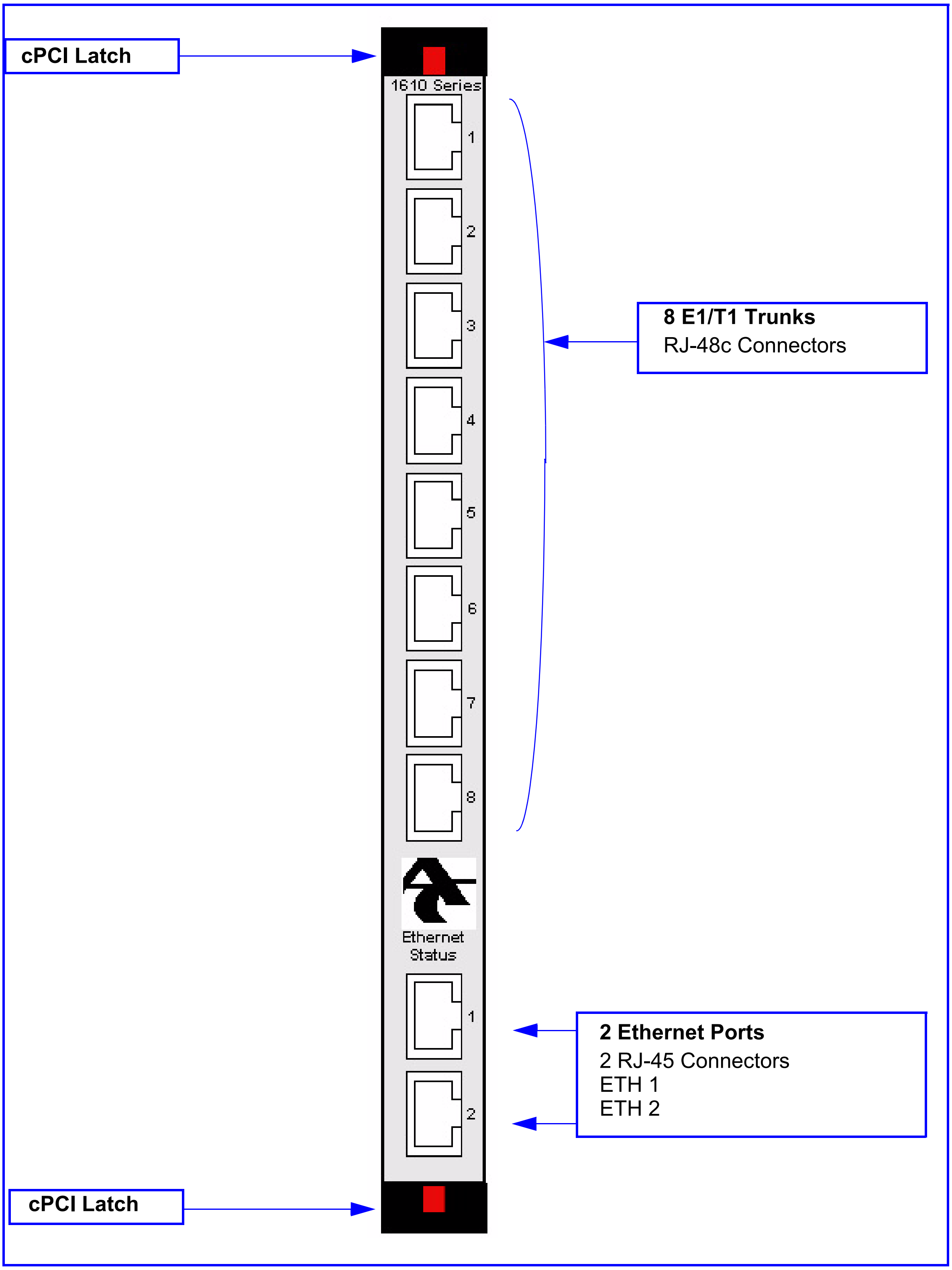


Figure 2-7: RTM Panel with 4 RJ-48c Trunk Connectors

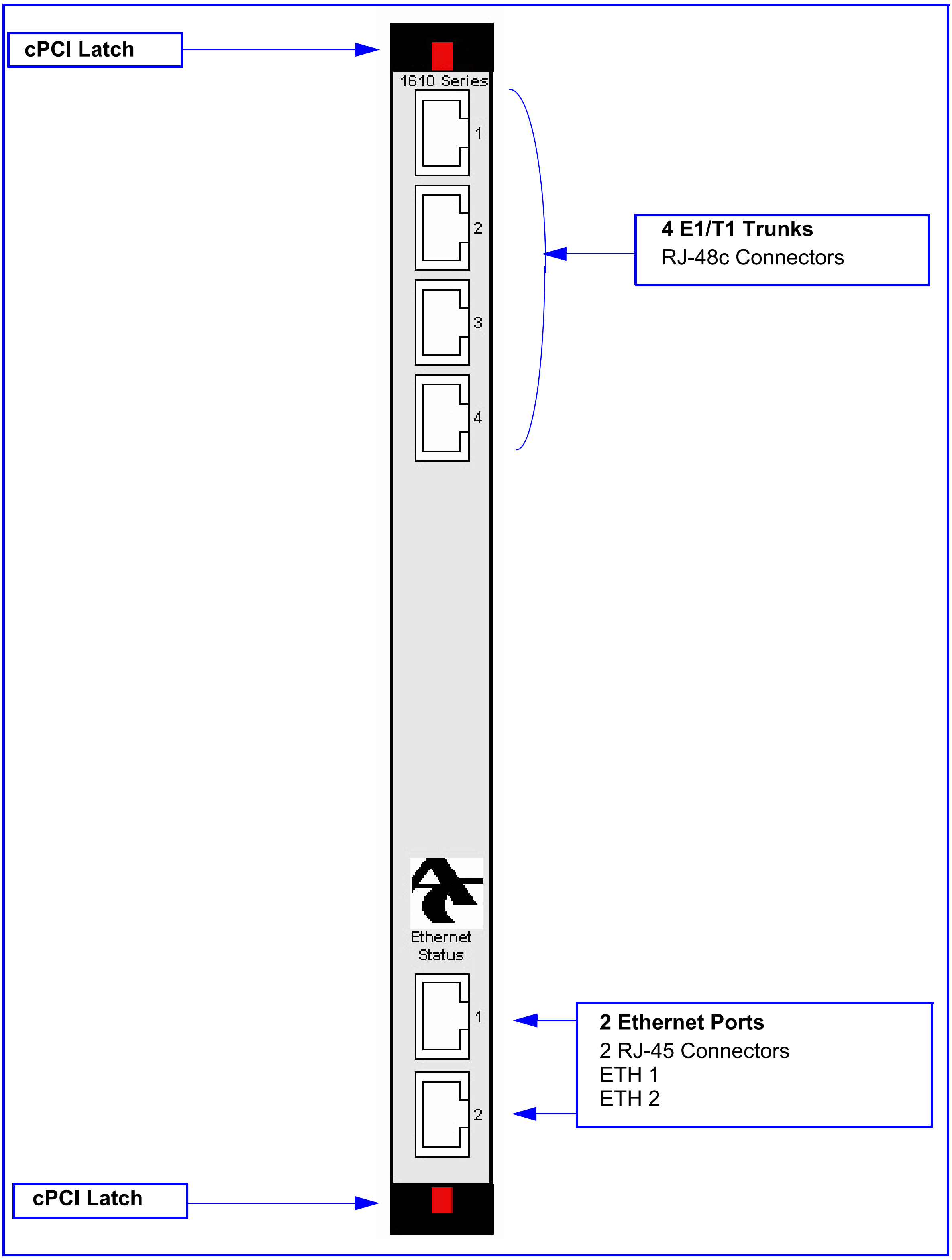


Figure 2-8: RTM Panel with 2 RJ-48c Trunk Connectors

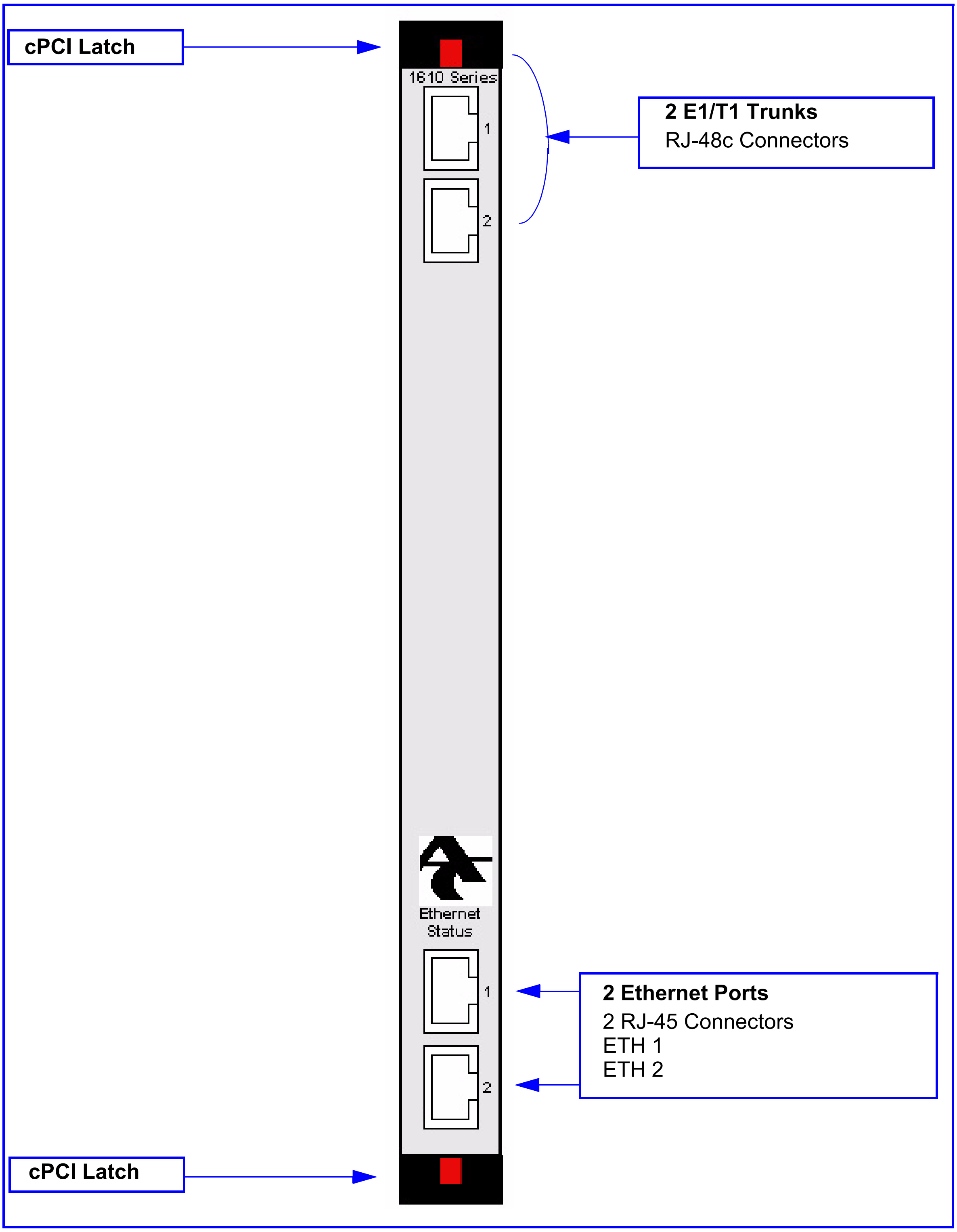
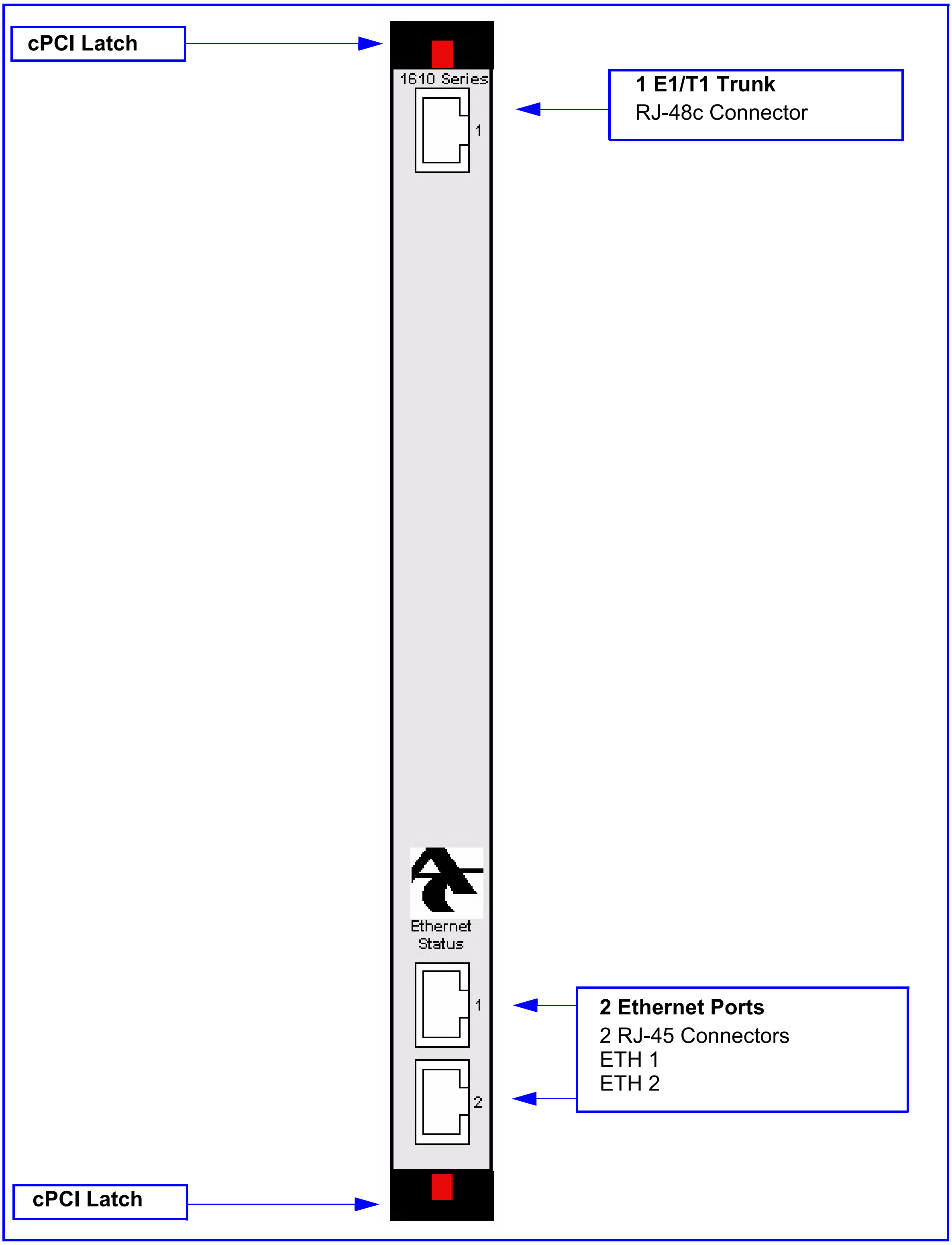


Figure 2-9: RTM Panel with 1 RJ-48c Trunk Connector



2.2.2 TP-1610 Board Front Panel LED Indicators

Refer to Table 2-2 through to Table 2-5 for LED indicator definitions.

Table 2-2: Board Status LED Indicators

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

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

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

Table 2-3: Trunk Status LED Indicators

Label	Color	Signal Description	
Trunk Status 1 to 8 Trunk Status 9 to 16	Green	Trunk is synchronized (normal operation)	Bi-color indicator
	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 operation, the E1/T1 bi-color LED illuminates Green for each trunk. Any other condition, either in the E1/T1 cable, in the Mediant 2000device, or on the remote side, causes the E1/T1 bi-color LED to light up Red, indicating a loss due to any of the 4 signals listed and described in Table 2-3.

Table 2-4: Ethernet LED Indicators

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

Table 2-5: Auxiliary LED Indicators

Label	Color	Function
PWR	**Green**	Power is supplied to the board
SWAP READY	**Blue**	The cPCI board can now be removed. Refer to Note 1.
		The cPCI board was inserted successfully. Refer to Note 2.

Note 1: Before removing the board, wait for the Blue LED to flash ON and then stay OFF. Note that when using the board with a host-supported standard cPCI chassis, the blue LED functionality is dependent on the host software.

Note 2: When inserting the board into the system, if the board has any abnormal physical or electrical condition, then the Blue LED illuminates ON, indicating a fault. If the board is NOT powered up, the Blue LED does not function.

Note 3: For more information of board replacement, refer to Section 3.4 on page 43.

2.3 Optional CPU Board

The Mediant 2000 device features a second cPCI slot that can be optionally used for a customer's CPU board. The CPU board can be used for general applications such as a Softswitch, Application Server, etc. The following CPU boards were tested for compliancy with AudioCodes' Mediant 2000 Media Gateway:

- Sun™: CP2080 + PMC-233 (Ramix™ disk on board)
- Intel™ Ziatech™: ZT5515-A-1B

Reader's Notes

3 Hardware Installation

This section describes the installation procedures for the Mediant 2000 system, as well as board replacement procedures. The Mediant 2000 can be installed either as a desktop system or as a chassis in a standard 19-inch rack.

There are no DIP switches to be set.

Caution Electrical Shock

The equipment must only be installed or serviced by qualified service personnel.

Electrical Grounding

The unit must be permanently connected to ground via the screw provided at the back on the unit. Use 14-16 AWG wire and a proper ring terminal for the grounding.

Electrical Component Sensitivity

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.

➢ **To install the Mediant 2000 take these 3 steps:**

Step 1. Unpack the Mediant 2000 (refer to Section 3.1 below).

Step 2. Mount the Mediant 2000 (refer to Section 3.2 on page 34).

Step 3. Cable the Mediant 2000 (refer to Section 3.3 on page 36).

3.1 Unpacking

➢ **To unpack the Mediant 2000 take these 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.

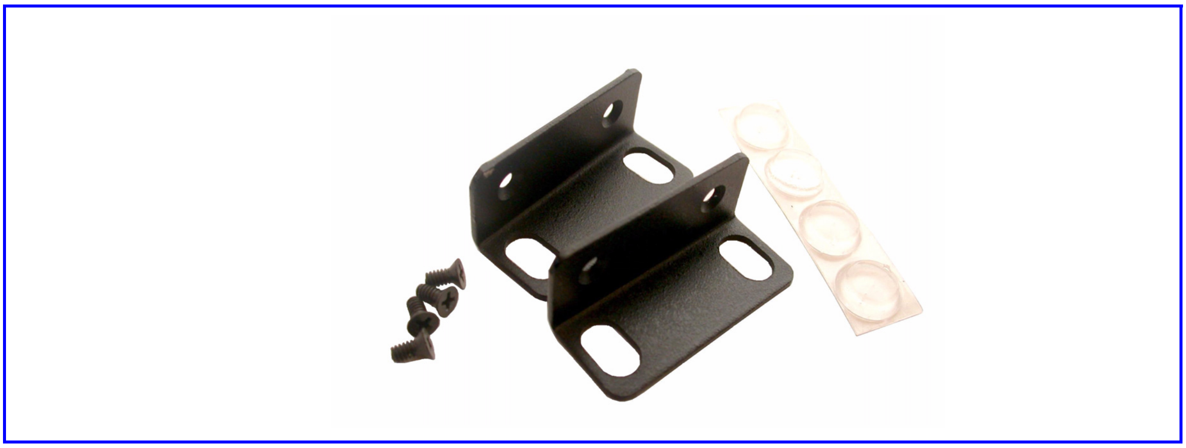
Step 6. Retain any diskettes or CDs.

3.1.1 Package Contents

Ensure that the Mediant 2000 package includes the following items (in addition to the device):

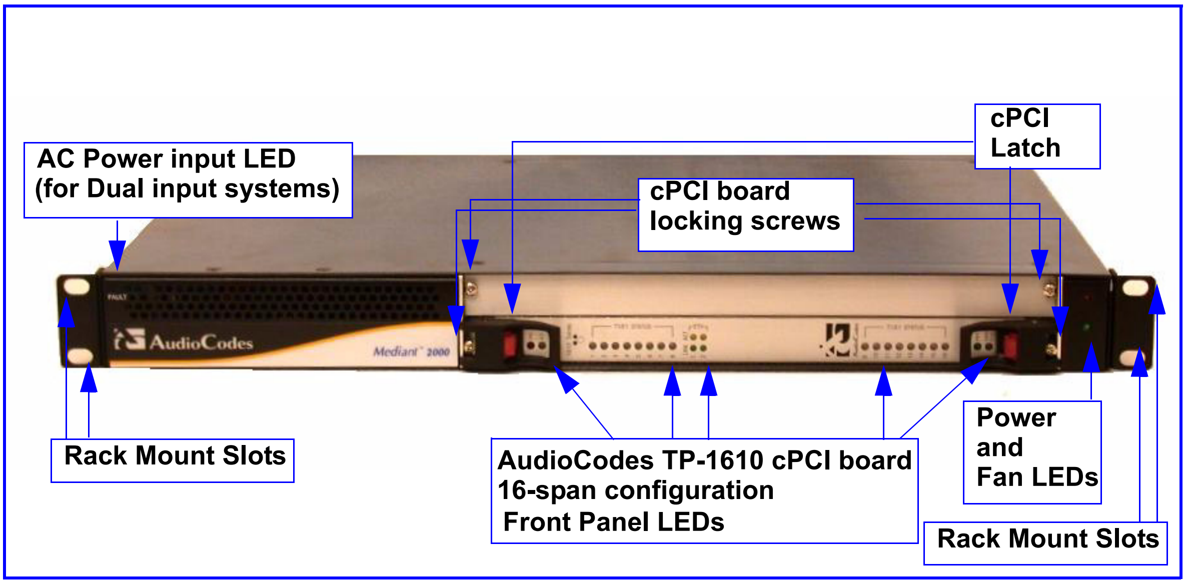
- For the dual AC power supply version of the Mediant 2000, two AC power cables are supplied; for the single AC power supply version of the Mediant 2000, one AC power cable is supplied; for the DC power supply version of the Mediant 2000, one DC power cable is supplied.
- CD (software and documentation).
- Small plastic bag containing (refer to Figure 3 1):
  - Two brackets and four bracket-to-device screws for 19-inch rack installation option.
  - Four non-slip pads for desktop / shelf installation option.

Figure 3-1: Plastic Bag Contents



3.2 Mounting the Mediant 2000

Figure 3-2: Mediant 2000 Front Panel



The 8-trunk device's front panel issimilar to the 16-trunk device's front panel (refer to Figure 3-2), except for trunks 9-16 LEDs.

The 1-trunk, 2-trunk and 4-trunk devices' front panels are identical in appearance to the 8-trunk device's front panel.

If the device supports 1 trunk, only the first LED is functional; if 2 trunks, only the first two LEDs are functional; if 4 trunks, only the first four LEDs are functional.

AudioCodes' Mediant 2000 can be mounted, as required, on a desktop or in a standard 19-inch rack. Attach cables as described in Section 3.3 on page 36.

3.2.1 Mounting a Mediant 2000 in a 19-inch Rack

Users can mount the device on a standard 19-inch rack shelf preinstalled in the rack (preferred method), or by attaching the device directly to the rack's frame using the 2 brackets and screws supplied.

Rack Mount Safety Instructions (UL)

Note: When mounting the chassis on a rack, be sure to implement the following Safety instructions:

- Elevated Operating Ambient - If installed in a closed or multi-unit rack assembly, the operating ambient temperature of the rack environment may be greater than room ambient. Therefore, consideration should be given to installing the equipment in an environment compatible with the maximum ambient temperature (Tma) specified by the manufacturer.
- Reduced Air Flow - Installation of the equipment in a rack should be such that the amount of air flow required for safe operation on the equipment is not compromised.
- Mechanical Loading - Mounting of the equipment in the rack should be such that a hazardous condition is not achieved due to uneven mechanical loading.
- Circuit Overloading - Consideration should be given to the connection of the equipment to the supply circuit and the effect that overloading of the circuits might have on overcurrent protection and supply wiring. Appropriate consideration of equipment nameplate ratings should be used when addressing this concern.
- Reliable Earthing - Reliable earthing of rack-mounted equipment should be maintained. Particular attention should be given to supply connections other than direct connections to the branch circuit (e.g., use of power strips.)

Before rack mounting the chassis, attach the two (supplied) brackets to the front sides of the device. Note that there is an option for two additional brackets to attach the rear sides of the device to the rack.

➢ To attach the two supplied brackets to the front sides of the device, take these 3 steps:

Step 1. Remove the 2 screws nearest the front panel on either side of the device.

Step 2. Align a bracket over 2 holes on one side (so that the bracket's larger holes face front) and with the 2 supplied replacement screws, screw in the bracket.

Step 3. Perform the same procedure on the other side.

➢ To attach the device directly to a 19-inch rack's frame take these 3 steps:

Step 1. Position the device in your 19-inch rack and hoizontally align the left-hand and right-hand bracket holes to selected holes in the vertical tracks of the 19-inch rack.

Step 2. Use standard 19-inch rack bolts (not provided) to fasten the device to the rack tracks.

Step 3. AudioCodes recommends using two additional (not supplied) rear mounting brackets to better support the unit.

> **Note**: Users assembling the rear brackets by themselves should note the following:
>
> - The distance between the two rear screws is 26.5 mm.
> - To attach the rear brackets use 4-40 UMC screws.

➢ To place the device on a 19-inch rack's shelf take these 2 steps:

Step 1. Place the device on the shelf.

Step 2. Follow the steps (optional) for fastening the device to the frame of the rack (as described above) while it is placed on the shelf. This prevents the device from sliding when inserting cables into connectors on the rear panel.

3.2.2 Installing the Mediant 2000 on a Desktop

No brackets are required. Simply place the device on the desktop in the position you require and attach cables as described in Section 3.3 below.

3.3 Cabling the Mediant 2000

Refer to Section 3.3.2 on page 38 for the rear panel cables and cabling procedures. Refer to Figure 3-3 below and Figure 3-9, on page 41 for the rear connectors on the 16-trunk version. Refer to Figure 3-4 below and Figure 3-10, on page 41 for the rear connectors on the 8-trunk version.

> **Note**: The **Mediant 2000** is available in all configuration combinations, i.e., AC or DC, as a 16-trunk, 8-trunk, 4-trunk, 2-trunk or 1-trunk configuration. The 16-trunk AC and the 8-trunk DC configuration combinations are illustrated here as representative of the range.

Figure 3-3: Mediant 2000 Rear Connectors (16 Spans Dual AC power)

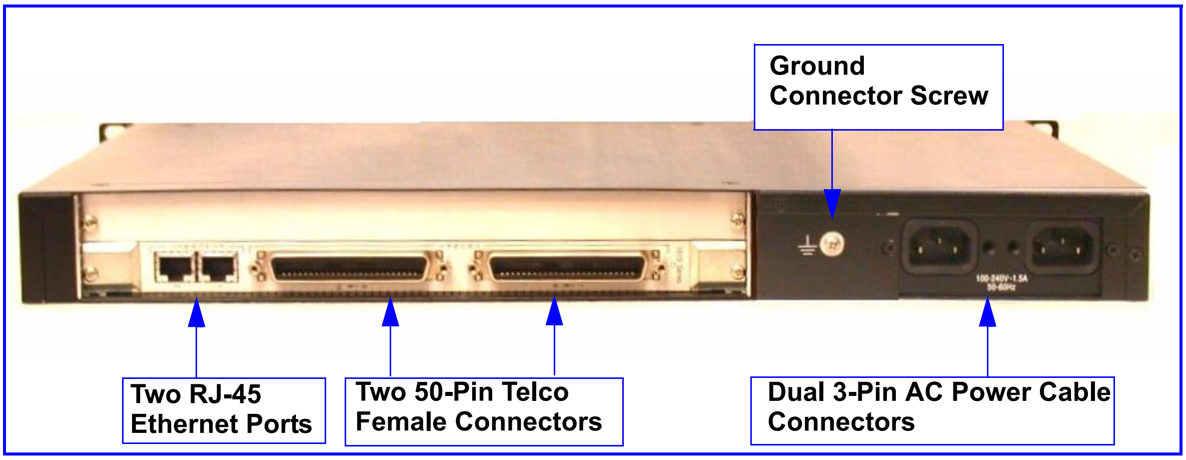
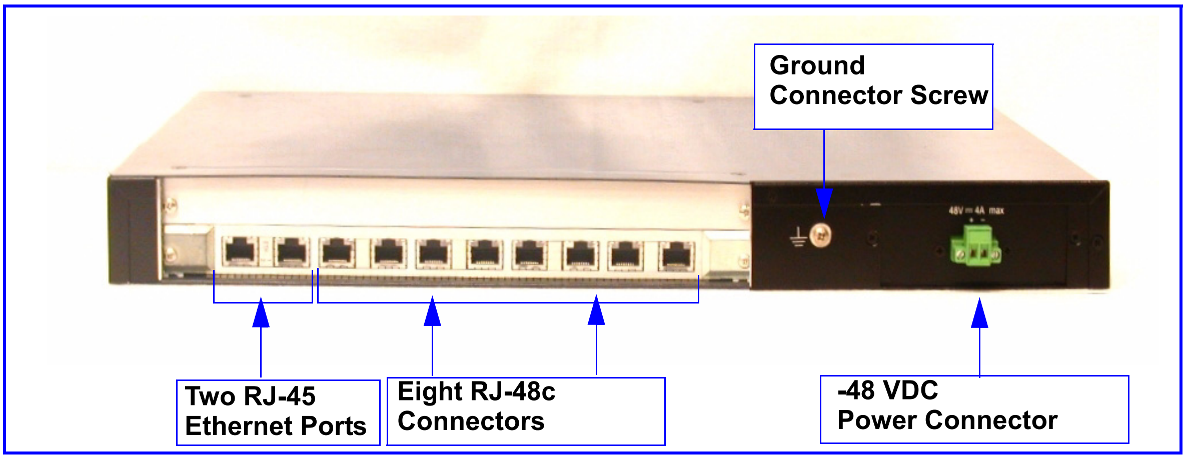

Ground Connector Screw
Two RJ-45 Ethernet Ports
Two 50-Pin Telco Female Connectors
Dual 3-Pin AC Power Cable Connectors

Figure 3-4: Mediant 2000 Rear Connections (8 Spans, DC Power)



3.3.1 Power Supply Cabling

➢ To cable the Power Supply , take these 3 Steps:

Step 1. Permanently connect the unit to a suitable ground with the grounding screw on the rear connector panel, using 14-16 AWG wire.

Step 2. Connect the power supply using one of the following:

Step 2a. **If you are using the 110/220 VAC power supply:**

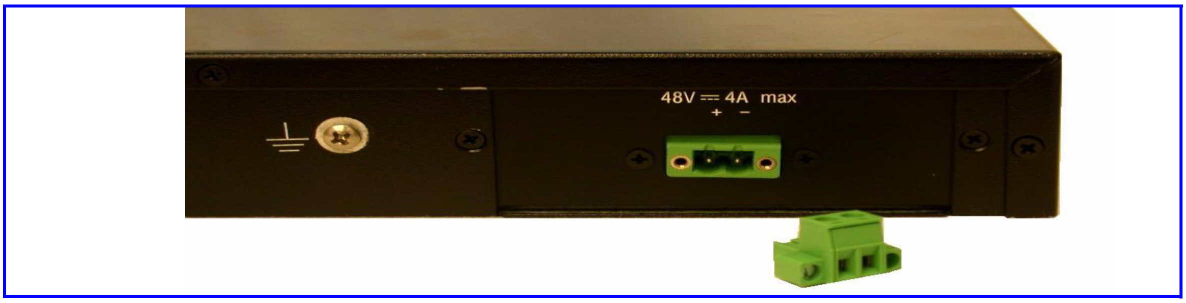
Attach a certified 100/240 VAC power cable to the rear AC socket (or, optionally, two cables to dual AC sockets) and connect to the correct AC power supply.

Step 2b. **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 (5.08mm) of Phoenix Contact). Use 14-16 AWG cables for the DC connection. The DC input is floating and the maximum input current is 4 amps.

Step 3. Connect the power connectors, located on the device's rear panel, to the power source using AC or DC power cables. Refer to Figure 3-5 to view the DC power cabling setup.

Figure 3-5: DC Power Connector



Note: When cabling a device with a dual AC power supply:

- The LED on the left side of the device's front panel is activated only on devices featuring a dual AC power supply. It is not relevant to devices featuring a single AC power supply.
- If only one plug is connected to the AC power supply (while the other plug is left unconnected) the device's front panel LED illuminates Red, indicating a failure.
- When both AC power cables are connected, one of the plugs can be disconnected under power without affecting operation.
- The UPS can be connected to either of the AC connections.
- The dual AC connections provide load-sharing redundancy.

3.3.2 Connecting the E1/T1 Trunk Interfaces

➢ To connect the E1/T1 trunk interfaces, take these 6 steps:

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

- **50-pin Telco connectors (for 16-Spans) Go to Step 2a.**
- **RJ-48c connectors (for 1, 2, 4 and 8-Spans) Go to Step 2b on Page 39.**

Step 2a. If you are using 50-pin Telco connectors:

Connect an E1/T1 trunk cable to the TP-1610 RTM's 50-pin female Telco connector labeled Trunks 1-8 and connect another E1/T1 trunk cable to the RTM's 50-pin female Telco connector labeled Trunks 9-16. Refer to Figure 3-9, on page 41, and detailed in Figure 3-6 on page 39. The pinout of the 50-pin connectors is shown in Table 3-1 on page 39. **Now continue with Step 3 on Page 39.**

Note: The user's 50-pin male connector of Trunks 1 to 8 is connected to the 50-pin female connector (DDK 57AE-40500-21D) 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 3-1 and Figure 3-6 shown below.

Figure 3-6: 50-Pin Female Telco Board-Mounted Connector

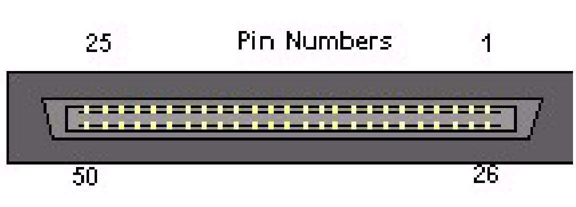


Table 3-1: Connections on Each 50-Pin Telco Connector

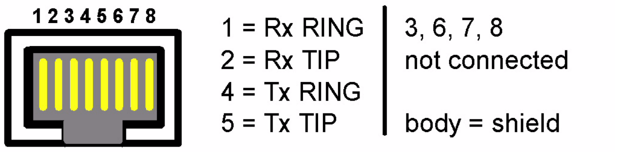
E1/T1 Number		Tx Pins (Tip/Ring)	Rx Pins (Tip/Ring)
1 to 8	9 to 16		
1	9	27/2	26/1
2	10	29/4	28/3
3	11	31/6	30/5
4	12	33/8	32/7
5	13	35/10	34/9
6	14	37/12	36/11
7	15	39/14	38/13
8	16	41/16	40/15

Step 2b. **If you are using RJ-48c connectors:**

Connect the E1/T1 trunk cables to the Mediant 2000's E1/T1 interfaces on the RTM panel (refer to Figure 3-7). There are up to eight RJ-48c connectors labeled Trunks 1 to 8 on the RTM, shown in Figure 3-7 below.

The RJ-48c connectors are wired according to Figure 3-7. **Now continue with Step 3 on Page 39.**

Figure 3-7: RJ-48c Trunk Connectors

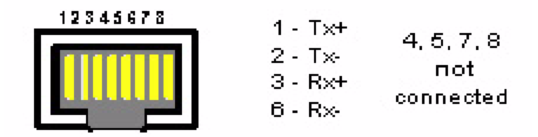


Step 3. On the RTM (shown in Figure 3-9 and Figure 3-10 below), connect the Category 5, LAN cables to the Ethernet 1 and optionally Ethernet 2 RJ-45 interfaces. Connect the other end of the Category 5 LAN cable to your IP network.

Step 4. 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 3-8.

Figure 3-8: RJ-45 Ethernet/LAN Network Connectors



Step 5. Restart the applications and run the system.

Step 6. Power up the Mediant 2000 The Ready and LAN LEDs on the front panel turn to green (after a self-testing period of about 60 seconds). Any malfunction causes the Ready LED to turn red (refer to Section 2.2.2 on page 30 for details on the Mediant 2000 LEDs).

When you have completed the hardware setup, proceed to the software initialization procedure (Section 4 on page 45).

The **Mediant 2000** hardware installation is now complete.

Figure 3-9 and Figure 3-10 display the Mediant 2000 rear view with connected cables.

Figure 3-9: Mediant 2000 Rear with Connected Cables
(16 Spans and Dual AC)

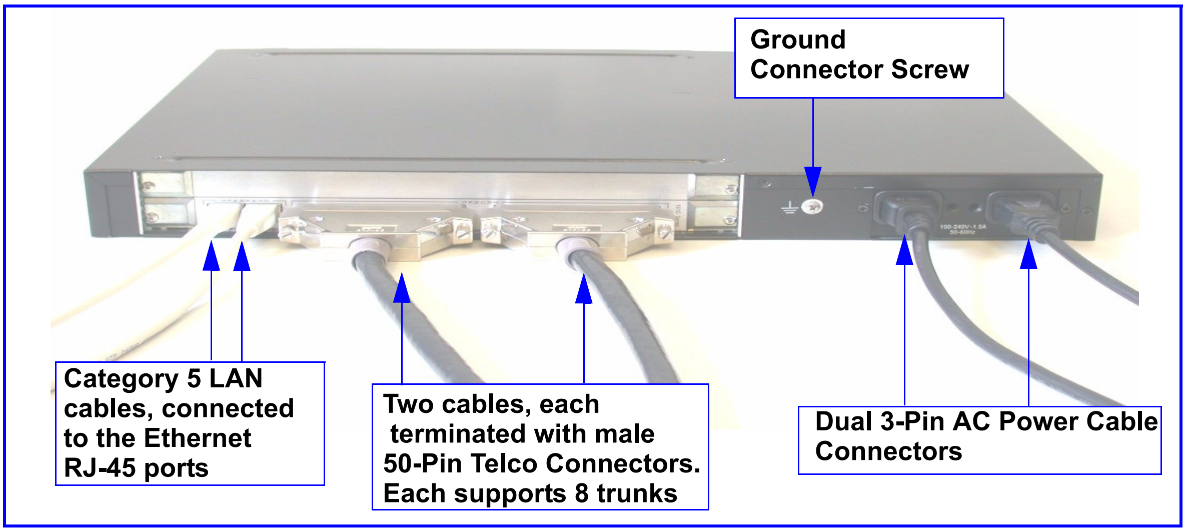
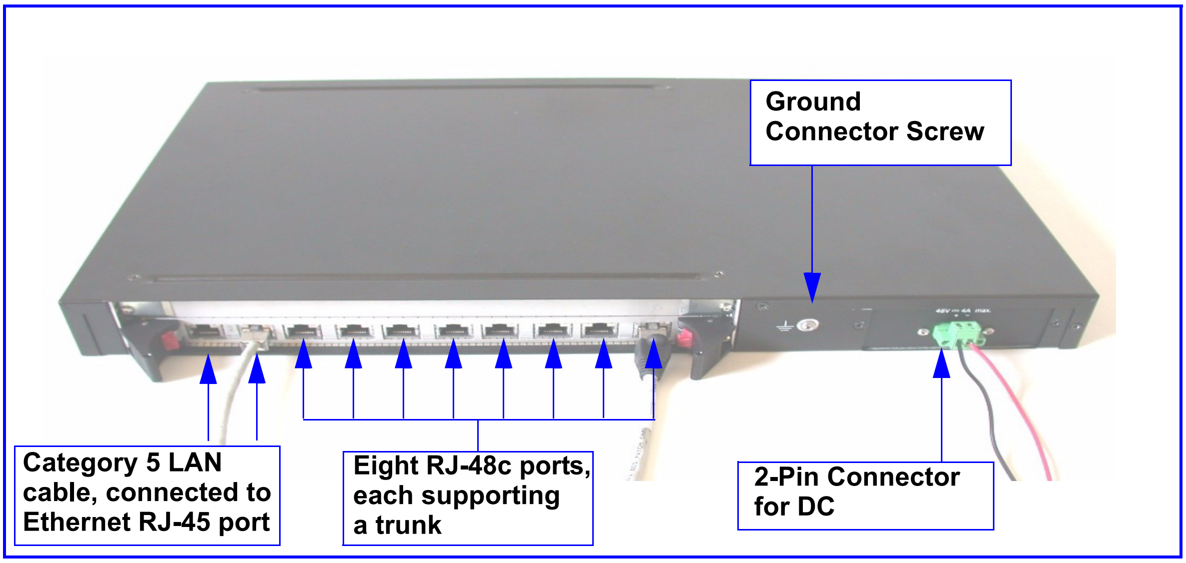

Ground Connector Screw
Category 5 LAN cables, connected to the Ethernet RJ-45 ports
Two cables, each terminated with male 50-Pin Telco Connectors. Each supports 8 trunks
Dual 3-Pin AC Power Cable Connectors

Figure 3-10: Mediant 2000 Rear with Connected Cables
(8 Spans and DC)


Ground Connector Screw
Category 5 LAN cable, connected to Ethernet RJ-45 port
Eight RJ-48c ports, each supporting a trunk
2-Pin Connector for DC

3.4 Board Replacement

3.4.1 Preliminaries

Observe the general safety precautions against personal injury and equipment damage outlined in the regional Installation Safety Manual at all times.

Note: 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.

➢ **Before removing or replacing boards from the chassis, take these 2 steps:**

Step 1. Locate the 2 ESD (electrostatic discharge) connections on the Mediant 2000 chassis. One is located on the front right bottom corner (when facing the front cage), protruding from the flange. The other is located on the rear left corner (when facing the front cage), next to the ground connections.

Step 2. Attach a wrist strap for electrostatic discharge (ESD) and connect it to the ESD connections in either the front or the back of the chassis using a banana plug or an alligator clip.

> **Note**: Do not set components down without protecting them with a static bag.

The TP-1610 board is hot swappable and can therefore be removed from a slot (and inserted into a slot) while the Mediant 2000 Media Gateway is under power.

For details on removing/inserting the optional CPU board, refer to the directions accompanying it.

3.4.2 Performing Graceful Lock

The Mediant 2000 Media Gateway components are hot swappable which means that they can be removed from the chassis without taking the Mediant 2000 out of service. However, currently, the Mediant 2000 is a single gateway which is configured into the network with an N+1 redundancy meaning that other gateways in the network are available to take the load in the event of a card failure. Therefore, AudioCodes recommends you power down the chassis before replacing the components, with the exception of cleaning or replacing the fan filters, which can be carried out while the system is fully functioning.

Prior to powering down the Mediant 2000, it should be locked gracefully (with no active calls forced to terminate) using the element management system employed in your system. For more information on performing graceful (EMS) lock, refer to the user documentation accompanying the element management system employed in your system.

Graceful shutdown parameters, also known as node maintenance, is provided via an SNMP interface. For more information about these parameters, refer to Section 6.2.1 and Section 6.2.2 on page 110.

3.4.3 Removing Boards

➢ **To remove the front TP-1610 board from the device, take these 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 device, take these 4 steps:

Step 1. Remove the cables attached to the RTM.

Step 2. Unfasten the screws on the brackets at both ends of the panel that secure the RTM to the device.

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 RTM board out of the slot.

3.4.3.1 Inserting Boards

> **Note**: Make a note of the MAC address of the replacement board as it is needed for setting the correct parameter configuration for the replacement board in the element management system you are utilizing.

➢ To insert the TP-1610 board into a cPCI chassis, take these 7 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 with the grooves on each end.

Step 3. Ease the board all the way into the slot until the ejector/injector latches touch the chassis. The Blue hot-swap 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 hot-swap blue LED to turn off.

Step 6. Fasten the screws on the front panel of the board to secure the board to the chassis and to ensure that the board has a chassis ground connection.

Step 7. Reattach the cables. Refer to Section 3.3 on page 36.

➢ To insert the TP-1610 RTM into a cPCI chassis, take these 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 with 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 panel of the board to secure the board to the chassis and to ensure that the board has a chassis ground connection.

Step 6. Reattach the cables. (Refer to Section 3.3 on page 36.)

3.4.3.2 Configuring and Unlocking the Mediant 2000

The MAC address of the replacement board should be updated in the parameter configuration in the element management system you are utilizing.

The Mediant 2000 should be unlocked using the element management system (EMS) employed in your system. For more information on performing graceful lock, refer to Section 3.4.2 on page 42 or the user documentation accompanying the element management system (EMS) employed in your system.

Reader's Notes

4 AudioCodes’ Software Package

After installing the device and powering it up, you are ready to install the utilities that are included in the AudioCodes-supplied software package. You install this software package on the host PC/machine to be used to manage the device. The software package is supplied to customers by AudioCodes on a CD accompanying the device.

➢ **To get started, take these 3 basic steps:**

Step 1. Unzip the software package (go to Section 4.1)

Step 2. Check the software package contents (go to Section 4.1.3 on page 46)

Step 3. Access the device's Embedded Web Server (go to Section 4.2.4 on page 49 and Section 7 on page 113) to configure the device's parameters and to initialize the device)

4.1 Unzipping the Software Package

The software package is contained in AudioCodes' CD supplied with the device. It is also available on AudioCodes' FTP server or on AudioCodes' Web site.

- Customers using a Windows™ operating system unzip the software package from the self-extractable ***exe*** file (refer to Section 4.1.1).
- Customers using a Linux™/Solaris™ operating system unzip the software package from the ***tar.Z*** (refer to Section 4.1.2).

4.1.1 Unzipping When Using a Windows™ Operating System

➢ **To unzip when using a Windows™ operating system, take these 3 steps:**

Step 1. Execute the ***exe*** self-extractable ***zip*** file.

Step 2. In the dialog box opened after executing the ***exe*** file, enter the full path of the directory that you intend to be the root directory for the installation.

Step 3. Click **UnZip**. The ***exe*** file installs the extracted files into the specified root directory.

4.1.2 Unzipping When Using a Linux™/Solaris™ Operating System

➢ **To unzip when using a Linux™/Solaris™ operating system:**

- Use the commands: **uncompress tar.Z; tar -xvf xxxx.tar**

4.1.3 Software Directory Structure

Ensure the software package contains the contents of Table 4-1 and Table 4-2.

Table 4-1: Software Package Contents

Contents	Description
VoPLibrary sources	The VoPLibrary is AudioCodes' proprietary API for controlling and managing the device. Detailed information on the VoPLibrary can be found in AudioCodes' VoPLib API Reference Manual, Document #: LTRT-84001 and VoPLib User's Manual, Document #: LTRT-84401.
Utilities	These are AudioCodes' applications that are used for various purposes. For example, the folder "Downloadables_Construction_Utility" is used to build configuration files that can then be downloaded to the device.
Device firmware	The package firmware file is used for firmware updates. When the device is supplied to customers, it contains pre-installed firmware.
Example configuration files	Examples of configuration files are provided. Users can utilize these examples as a baseline for creating customized configuration files.
Documentation	All relevant documentation is found inside the package

.

Table 4-2: Software Directory Description (continues on page 46 to page 47)

Directory	Description
.\VoP_API_Library\VoPLib	Contains the TrunkPack-VoP Library (including the source and header files, as well as the library in debug and release versions)
.\Firmware	Contains the TrunkPack-VoP board's downloadable software file
.\Auxiliary_Files\Sample_Call_Progress_Files	Contains examples of TrunkPack-VoP board's Call Progress configuration files
.\Auxiliary_Files\Sample_CAS_Protocol_Files	Contains the TrunkPack-VoP board's CAS protocol files
.\Auxiliary_Files\Sample_Ini_Files	Contains the examples of TrunkPack-VoP ***ini*** files
.\Auxiliary_Files\MIB_Files	SNMP MIBs files: ACL.my, RTP.my, ds1.my, MIB_2.my, V2_MIB.my
.\VoP_API_Library\VoPLib\Lib\Debug	Contains the voiplib.lib files for Windows™ NT\2000 for debug version
.\VoP_API_Library\VoPLib\Lib\Release	Contains the voiplib.lib files for Windows™ NT\2000 for release version
.\VoP_API_Library\windrvr_5 and **.\VoP_API_Library\windrvr_6**	Contains the WinDriver PCI driver files and utilities for various operating systems (AudioCodes recommends using windrver_6)
.\VoP_API_Library\VoPLib_Tcl_Extension	TCL based demo program named Apirunce. Apirunce is available for Linux™/Solaris™ and Windows OSs.
.\VoP_API_Library\VoPShell	Console based demo program

Table 4-2: Software Directory Description (continues on page 46 to page 47)

Directory	Description
.\VoP_API_Library\VoPLib_API_Examples	Contains examples of Code; Utilities
.\Utilities\MGCPTester	MGCP simulator demo program
.\Utilities\MGCP_Call_Manager_Demo	MGCP Call Manager demo program
.\Utilities\MEGACOTester	MEGACO simulator demo program
.\Utilities\BootP_TFTP_Server	Contains a demo BootP/TFTP server
.\Utilities\DConvert	Contains the Downloadables Construction utility
.\Utilities\PSTN_Trace_Utility	Contains the PSTN Trace utility
.\Utilities\PCI_Diagnostic_Utility	Contains the Diagnostic utility
.\Utilities\AcSyslog	Contains the Syslog
.\Utilities\SpecialID_Utility	Contains a utility to burn customer defined ID
.\Docs	Contains version documentation

Note: All the demo programs described above are for reference only. AudioCodes cannot guarantee flawless operation and stability of these applications.

4.2 Call Control Protocols

The TP-1610 can be controlled by the following call control protocols:

- **VoPLib over PCI/cPCI** - The TP-1610 can be controlled directly through its cPCI interface by AudioCodes' proprietary API. This API is composed of two software modules, the VoPLib, providing control over all VoIP-related functionality, and the PSTNLib, providing control over all PSTN interface functionality. When using the VoPLib over PCI/cPCI, all commands sent to the device and all status indications from it are conveyed through the PCI/cPCI interface. The device must therefore reside in the same machine on which the control application is running. For more information, refer to AudioCodes' VoPLib API Reference Manual, Document #: LTRT-84001 and VoPLib User's Manual, Document #: LTRT-84401.

- **TPNCP (TrunkPack Network Control Protocol)** - This AudioCodes proprietary network-based protocol enables devices to be controlled over the network by a remote host using an API that is almost identical to that used in the "VoPLib/PSTNLib over PCI" mode. (This is effectively "VoPLib/PSTNLib over the Network".) All commands and status indications are sent/received through the IP network (using the integral network interface) using TPNCP. When using TPNCP, the device does not have to reside in the same machine that the control application is running on. For more information, refer to AudioCodes' VoPLib API Reference Manual, Document #: LTRT-84001 and VoPLib User's Manual, Document #: LTRT-84401.

- **MGCP (Media Gateway Control Protocol)** - MGCP is a standard, network-based, control protocol (based on the IETF's RFC 2705). Since MGCP is a standards-based control protocol, AudioCodes does not provide any special software library. Users can choose from many such stacks that are generally available in the market for constructing their own Call Agent. For more information, refer to Section 5.2 on page 61.

- **MEGACO (Media Gateway Control) Protocol** - MEGACO is a standard, network-based, control protocol (based on the IETF's RFC 3015 and ITU-T: H.248). Since this is a standards-based control protocol, AudioCodes does not provide any special software library. Users can choose from many such stacks that are generally available in the market to construct their own Call Agent. For more information, refer to Section 5.3 on page 80.

Notes:

- In general, AudioCodes' VoPLib API provides more control functionality than that provided by MGCP and MEGACO.
- AudioCodes' VoPLib API handles call control, media streaming, as well as configuration functionalities.
- When this text refers to VoPLib users, both TPNCP and VoPLib over PCI/cPCI users are referred to in the reference.
- When either MGCP or MEGACO operate in parallel with TPNCP, only limited functionality is allowed via TPNCP. This limited TPNCP functionality should be used only for Provisioning and Monitoring (audit, query, etc.). Contact AudioCodes for more information.

Note: MGCP and MEGACO cannot coexist in the same device.

For more information on call control protocols, refer to AudioCodes' VoPLib API Reference Manual, Document #: LTRT-84001 and VoPLib User's Manual, Document #: LTRT-84401.

4.2.1 Management Protocols

The device's status can be accessed, viewed and monitored and its configuration modified using SNMP (refer to Section 7.1 on page 113 under 'Management') and AudioCodes' Embedded Web Server (refer to Section 7.4 on page 127 under 'Management')

4.2.2 Simple Network Time Protocol Support

The AudioCodes Simple Network Time Protocol (also known as SNTP) client functionality generates requests and reacts to the resulting responses using the NTP version 3 protocol definitions as defined in IETF's RFC1305. Through these requests and responses, the AudioCodes NTP client is able to synchronize the system time to a time source within the network thereby eliminating any potential issues should the local system clock 'drift' during operation. By synchronizing time to a network time source, traffic handling, maintenance, and debugging actions become simplified for the network administrator.

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

The client requests a time update from a specified NTP server at a specified update interval. In most situations this update interval should be every 24 hours based upon when the system was restarted. The NTP server identity (as an IP address) and the update interval are configurable parameters that can be specified either in the ***ini*** file or via an SNMP MIB object.

When the client receives a response to its request from the identified NTP server it must be interpreted based upon time zone, or location, offset that the system is to a standard point of reference called the Universal Time Coordinate (UTC). The time offset that the NTP client should use is configurable parameter that can be specified either in the ***ini*** file or via an SNMP MIB object.

If needed, the clock update is performed by the client as the final step of the update process. The update is done in such a way as to be transparent to the end users. For instance, the response of the server may indicate that the clock is running too fast on the client. The client will slowly rob bits off of the clock counter in an attempt to update the clock to the correct time. If the clock is running too slow then in an effort to catch the clock up, bits will be added to the counter, causing the clock to update quicker and catch up to the correct time. The advantage of this method is that it does not introduce any disparity in the system time, that is noticeable to an end user, or that could corrupt call timeouts and timestamps.

4.2.3 Boot Software and Firmware

The TP-1610 runs two distinct software programs: Boot software and firmware.

- **Boot software** - Boot software (also known as flash software) resides in the device's non-volatile memory. When the device is reset, Boot software is initialized and the operational software is loaded into the SDRAM from either the PCI host, a TFTP server or integral non-volatile memory. Boot software is also responsible for obtaining the device's IP parameters and ***ini*** file name (used to obtain the device's configuration parameters) via integral BootP or DHCP clients. The Boot software version can be viewed on the Embedded Web Server’s GUI (Section 7.4 on page 127). The last step the Boot software performs is to jump to the first line of code in the operational software.
- ***cmp* and *hex* Firmware Files** - The firmware, in the form of a ***cmp*** file (the software image file) and ***hex*** file (the uncompressed software image file), is supplied in AudioCodes' software package contained on the CD accompanying the device. These files contain the device's main software, providing all the services described in this manual. The ***cmp*** file is usually burned into the device's flash memory so that it does not need to be externally loaded each time the device is reset (except when the board is controlled via PCI).

4.2.4 Provisioning and Configuring

4.2.4.1 Configuration Parameters

The configuration of a device's parameters (IP parameters, control protocol related parameters, PSTN parameters, etc.) determines the device's operation. Parameter configurations are stored in non-volatile memory. After the device is reset (and assuming that you don't want to modify them), they are reloaded from non-volatile memory. Refer to Section 8 on page 161 for more information on configuring/modifying parameters. The following methods are used for configuring the device:

- **VoPLib** - Using AudioCodes' VoPLib, you can configure the device's parameters from acOpenBoard, acOpenRemoteBoard, and acSetInfrastructure APIs. (Refer to AudioCodes' VoPLib API Reference Manual, Document #: LTRT-84001 and VoPLib User's Manual, Document #: LTRT-84401).
- ***ini* File** - The ***ini*** file is an AudioCodes proprietary text file that includes the device's configuration parameters (for more information on the ***ini*** file, refer to Section 8 on page 161).
- **Web Server GUI** - Users can use a regular Web browser to access the device's Web server and configure the device's parameter fields in its GUI.

- **SNMP** - The board supports several SNMP MIBs, which include both standard MIBs and proprietary MIBs. Through these MIBs, the user can set the board’s configuration parameters.

> **Note:** Any of the above configuration methods by itself might not cover all of the configuration parameters.

4.2.4.2 Configuration Files

Configuration files (rather than configuration parameters) are used to configure a device in some circumstances. Configuration files are of the following types (for more information, refer to Section 8.2.1 on page 200):

- **Voice Prompt file** - contains a series of Voice Prompts that the device can play
- **Call Progress Tone and User Defined Tone files** - contains a definition of the call progress tones that the device can play and detect
- **CAS file** - used for CAS protocol configuration
- **Prerecorded Tones file** – contains prerecorded tones that can be played using regular tone generation request.

The configuration files can be loaded into the device via AudioCodes' BootP/TFTP Server or the Embedded Web Server. Most can be burned into the device's non-volatile flash memory so that reloading is not required after every device reset. Configuration files are supplied by AudioCodes in the software package on the CD accompanying the device. Configuring files can also be modified/generated by the users (with AudioCodes' utilities supplied as part of the software package on the CD accompanying the device).

4.2.5 Device Startup

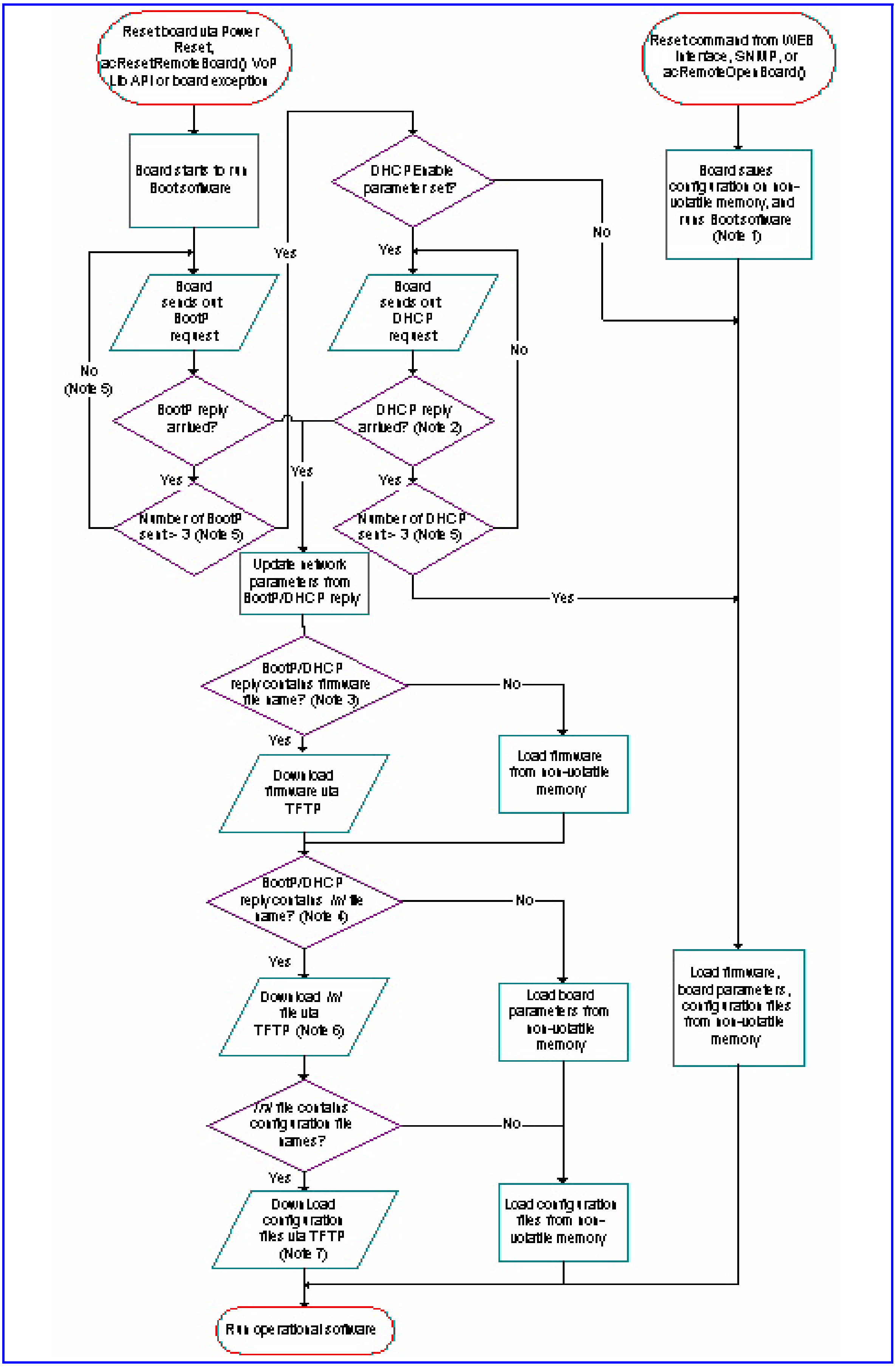
The device's startup process begins when the device is reset. The startup process ends when the operational software is running. The startup process includes how the device obtains its IP parameters, firmware and configuration files.

The device is reset when one of the following scenarios occurs:

1. The power is reset
2. acOpenRemoteBoard() is called with RemoteOpenBoardOperationMode set to Full Configuration Mode
3. acOpenBoard is called; VoPLib users who control the device via PCI call this API to initialize the device
4. acResetRemoteBoard() in the VoPLib API is called
5. There is a device irregularity
6. Users perform a reset in the Embedded Web Server GUI or SNMP manager

The flow chart in Figure 4-1 illustrates the process that occurs in scenarios 1-5. For a detailed explanation on the processes, refer to AudioCodes' VoPLib API Reference Manual, Document #: LTRT-84001 and VoPLib User's Manual, Document #: LTRT-84401.

Figure 4-1: Device Startup Process


Reset board via Power Reset, acResetRemoteBoard() VoP Lib API or board exception
Reset command from WEB Interface, SNMP, or acRemoteOpenBoard()
Board starts to run Boot software
DHCP Enable parameter set?
Board saves configuration on non-volatile memory, and runs Boot software (Note 1)
Yes
Yes
No
Board sends out BootP request
Board sends out DHCP request
No
No (Note 5)
BootP reply arrived?
DHCP reply arrived? (Note 2)
Yes
Yes
Yes
Number of BootP sent > 3 (Note 5)
Number of DHCP sent > 3 (Note 5)
Update network parameters from BootP/DHCP reply
Yes
BootP/DHCP reply contains firmware file name? (Note 3)
No
Yes
Download firmware via TFTP
Load firmware from non-volatile memory
BootP/DHCP reply contains ini file name? (Note 4)
No
Yes
Download ini file via TFTP (Note 6)
Load board parameters from non-volatile memory
Load firmware, board parameters, configuration files from non-volatile memory
ini file contains configuration file names?
No
Yes
Download configuration files via TFTP (Note 7)
Load configuration files from non-volatile memory
Run operational software

Note 1: Resetting via the Embedded Web Server, SNMP and acOpenRemoteBoard() does not cause a BootP command to be issued and the entire configuration (including IP parameters) is obtained from non-volatile flash memory.

Note 2: A DHCP server is usually used to obtain only IP parameters. It is not used to obtain the ***ini*** or firmware file. These can be loaded later via the Web and the configuration can be modified, if required, via SNMP or Web.

Note 3: The BootP/DHCP server should not be defined with the firmware's filename since the firmware is usually stored in non-volatile flash memory. An exception to this rule is when you need to upgrade the firmware.

Note 4: The BootP/DHCP server should not be defined with an ***ini*** file name since the ***ini*** file is stored in non-volatile flash memory. Exceptions to this rule are when you need to modify configuration parameters or when you're working with a large Voice Prompt file that is not stored in non-volatile flash memory and must be loaded after every reset.

Note 5: The default time duration between BootP/DHCP requests is set to 1 second. This can be changed by the ***ini*** file parameter BootPDelay. Also, the default number of requests is 3 and can be changed by the ***ini*** file parameter Boot-PRetries. (Both parameters can also be set using the Command Line Switches in the BootP/TFTP Server).

Note 6: The ***ini*** file configuration parameters are stored in non-volatile flash memory after the file is loaded. When a parameter is missing from the ***ini*** file, a default value is assigned to this parameter and stored in non-volatile memory (thereby overriding any previous value set for that parameter).

Note 7: By default, the configuration files are stored in non-volatile flash memory. Use the ***ini*** file parameter, 'SaveConfiguration=0', to refrain from storing the configuration files in the non-volatile flash memory after loading.

4.2.6 BootP/DHCP Server

AudioCodes' devices use BootP (Bootstrap protocol) and DHCP to configure the device's initial parameters. BootP/DHCP servers enable network administrators to manage the basic configuration of the TP-1610 from a central PC

RFCs (IETF Requests for Comment) 951, 1542, and 2132 describe BootP in detail. AudioCodes has extended the protocol to enable the BootP/DHCP server to configure additional parameters specific to the TP-1610.

As the flow chart in Figure 4-1, on page 51 illustrates, a BootP/DHCP request is issued after a power reset, a device exception, or when calling acResetRemoteBoard() API (assuming that the device was not reset by an acOpenBoard() API).

Note: The BootP server is normally used to configure the initial parameters of the TP-1610. Thereafter, the BootP server need no longer be required as all parameters can be stored in the device's non-volatile flash memory and used when the BootP server is inaccessible. The BootP server will be required again (for example) to change the IP address of the TP-1610.

4.2.6.1 BootP/DHCP Server Parameters

The BootP/DHCP server can be used to provision the following parameters (included in the BootP/DHCP reply. Note that some parameters are optional):

- **IP address, IP subnet mask** - These parameters are mandatory and are supplied by the server to the device every time a BootP/DHCP process takes place.
- **Default Gateway IP address** - This configuration parameter is optional. The default Gateway IP address is supplied to the device by the BootP/DHCP server only if the field is defined/configured in the server.
- **TFTP server IP address** - This optional parameter contains the address of the TFTP server from which the firmware file and ***ini*** file are loaded.
- **DNS Server IP Address (Primary and Secondary)** - These optional parameters contain the IP addresses of the DNS servers. These parameters are available only in DHCP and from Boot version 1.92. A DNS server can only be used by an MGCP/ MEGACO configured device.
- **Firmware file name** - When the device detects that this optional parameter is defined/configured in the BootP/DHCP server, it initiates a TFTP process to load the file.
- **Command Line Switches**

In AudioCodes' BootP/TFTP Server, you can add command line switches in the **Boot File** field (in the Client Configuration screen). Only the ***cmp*** file (the compressed firmware file) can be burned to your device's flash memory. You cannot burn the ***hex*** file (the uncompressed firmware file) because it is too big. Adding the **-fb** command line switch after ***cmp*** in the **BootP File Name** field burns the image file into your device's flash. When the regular ***hex*** extension is used, the loaded image is the one used by the device, but it is not stored in the device's flash memory. If the firmware file name is not specified in the BootP/TFTP server, the device uses the last image stored in its flash memory.

➢ To use a command line switch:

Step 1. In the **Boot File** field, leave the file name defined in the field as it is (e.g., ***ramxxx.cmp***).

Step 2. Place your cursor after ***cmp***.

Step 3. Press the space bar.

Step 4. Type in the switch you require (refer to Table 4-3 below).

Example: **ramxxx.cmp -fb** to burn flash memory

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

Table 4-3 lists and describes the switches available:

Table 4-3: Command Line Switch Descriptions

Switch	Description
-fb	Burn ***ram.cmp*** in flash (only for ***cmp*** files)
-em#	Use this switch to set Ethernet mode. 0 = 10 Base-T half-duplex 1 = 10 Base-T full-duplex 2 = 100 Base-T half-duplex 3 = 100 Base-T full-duplex 4 = auto-negotiate (default) Auto-negotiate falls back to half-duplex mode when the opposite port is not in auto-negotiate but the speed (10 Base-T or 100 Base-T) in this mode is always configured correctly.
-br	BootP retries: 1 = 1 BootP retry, 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 Use this switch to set the number of BootP retries that the device sends during start-up. The device stops issuing BootP requests when either an AA122BootP reply is received or Number Of Retries is reached. This switch takes effect only from the next device reset.
-bd	BootP delays. 1 = 1 sec (default), 2 = 10 sec, 3 = 30 sec, 4 = 60 sec, 5 = 120 sec. This sets the delay from the device's reset until the first BootP request is issued by the device. The switch only takes effect from the next reset of the device.
-bs	Selective BootP
-be	Use -be 1 for the device to send client information that can be viewed in the main screen of the BootP/TFTP Server, under column 'Client Info' (refer to Figure A-3, on page 235, showing BootP/ TFTP Server's main screen with the column 'Client Info' on the extreme right). 'Client Info' can include IP address, number of channels (in the case of AudioCodes' media gateways), which ***cmp*** file is burned into the device's flash memory, etc.

Note: After programming a new ***cmp*** software image file, all configuration parameters and tables are erased. Reprogram them by downloading the ***ini*** file.

- **Configuration (*ini*) file name** - The ***ini*** file is AudioCodes' proprietary configuration file with an ***ini*** extension, containing configuration parameters and tables. For more information on this file, refer to Section 4.2.4 on page 49. When the device detects that this optional parameter field is defined in the BootP server, it initiates a TFTP process to load the file into the device. The new configuration contained in the ***ini*** file can be stored in the device's integral non-volatile flash memory. If it is stored in flash memory, it is loaded into the device whenever the ***ini*** file's name is defined in the BootP server (or when a BootP server is not detected or not searched for).

4.2.6.2 Host Name Support

From Boot software version 1.92, the device registers a device-specific Host Name on the DNS server by defining the Host Name field of the DHCP request. The host name is set to **acl_nnnnnnn**, where nnnnnnn is the serial number of the device (the serial number is equal to the last 6 digits of the MAC address converted from HEX to decimal). The DHCP server registers this Host Name on the DNS server. This feature allows users to configure the device via the Web GUI by providing the following URL: http://ACL_<serial number> (instead of using the boards' IP address).

4.2.6.3 Selective BootP

The Selective BootP mechanism, available from Boot version 1.92, allows the integral BootP client to filter out unsolicited BootP replies. This can be beneficial for environments where more than one BootP server is available and only one BootP server is used to configure devices.

- To activate this feature, add the command line switch **-bs 1** to the Firmware File Name field.
- To deactivate, use **-bs 0**. When activated, the device accepts only BootP replies containing the text **AUDC** in the Vendor Specific Information field.

4.2.6.4 Vendor Specific Information

The device uses the Vendor Specific Information field in the BootP server to provide device-related initial startup parameters (according to RFC 1533). This field is not available in DHCP servers. The field is disabled by default.

To enable / disable this feature user can do one of the following:

a Set the ***ini*** file parameter "ExtBootPReqEnable = **0** to disable, or **1** to enable.

b Use the **-be** command line switch in the Boot file field in the BootP server as follows: **ramxxx.cmp -be 0** to disable, or **-be 1** to enable.

Table 4-4 details the Vendor Specific Information field according to device types:

Table 4-4: Vendor Specific Information Field

Tag #	Description	Value	Length
220	Board Type	02 - TP-1610	1
221	Current IP Address	XXX.XXX.XXX.XXX	4
222	Burned Boot Software Version	X.XX	4
223	Burned CMP Software Version	XXXXXXXXXXXX	12
224	Geographical Address	0 - 31	1
225	Chassis Geographical Address	0 - 31	1
226	TPM ID	N/A	1
227	Rear I/O Version	N/A	1
228	In door - Out door (In door is valid for FXS only. FXO is always Out door.)	N/A	1
229	E&M	N/A	1
230	Analog Channels	N/A	1

The structure of the Vendor Specific Information field is demonstrated in Table 4-5.

Table 4-5: Vendor Specific Information Fields

Vendor-Specific Information Code	Length Total	Tag Num	Length	Value	Tab Num	Length	Value	Tag Num	Length	Value (1)	Value (2)	Value (3)	Value (4)	Tag End
42	12	220	1	02	227	1	1	221	4	10	2	70	1	255

4.2.7 Microsoft™ DHCP/BootP Server

The device can be configured with a 3rd party BootP server (besides AudioCodes' BootP/ TFTP Server), including the Microsoft™ DHCP server, to provide the TP-1610 with an IP address and other initial parameter configurations.

To configure the Microsoft™ Windows™ NT DHCP Server to configure an IP address to BootP clients, add a reservation for each BootP client.

For information on how to add a reservation, view the "Managing Client Reservations Help" topic in the DHCP Manager.

The reservation builds an association between MAC address (12 digits), provided in accompanying product documentation) and the IP address. Windows™ NT Server provides the IP address based on the TP-1610 MAC address in the BootP request frame.

To configure the Microsoft™ Windows™ NT DHCP server to provide Boot File information to BootP clients, edit the BootP Table in the DHCP Manager. The BootP Table is located in the Server Properties dialog, accessed from the Server menu. For information on editing the BootP Table, view the "BootP Table" Help topic in the DHCP Manager.

The following parameters must be specified:

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

Note: The BootP File field should normally not be used. The field is only used for software upgrade (refer to Section 4.3 on page 56).

4.3 Upgrading Device Software

To upgrade the device's firmware, load the upgraded firmware ***cmp*** file into the device (and optionally burn it into integral non-volatile flash memory) using either:

1 **AudioCodes Embedded Web Server** - For a complete description of this option please refer to Section 7.4 on page 127.

2 **AudioCodes BootP/TFTP Server** - By using the **-fb** BootP command line switch, the user can direct the board to burn the firmware on the non-volatile memory. The board thereby downloads the specified firmware name via TFTP and also "burns" the firmware on the non-volatile memory. See Section 4.2.6 on page 52.

Note: Upgrading the device's firmware requires reloading the ***ini*** file and reburning the configuration files.

4.4 License Key

The License Key is a string stored in the TP-1610's non-volatile flash memory, defining the features and capabilities allowed by the specific license purchased by the customer. Customers specify the features and capabilities they require at the time they order the device. The device only allows users to utilize those features allowed by the integral License Key. The License Key is valid for firmware versions 4.2 and above.

The device is supplied by AudioCodes already pre-configured with a License Key according to the customer's order. Users can verify which features are allowed by the license using the Embedded Web Server GUI. (Refer to Section 7.5.6.3 on page 157).

The License Key can be updated to expand the features and capabilities of the TP-1610.

The License Key is upgraded via a file, supplied to customers by AudioCodes, containing a key for one or more devices. Users must specify this file as an ***ini*** file and load it into the device instead of an ***ini*** file. It can be loaded using either:

1. AudioCodes' Embedded Web Server GUI (refer to Section 7.4 on page 127)
2. AudioCodes' BootP/TFTP Server (refer to Section 4.2.6 on page 52)
3. VoP Lib acUpdateLicenseKey() API

When the key is successfully loaded to the device, the following message is issued in the Syslog server:

"S/N___ Key Was Updated. The Board Needs to be Reloaded with ***ini*** file\n"

(S/N is the serial number of the board).

After this message is issued, reset the device and load an ***ini*** file.

Note: The License Key is an encrypted key. Each device utilizes a unique License Key. The License Key is provided by AudioCodes only.

Reader's Notes

5 Standard Control Protocols

5.1 General

AudioCodes' Mediant 2000 can be controlled from a Media Gateway Controller (MGC)/ Call Agent using standard MGCP (Media Gateway Control Protocol), MEGACO (Media Gateway Control) protocol and AudioCodes proprietary TPNCP (TrunkPack Network Control Protocol).

For information on TPNCP, refer to the section on TPNCP in AudioCodes' "VoPLib User's Manual", Document #: LTRT-84401).

5.2 MGCP Control Protocol

5.2.1 MGCP Overview

MGCP (Media Gateway Control Protocol) is a standards-based network control protocol (based on the IETF's RFC 2705). MGCP assumes a call control architecture where the call control intelligence is outside the device and handled by an external Call Agent. MGCP is a master/slave protocol, where the device is expected to execute commands sent by the Call Agent.

Since this is a standards-based control protocol, AudioCodes does not provide any special software library to enable users to construct their own Call Agent. (The user is able to choose any one of many such stacks available in the market).

Note: MGCP and MEGACO protocols cannot coexist on the same device.

5.2.2 MGCP Operation

5.2.2.1 Executing MGCP Commands

MGCP commands, received from an external Call Agent through the IP network, are decoded and executed in the device. Commands can create new connections, delete connections, or modify the connection parameters.

Several commands support the basic operations required to control a device:

- **Connection commands -** Allow the application to create new connections, delete existing connections inside the device, and modify connection parameters.
- **Notify commands -** Using notifications, the device can inform the Call Agent of events occurring on one of the Endpoints. Notify commands can also generate signals on the Endpoints.
- **Audit commands -** These commands are used to query the device about Endpoint configuration and state. This information helps in managing and controlling the device.

5.2.2.2 Setting MGCP Call Agent IP Address

Users can set the IP address and port of the MGCP Call Agent using the CallAgentIP and Call Agent port parameters. A redundant Call Agent can also be specified using the parameters RedundantAgentIP and RedundantAgentPort. The redundant Call Agent is used by the device when a loss of connection with the primary Call Agent occurs. This setting is used only until the first command is received from an actual Call Agent (that is, only for the RSIP message). From then on, the device uses the address of the “real” Call Agent. If MGCP is not to be used, or the Call Agent address is not known at the time the device is opened, set this parameter to **0**.

5.2.3 Using DNS with MGCP

Instead of defining an IP address, users may use a domain name for the Call Agent IP using the 'CallAgentDomainName' and 'RedundantCallAgentDomainName' parameters. DNS(Domain Name System) converts domain names into IP addresses. When the DNS is defined, 'DNSPRISERVERIP' and 'DNSSECSERVERIP' parameters must be configured. While working with domain name, the media gateway device resolves the name during device initialization only.

Using the DNS format for notified an entity is not supported. the device rejects all commands with notified entity in the DNS format.

DNS ***ini*** file configuration:

```
CallAgentDomainName = 'domain name'

RedundantCallAgentDomainName = 'domain name'

DNSPRISERVERIP = IP address

DNSSECSERVERIP = IP address

CallAgentIP = 0

RedundantAgentIP = 0
```

5.2.4 MGCP KeepAlive Mechanism

The device does not initialize commands unless it is asked to do so. Therefore, there is an interval of time until the device notices that its Call Agent is no longer active (the default time interval is 12 seconds). The KeepAlive mechanism maintains a constant connection with the Call Agent. In case of Call Agent failure, the device enters disconnected mode and switches over to its redundant Call Agent. Moreover, since constant transportation is running between the Call Agent and device, using KeepAlive gives VoIP networks the ability to work with NAT machines.

While the KeepAlive mechanism is enabled, the device sends an RSIP command when it detects a time interval without commands received from the Call Agent.

The KeepAlive mechanism deactivates itself when the device looses connection with the Call Agent. KeepAlive messages are sent immediately following the reestablishment of the connection and when no other commands are received during the KeepAlive interval.

ini file parameters:

KeepAliveEnabled = 1(on) or 0(off, default) This parameter can be used to enable a KeepAlive message (NOP ServiceChange).

KeepAliveInterval = 12 (default, seconds) This parameter is used to define the interval in seconds of a KeepAlive message

KeepAlive examples:

While working in endpoint naming conversions:

RSIP 2200 *@AudioCodes.Com MGCP 1.0

RM: X-KeepAlive

While working in trunk naming conversions:

RSIP 2420 ds/tr/*/*@AudioCodes.Com MGCP 1.0

RM: X-KeepAlive

5.2.5 SDP Support in MGCP

MGCP supports basic SDP (Session Description Protocol), as defined in RFC 2327. It also supports SDP-ATM, as defined in RFC 3108. However, the only supported attributes in the SDP are:

- **RTPMAP**
  Used for dynamic payload mapping, to map the number to the coder. The format is:
  a=rtpmap: 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)

- **FMTP**
  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 parameter of the G.723 coder.
  Other supported parameters are:
  mode-set - Defines which mode is used for the AMR and the X-NETCODER coder (0-7)
  annexa - Refers to G.723 if silence suppression is on (yes or no)
  annexb - Refers to G.729 if silence suppression is on (yes or no)

5.2.6 Fax Transport Type Setting with Local Connection Options

In addition to the T.38 fax package described in Section 5.2.8.13 on page 72, the parameter "x-faxtranstype" can set the Fax Transport Type of each connection to either Transparent, Relay or Transparent with Events. If users do not placed this parameter in the Local Connection Options, (LCO) command, the default value configured by ***ini*** file parameter is set.

Table 5-1: Fax Transport Type

Fax Mode	Description
x-faxtranstype:transparent	Fax events are ignored
x-faxtranstype:relay	Fax are transmitted on T.38\FRF11
x-faxtranstype:transparentwithevents	Fax will be transmitted in-band and fax events are detected

5.2.7 TGCP Compatibility

To use TGCP conventions, the user must set the device to the TGCP profile, e.g. adding MGCPCompatibilityProfile = 32 to the device's ***ini*** file.

The following lists the supported TGCP additions:

- **Endpoint Naming Scheme** - Supports wild card and Endpoint naming conventions.
- **Endpoint Name Retrieval** - Wild-carded Audit endpoint command supports MaxEndPointIds, and NumEndPoints parameters.
- **Supported Versions** - The RestartInProgress response and the AuditEndpoint command have been extended with a VersionSupported parameter to enable Media Gateway Controllers and devices to determine which protocol versions each supports.
- **Error Codes** - Supports 532 and 533 error codes.

5.2.7.1 Mapping of Payload Numbers to Coders

Table 5-2 shows the default mapping between payload numbers and coders, when the dynamic payload assignment is not used. This is a general table. Coders are supported according to selected DSPVersion templates - DSPVersionTemplateNumber ***ini*** file parameter.

Table 5-2: MGCP Mapping of Payload Numbers to Coders (continues on page 62 to page 63)

Default Payload Number	Encoding Name	Coder
0	"PCMU", "G711", "G.711","G.711U", "G.711MULAW", "G711MULAW"	G711Mulaw
2	"G726_32"	G726_32
3	"GSM"	GSM
84	"GSM-EFR"	GSM-EFR
4	"G723" "G.723" , "G723", "G723HIGH"	G723 (High)
80	G723LOW	G723 (Low)
8	"PCMA", "G.711A","G.711ALAW"	G711Alaw_64
15	"G728"	G728
18	"G729","G.729","G729A"	G729
35	"G726_16"	G726_16
36	"G726_24"	G726_24
38	"G726_40"	G726_40
39	"X-G727_16"," G727"	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

Table 5-2: MGCP Mapping of Payload Numbers to Coders
(continues on page 62 to page 63)

Default Payload Number	Encoding Name	Coder
45	"X-G727_40_16"	G727_40_16
46	"X-G727_40_24"	G727_40_24
47	"X-G727_40_32"	G727_40_32
49	"X-NETCODER",“NETCODER“, “NETCODER_4_8”	NetCoder_4_8
50	"X-NETCODER",“NETCODER_5_6”	NetCoder_5_6
51	"X-NETCODER", “NETCODER_6_4”	NetCoder_6_4
52	"X-NETCODER",“NETCODER_7_2”	NetCoder_7_2
53	"X-NETCODER",“NETCODER_8”	NetCoder_8
54	"X-NETCODER",“NETCODER_8_8”	NetCoder_8_8
55	"X-NETCODER",“NETCODER_9_6”	NetCoder_9_6
56	"X-CCD", “TRANSPARENT”	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”	G.729E
64	"AMR",“AMR_4_75”,”AMR475”	AMR (4.75)
65	"AMR",“AMR_5_15” , “AMR515”	AMR (5.15)
66	"AMR",“AMR_5_9” ,”AMR590”	AMR (5.9)
67	"AMR", “AMR_6_7“, “AMR670”	AMR (6.7)
68	"AMR", “AMR_7_4”, “AMR740”	AMR (7.4)
69	"AMR",“AMR_7_95”, “AMR795”	AMR (7.95)
70	"AMR",“AMR_10_2”, “AMR1020”	AMR (10.2)
71	"AMR",“AMR_12_2”,”AMR1220”	AMR (12.2)
96	“telephone-event”	RFC 2833
104	“RED”	Redundancy per RFC 2198
13	“CN”,” COMFORT-NOISE”	Comfort Noise
No PayLoad	“IMAGE/T38”	T.38 Fax

5.2.8 Supported MGCP Packages

Events and signals are grouped in packages. Each package supports several events and signals. The TrunkPack series MGCP client supports LINE, DTMF, Fax Package Definition, Media Format Parameter Package, Extended line package, Announcement package, Trunk, Hand Set Emulation and Generic packages.

Note that not all commands/events listed below are applicable to all TrunkPack series devices. For example, hu, hd, hf (all related to on/off hook transitions) are applicable only to devices containing an analog PSTN interface.

Notes for the tables on MGCP Packages (Table 5-3, “Generic Media Package - G,” on page 64 to Table 5-15, “V5 Package Definition,” on page 73):

R: An x appears in this column if the event can be requested by the Call Agent.

S: If nothing appears in this column for an event, then the event cannot be signaled on command by the Call Agent.

Otherwise, the following symbols identify the type of event:

OO signal: The On/Off signal is turned ON until commanded by the Call Agent to switch it OFF, and vice versa.

TO signal: The Timeout signal lasts for a given duration unless it is superseded by a new signal.

BR signal: The Brief signal event has a short, known duration.

Duration: Specifies the duration of TO signals. Signal duration can be changed by adding time out parameter to signal e.g. L/dl(to=18000) , time units are 1 msecs.

5.2.8.1 Generic Media Package - G

Table 5-3: Generic Media Package - G

Symbol	Definition	R	S	Duration
mt	Modem detected	x		
ft	Fax tone detected	x		
rt	Ring back tone		TO	
rbk	Ring back on connection		TO	180 sec

5.2.8.2 DTMF Package - D

Table 5-4: DTMF Package - D

Symbol	Definition	R	S	Duration
0	DTMF 0	x	BR	
1	DTMF 1	x	BR	
2	DTMF 2	x	BR	
3	DTMF 3	x	BR	
4	DTMF 4	x	BR	
5	DTMF 5	x	BR	
6	DTMF 6	x	BR	
7	DTMF 7	x	BR	
8	DTMF 8	x	BR	
9	DTMF 9	x	BR	
#	DTMF #	x	BR	
*	DTMF *	x	BR	
a	DTMF A	x	BR	
b	DTMF B	x	BR	
c	DTMF C	x	BR	
d	DTMF D	x	BR	
t	Inter-digit Timer	x		4 sec
x	Wildcard, match any digit 0 to 9	x		
of	Report Failure	x		

5.2.8.3 Line Package - L

Table 5-5: Line Package - L (continues on page 65 to page 66)

Symbol	Definition	R	S	Duration
0-9, #, *, ABCD	DTMF tones		BR	
hd*	Off hook transition	x		
hu*	On hook transition	x		
hf	Flash hook	x		
bz	Busy tone		TO	30 sec
ft	Fax tone event	x		
mt	Modem tones	x		
dl	Dial tone		TO	16 sec
ro	Reorder tone		TO	30 sec
rt	Ring back tone		TO	180 sec

Table 5-5: Line Package - L (continues on page 65 to page 66)

Symbol	Definition	R	S	Duration
rg	Ringing		TO	180 sec
cf	Confirmation tone		BR	
oc	Report on completion of TO	x		
wt, wt1, wt2,wt3,wt4	Call waiting tones	x	BR	
ci (ti,nu,na)	Caller ID (ci(time, number, name) Time = MM/DD/HH/MN		BR	
sup(addr("digits")	DTMF dialing		BR	
of	Report Failure	x		
Lsa	Line Side Answer Supervision	x	TO	Infinite
OSI	Network Disconnect		TO	900 ms
VMWI	Visual Message Waiting Indicator	x	OO	

* Persistence Events

5.2.8.4 Handset Emulation Package - H

Table 5-6: Handset Emulation Package - H

Symbol	Definition	R	S	Duration/ Comment
hd	Off hook transition	x	OO	
hu	On hook transition	x	OO	
hf	Flash hook		BR	
bz	Busy tone	x		
wt, wt1, wt2,wt3,wt4	Call waiting tones	x	BR	
dl	Dial tone (350 Hz & 440 Hz)	x		
nbz	Network busy (fast cycle busy)	x		
rg	Ringing	x		
ro	Reorder tone	x		
oc	Report on completion	x		
ot	Off hook warning tone	x		
sup(addr ("digits"))	DTMF dialing		BR	Example: Supp(addr(2,3,5))
of	Report Failure	x		
Lsa	Line Side Answer Supervision	x	TO	Infinite
OSI	Network Disconnect	x	TO	900 ms

5.2.8.5 Trunk Package - T

Table 5-7: Trunk Package - T

Symbol	Definition	R	S	Duration/ Comment
co1	Continuity tone		TO	2 seconds
co2	Continuity test		TO	2 seconds
lb	Loopback	x	OO	Supported via 'Connection Mode'
om	Old milliwatt tone	x	OO	
nm	New milliwatt tone	x	OO	
ro	Reorder tone	x	TO	30 seconds
of	Report failure	x		

5.2.8.6 PacketCable (NCS) Line Package - L

Table 5-8: PacketCable (NCS) Line Package - L
(continues on page 67 to page 68)

Symbol	Definition	R	S	Duration/Comment
0-9,*,#,a,b,c,d	DTMF tones	x	BR	
aw	Answer tone	x		
bz	Busy tone		TO	30 seconds
cf	Confirmation tone		BR	
ci(ti, nu,na)	Caller ID		BR	**ti** denotes time, **nu** denotes number, and **na** denotes name
dl	Dial tone		TO	
ft	Fax tone	x		
hd	Off-hook transition	P,S		
hf	Flash hook	P		
hu	On-hook transition	P,S		
mt	Modem tones	x		
mwi	Message waiting indicator		TO	16 seconds
oc	Operation complete	x		
of	Operation failure	x		
ot	Off-hook warning tone	x		Time-out = infinite
r0, r1, r2, r3, r4, r5, r6 or r7	Distinctive ringing (0...7)		TO	
rg	Ringing		TO	180 seconds

Table 5-8: PacketCable (NCS) Line Package - L
(continues on page 67 to page 68)

Symbol	Definition	R	S	Duration/Comment
ro	Reorder tone		TO	180 seconds
rt	Ring back tone		TO	30 seconds
sl	Stutter dial tone		C,TO	180 seconds
wt, wt1, wt2, wt3, wt4	Call waiting tones	x	BR	
x	DTMF tones wildcard	x		Matches any of the digits “0-9”
OSI	Network Disconnect		TO	900 ms
VMWI	Visual Message Waiting Indicator	x	OO	

5.2.8.7 Announcement Package - A

Table 5-9: Generic Media Package - G

Symbol	Definition	R	S	Duration/ Comment
Ann (index)	Play an announcement		TO	Variable
oc	Report on completion			
of	Report failure	x		

5.2.8.8 RTP Package - R

Table 5-10: RTP Package - R

Symbol	Definition	R	S	Duration/ Comment
co1	Continuity Tone (single or return tone)	C	TO	2 sec
co2	Continuity Test (go tone, in dual tone procedures)	C	TO	2 sec
ma	Media Start	C	X	
Rto	RTP/RTCP Timeout	C	X	

RTP/RTCP Timeout (rto(<timeout>,st=<start-time>)):

- **time out -** optional parameter, increase in 100 msec steps. Maximum value is 12800 msec.
- **start-time -** optional parameter, default value is "ra".

If the user does not utilize the event parameters, defaults could be set through ***ini*** file:

- **timeout -** "BrokenConnectionEventTimeOut". Default value is 300 msec. Parameter can be changed in 100 msec steps.
- **Start-time -** "BrokenConnectionEventActivationMode". Default value is 1- start after first incoming rtcp packet. While set to zero the timer will start at once.

Event example

RQNT 2001 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0

X: 1

R: r/rto(N)

In this case a notification occurs if there is a period of time when no RTP or RTCP packets have been received for BrokenConnectionEventTimeOut*100.

The resulting NTFY with observed events would be as follows:

NTFY 3002 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0

X: 1

O: r/rto(300)

Another option could be:

RQNT 2001 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0

X: 1

R: r/rto(N)(4000,st=im)

In case no rtp is received 4 seconds from the time the event was received, remote disconnected event is generated:

NTFY 3002 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0

X: 1

O: r/rto(300)

Continuity Test (go tone, in dual tone procedures) and Continuity Tone (single or return tone):

Continuity tone generation is configuration dependent. To generate continuity tones (if desired), they must be defined by adding EnableContinuityTones = 1 to the device configuration file or define the tones as part of the call progress tone.

5.2.8.9 CAS Package - MS

Table 5-11: CAS Package - MS

Symbol	Definition	R	S	Duration/Comment
ans	Call answer	P	BR	
bl	Block	S	BR	
bz	Busy Tone	-	TO	
inf	Information Digits	X	-	
oc	Operation complete	X	-	
of	Operation fail	X	-	
rel	Release call	P	BR	
res	Resume call	P	BR	
rlc	Release complete	P,S	BR	
ro	Reorder tone	-	TO	
rt	Ring back tone	-	TO	
sup	Call setup	P,S	TO	
sus	Suspend call	P	BR	

5.2.8.10 CAS Package - DT

Table 5-12: CAS Package - DT

Symbol	Definition	R	S	Duration/Comment
ans	Call answer	P	BR	
bl	Block	S	BR	
bz	Busy tone	-	TO	
dl	Dial tone	-	TO	
oc	Operation complete	X	-	
of	Operation fail	X	-	
rel	Release call	P	BR	
res	Resume call	P	BR	
rlc	Release complete	P,S	BR	
ro	Reorder tone	-	TO	
rt	Ring back tone	-	TO	
sup	Call setup	P,S	TO	
sus	Suspend call	P	BR	

5.2.8.11 ISUP Trunk Package - IT

Table 5-13: ISUP Trunk Package - IT

Symbol	Definition	R	S	Duration/Comment
co1	Continuity tone 1		TO	Time-out = 2 seconds
co2	Continuity tone 2		TO	Time-out = 2 seconds
ft	Fax tone		-	
ma	Media start	C	-	
mt	Modem tone		-	
oc	Operation complete		-	
of	Operation failure		-	
ro	Reorder tone	-	TO	Time-out = 30 seconds
rt	Ring back tone	-	TO,C	Time-out = 180 seconds

5.2.8.12 Media Format Parameter Package - FM

Supported FMTP Formats

According to the Media Format Parameter Package, AudioCodes supports the following FMTP formats:

- L:a:codec1;codec2, fmtp:"codec1 formatX", fmtp:"codec2 formatY"
- L:a:codec1;codec2, fmtp:"codec1 formatX";"codec2 formatY"
- L:a:codec1;codec1, fmtp:"codec1 formatX"
- L:a:codec1;codec1, fmtp:"codec1:2 formatX"

Redundancy

- fmtp “red codername1/codername2/…./codernameN”

AMR Family

- fmtp: “AMR mode-set=0” (bitrate=4.75)
- fmtp: “AMR mode-set=1” (bitrate=5.15)
- fmtp: “AMR mode-set=2” (bitrate=5.9)
- fmtp: “AMR mode-set=3” (bitrate=6.7)
- fmtp: “AMR mode-set=4” (bitrate=7.4)
- fmtp: “AMR mode-set=5” (bitrate=7.95)
- fmtp: “AMR mode-set=6” (bitrate=10.2)
- fmtp: “AMR mode-set=7” (bitrate=12.2)

G723 Family

- fmtp: “G723 bitrate=5.3” Low
- fmtp: “G723 bitrate=6.3” High
- fmtp: “G723 annexb=yes” VAD on - Voice Activity Detection on
- fmtp: “G723 annexb=no” VAD off - Voice Activity Detection off

NetCoder Family

- fmtp: “NETCODER mode-set=0” (bitrate=4.75)
- fmtp: “NETCODER mode-set=1” (bitrate=5.15)
- fmtp: “NETCODER mode-set=2” (bitrate=5.9)
- fmtp: “NETCODER mode-set=3” (bitrate=6.7)
- fmtp: “NETCODER mode-set=4” (bitrate=7.4)
- fmtp: “NETCODER mode-set=5” (bitrate=7.95)
- fmtp: “NETCODER mode-set=6” (bitrate=10.2)

G729 Family

- fmtp: “G729 annexb=yes” (VAD on - Voice Activity Detection on)
- fmtp: “G729 annexb=no” (VAD off - Voice Activity Detection off)

5.2.8.13 Fax Package Definition - FXR

Table 5-14: Fax Package Definition - FXR

Symbol	Definition	R	S	Duration/Comment
gwfax	Gateway controlled fax	x		Device controlled fax handling (See below)
nopfax	No special fax handling	x		No special fax handling upon fax (See below)
t38	T.38 fax relay	x		Call Agent controlled T.38 fax relay (See below)

Supported events parameters

- **Device Controlled Fax (gwfax)** - Device controlled fax handling. The device handled fax event is parameterized with one of the following:
  - Start device handled fax was initiated
  - Stop device handled fax ended normally
  - Failure - The procedure ended abnormally
- **No Special Fax Handling (nopfax)** - The no special fax handling event is parameterized with one of the following:
  - Start no special fax handling was in place “O: fxr/nopfax(start)”
- **T.38 fax relay (t38) Call Agent controlled T.38 fax relay** - The Call Agent controlled T.38 fax relay event is parameterized with one of the following:
  - Start Call Agent controlled T.38 fax relay was initiated
  - Stop Call Agent controlled T.38 fax relay
  - Failure Call Agent controlled T.38 fax relay ended abnormally

5.2.8.14 V5 Package Definition X-v5

Table 5-15: V5 Package Definition

Symbol	Definition	R	S	Duration/ Comment
prp	Wink signal	x	BR	
rp	Line polarity reversal		TO	Infinite

5.2.9 MGCP Endpoint Map

The Endpoint Name is a combination of the TrunkName (a string of up to 19 characters) followed by the Trunk Number representing the Trunk in the range 0 to 8, together with the EndpointPrefix (a string of up to 19 characters) followed by a number representing the B-channel number in the range 0 to 31 for E1, or 1 to 24 for T1.

The notation is shown as: EndPointPrefix/TRUNK NAME X/Y

For example: TS/TRUNK#2/31

Where, in this example, EndpointPrefix='TS' and TrunkName='Trunk#'

5.2.10 Compression Codecs

MGCP supports the compression Codecs listed in Table 5-2, "MGCP Mapping of Payload Numbers to Coders (continues on page 62 to page 63)," on page 62. The selected coder(s) should be used in the Local Connection Option line, **a** parameter.

For example: Create a connection command with G.711 coders will look like:

CRCX 10060 Acgw0@[10.1.37.5]

C: 35

L: a:G.711

5.3 MEGACO (Media Gateway Control) Protocol

5.3.1 MEGACO Overview

MEGACO (MEdia GAteway COntrol) Protocol is a standards-based network control protocol (based on IETF RFC 3015 and ITU-T H.248 V1). MEGACO assumes a call control architecture where the call control intelligence is outside the device and handled by an external Media Gateway Controller (MGC). MEGACO is a master/slave protocol, where the device 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 standards-based control protocol, AudioCodes does not provide any special software library to enable users to construct their own Call Agent. (Users can choose any of many such stacks are available in the market.)

Note: MGCP and MEGACO protocols cannot coexist on the same device.

5.3.2 Operation

5.3.2.1 Executing MEGACO Commands

MEGACO commands, received from an external Call Agent through the IP network, are decoded and executed in the device. 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 device:

- **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 device.
- **Connection commands** - The commands Add, Move, Modify and Subtract allow the creation and deletion of a call connection inside the device. 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 device 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 device about Termination configuration and state. This information helps in managing and controlling the device.

A MEGACO-configured device 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 device registration, the session can start. Subsequently, the device responds to MGC commands. Event notifications are sent only if the MGC requests them specifically.

5.3.2.2 KeepAlive Notifications From the Gateway

The Keep Alive notifications from the gateway to the MGC are implemented either using an AudioCodes proprietary mechanism via a NOP serviceChange command (controlled by ***ini file*** parameters), or using the standard inactivity timer package (H.248.14).

For the AudioCodes proprietary mechanism via a NOP serviceChange command there are two parameters:

- **KeepAliveEnabled -** activates or de-activate the Keep Alive function
- **KeepAliveInterval -** defines the inactivity period in seconds

If the KeepAlive mechanism is enabled, the device sends a NOP ServiceChange command when it detects a defined period without commands from the MGC (the default period is 12 seconds.) If no response is received from the MGC, the retransmission mechanism is initiated and eventually causes a new ServiceChange command to be sent to the next available MGC.

For the standard inactivity timer package (H.248.14) Inactivity detection is fully supported. The activation is by requesting the 'it/ito' event on the root termination. The 'mit' parameter of this event defines the inactivity period in 10 millisecond units. Note that this function is not set by configuration. The Call Manager must send a request for this event.

5.3.2.3 Setting MEGACO Call Agent IP Address and Port

Users can provide the device 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 device 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 device 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 does 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 device resolves the name on each disconnection, allowing the user to switch to another Call Agent.

5.3.2.4 Authorization Check of Call Manager IP Addresses

While the MEGACO specification specifies that only one Call Manager can send commands to the gateway at a time, AudioCodes gateways handle the Authorization

Check in either of these modes:

1. No authorization check is performed. This mode specifies that every command is accepted and executed.
2. The IP address of the Call Manager sending the incoming command is checked against the list of provisioned Call Managers. If it matches one on the list, the command is executed. If The Call Manager' IP Address is not found on the list, an error message is sent. This mode is set as the default.

These two modes are controlled by the ***ini*** file parameter 'MEGACOCheckLegalityOfMGC', for which the default value is 1.

5.3.2.5 Support of DiffServ Capabilities

The DiffServ value of the IP header can be set for both the control path and the media path. The range of the DiffServ parameter is between 0 and 63. It enables routers to differentiate between different streams. The value is set via SNMP, Web or ***ini*** file parameter. Note, that changing the value of the control path requires the Gateway to be reset.

The value of the control path is set via the ***ini*** file parameter 'ControlDIFFSERV', and the value for the media path is set via the ***ini*** file parameter 'IPDIFFSERV'.

5.3.2.6 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 5.3.2.7), a new EventDescriptor, or both. The embedded descriptor replaces the current descriptor.

5.3.2.7 Playing Signals

Signals in MEGACO reside in a SignalsDescriptor. Only one signal is allowed in the descriptor as the devices 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 do not 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 that 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 package's description in Section 5.3.2.7 on page 76).

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 **toneId** 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 5-3, "Generic Media Package - G," on page 64. When a CPT file is missing, the device 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}}}}}
```

5.3.2.8 Mediation

Mediation in MEGACO connects 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.

5.3.2.9 CAS/R2 Support in MEGACO

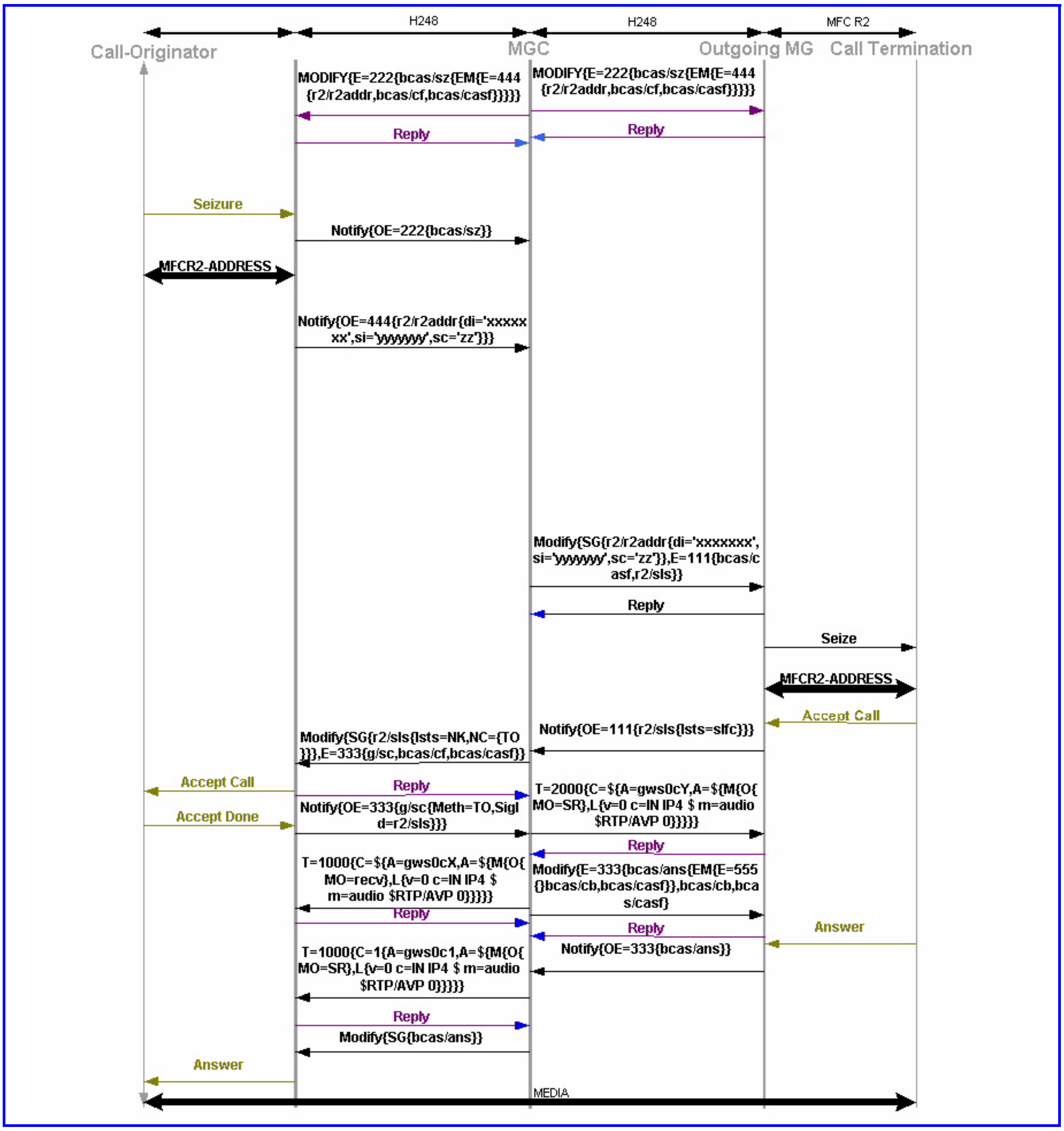
The CAS/R2 trunk protocols are supported in MEGACO by using the ‘bcas’ package and the ‘r2’ package (draft-ietf-megaco-r2package-02).

Using the Basic CAS Package (refer to Section 5.3.4.17 on page 90) and the R2 CAS Package (refer to Section 5.3.4.18 on page 90), AudioCodes' device converts MFCR2 protocol - which is a PSTN protocol - into MEGACO protocol, thereby bridging the PSTN world with the IP world.

When MEGACO and MFCR2 protocols share control of a channel, their timings are synchronized so that MEGACO commands do not cause damage to the MFCR2 protocol's negotiation. For example, MFCR2 protocol must work with the Echo Canceler in OFF state or else Multiple Frequency (MF) is not received correctly. Thus, if MEGACO protocol receives a command to open a channel with the Echo Canceler ON and MFCR2 protocol's negotiation is not yet finished, the entire negotiation could be damaged.

To avoid a collision, use the following call flow:

Figure 5-1: MEGACO-R2 Call Flow Diagram


H248
H248
MFC R2
Call-Originator
MGC
Outgoing MG
Call Termination
MODIFY{E=222{bcas/sz{EM{E=444 {r2/r2addr,bcas/cf,bcas/casf}}}}
MODIFY{E=222{bcas/sz{EM{E=444 {r2/r2addr,bcs/cf,bcas/casf}}}}
Reply
Reply
Seizure
Notify{OE=222{bcas/sz}}
MFCR2-ADDRESS
Notify{OE=444{r2/r2addr{di='xxxxxxx',si='yyyyyyy',sc='zz'}}}
Modify{SG{r2/r2addr{di='xxxxxxx', si='yyyyyyy',sc='zz'}},E=111{bcas/casf,r2/sls}}
Reply
Seize
MFCR2-ADDRESS
Accept Call
Notify{OE=111{r2/sls{lsts=slfc}}}
Modify{SG{r2/sls{lsts=NK,NC={TO}}},E=333{g/sc,bcas/cf,bcas/casf}}
Accept Call
Reply
T=2000{C=${A=gws0cY,A=${M{O{MO=SR},L{v=0 c=IN IP4 $ m=audio $RTP/AVP 0}}}}}
Accept Done
Notify{OE=333{g/sc{Meth=TO,Sigld=r2/sls}}}
Reply
T=1000{C=${A=gws0cX,A=${M{O{MO=recv},L{v=0 c=IN IP4 $ m=audio $RTP/AVP 0}}}}}
Modify{E=333{bcas/ans{EM{E=555{}bcas/cb,bcas/casf}},bcas/cb,bcas/casf}
Reply
Reply
Answer
Notify{OE=333{bcas/ans}}
T=1000{C=1{A=gws0c1,A=${M{O{MO=SR},L{v=0 c=IN IP4 $ m=audio $RTP/AVP 0}}}}}
Reply
Modify{SG{bcas/ans}}
Answer
MEDIA

5.3.2.10 RFC 2833 Support

DTMF Transport Type can be set to use RFC 2833 through configuration or dynamically through MEGACO commands.

> **Note**: RFC 2833 support is only applicable when running Voice Over IP traffic. It is not operational on Voice over ATM.

Configuration is performed through the ***ini*** file (the DTMFTransportType=3 parameter), or through the Web. This value is used by MEGACO as the default value, but commands can override it.

To enable RFC 2833 via a command, add a payload type in the media line of the SDP and define this payload type to be RF 2833 according to the following example:

```
v=0
c= IN IP4 $
m=audio $ RTP/AVP 0 97
a=rtpmap:97 telephone-event 0-15
```

’telephone-event’ is the name defined in RFC 2833, and 97 is used as the payload number (any number from the dynamic range can be used).

Negotiation is performed according to the following rules:

- If the remote side does not specify the ‘telephone-event’ in the SDP, the device uses the default value as the transport type.
- If the local and remote payload types are different, the remote payload number will be used.

Therefore, if a user wishes to activate the RFC 2833 only when both sides agree on it, users should configure the default value to be different to RFC 2833 (e.g., Transparent).

5.3.2.11 Silence Suppression Support

1 Silence suppression can be enabled in two ways:

   a Configure it to ON through one of the configuration tools. This is a static way, and applies to all calls.

   b Use the SDP attribute a=silencesupp:on both for the local and remote side. This is done on a per call basis.

2 Silence suppression can be disabled by:

   a Setting it to OFF in the ***ini*** file. This is a static way, and applies to all calls.

   b For G.729 or G.723 - If the remote descriptor contains the a=fmtp line with annexb=no (G.729) or annexa=no (G.723). Note that the default for the **annex** fields in the SDP is **Yes**. Therefore, if this line is omitted, the assumption is that this side supports the silence suppression according to the annex.

   c Using the SDP attribute a=silencesupp:off in the local or remote side. This is performed on a per call basis. Note that the silencesupp attribute is specified only in RFC 3108 (SDP for ATM). However, as parsers ignore fields they do not recognize, it is legal to use it for IP also, assuming that the call manager is capable of doing it.

   d In all other cases, the device default value is used.

Table 5-1 summarizes the operation of silence suppression:

Table 5-16: Silence Suppression Operation

CONFIG Setting	G.711	G.723	G.729
OFF	ON only if: - a=silencesupp:on AND - payload 13 was offered on both sides	ON only if: - a=silencesupp:on AND - remote SDP does not contain the line a=fmtp:4 annexa=no	ON only if: a=silencesupp:on AND - remote SDP does not contain the line a=fmtp:18 annexb=no
ON	OFF only if: - a=silencesupp:off	OFF only if: - a=silencesupp:off OR - remote SDP contains the line a=fmtp:4 annexa=no	OFF only if: - a=silencesupp:off OR - remote SDP contains the line a=fmtp:18 annexb=no

5.3.2.12 Fax T.38 and Voice Band Data Support (Bypass Mode)

Previous loads supported T.38 without MEGACO interference, if the device was configured to support T.38:

- FaxTransportType should be configured to T.38 Relay.
- Fax redundancy can be controlled using the configuration parameter FaxRelayRedundancyDepth. This parameter controls only non-V21 packets. For V21 packets that carry important data, the redundancy depth is hard-coded to be 4.
- The fax port is assumed to be the RTP port + 2, both for the local and remote side.

Following these rules, transition to T.38 is performed automatically upon detection.

Bypass (VBD) mode is also supported by using ***ini*** file parameters:

- FaxTransportType should be configured to be Bypass.
- The packetization period is configured by the parameter FaxModemBypassBasicRTPPacketInterval.
- The payload to be used is configured by the parameter FaxBypassPayLoadType and ModemBypassPayloadType.

Starting from this version, support of the Fax type (T.38, Bypass or Transparent) 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 device, 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 differ according to the user's requirements. Attributes to the voice ('a' lines) should be added after the first 'm' line. The 'm=image' line, however, is mandatory, and should appear in the identical format to the above.

The device returns a fully specified line with the local port used for the T.38.

- Fax redundancy can be requested by including the following attribute line after the ‘m=image’ line:

```
a=T38FaxUdpEC:T38UdpRedundancy
```

This parameter is only applicable for non-V21 packets. For V21 packets, the redundancy is hard coded 4.

Two modes of fax support are available. The modes are chosen by the value of bit 2 (value 4) of the MEGACO profiling parameter MGCPCompatibilityProfile. If this bit is not set, the device uses a positive negotiation:

- If the ‘m=image’ line is not received both in local AND in remote descriptors, the device works with the defaults defined in the device. For example, if the device is configured to work with T.38 (default setting) and the ‘m=image’ line is received in the local description only, the device still works with T.38.
- If the fax redundancy attribute line does not appear both in local and remote descriptors, the device uses the default value.

However, if bit 2 is set, the negotiation rules are as follows:

- If the 'm=image' line is not received both in local AND remote descriptors, T.38 is NOT used. In this case, if the local SDP "m=audio" line contains the G.711 coder, the fax and modem mode is Bypass (VBD), and the G.711 payload type is used for it. If the G.711 coder is not offered in the local SDP, the Fax and modem Transport Type is Transparent. Note, however, that the Fax and modem tones are not detected from the network side. This means that the gateway can work in VBD mode only when it is the answering side of the call.
- If the fax redundancy attribute line does not appear both in local and remote descriptors, redundancy for non-V21 packets is NOT used.

5.3.2.13 Reporting Fax Events

Some of the Fax events can be reported using the packages from H.248.2: "CTYP" and "IPFAX". The only Fax event reported by the "CTYP" package is the "V21flag", using the "ctyp/dtone" event.

The reported Fax states are "CONNECTED" and "EOF". "CONNECTED" is reported when the MEGACO application gets "EVENT_DETECT_FAX" from the board. "EOF" is reported when the MEGACO application gets "EVENT_END_FAX" from the board.

The number of Fax pages is reported in the statistics descriptor when this descriptor is requested. The number of Fax pages can also be audited during the Fax.

5.3.3 SDP Support in MEGACO

MEGACO supports basic SDP, as defined in RFC 2327. It also supports SDP-ATM, as defined in RFC 3108. The SDP parser can receive all lines defined in the RFC, but it will ignore all but the following lines: ‘v’, ‘c’, ‘m’, ‘a’.

In addition to the above four lines, the outgoing SDP can contain the ‘t’ ‘s’ ‘o’ lines, which are mandatory in some non-MEGACO applications. This option is controlled by the ***ini*** file parameter MGCPCompatibilityProfile, by adding the number 8 to the current value (2). For the SDP to have these lines, set the ***ini*** file parameter to 10.

In the ‘a’ line, the only supported attributes in SDP 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=rtpmap: 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)
- **FMTP**
  Defines the dynamic payload mapping for the session. For example for where 97 is the payload number to be used and the bitrate is a G.723 coder parameter, the following line should be used:

  a=fmtp: 97 bitrate=5.3

  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 is on (yes or no)
- **PTIME**
  Defines the packetization time for the session. For example for setting packetization time to 20 msec, the following line should be used:

  a=ptime: 20

5.3.3.1 Mapping Payload Numbers to Coders

Table 5-17 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 device defines which coder is supported on this device.

Table 5-17: MEGACO Mapping Payload Numbers to Coders (continues on page 82 to page 84)

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

Table 5-17: MEGACO Mapping Payload Numbers to Coders
(continues on page 82 to page 84)

Default Payload Number	Encoding Name	Coder
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
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

Table 5-17: MEGACO Mapping Payload Numbers to Coders (continues on page 82 to page 84)

Default Payload Number	Encoding Name	Coder
104	"RED"	Redundancy per RFC 2198
13	"CN"	Comfort Noise

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

5.3.4 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.letf.org/rfc/, 'RFC Index'.), according to the device type. For example, the Analog Line package is supported only for analog devices.

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

5.3.4.1 Generic Media Package - G

Table 5-18: 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 5-17, "MEGACO Mapping Payload Numbers to Coders (continues on page 82 to page 84)," on page 82 to Table 5-42, "MEGACO Endpoint Names (continues on page 95 to page 96)," on page 95)**:**

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 superseded by a new signal.

BR signal: The Brief signal event has a short, known duration.

Duration: Specifies the duration of TO signals.

5.3.4.2 Base Root Package - ROOT

Table 5-19: Base Root Package - ROOT

Symbol	Definition	Type
maxNumberOfContexts	Maximum number of Contexts in the device	Property
maxTerminationsPerContext	Maximum 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

5.3.4.3 Tone Generator Package - ToneGen

Table 5-20: Tone Generator Package - ToneGen

Symbol	Definition	Type	S	Duration
pt	Plays audio tone	Signal	TO	

5.3.4.4 Tone Detection Package - ToneDet

Table 5-21: Tone Detection Package - ToneDet

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

5.3.4.5 DTMF Generator Package - DG (Extends ToneGen)

Table 5-22: DTMF Generator Package - DG (continues on page 85 to page 86)

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	

Table 5-22: DTMF Generator Package - DG (continues on page 85 to page 86)

Symbol	Definition	Type	S	Duration
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	

5.3.4.6 DTMF Detection Package - DD (Extends ToneDet)

Table 5-23: DTMF Detection Package - DD

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

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

Table 5-24: 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	BR	1sec	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

5.3.4.8 Call Progress Tones Detection Package - CD (Extends ToneDet)

Table 5-25: 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

5.3.4.9 Basic Continuity Package - CT

Table 5-26: 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	User Defined CO1
rsp	Responds to continuity test	Signal	TO	2 sec	

5.3.4.10 Network Package - NT

Table 5-27: Network Package - NT

Symbol	Definition	Type
jit	Maximal jitter buffer size	Property
netfail	Network failure	Event
qualert	Quality alert - Not supported	Event
dur	Termination's InContext duration	Statistics
os	Octets sent	Statistics
or	Octets received	Statistics

5.3.4.11 RTP Package - RTP (Extends - NT)

Table 5-28: RTP Package - RTP

Symbol	Definition	Type
pltrans	PayLoad Transition - Not supported	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

5.3.4.12 TDM Circuit Package - TDMC (Extends - NT)

Table 5-29: TDM Circuit Package - TDMC

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

5.3.4.13 Generic Announcement Package - AN

Table 5-30: Generic Announcement Package

Symbol	Definition	Type	Supported Parameters
apf	Initiates the play of a fixed announcement	Signal	An - Announcement number Di - The direction of the announcement Noc - Number of cycles
apv	Initiates the play of a variable announcement	Signal	Handled in the same manner as apf

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

Table 5-31: 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

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

Table 5-32: 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

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

Table 5-33: 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

5.3.4.17 Basic CAS Package - CAS

Table 5-34: Basic CAS Signal/Events

Symbol	Definition	Type	S	Duration	Map to CPT File	Symbol
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	

5.3.4.18 R2 CAS Package - R2

Table 5-35: 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	

5.3.4.19 MF Generator Package - MFG (Extends - ToneGen)

Table 5-36: MF Generator Package - MFG (continues on page 90 to page 91)

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	
mfa	MF A	Signal	BR	
mfb	MF B	Signal	BR	
mfc	MF C	Signal	BR	
mfd	MF D	Signal	BR	

Table 5-36: MF Generator Package - MFG (continues on page 90 to page 91)

Symbol	Definition	Type	S	Duration
mfe	MF E	Signal	BR	
mff	MF F	Signal	BR	
mfg	MF G	Signal	BR	
mfh	MF H	Signal	BR	

5.3.4.20 MF Detection Package - MFD (Extends - ToneDet)

Table 5-37: 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

5.3.4.21 Inactivity Timer Package - IT

Table 5-38: Inactivity Timer Package - IT

Symbol	Definition	Type
ito	Detects that inactivity timer has expired	Event

5.3.4.22 Basic Call Progress Tones Generator with Directionality Package - BCG (Extends ToneGen)

Table 5-39: Basic Call Progress Tones Generator with Directionality Package - BCG

Symbol	Definition	Type	S	Duration	Map to DPT File
bdt	Dial tone	Signal	TO	180 sec	1
brt	Ringing tone	Signal	TO	180 sec	2
bbt	Busy tone	Signal	TO	180 sec	3
bct	Congestion tone	Signal	TO	180 sec	4
bsit	Special Information tone	Signal	BR	2 sec	5
bwt	Warning tone	Signal	BR	1 sec	6
bpt	Payphone Recognition tone	Signal			Not supported
bcw	Call Waiting tone	Signal	BR	1 sec	9
bcr	Caller Waiting tone	Signal	TO	180 sec	15
bpy	Pay tone	Signal	TO	180 sec	Not supported

5.3.4.23 Call Type Discrimination Package - CTYP

Table 5-40: Call Type Discrimination Package - CTYP

Symbol	Definition	Type
Dtone	Discriminating tone detected	Event

5.3.4.24 IP Fax Package - IPFAX

Table 5-41: IP Fax Package - IPFAX

Symbol	Definition	Type
faxconnchange	Fax connection state changed (Only 'Connected' and 'EOF' are supported)	Event
pagestrans	Number of pages transferred	Statistics

5.3.5 MEGACO Profiling

Profiling of various MEGACO features is controlled via the ***ini*** file parameter MGCPCompatibilityProfile. Initially, only value **2** has been supported. (Values **0** and **1** are obsolete). **2** is the default value, and it is for supporting MEGACO version 1. Additional features are:

- **Bit 2 (Value 4)** -Controls the type of support for the Fax T.38 negotiation. (Refer to Section 5.3.2.12 on page 80.)
- **Bit 3 (Value 8)** - Enables the extra lines in the outgoing SDP ('t' 's' 'o' lines). (Refer to Section 5.3.3 on page 81.)
- **Bit 4 (Value 16)** - Enables the following features:
  - In the serviceChange request, the Timestamp parameter is omitted.
  - The audit command on ROOT termination with packages descriptor returns the total supported packages for the device.
  - The default packetization period (ptime) for the transparent coder is 10 milliseconds. Using the SDP attribute ptime can change this.
  - The packetization period for Bypass Fax mode is the same as the packetization period used for voice. If this bit is not set, the packetization period for the Fax Bypass is taken from the ***ini*** file.
  - When sending a notification transaction request, the device does not mark it as optional.

5.3.6 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. From version 4.4, the terminations names are defined by a new set of pattern parameters, as described in the next section. Backward Compatibility is kept for the previous Terminations. (Refer to Section 5.3.6.3 on page 94.),

5.3.6.1 Termination Name Patterns

Each termination type name is defined by an ***ini*** file or SNMP parameter. The pattern may contain acceptable characters as defined in MEGACO. The '*' character is used to represent the place where a digit should be. Therefore, it can not be part of the name itself. All other characters, including slash, are considered text.

For example: The pattern "gws*c*" matches the termination name "gws0c1" and also "gws10c20". The trunk numbers, in this case, are 0 and 10 and the channels are 1 and 20.

The possible patterns are:

- **PHYSTERMNAMEPATTERN -** Pattern of the physical terminations.
- **LOGICALRTPTERMPATTERN -** Pattern for ephemeral terminations based on RTP stream.

The starting number of each level can be controlled by a set of parameters:

- **EP_NUM -** Controls the numbering of the physical terminations name pattern. (All but MP)
- **EP_NUM_0** - Defines the starting termination number
- **EP_NUM_1** - Defines the starting channel number
- **RTP_NUM - Defines the starting number for the RTP terminations. (The default is 0)**

5.3.6.2 Old Termination Naming Method

Physical Termination names have up to three components: Gateway (in this case the Device) name, Trunk name and Endpoint name (for non-trunking gateways, the trunk field does not 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 device 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.

Note also that for trunking gateways, the 'TrunkName' can NOT be null. The default values for the Termination name parts depend on the device type:

- **For Analog boards** - (such as those used in AudioCodes' analog Media Gateways, e.g., MP-104, MP-108, and MP-124), the default gateway name is Null, the trunk name is not used, and the endpoint name is 'line/'. Therefore, the first line on an analog board when using defaults is 'line/1'.
- **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 B-channel 1 of trunk 0 will be 'tgw/s0/c1'.

PSTN Interface - mapping Trunk/B-channel pairs to Endpoints is hardware-specific (refer to Table 5-42.) Note that the number of supported terminations per device is equal to the channel density of the device.

5.3.6.3 Backward Compatibility

The connection between the old naming parameters and the new ones is done by creating the name pattern from the old name parameters. Lets assume for example that the old name parameters are:

GATEWAYNAME = "gw"

TRUNKNAME = "s"

ENDPOINTNAME = "chan"

This is equivalent to the following new name parameters:

PHYSTERMNAMEPATTERN = "gws*chan*"

LOGICALRTPTERMPATTERN = "gwrtp/*"

5.3.6.4 Terminations Mapping to a PSTN Interface

Table 5-42 describes the mapping between Endpoints and channels for a PSTN interface, according to the interface type used. Table 5-42 assumes the following initial values:

Channel density - 60

Gateway name = 'Acgw/'

Trunk name = 'T'

Endpoint name = '/C',

Table 5-42 lists only the names for a two trunks unit.

Table 5-42: MEGACO Endpoint Names (continues on page 95 to page 96)

Endpoint Name	E1 - PRI/CAS E1 - Transparent	E1 - Transparent 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
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

Table 5-42: MEGACO Endpoint Names (continues on page 95 to page 96)

Endpoint Name	E1 - PRI/CAS E1 - Transparent	E1 - Transparent 62	T1 - PRI	T1 - CAS T1 - Transparent
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
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			

6 Device Management

Two types of device managements are detailed in this section:

- **SNMP-Based Client Program** - Refer to Section 6.1 below
- **AudioCodes Web interface** - Refer to Section 6.3 on page 110

6.1 Developing an SNMP-Based Client Program to Manage Your Device/s

Simple Network Management Protocol (SNMP) is a standard network-based control protocol used to manage elements in a network. The SNMP Manager (usually implemented by a Network Manager (NM) or an Element Manager (EM)) connects to an SNMP Agent (embedded on a remote Network Element (NE)) to perform network element Operation, Administration and Maintenance (OAM).

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

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

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

The Mediant 2000 contains an embedded SNMP Agent supporting both general network MIBs (such as the IP MIB), VoP-specific MIBs (such as RTP) and AudioCodes' proprietary MIBs (AcBoard, acGateway, AC-ALARM-MIB and other MIBs) enabling a deeper probe into the inter-working of the Gateway. All supported MIB files are supplied to Customers as part of the release.

6.1.1 About SNMP

6.1.1.1 SNMP Message Standard

Four types of SNMP messages are defined:

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

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

- **Get Request** - Specific values can be fetched via the "get" request to determine the performance and state of the device. Typically, many different values and parameters can be determined via SNMP without the overhead associated with logging into the device, or establishing a TCP connection with the device.
- **Get Next Request** - Enables the SNMP standard network managers to "walk" through all SNMP values of a device (via the "get-next" request) to determine all names and values that 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.
- **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 or alert the manager of a problem with the device. This typically requires each device on the network to be configured to issue SNMP traps to one or more network devices that are awaiting these traps.

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

6.1.1.2 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 four main 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. For "Discrete" objects, this instance number is zero. For "Table" objects, this instance number is the index into the SNMP table. SNMP tables are special types of SNMP objects, which allow parallel arrays of information to be supported. Tables are distinguished from scalar objects, such that tables can grow without bounds. For example, SNMP defines the "ifDescr" object (as a standard SNMP object) that indicates the text description of each interface supported by a particular device. Since network devices can be configured with more than one interface, this object can only be represented as an array.

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

6.1.1.3 SNMP Extensibility Feature

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

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

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

6.1.2 Carrier Grade Alarm System

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

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

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

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

6.1.2.1 Active Alarm Table

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

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

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

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

6.1.2.2 Alarm History

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

- acAlarmHistoryTable in the enterprise AcAlarm
- nlmLogTable and nlmLogVariableTable in the standard NOTIFICATION-LOG-MIB

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

6.1.3 Cold Start Trap

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

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

6.1.4 Performance Measurements for a Third-Party Performance Monitoring System

Performance Measurements are available for a Third-Party Performance Monitoring System through an SNMP interface and can be polled at scheduled intervals by an external poller or utility in the management server or other off board system.

The media server provides performance measurements in the form of two types:

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

The media server performance measurements is provided by two AudioCodes MIBs (acPerfMediaGateway and acPerfMediaServices). The first MIB is generic-type of PM MIB that is available on all AudioCodes devices. The second PM MIB is media server specific.

The generic PM MIB covers:

- Control protocol
- RTP stream
- System packets statistics

This AudioCodes PM enterprise MIB supports statistics which apply to the media server and other AudioCodes products like gateways.

6.1.5 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 according to 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 contains objects relevant to E1/T1 Trunk interfaces.
- **NOTIFICATION-LOG-MIB** - This standard MIB (RFC 3014 - iso.org.dod.internet.mgmt.mib-2) is supported as part of AudioCodes' implementation of Carrier Grade Alarms.
- **ALARM-MIB** - This is an IETF proposed MIB also supported as part of AudioCodess' implementation of Carrier Grade Alarms. This MIB is still not standard and therefore is under the audioCodes.acExperimental branch.
- **SNMP-TARGET-MIB** - This MIB is partially supported, (RFC 2273). It allows for configuration of trap destinations and trusted managers only.
- **SNMP Research International enterprise MIBs** - AudioCodes' boards support two SNMP Research International MIBs: SR-COMMUNITY-MIB and TGT-ADDRESS-MASK-MIB. These MIBs are used in configuration of SNMPv2c community strings and trusted managers.

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

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

Note: The AcBoard MIB is in state of deprecation. It is still supported but it is being replaced by five newer AudioCodes proprietary MIBs.

The AcBoard MIB has the following Groups:

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

As noted above, five new MIBs cover the general parameters in the board.

They each contain a Configuration subtree, for configuring the related parameters. In some there also are Status and Action subtrees.

The 5 MIBs are:

- **AC-ANALOG-MIB**
- **AC-CONTROL-MIB**
- **AC-MEDIA-MIB**
- **AC-PSTN-MIB**
- **AC-SYSTEM-MIB**

Other AudioCodes proprietary MIBs are:

- **acGateway MIB -** This proprietary MIB contains objects related to configuration of the board when applied as a SIP or H.323 gateway only. This MIB complements the other AudioCodes proprietary MIBs.

  The acGateway MIB has the following groups:

  - **Common** - for parameters common to both SIP and H.323.
  - **SIP** - for SIP only parameters.
  - **H.323** - for H.323 only parameters.

- **AcAtm -** This proprietary MIB contains objects related to configuration and status of the board when applied as an ATM gateway only. This MIB complements the other AudioCodes proprietary MIBs.

  The acAtm MIB has the following groups:

  - **acAtmConfiguration-** for configuring ATM related parameters.
  - **acAtmStatus** - for the Status of ATM connections.

- **AcAlarm -** This is AudioCodes' proprietary carrier grade alarm MIB. It is a simpler implementation of the notificationLogMIB and the IETF suggested alarmMIB (both also supported in all AudioCodes boards).

  The acAlarm MIB has the following groups:

  - **ActiveAlarm** - straight forward (single indexed) table listing all currently active Alarms together with their bindings (the AudioCodes Alarm bindings are defined in acAlarm. acAlarmVarbinds and also in acBoard.acTrap. acBoardTrapDefinitions. oid_1_3_6_1_4_1_5003_9_10_1_21_2_0).
  - **acAlarmHistory** - straight forward (single indexed) table listing all recently raised Alarms together with their bindings (the AudioCodes Alarm bindings are defined in acAlarm. acAlarmVarbinds and also in acBoard.acTrap. acBoardTrapDefinitions. oid_1_3_6_1_4_1_5003_9_10_1_21_2_0).

  The table size can be altered via notificationLogMIB.notificationLogMIBObjects.nlmConfig.nlmConfigGlobalEntryLimit or notificationLogMIB.notificationLogMIBObjects.nlmConfig.nlmConfigLogTable.nlmConfigLogEntry.nlmConfigLogEntryLimit.

  The table size can be an value between 50 to 1000 and is 500 units by default.

- **Traps**

  Full proprietary trap definitions and trap Varbinds are found in AcBoard MIB and AcAlarm MIB.

  The following proprietary traps are supported in Mediant 2000:

  - **acBoardEvResettingBoard** - Sent after a board is reset.
  - **acBoardEvBoardstarted** - Sent after a board is successfully restored and initialized following reset.

- **acBoardConfigurationError** - Sent when a board's settings are illegal - the trap contains a message stating/detailing/explaining the illegality of the setting.
- **acBoardFatalError** - Sent whenever a fatal board error occurs.
- **acFeatureKeyError** - Development pending. Intended to relay Feature Key errors etc.
- **acgwAdminStateChange** - Sent when Graceful Shutdown commences and ends.
- **acBoardControllerFailureAlarm** - The GK/Proxy is not found or registration failed. internal routing table may be used for routing.
- **acBoardEthernetLinkAlarm** - Ethernet Link or links are down.
- **acActiveAlarmTableOverflow** - An active alarm could not be placed in the active alarm table because the table is full.
- **acAtmPortAlarm** - ATM Port Alarm.
- **acAudioProvisioningAlarm** - Raised if the Media Server is unable to provision its audio.

In addition to the listed traps the Board also supports the following standard traps:

- **dsx1LineStatusChange**
- **coldStart**
- **authenticationFailure**

Note 1: The following are special notes pertaining to MIBs:

- A detailed explanation of each parameter can be viewed in an SNMP browser in the MIB Description field.
- Not all groups in the MIB are functional. Refer to version release notes.
- Certain parameters are non-functional. Their MIB status is marked 'obsolete'.
- When a parameter is SET to a new value via SNMP, the change may affect board functionality immediately or may require that the board be soft reset for the change to take effect. This depends on the parameter type.

Note 2: The current (updated) board configuration parameters are programmed into the board provided that the user does not load an ***ini*** file to the board after reset. Loading an ***ini*** file after reset overrides the updated parameters.

Additional MIBs are to be supported in future releases.

6.1.6 SNMP Interface Details

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

Currently, both SNMP and ***ini*** file commands and downloads are not encrypted. For ***ini*** file encryption, refer to Section E.2.6 on page 246.

6.1.6.1 SNMP Community Names

By default, the board uses a single, read-only community string of "public" and a single read-write community string of "private".

One can configure up to 5 read-only community strings and up to 5 read-write community strings, and a single trap community string is supported:

6.1.6.1.1 Configuration of Community Strings via the *ini* File

SNMPREADONLYCOMMUNITYSTRING_<x> = '#######'

SNMPREADWRITECOMMUNITYSTRING_<x> = '#######'

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

6.1.6.1.2 Configuration of Community Strings via SNMP

To configure read-only and read-write community strings, the EM must use the srCommunityMIB. To configure the trap community string, the EM must also use the snmpVacmMIB and the snmpTargetMIB.

➢ To add a read-only community string, v2user, take this step:

- Add a new row to the srCommunityTable with CommunityName v2user and GroupName ReadGroup.

➢ To delete the read-only community string, v2user, take these 2 steps:

Step 1. If v2user is being used as the trap community string, follow the procedure for changing the trap community string. (See below.)

Step 2. Delete the srCommunityTable row with CommunityName v2user.

➢ To add a read-write community string, v2admin, take this step:

- Add a new row to the srCommunityTable with CommunityName of v2admin and GroupName ReadWriteGroup.

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

Step 1. If v2admin is being used as the trap community string, follow the procedure for changing the trap community string. (See below.)

Step 2. Delete the srCommunityTable row with a CommunityName of v2admin and GroupName of ReadWriteGroup.

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

Step 1. Follow the procedure above to add a read-write community string to a row for v2mgr.

Step 2. Set up the EM such that subsequent set requests use the new community string, v2mgr.

Step 3. If v2admin is being used as the trap community string, follow the procedure to change the trap community string. (See below.)

Step 4. Follow the procedure above to delete a read-write community name in the row for v2admin.

➢ To change the trap community string, take these 2 steps:

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

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

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

Step 2. Modify the SecurityName field in the sole row of the snmpTargetParamsTable.

6.1.6.2 Trusted Managers

By default, the agent accepts get and set requests from any IP address, as long as the correct community string is used in the request. Security can be enhanced via the use of Trusted Managers. A Trusted Manager is an IP address from which the SNMP agent will accept and process get and set requests. An EM can be used to configure up to 5 Trusted Managers.

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

6.1.6.2.1 Configuration of Trusted Managers via *ini* File

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

SNMPTRUSTEDMGR_X = D.D.D.D

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

6.1.6.2.2 Configuration of Trusted Managers via SNMP

To configure Trusted Managers, the EM must use the srCommunityMIB, the snmpTargetMIB and the TGT-ADDRESS-MASK-MIB.

➢ To add the first Trusted Manager, take these 3 steps:

The following procedure assumes that there is at least one configured read-write community. There are currently no Trusted Managers. The taglist for columns for all srCommunityTable rows are currently empty.

Step 1. Add a row to the snmpTargetAddrTable with these values: Name=mgr0, TagList=MGR, Params=v2cparams.

Step 2. Add a row to the tgtAddressMaskTable table with these values: Name=mgr0, tgtAddressMask=255.255.255.255:0. The agent will not allow creation of a row in this table unless a corresponding row exists in the snmpTargetAddrTable.

Step 3. Set the value of the TransportLabel field on each non-TrapGroup row in the srCommunityTable to MGR.

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

The following procedure assumes that there is at least one configured read-write community. There are currently one or more Trusted Managers. The taglist for columns for all rows in the srCommunityTable are currently set to MGR. This procedure must be done from one of the existing Trusted Managers.

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

Step 2. Add a row to the tgtAddressMaskTable table with these values: Name=mgrN, tgtAddressMask=255.255.255.255:0.

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

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

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

- Remove the appropriate row from the snmpTargetAddrTable.

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

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

The following procedure assumes that there is at least one configured read-write community. There is currently only one Trusted Manager. The taglist for columns for all rows in the srCommunityTable are currently set to MGR. This procedure must be done from thefinal Trusted Manager.

Step 1. Set the value of the TransportLabel field on each row in the srCommunityTable to the empty string.

Step 2. Remove the appropriate row from the snmpTargetAddrTable

The change takes affect immediately. All managers can now access the board.

6.1.6.3 SNMP Ports

The SNMP Request Port is 161 and Trap Port is 162

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

SNMPPort = <port_number>
Valid UDP port number; default = 161

This parameter specifies the port number for SNMP requests and responses.

Usually it should not be specified. Use the default.

6.1.6.4 Multiple SNMP Trap Destinations

An agent can now send traps to up to five managers. For each manager the user needs to set the following parameters which are defined in the snmpManagersTable in the acBoardMIB:

1. snmpTrapManagerSending
2. snmpManagerIsUsed
3. snmpManagerTrapPort
4. snmpManagerIP

When snmpManagerIsUsed is set to zero (not used) the other three parameters are set to zero. (The intent is to have them set to the default value which means TrapPort to be set to 162. This will occur in a later release.)

- **snmpManagerIsUsed Default = Disable(0)**

  The allowed values are 0 (disable or no) and 1 (enable or yes).

- **snmpManagerIp Default = 0.0.0.0**

  This is known as SNMPMANAGERTABLEIP in the ***ini*** file and is the IP address of the manager.

- **snmpManagerTrapPort Default = 162**

  The valid port range for this is 100-4000.

- **snmpManagerTrapSendingEnable Default = Enable(1)**

  The allowed values are 0 (disable) and 1 (enable).

Note 1: Each of these MIB objects is independent and can be set regardless of the state of snmpManagerIsUsed.

Note 2: If the IsUsed parameter is set to 1, then the IP address for that row should be supplied in the same SNMP PDU.

6.1.6.4.1 Configuration via the *ini* File

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

SNMPMANAGERTRAPSENDINGENABLE_<x> = 0 or 1 indicates if traps are to be sent to the specified SNMP trap manager. A value of ‘1’ means that it is enabled, while a value of ‘0’ means disabled.

Where <x> = a number 0, 1, 2 and is the array element index. Currently up to 5 SNMP trap managers can be supported.

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

```
; SNMP trap destinations

; The board maintains a table of trap destinations containing 5
;rows. The rows are numbered 0..4. Each block of 4 items below
;apply to a row in the table.
```

```
; To configure one of the rows, uncomment all 4 lines in that
;block. Supply an IP address and if necessary, change the port
;numer.

; To delete a trap destination, set ISUSED to 0.

; -change these entries as needed

;SNMPMANAGERTABLEIP_0=

;SNMPMANAGERTRAPPORT_0=162

;SNMPMANAGERISUSED_0=1

;SNMPMANAGERTRAPSENDINGENABLE_0=1

;

;SNMPMANAGERTABLEIP_1=

;SNMPMANAGERTRAPPORT_1=162

;SNMPMANAGERISUSED_1=1

;SNMPMANAGERTRAPSENDINGENABLE_1=1

;

;SNMPMANAGERTABLEIP_2=

;SNMPMANAGERTRAPPORT_2=162

;SNMPMANAGERISUSED_2=1

;SNMPMANAGERTRAPSENDINGENABLE_2=1

;

;SNMPMANAGERTABLEIP_3=

;SNMPMANAGERTRAPPORT_3=162

;SNMPMANAGERISUSED_3=1

;SNMPMANAGERTRAPSENDINGENABLE_3=1

;

;SNMPMANAGERTABLEIP_4=

;SNMPMANAGERTRAPPORT_4=162

;SNMPMANAGERISUSED_4=1

;SNMPMANAGERTRAPSENDINGENABLE_4=1
```

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

6.1.6.4.2 Configuration via SNMP

To configure trap destinations, the EM must use the snmpTargetMIB.

Up to 5 trap destinations can be configured.

➢ To add a trap destination, take this step:

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

All changes to the trap destination configuration take effect immediately.

➢ To delete a trap destination, take this step:

- Remove the appropriate row from the snmpTargetAddrTable.

➢ To modify a trap destination, take this step:

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

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

➢ To disable a trap destination, take this step:

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

➢ To enable a trap destination, take this step:

- Change TagList on the appropriate row in the snmpTargetAddrTable to "AC_TRAP".

6.1.6.5 SNMP Manager Backward Compatibility

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

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

6.2 Administrative State Control

6.2.1 Node Maintenance

Node maintenance for the AudioCodes Media Server is provided via an SNMP interface. The acBoardMIB provides two parameters for graceful and forced shutdowns of the AudioCodes Media Server. (Refer to the note is Section 6.2.2 below.) These parameters are in the acBoardMIB as acgwAdminState and acgwAdminStateLockControl.

The acgwAdminState is used either to request (set) a shutdown (0), undo shutdown (2), or to view (get) the gateway condition (0 = locked, 1 = shutting down, 2 = unlocked).

The acgwAdminStateLockControl is used to set a time limit for the shutdown (in seconds) where 0 means shutdown immediately (forced), -1 means no time limit (graceful) and x where x>0 indicates a time limit in seconds (timed limit is considered a graceful shutdown).

The acgwAdminStateLockControl should be set first followed by the acgwAdminState.

6.2.2 Graceful Shutdown

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

Note: Graceful shutdown is currently supported in MEGACO (H.248) only.

The possible values received on a get request are:

- **locked(0)** - The board is locked
- **shuttingDown(1)** - The board is in the process of performing a graceful lock
- **unlocked(2)** - The board is unlocked

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

When the board changes to either shuttingDown or locked state, an adminStateChange alarm is raised. When the board changes to an unlocked state, the adminStateChange alarm is cleared.

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

- **1 -** Perform a graceful lock. Calls are allowed to complete. No new calls are allowed to be originated on this device.
- **0 -** Perform a force lock. Calls are immediately terminated.
- **Any number greater than 0 -** Time in seconds before the graceful lock turns into a force lock.

6.3 AudioCodes' Embedded Web Server

AudioCodes' boards and modules contain an Embedded Web Server to be used for device configuration and for run-time monitoring. AudioCodes' Embedded Web Server enables users equipped with any standard Web-browsing application such as Microsoft™ Internet Explorer™ (Ver. 5.0 and higher) or Netscape™ Navigator™ (Ver. 7.0 and higher) to:

- **Provision devices** (refer to Section 6.4.4 on page 119)
- **Verify configuration changes in the Status screens** (refer to Section 6.4.5 on page 130)
- **Load the *ini* file** (refer to Section 6.4.6.1 on page 136)
- **Load the CMP, Voice Prompt, Prerecorded Tones, CPT and CAS Files** (refer to Section 6.4.6.2 on page 140)

6.3.1 Embedded Web Server Protection & Security Mechanisms

Access to the Embedded Web Server is controlled by the following protection and security mechanisms:

- **Username and Password** - Refer to Section 6.3.1.1 below
- **Limiting the Web Server GUI to Read-Only Mode** - Refer to Section 6.3.2.1 below
- **Disabling the Web Server GUI** - Refer to Section 6.3.2.2 on page 111

6.3.1.1 Username and Password

A unique Username and Password combination protect the Embedded Web Server. The first time you access the Web GUI, you're prompted to provide a Username and Password to obtain access. Operations subsequent to the first-time access are automatically negotiated (and authenticated) by the browser on behalf of the user so that the user does not have to re-enter Username and Password for each operation.

The default Username and Password for all devices is:

- **Username = Admin**
- **Password = Admin**

For more information about changing the Password and Username or reseting them to the defaults, refer to Section 6.4.4.7 on page 129.

6.3.2 Web Configuration

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

6.3.2.1 Limiting the Embedded Web Server to Read-Only Mode

Users can limit the Web Server to read-only mode by changing the default of ***ini*** file parameter DisableWebConfig. Use the read-only mode feature as a security measure. This security level provides protection against unauthorized access (such as Internet hacker attacks), particularly important to users without a firewall.

➢ To limit the Web Server to read-only mode:

- Set the ***ini*** file parameter DisableWebConfig to **1** (Default = 0, i.e., read-write mode) and send the modified ***ini*** file to the device. All Web pages are presented in read-only mode. The ability to modify configuration data is disabled. In addition, users will NOT have access to any file loading page, to the "Change Password" page, to the "SaveConfiguration", or to the "Reset" page.

6.3.2.2 Disabling the Embedded Web Server

You can deny access to the device's Web Server by changing the default of ***ini*** file parameter DisableWebTask. The ability to disable access to the device's Web Server via HTTP provides a high level of security in which protection against unauthorized access (such as Internet hacker attacks) is included. This is particularly important to users without a firewall.

➢ To disable the Embedded Web Server:

- Set the ***ini*** file parameter DisableWebTask to **1** (Default = 0, i.e., web task enabled). Access to the device's Web Server is denied.

6.3.3 Correlating PC /Mediant 2000 IP Address & Subnet Mask

Before using the Web browser to access the Mediant 2000 device's Embedded Web Server, change the PC's IP address and Subnet Mask to correspond with the Mediant 2000's factory default IP address and Subnet Mask shown in Table 6-1. For details on changing the IP address and Subnet Mask, refer to the Help information provided by the Operating System used.

Table 6-1: Default IP Address and Subnet Mask

E1/T1 Trunks	IP Address	Subnet Mask
Trunks 1-8	10.1.10.10	255.255.0.0
Trunks 9-16	10.1.10.11	255.255.0.0

Note 1: The two IP addresses refer to two different modules residing on the same (TP-1610) board inside the Mediant 2000 (one IP address for the module containing Trunks 1-8 and another for the module containing Trunks 9-16).

Note 2: Note and retain the IP Address and Subnet Mask that you assign to the device. Do the same when defining Username and Password (refer to Section 6.3.1.1 on page 111). If the Embedded Web Server is unavailable (for example, if you have lost your Username and Password), use AudioCodes' BootP/TFTP Server to access the device, "reflash" the files and reset the password. For more information on the BootP/TFTP Server, refer to Appendix A on page 211.

6.3.4 Accessing the Embedded Web Server

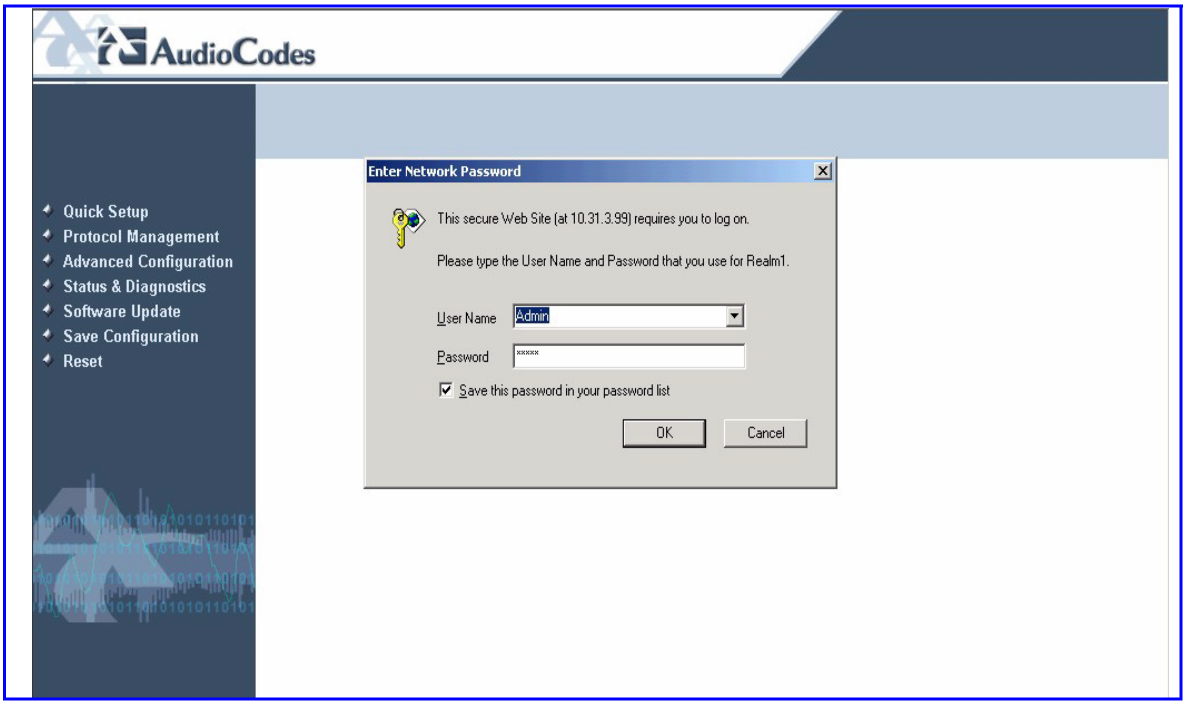
➢ To access the Embedded Web Server, take these 2 steps:

Step 1. Open any standard Web-browser 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 6-1: Enter Network Password Screen



This secure Web Site (at 10.31.3.99) requires you to log on.

Please type the User Name and Password that you use for Realm1.

User Name: Admin

Password: *****

☑ Save this password in your password list

6.3.5 Using Internet Explorer to Access the Embedded Web Server

Internet explorer's security settings may block access to the Gateway's Web browser if they're configured incorrectly. If this happens, the following message appears:

Unauthorized

Correct authorization is required for this area. Either your browser does not perform authorization or your authorization has failed. RomPager server.

➢ To troubleshoot blocked access to Internet Explorer, take these 5 steps:

Step 1. Delete all cookies from the Temporary Internet files folder. If this does not clear up the problem, the security settings may need to be altered. (Continue to Step 2).

Step 2. In Internet Explorer, from the Tools menu, select **Internet Options**. The Internet Options dialog box appears.

Step 3. Select the Security tab, and then, at the bottom of the dialog box, click the **Custom Level** button. The Security Settings dialog box appears.

Step 4. Scroll down until the Logon options are displayed and change the setting to **Prompt for user name and Password**. Then Click **OK**.

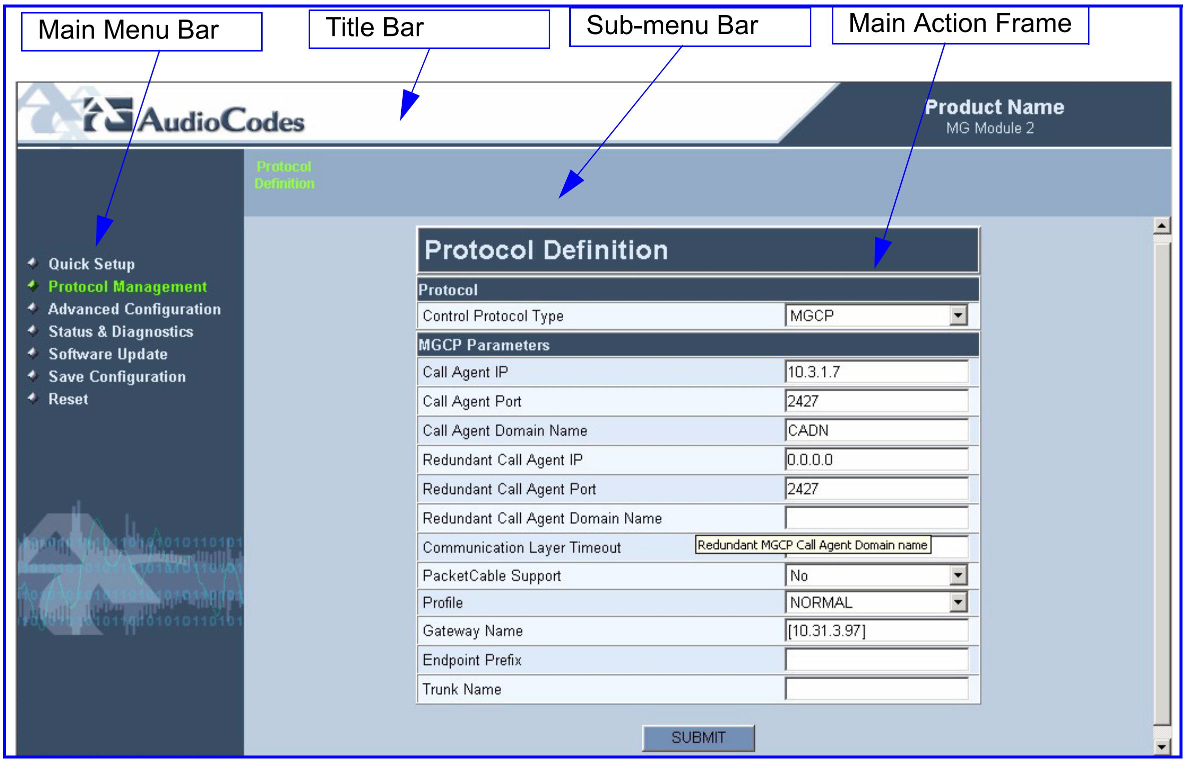
Step 5. Restart the browser. This fixes any issues related to domain use logon policy.

6.4 Getting Acquainted with the Web Interface

6.4.1 About the Web Interface Screen

Figure 6-2 is an example of the General layout of the Web Interface screen.

Figure 6-2: Web Interface Screen - Example



Protocol Definition

Protocol	
Control Protocol Type	MGCP
MGCP Parameters	
Call Agent IP	10.3.1.7
Call Agent Port	2427
Call Agent Domain Name	CADN
Redundant Call Agent IP	0.0.0.0
Redundant Call Agent Port	2427
Redundant Call Agent Domain Name	
Communication Layer Timeout	Redundant MGCP Call Agent Domain name
PacketCable Support	No
Profile	NORMAL
Gateway Name	[10.31.3.97]
Endpoint Prefix	
Trunk Name	

The Web Interface screen contains the following parts:

- **Title bar** - contains the corporate logo, background images and product name
- **Main menu bar** - always appears to the left on every screen for quick access to the other main modules
- **Sub-menu bar -** always appears at the top on every screen and contains links to the sub-menus of the main module selected in the main menu bar to the left
- **Main action frame** - The main area of the screen in which information is viewed and configured

The Web interface is divided into the following 7 modules in the main menu bar to the left:

- **Quick Setup -** Use this module to configure the device's basic settings. (For the full list of configurable parameters go directly to the Protocol Management and Advanced Configuration menus.)
- **Protocol Management -** Use the menus in this module to configure the device's control protocol parameters.
- **Advanced Configuration -** Use the menus in this module to set the device's advanced configuration parameters (for advanced users only).
- **Status & Diagnostics -** Use the menus in this module to view and monitor the device's channels, Syslog messages and hardware / software product information.
- **Software Update -** Use the menus in this module when you want to download new software or configuration files onto the device.

- **Save Configuration -** Use this menu to save configuration changes to the non-volatile flash memory.
- **Reset -** Use this menu to remotely reset the device.

Note: When positioning your curser over a parameter name for more than 1 second, a short description of this parameter is displayed.

6.4.2 Saving Changes

To save changes to the volatile memory (RAM) press the **Submit** button (changes to parameters with on-the-fly capabilities are immediately available, other parameters are updated only after a device reset). Parameters that are only saved to the volatile memory revert to their previous settings after hardware reset (software reset i.e., via the Web Interface offers the option to save the changes to the non-volatile memory prior to the reset). To save changes so they are available after a power fail, you must save the changes to the non-volatile memory (flash). When **Save Configuration** is performed, all parameters and HTTP loaded files are saved to the flash memory.

➢ To save the changes to flash, take the next 2 steps:

Step 1. From the main menu on the left, click the Save Configuration link. The Save Configuration screen appears.

Step 2. Click the **Save Configuration** button in the middle of the screen. A confirmation message appears when the save is complete.

➢ To quickly setup a device, take these 17 steps:

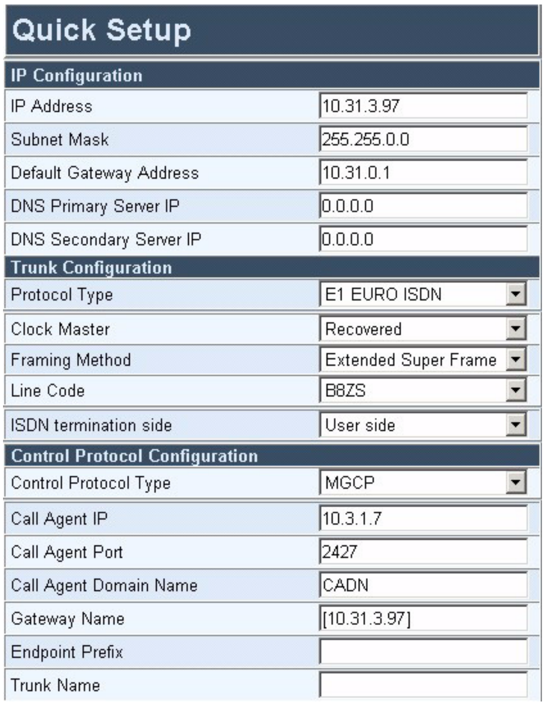
Step 1. Access the Web Server Interface (refer to Section 6.3.4 on page 112.)

Step 2. Enter the **Username** (default: **Admin**) and **Password** (default: **Admin**).

Note: The Username and Password fields are case-sensitive.

Step 3. Click **OK.** The Quick Setup screen appears.

Figure 6-3: Quick Setup Screen



Quick Setup

IP Configuration	
IP Address	10.31.3.97
Subnet Mask	255.255.0.0
Default Gateway Address	10.31.0.1
DNS Primary Server IP	0.0.0.0
DNS Secondary Server IP	0.0.0.0
Trunk Configuration	
Protocol Type	E1 EURO ISDN
Clock Master	Recovered
Framing Method	Extended Super Frame
Line Code	B8ZS
ISDN termination side	User side
Control Protocol Configuration	
Control Protocol Type	MGCP
Call Agent IP	10.3.1.7
Call Agent Port	2427
Call Agent Domain Name	CADN
Gateway Name	[10.31.3.97]
Endpoint Prefix	
Trunk Name	

Step 4. In the Quick Setup screen, enter or modify appropriate information for the IP Configuration, Trunk Configuration and Control Protocol (per type).

Step 5. In the **IP Configuration** section, **IP Address** and **Subnet Mask** fields, enter the appropriate addresses, which must correspond with your network IP Address settings. Refer to Section 6.3.3 on page 112.

Step 6. For the **Default Gateway Address**, **DNS Primary Server IP** and **DNS Secondary Server IP** fields, enter appropriate addresses. (If your network features a DNS server, clarify with your Network Administrator).

Step 7. In the **Trunk Configuration** section, **Protocol Type** drop-down menu, select the appropriate option. In the **Clock Master Type** (the trunk clock source) drop-down menu, select either **Recovered** (the clock is recovered from the trunk; default) or **Generated** (the trunk clock source is provided by the internal/TDM bus clock source. The clock source depends on the parameter 'TDM Bus Clock Source' (**Advanced Configuration > TDM Bus Settings**).

Step 8. For the **Framing Method** dropdown menu, select the appropriate option. (For E1 trunks, always set the Framing Method to **Extended Super Frame**).

Note: The Trunk Configuration parameters are global, and apply to all trunks. To configure trunks per trunk, refer to Section 6.4.4.3 on page 122**.**

Step 9. For the **Line Code** dropdown menu, select either **B8ZS** (bipolar 8-zero substitution) for T1 trunks only; **HDB3** (high-density bipolar 3) for E1 trunks only; or **AMI**, which applies to both T1 and E1.

Step 10. In the **Control Protocol Type** section, for the **Call Agent IP** field, if your network does not feature a DNS server that automatically defines the Call Agent's IP address, enter the appropriate IP address. If you have a DNS server, the field is optional.

Step 11. For the **Call Agent Port** field, enter the appropriate port ID. The default is **2427** for MGCP and **2944** for MEGACO.

Step 12. For the **Call Agent Domain Name** field, when using the DNS server option, enter the Domain Name of the Call Agent operating with the Mediant 2000. The DNS server automatically detects the Call Agent's IP address from the Domain Name.

Step 13. For the **Gateway Name** field, assign a name to the device. (For example: gateway1.com). Ensure that the name you choose is the one that the Call Manager/Agent is configured with to identify your Mediant 2000.

Step 14. If you are working with MGCP, for the **Endpoint Prefix** field, enter an appropriate definition. Ensure that the prefix you choose is the name that the Call Manager/ Agent is configured with to identify your Mediant 2000.

Step 15. For the **Trunk Name** field, enter an appropriate definition. Ensure that the name you choose is the name that the Call Manager/Agent is configured with to identify your Mediant 2000.

Step 16. If you are working with MEGACO, for the **Endpoint Name** field, enter an appropriate definition. Ensure that the name you choose is the name that the Call Manager/Agent is configured with to identify your Mediant 2000.

Step 17. 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. On the TP-1610, the Ready and LAN LEDs are lit green.

6.4.3 Protocol Management

The Protocol Management screen offers access to the Protocol Definition screen in which users 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 link. The Protocol Management screen with the sub-menu bar on the top is displayed.

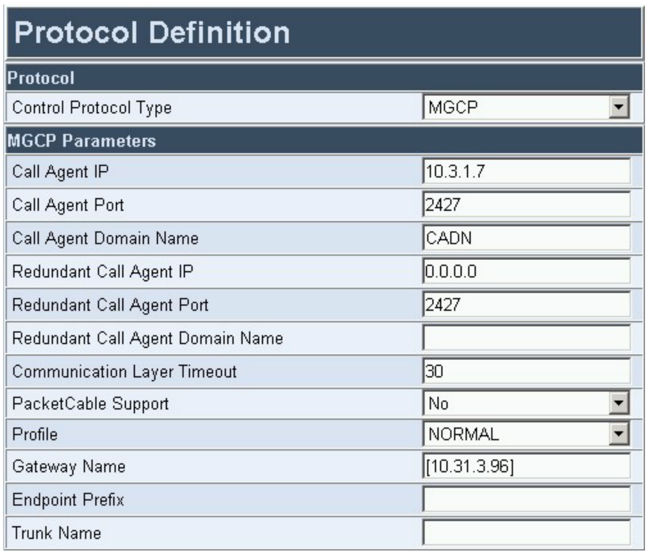
6.4.3.1 Protocol Definition

➢ To define the control protocol type, take these 5 steps:

Step 1. From the main menu list on the left, click on the Protocol Management link. The Protocol Management screen appears.

Step 2. On the sub-menu bar on the top, click the Protocol Definition link. The Protocol Definition screen appears. The first-time you access the device's Embedded Web Server , the displayed parameters are the factory-configured default values.

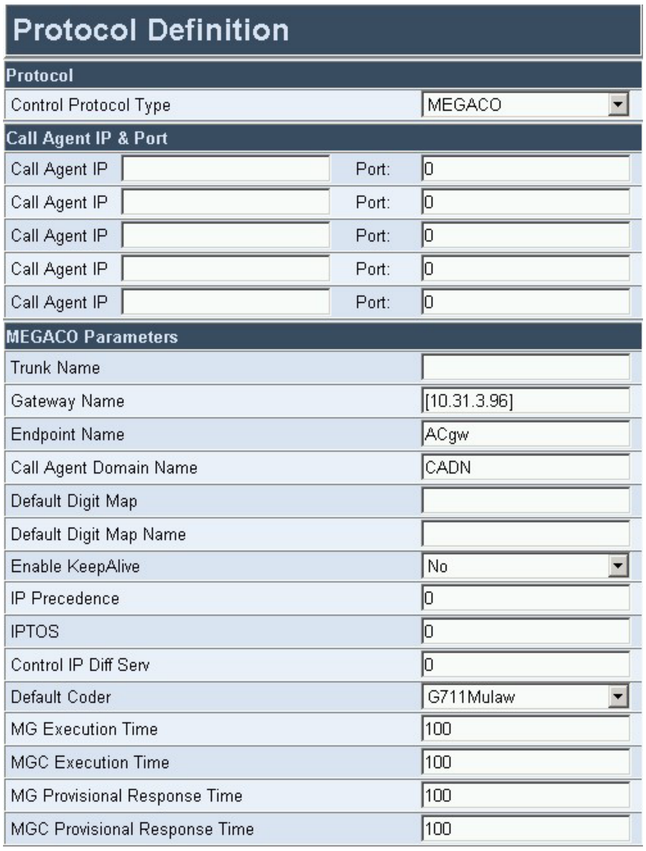
Figure 6-4: MGCP Protocol Definition Screen



Protocol Definition

Protocol	
Control Protocol Type	MGCP ▾
MGCP Parameters	
Call Agent IP	10.3.1.7
Call Agent Port	2427
Call Agent Domain Name	CADN
Redundant Call Agent IP	0.0.0.0
Redundant Call Agent Port	2427
Redundant Call Agent Domain Name	
Communication Layer Timeout	30
PacketCable Support	No ▾
Profile	NORMAL ▾
Gateway Name	[10.31.3.96]
Endpoint Prefix	
Trunk Name	

Figure 6-5: MEGACO Protocol Definition Screen



Protocol Definition

Protocol			
Control Protocol Type			MEGACO ▾
Call Agent IP & Port			
Call Agent IP		Port:	0
Call Agent IP		Port:	0
Call Agent IP		Port:	0
Call Agent IP		Port:	0
Call Agent IP		Port:	0
MEGACO Parameters			
Trunk Name			
Gateway Name			[10.31.3.96]
Endpoint Name			ACgw
Call Agent Domain Name			CADN
Default Digit Map			
Default Digit Map Name			
Enable KeepAlive			No ▾
IP Precedence			0
IPTOS			0
Control IP Diff Serv			0
Default Coder			G711Mulaw ▾
MG Execution Time			100
MGC Execution Time			100
MG Provisional Response Time			100
MGC Provisional Response Time			100

Step 3. From the **Control Protocol Type** dropdown list, select the appropriate option. The parameter fields are modified accordingly.

Step 4. Use Table 7-7 on page 173 and Table 7-8 on page 175 as a reference when configuring/modifying the MGCP or MEGACO parameters in the Protocol Definition screen.

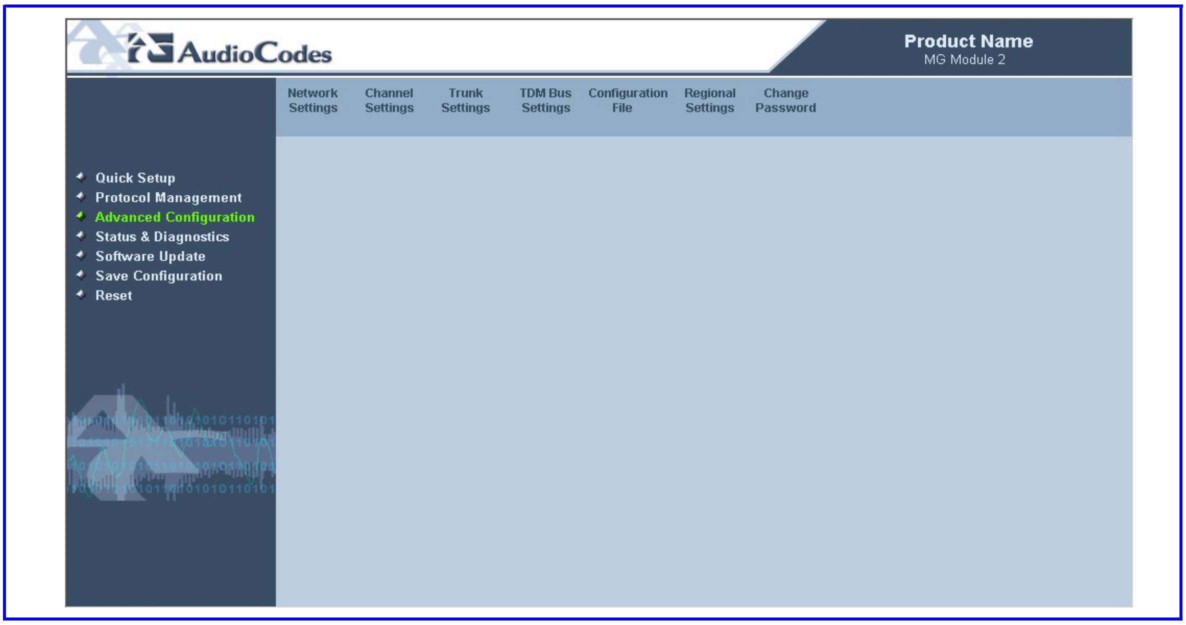
Step 5. After configuring/modifying the parameter fields, click the **SUBMIT** button. The changes are entered into the system and the screen is refreshed.

6.4.4 Advanced Configuration Screen

➢ To access the Advanced Configuration screen take this step:

- To access the device's advanced configuration parameters, from the main menu list on the left, click the Advanced Configuration link. The Advanced Configuration Parameters screen appears with the sub-menu bar on the top displaying the following menu options:
  - Network Settings - Refer to Section 6.4.4.1 on page 119
  - Channel Settings - Refer to Section 6.4.4.2 on page 120
  - Trunk Settings - Refer to Section 6.4.4.3 on page 122
  - TDM Bus Settings - Refer to Section 6.4.4.4 on page 125
  - Configuration File - Refer to Section 6.4.4.5 on page 126
  - Regional Settings - Refer to Section 6.4.4.6 on page 127
  - Change Password - Refer to Section 6.4.4.7 on page 129

Figure 6-6: Advanced Configuration Parameters Screen



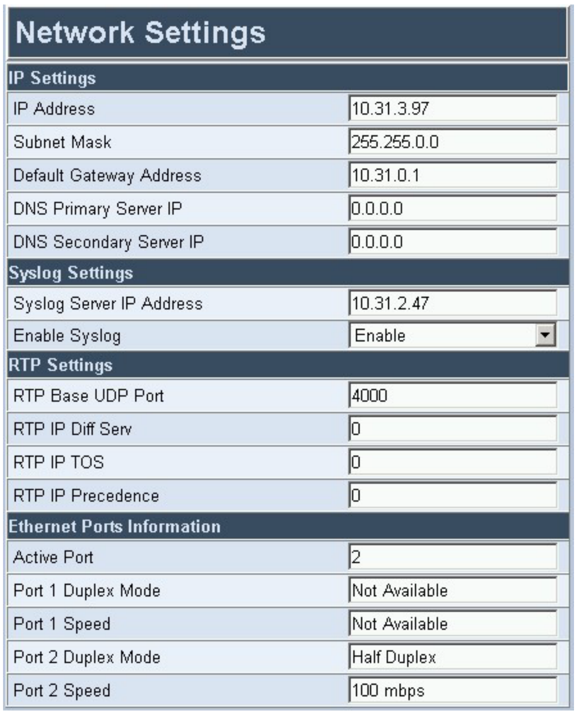
6.4.4.1 Network Settings

➢ To configure the Network Settings, take these 4 steps:

Step 1. From the main menu list on the left, click on the Advanced Configuration link. The Advanced Configuration screen appears.

Step 2. From the sub-menu bar on the top, click the Network Settings link. The Network Settings screen appears.

Figure 6-7: Network Settings Screen



Network Settings

IP Settings	
IP Address	10.31.3.97
Subnet Mask	255.255.0.0
Default Gateway Address	10.31.0.1
DNS Primary Server IP	0.0.0.0
DNS Secondary Server IP	0.0.0.0
Syslog Settings	
Syslog Server IP Address	10.31.2.47
Enable Syslog	Enable
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	Half Duplex
Port 2 Speed	100 mbps

Step 3. Use Table 7-1 on page 146 as a reference when configuring/modifying the network parameter fields in the Network Settings screen.

Step 4. After configuring/modifying the parameter fields, click the **SUBMIT** button. The changes are entered into the system and the screen is refreshed.

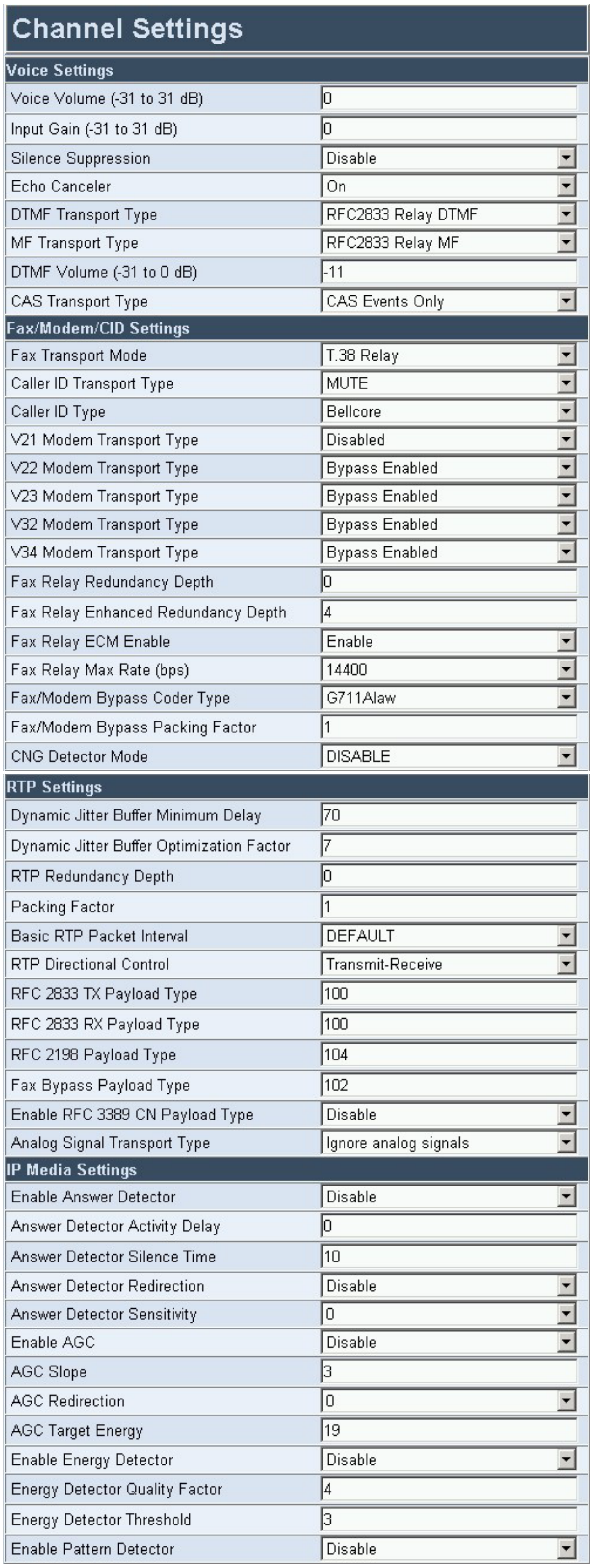
6.4.4.2 Channel Settings

➢ **To configure Channel Settings, take these 4 steps:**

Step 1. From the main menu list on the left, click on the Advanced Configuration link. The Advanced Configuration screen appears.

Step 2. From the sub-menu bar on the top, click the Channel Settings link. The Channel Settings screen appears.

Figure 6-8: Channel Settings Screen



Channel Settings

Voice Settings

Voice Volume (-31 to 31 dB)	0
Input Gain (-31 to 31 dB)	0
Silence Suppression	Disable
Echo Canceler	On
DTMF Transport Type	RFC2833 Relay DTMF
MF Transport Type	RFC2833 Relay MF
DTMF Volume (-31 to 0 dB)	-11
CAS Transport Type	CAS Events Only

Fax/Modem/CID Settings

Fax Transport Mode	T.38 Relay
Caller ID Transport Type	MUTE
Caller ID Type	Bellcore
V21 Modem Transport Type	Disabled
V22 Modem Transport Type	Bypass Enabled
V23 Modem Transport Type	Bypass Enabled
V32 Modem Transport Type	Bypass Enabled
V34 Modem Transport Type	Bypass Enabled
Fax Relay Redundancy Depth	0
Fax Relay Enhanced Redundancy Depth	4
Fax Relay ECM Enable	Enable
Fax Relay Max Rate (bps)	14400
Fax/Modem Bypass Coder Type	G711Alaw
Fax/Modem Bypass Packing Factor	1
CNG Detector Mode	DISABLE

RTP Settings

Dynamic Jitter Buffer Minimum Delay	70
Dynamic Jitter Buffer Optimization Factor	7
RTP Redundancy Depth	0
Packing Factor	1
Basic RTP Packet Interval	DEFAULT
RTP Directional Control	Transmit-Receive
RFC 2833 TX Payload Type	100
RFC 2833 RX Payload Type	100
RFC 2198 Payload Type	104
Fax Bypass Payload Type	102
Enable RFC 3389 CN Payload Type	Disable
Analog Signal Transport Type	Ignore analog signals

IP Media Settings

Enable Answer Detector	Disable
Answer Detector Activity Delay	0
Answer Detector Silence Time	10
Answer Detector Redirection	Disable
Answer Detector Sensitivity	0
Enable AGC	Disable
AGC Slope	3
AGC Redirection	0
AGC Target Energy	19
Enable Energy Detector	Disable
Energy Detector Quality Factor	4
Energy Detector Threshold	3
Enable Pattern Detector	Disable

Step 3. Use Table 7-3 on page 152 as a reference when configuring/modifying the channel parameter fields in the 'Channel Settings' screen.

Step 4. After configuring/modifying the parameter fields, click the **SUBMIT** button. The changes are entered into the system and the screen is refreshed.

6.4.4.3 Trunk Settings

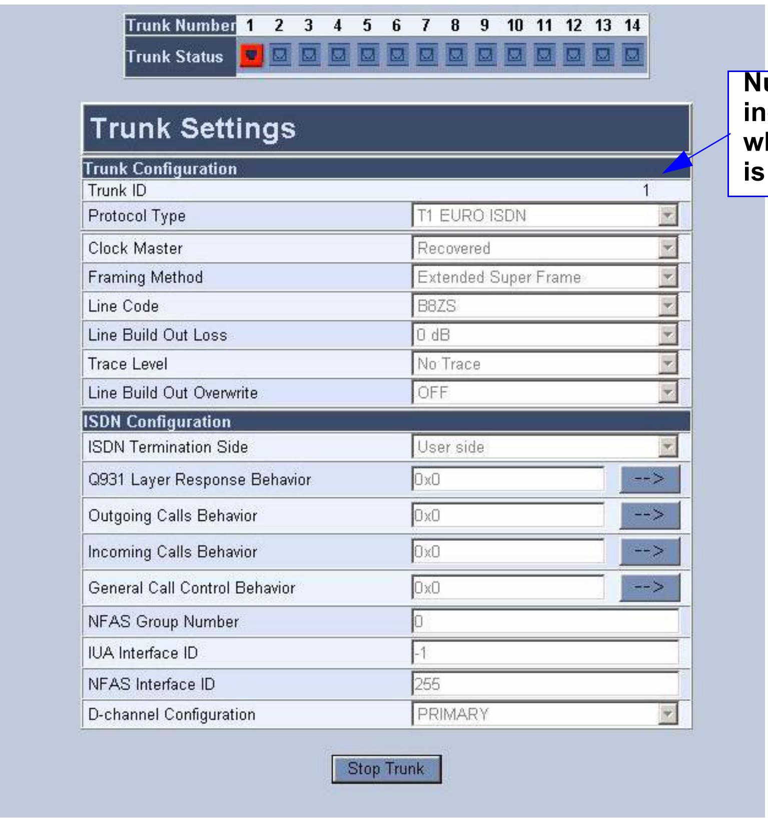
➢ To view the Trunk Settings, take these 11 steps:

Step 1. From the main menu list on the left, click on the Advanced Configuration link. The Advanced Configuration screen.

Step 2. From the Advanced Configuration screen, click the **Trunk Settings** option in the sub-menu bar on the top. The Trunk Settings screen appears.

Initially, the screen appears with the parameters fields grayed (indicating read-only). The Stop Trunk button appears at the bottom of the screen.

Figure 6-9: Trunk Settings Screen



Trunk Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Trunk Status	🟥	▣	▣	▣	▣	▣	▣	▣	▣	▣	▣	▣	▣	▣

Trunk Settings

Trunk Configuration	
Trunk ID	1
Protocol Type	T1 EURO ISDN
Clock Master	Recovered
Framing Method	Extended Super Frame
Line Code	B8ZS
Line Build Out Loss	0 dB
Trace Level	No Trace
Line Build Out Overwrite	OFF
ISDN Configuration	
ISDN Termination Side	User side
Q931 Layer Response Behavior	0x0 -->
Outgoing Calls Behavior	0x0 -->
Incoming Calls Behavior	0x0 -->
General Call Control Behavior	0x0 -->
NFAS Group Number	0
IUA Interface ID	-1
NFAS Interface ID	255
D-channel Configuration	PRIMARY

Stop Trunk

Number indicates which Trunk is selected

Step 3. To configure a specific trunk's parameters or make a change to any of the parameters, from the Trunks displayed on the top, select a Trunk by clicking the 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 6-2 shows the possible indicators and their descriptions.

Table 6-2: Trunk Status Color Indicator Key

Indicator	Color	Description	
▣	**Grey**	**Disabled**	
🟩	**Green**	**Active - OK**	
🟨	**Yellow**	**RAI Alarm**	Remote Alarm Indication (the yellow alarm)
🟥	**Red**	**LOS**	Loss of Signal
🟦	**Blue**	**AIS Alarm**	Alarm Indication Signal (the blue alarm)
🟧	**Orange**	**D-Channel Alarm** **(ISDN only)**	**D-Channel Alarm** **(ISDN only)**

When modifying the Protocol Type, there are three different menu types that match the following 3 coder families:

- Transparent
- ISDN
- CAS

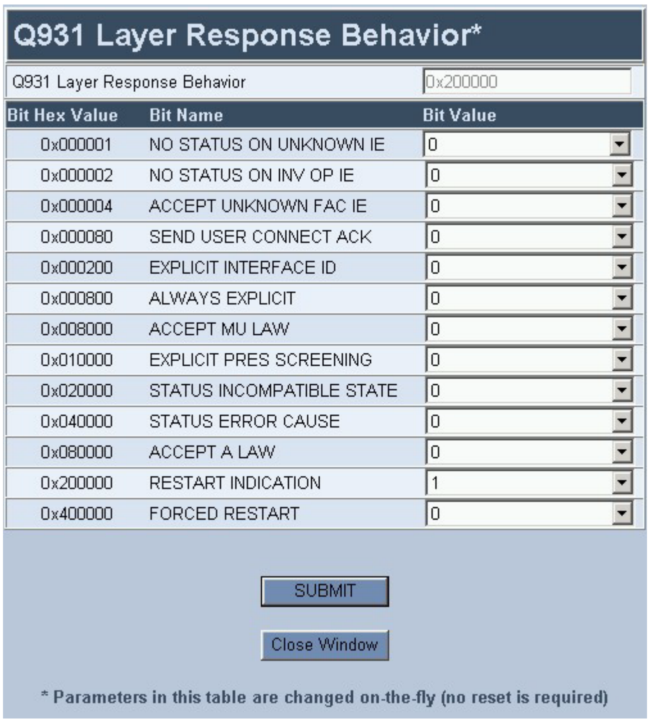
When traversing between these 3 coder families, the menu is modified to include additional parameters appropriate to the family type selected.

Step 4. At the bottom of the screen, click the **Stop Trunk** button to return the screen to a modifiable state. The parameters are no longer grayed. When configuring the device for the first time, or when no protocol is specified for any trunk, at the bottom of the screen, two buttons appear: the **Apply Trunk Settings** button and the **Apply to all Trunks** button. When protocols are specified, only the **Apply Trunk Settings** button appears.

Step 5. In the **Trunk Configuration** section, from the **Protocol Type** and **Framing Method** drop-down lists, select the appropriate option. Since in the example displayed in Figure 6-9, **Protocol Type** is configured as E1 EURO ISDN, the ISDN parameters are displayed. Configure the parameter **ISDN Termination Side** as “User side” when the PSTN or PBX side is configured as “Network side”, and vice versa. If you do not know the Mediant 2000 ISDN termination side, choose “User side” and refer to the Status & Diagnostics screen. If the D-channel alarm is indicated, choose “Network side”. For E1 trunks, always set the **Framing Method** to Extended Super Frame.

Step 6. For the different behavior bits configuration, enter the direct Hex value of the bits in the text box,
or,
configure the bit map directly by clicking the arrow button, a new window appears, with the specific bits (refer to Figure 6-10).

Figure 6-10: Q931 Bit Map Screen



Q931 Layer Response Behavior*

Q931 Layer Response Behavior	0x200000

Bit Hex Value	Bit Name	Bit Value
0x000001	NO STATUS ON UNKNOWN IE	0
0x000002	NO STATUS ON INV OP IE	0
0x000004	ACCEPT UNKNOWN FAC IE	0
0x000080	SEND USER CONNECT ACK	0
0x000200	EXPLICIT INTERFACE ID	0
0x000800	ALWAYS EXPLICIT	0
0x008000	ACCEPT MU LAW	0
0x010000	EXPLICIT PRES SCREENING	0
0x020000	STATUS INCOMPATIBLE STATE	0
0x040000	STATUS ERROR CAUSE	0
0x080000	ACCEPT A LAW	0
0x200000	RESTART INDICATION	1
0x400000	FORCED RESTART	0

* Parameters in this table are changed on-the-fly (no reset is required)

Step 7. For the **Clock Master** (the trunk clock source) drop-down list, select either 'Recovered' (the clock is recovered from the trunk; default) or 'Generated' (the trunk clock source is provided by the internal/TDM bus clock source) the above selection depends on the parameter 'TDM Bus Clock Source'. Refer to Section 6.4.4.4 on page 125.

Step 8. For the **Line Code** drop-down list, select either **B8ZS** (bipolar 8-zero substitution) for T1 trunks only; **HDB3** (high-density bipolar 3) for E1 trunks only; or **AMI** for both T1 and E1.

Step 9. After modifying any parameters, do one of the following:

- To apply the changes to the selected trunk only, click the **Apply Trunk Settings** button.
- When configuring the device for the first time, or when no protocol is specified for any trunk, To apply the changes to all trunks, click the **Apply To All Trunks** button.

The screen is refreshed. Parameters become read-only (shown grayed). The **Stop Trunk** button appears at the bottom of the screen.

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

Step 10. To commit the changes to the Flash memory, in the main menu on the left, click the Save Configuration link. The Save Configuration screen appears.

Step 11. Click the **Save** button. The configuration is saved to Flash memory.

> **Note 1:** A device reset may be needed in certain circumstances for the setup to be activated, if a reset is needed the Web interface will alert the user. In case such a device reset is needed, click the **Reset** link in main menu to the left, choose the **Burn** option and click the **Reset** button to restart the device with the new configuration. (Refer to Section 6.4.8 on page 143.)
>
> **Note 2:** If you are modifying multiple screens, and a reset is required in a certain screen, perform the reset after you are finished modifying all of the screens you intended and NOT after each screen.

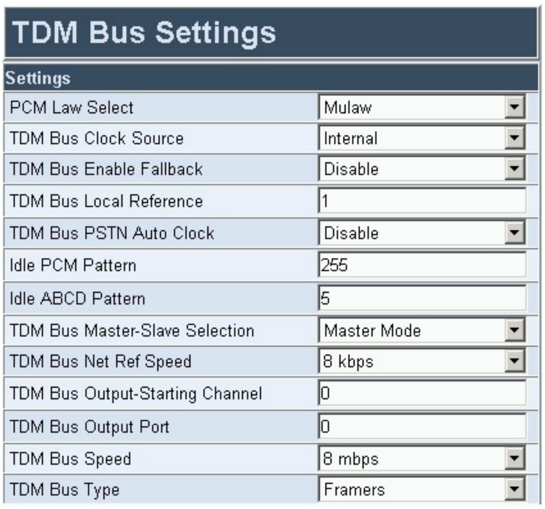
6.4.4.4 TDM Bus Settings

➢ To configure the TDM Bus settings, take these 6 steps:

Step 1. From the main menu list on the left, click on the Advanced Configuration link. The Advanced Configuration screen appears.

Step 2. From the Advanced Configuration screen, click the **TDM Bus Settings** option in the sub-menu bar on the top. The TDM Bus Settings screen appears.

Figure 6-11: TDM Bus Settings Screen



TDM Bus Settings

Settings

PCM Law Select	Mulaw
TDM Bus Clock Source	Internal
TDM Bus Enable Fallback	Disable
TDM Bus Local Reference	1
TDM Bus PSTN Auto Clock	Disable
Idle PCM Pattern	255
Idle ABCD Pattern	5
TDM Bus Master-Slave Selection	Master Mode
TDM Bus Net Ref Speed	8 kbps
TDM Bus Output-Starting Channel	0
TDM Bus Output Port	0
TDM Bus Speed	8 mbps
TDM Bus Type	Framers

Step 3. Use Table 7-2 on page 147 and Table 7-4 on page 158 as a reference when configuring/modifying the parameter fields in the 'TDM Bus Settings' screen.

Step 4. After configuring/modifying the parameter fields, click the **Submit** button. The changes are entered into the system and the screen is refreshed.

Step 5. To commit the changes to flash memory, on the main menu to the left, click the **Reset** link. The Reset screen appears.

Step 6. Select the **Burn** option and click the **Reset** button. (Refer to Section 6.4.8 on page 143.)

.

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 142) to retain the changes to the device's Flash memory.

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

6.4.4.5 Configuration File

The Configuration File screen enables you to restore/change (download a new ***ini*** file to the Device) or backup the current configuration file that the device is using (make a copy of the VoIP device's ***ini*** file and store it in a directory on your PC).

- **Restore your configuration** - If the VoIP device has been replaced or has lost its programming information, you can restore the VoIP device configuration file from a previous backup or from a newly created ***ini*** file. To restore the VoIP Device configuration from a previous backup you must have a backup of the VoIP device information stored on your PC.
- **Back up your configuration** - If you want to protect your VoIP device programming. . The generated backup ***ini*** file contains values that have been changed by the user and are other than the default values.

In the Configuration File screen, you can bring an ***ini*** file from the device to a directory in your PC, and send the ***ini*** file from your PC to the device and to other devices.

Protect the device configuration by bringing the ***ini*** file from the device to your PC. Later, if another device is replaced or loses its programming data, you'll be able to restore / send the ***ini*** file backed up on your PC to the device.

The ***ini*** file is AudioCodes' proprietary configuration text file containing configuration parameters and data. Sending the ***ini*** file to the device only provisions parameters that are contained in the ***ini*** file.

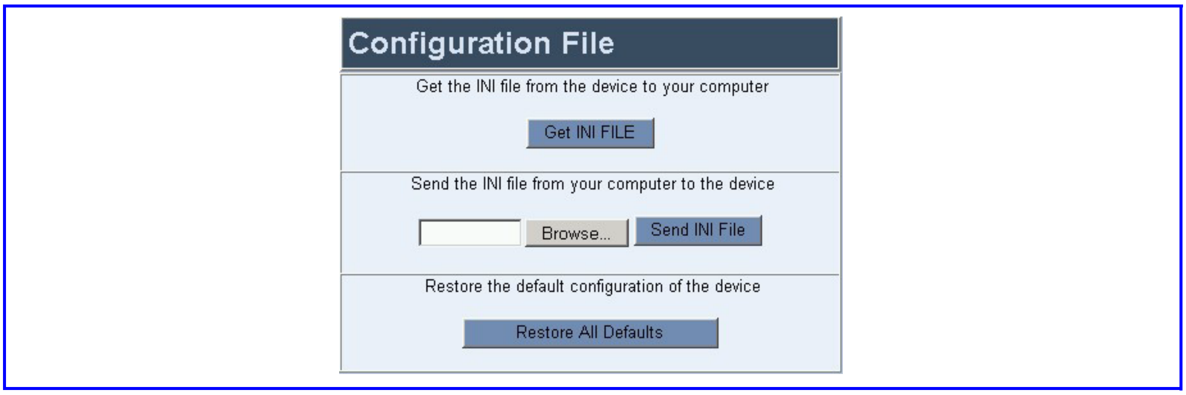
The ***ini*** file with parameters set at their default values is on the CD accompanying the device. The ***ini*** file can also be received as an e-mail attachment from AudioCodes' Technical Support. Users can also generate their own ***ini*** file using AudioCodes' DConvert utility (refer to Appendix E on page 237).

➢ To save the *ini* file to the PC, take these 3 steps:

Step 1. From the main menu list on the left, click on the Advanced Configuration link. The Advanced Configuration screen appears.

Step 2. From the Advanced Configuration screen, click the Configuration File link in the sub-menu bar on the top. The Configuration File screen appears.

Figure 6-12: Configuration File Screen



Configuration File

Get the INI file from the device to your computer

Send the INI file from your computer to the device

Restore the default configuration of the device

Step 3. Click the **Get *ini* File** button. You are prompted to select a location in which to save it.

> **Note:** The ***ini*** file that you save from the device to the PC contains only those parameters whose values you modified following receipt of the device. It does not contain parameters unchanged at their (original) default value. If a user changes a parameter's default value and later reverts back to the default, the parameter is not included in the ***ini*** file saved from the device to the PC.

➢ To load an *ini* file from the PC to the device, take these 4 steps:

Step 1. Click on the **Browse** button next to the **Send the *ini* file from your computer to the device** field and navigate to the location of the predefined ***ini*** file. Refer to Figure 6-12.

Step 2. Click the **Send File** button. The file loading process is activated. When the loading is complete, a verification message is displayed at the bottom of the screen: **File XXXX was successfully loaded into the device**.

Step 3. From the main menu list on the left, click **Reset**. The Reset screen appears.

Step 4. Select the **Burn** option and click the **Restart** button. Wait for the device to reset. After self-testing, the Ready and LAN LEDs on the device's front panel are lit green. Any malfunction causes the Ready LED to change to red.

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

6.4.4.6 Regional Settings

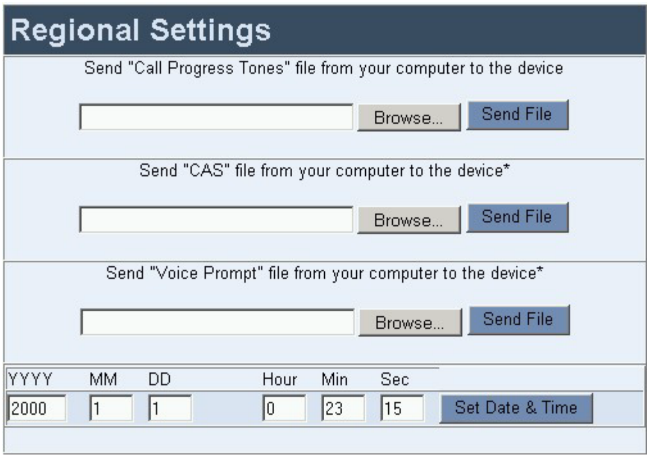
From the Regional Settings screen users can send a Call Progress Tones ***dat*** file, a CAS ***dat*** file and/or a Voice Prompts ***dat*** file to the device from their PC.

➢ To access the Regional Settings screen, take these 2 steps:

Step 1. From the main menu list on the left, click on the Advanced Configuration link. The Advanced Configuration screen appears.

Step 2. From the sub-menu bar on the top, click the Regional Settings link. The Regional Settings screen appears.

Figure 6-13: Regional Settings Screen - Sending CPT, CAS and/or Voice Prompt File to the Device



Regional Settings

Send "Call Progress Tones" file from your computer to the device

Send "CAS" file from your computer to the device*

Send "Voice Prompt" file from your computer to the device*

YYYY	MM	DD	Hour	Min	Sec
2000	1	1	0	23	15

The files are available on the CD accompanying your device. They can also be received as an e-mail attachment from AudioCodes' Technical Support. A Call Progress Tones ***txt*** file can be modified and converted into the binary ***dat*** file (refer to Figure E-2, on page 238). When modifying the Call Progress Tones and Distinctive Ringing File, only the ***dat*** file can be sent from your PC to the device. (Refer to Section 7.4 on page 195 and Appendix E on page 237.)

- The Call Progress Tones ***dat*** file is a region-specific, telephone exchange-dependent file. It provides call status/call progress to Customers, operators, and connected equipment. Default Tone: U.S.A.
- E1/T1 CAS signaling files, such as ***E_M_WinkTable.dat***. These files are not needed for CAS protocols only.
- The ***dat*** Voice Prompts file is played by the device during the phone conversation on Call Agent request. Download if you have an application requiring Voice Prompts. The Voice Prompt buffer size in the board is 10 Mbyte. The Voice Prompt buffer size is also controlled by the feature key. For more information contact an AudioCodes representative.

➢ To send a Call Progress Tone, CAS, or Voice Prompt file to the board, take these 7 steps:

Step 1. From the main menu list on the left, click on the Advanced Configuration link. The Advanced Configuration screen appears.

Step 2. From the sub-menu bar on the top, click the Regional Settings link. The Regional Settings screen appears. (Refer to Figure 6-13.)

Step 3. Click the **Browse** button to locate the predefined **Call Progress Tone**, **CAS**, or **Voice Prompt** file as appropriate. (A new software file package may be issued from AudioCodes or your local supplier.)

Step 4. 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.

Step 5. Click the **Send File** button. The file loading process is activated. When the loading is complete, a verification message is displayed at the bottom of the screen: **File XXX was successfully loaded into the device.**

Step 6. For CPT file downloading only - (The rest of files do not require a device reset.) From the main menu list on the left, click **Reset**. The Reset screen appears.

Step 7. Select the **Burn** option and click the **Restart** button. Wait for the device to reset. After self-testing, the Ready and LAN LEDs on the device's front panel are lit green. Any malfunction causes the Ready LED to change to red.

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

Step 1. Enter the date and/or time using the **YYYY**, **MM**, and **DD** field for Year, Month and Day and **HH**, **MM**, and **SS** fields for Hour, Minutes and Seconds.

Step 2. Click the **Set Date** and **Time** button. The date and time is set on the device, accordingly.

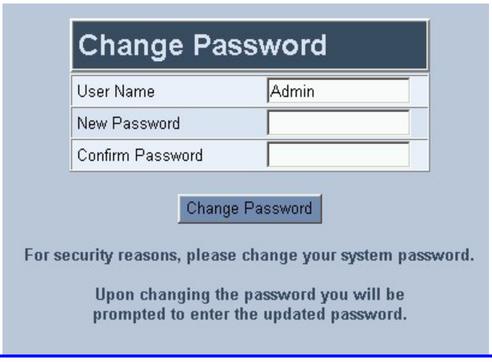
6.4.4.7 Change Password

➢ To change the Password, take these 4 steps:

Step 1. From the main menu list on the left, click on the Advanced Configuration link. The Advanced Configuration screen appears.

Step 2. From the sub-menu bar on the top, click the Change Password link. The Change Password screen appears.

Figure 6-14: Change Password Screen



Change Password

User Name	Admin
New Password	
Confirm Password	

For security reasons, please change your system password.

Upon changing the password you will be prompted to enter the updated password.

Step 3. Enter a User Name and New Password into the fields and confirm the New Password in the **Confirm Password** field. Note that you can configure the User Name and New Password up to 7 characters. They are case-sensitive.

Step 4. Click the **Change Password** button. You are prompted to enter a new username and password. The new username/password takes effect immediately.

When making a change, note that the Password and Username can be up to 7 characters and that they are case-sensitive. The new password takes effect immediately.

➢ To reset the username and password to their defaults:

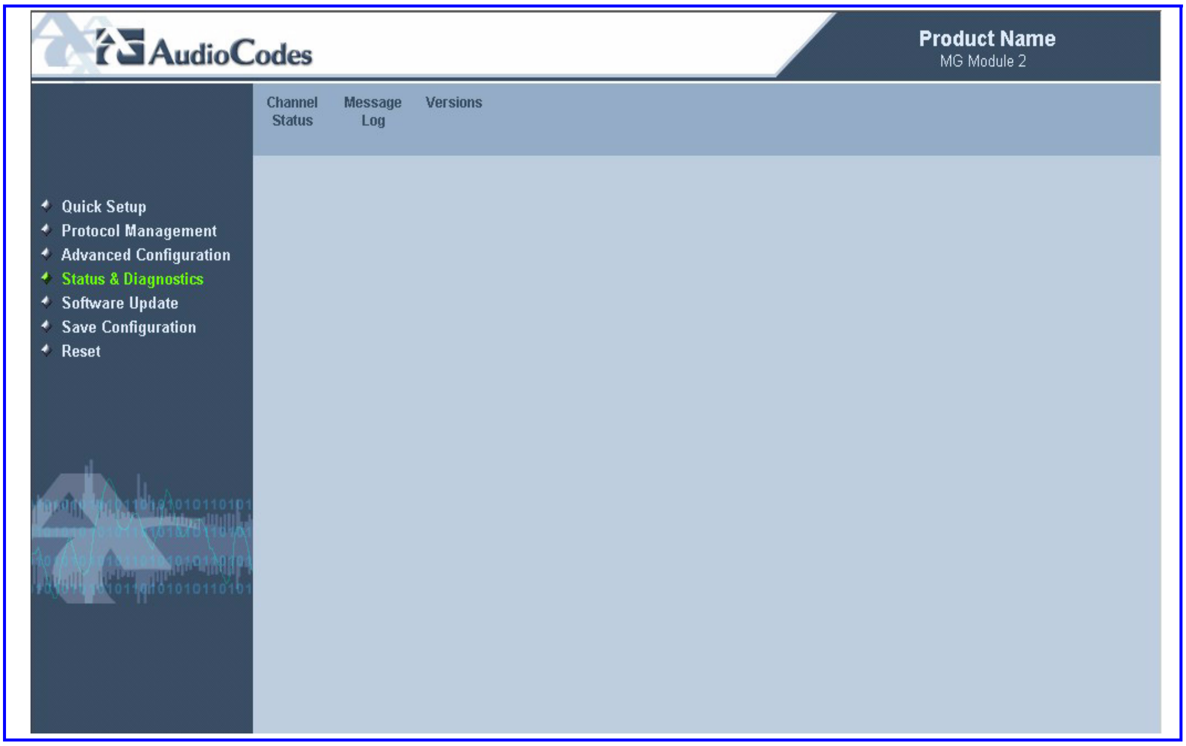
- Set the ***ini*** file parameter ResetWebPassword to **1** and use AudioCodes' BootP/TFTP Server to load the ***ini*** file to the device (refer to Appendix A on page 211). After loading, the username and password automatically revert to their default values (Admin).

6.4.5 Status and Diagnostic Menu

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

- From the main menu list on the left, click on the Status and Diagnostics link. The Status and Diagnostics screen with the sub-menu bar on the top is displayed.

Figure 6-15: Status and Diagnostic Menu Screen



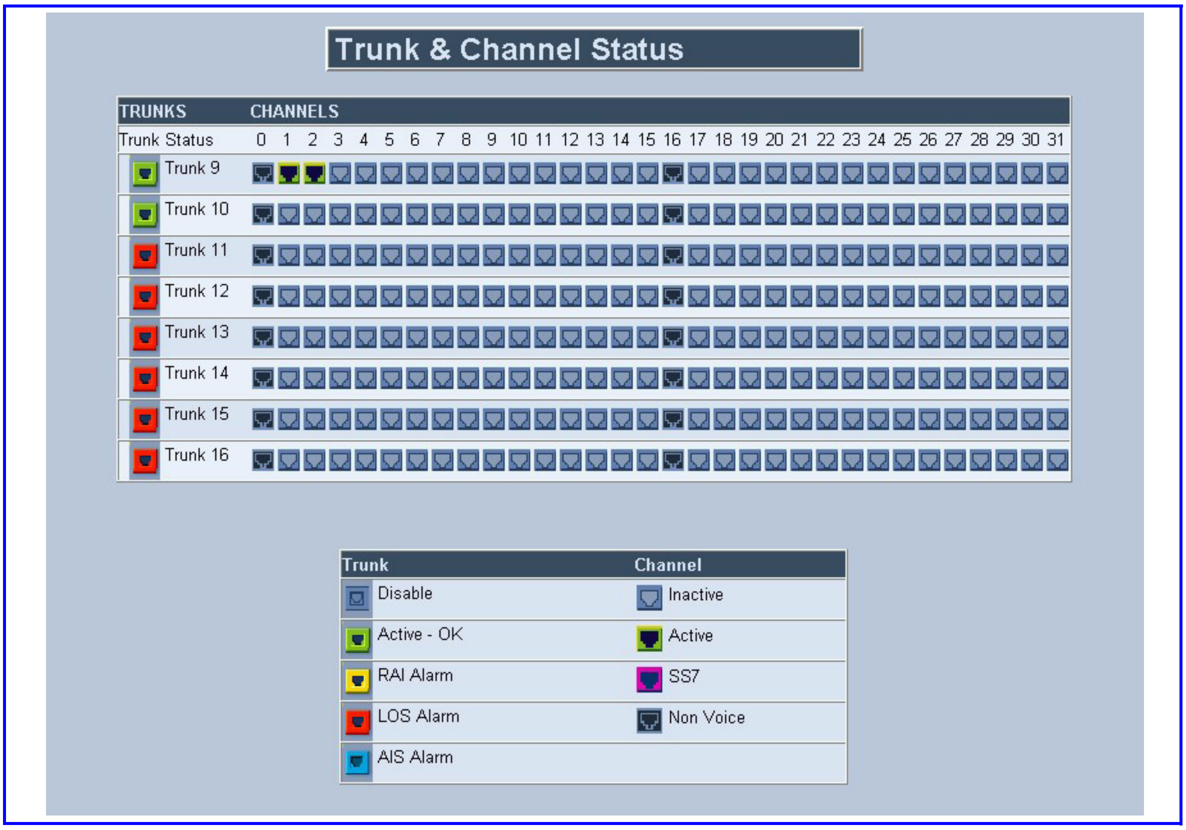
6.4.5.1 Trunk and Channel Status

➢ To access the Trunk and Channel Status screen, take these 3 steps:

Step 1. From the main menu list on the left, click on the Status and Diagnostics link. The Status and Diagnostics screen appears.

Step 2. From the sub-menu bar on the top, click the Channel Status link. The Trunk and Channel Status screen appears. The screen is Read-only.

Figure 6-16: Trunk and Channel Status Screen



Trunk & Channel Status

TRUNKS	CHANNELS																															
Trunk Status	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
🟩 Trunk 9	⬛	🟩	🟩	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
🟩 Trunk 10	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
🟥 Trunk 11	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
🟥 Trunk 12	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
🟥 Trunk 13	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
🟥 Trunk 14	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
🟥 Trunk 15	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
🟥 Trunk 16	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜

Trunk	Channel
⬜ Disable	⬜ Inactive
🟩 Active - OK	🟩 Active
🟨 RAI Alarm	🟪 SS7
🟥 LOS Alarm	⬛ Non Voice
🟦 AIS Alarm	

The number of trunks and channels that appear on the screen depends of the system configuration.

The Trunk and Channel Status indicators can appear colored. Table 6-3 shows the possible indicators and their descriptions.

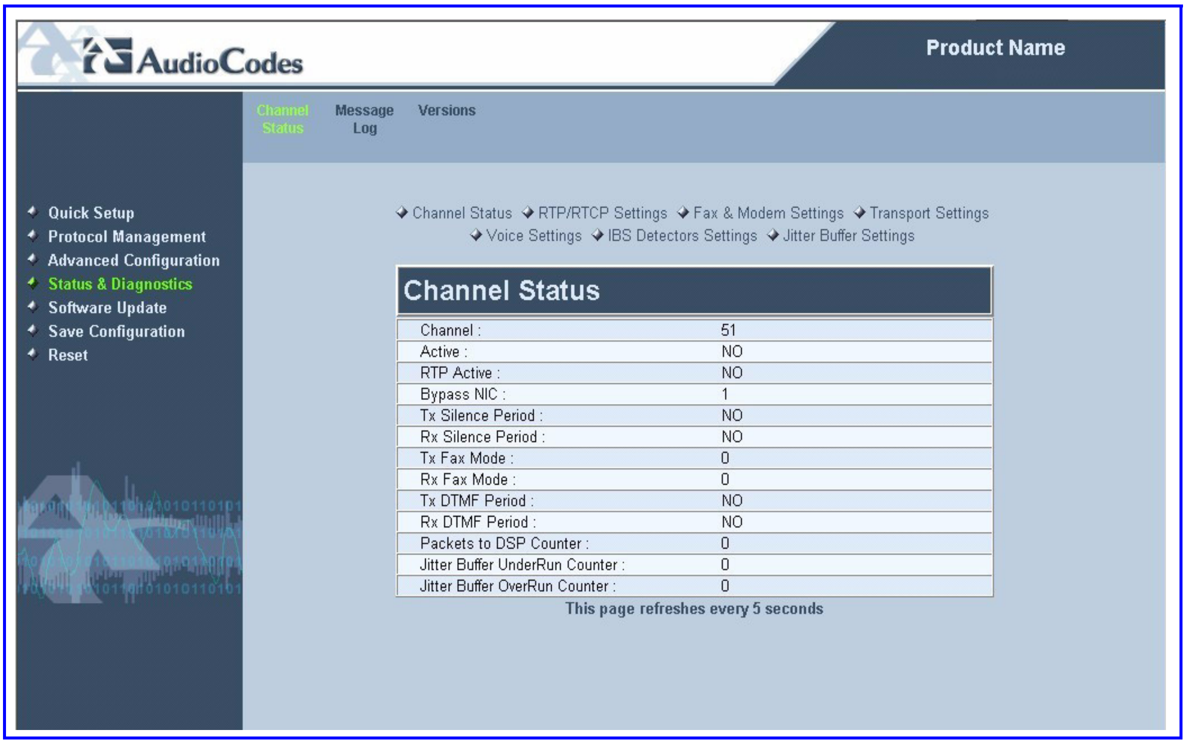
Table 6-3: Trunk and Channel Status Color Indicator Key

Trunk			Channel		
Indicator	Color	Description	Indicator	Color	Description
⬜	Grey	Disabled	⬜	Grey	In-Active
🟩	Green	Active - OK	🟩	Green	Active
🟨	Yellow	RAI Alarm	🟥	Red	PCI mode
🟥	Red	LOS Alarm	🟪	Purple	SS7
🟦	Blue	AIS Alarm	⬛	Black	Non-Voice
🟧	Orange	D-Channel Alarm			

Step 3. To display a screen with a summary of parameter information relevant to a channel, click on the channel.

The following 'per channel' screen information is available when clicking on a specific channel:

Figure 6-17: Channel Status Screen

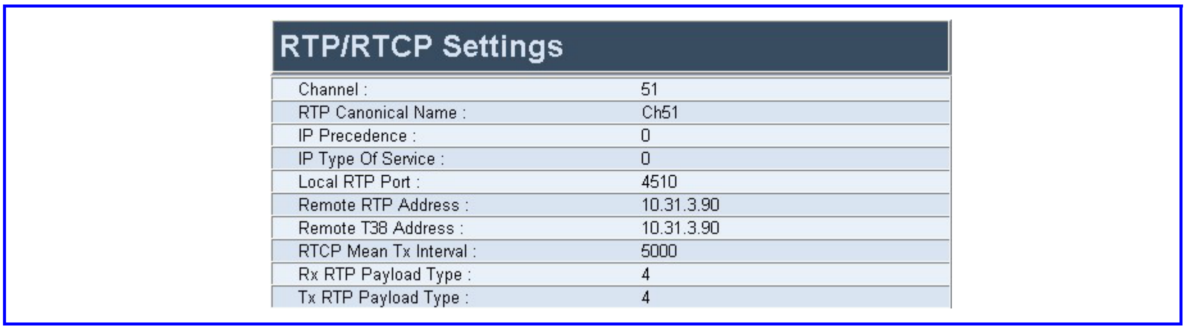


Channel Status

Channel :	51
Active :	NO
RTP Active :	NO
Bypass NIC :	1
Tx Silence Period :	NO
Rx Silence Period :	NO
Tx Fax Mode :	0
Rx Fax Mode :	0
Tx DTMF Period :	NO
Rx DTMF Period :	NO
Packets to DSP Counter :	0
Jitter Buffer UnderRun Counter :	0
Jitter Buffer OverRun Counter :	0

This page refreshes every 5 seconds

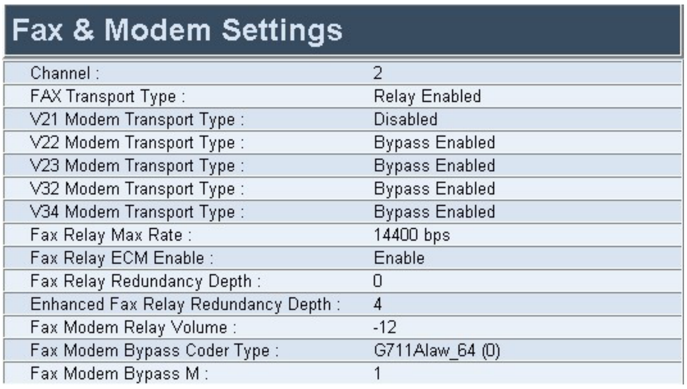
Figure 6-18: RTP/RTCP Settings Screen



RTP/RTCP Settings

Channel :	51
RTP Canonical Name :	Ch51
IP Precedence :	0
IP Type Of Service :	0
Local RTP Port :	4510
Remote RTP Address :	10.31.3.90
Remote T38 Address :	10.31.3.90
RTCP Mean Tx Interval :	5000
Rx RTP Payload Type :	4
Tx RTP Payload Type :	4

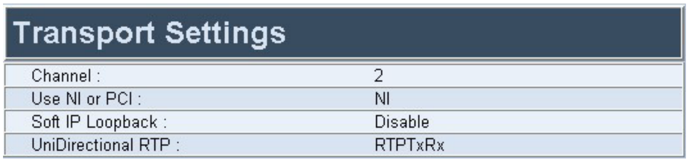
Figure 6-19: Fax & Modem Settings Screen



Fax & Modem Settings

Channel :	2
FAX Transport Type :	Relay Enabled
V21 Modem Transport Type :	Disabled
V22 Modem Transport Type :	Bypass Enabled
V23 Modem Transport Type :	Bypass Enabled
V32 Modem Transport Type :	Bypass Enabled
V34 Modem Transport Type :	Bypass Enabled
Fax Relay Max Rate :	14400 bps
Fax Relay ECM Enable :	Enable
Fax Relay Redundancy Depth :	0
Enhanced Fax Relay Redundancy Depth :	4
Fax Modem Relay Volume :	-12
Fax Modem Bypass Coder Type :	G711Alaw_64 (0)
Fax Modem Bypass M :	1

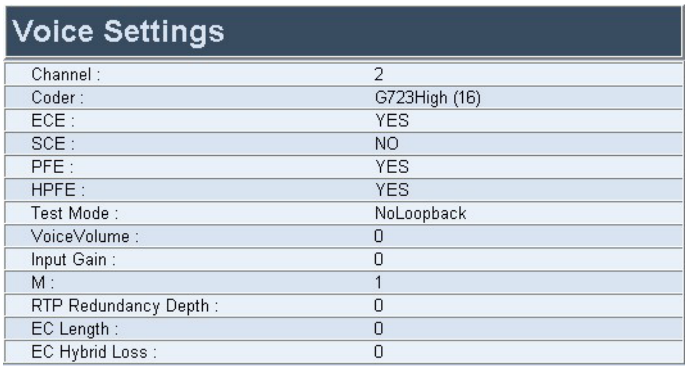
Figure 6-20: Transport Settings Screen



Transport Settings

Channel :	2
Use NI or PCI :	NI
Soft IP Loopback :	Disable
UniDirectional RTP :	RTPTxRx

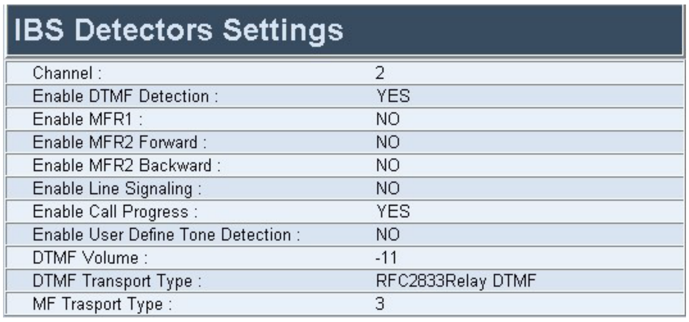
Figure 6-21: Voice Settings Screen



Voice Settings

Channel :	2
Coder :	G723High (16)
ECE :	YES
SCE :	NO
PFE :	YES
HPFE :	YES
Test Mode :	NoLoopback
VoiceVolume :	0
Input Gain :	0
M :	1
RTP Redundancy Depth :	0
EC Length :	0
EC Hybrid Loss :	0

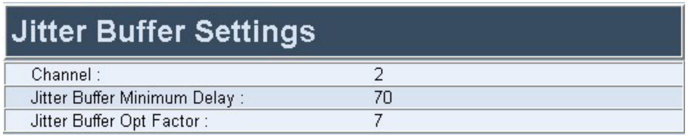
Figure 6-22: IBS Detector Settings Screen



IBS Detectors Settings

Channel :	2
Enable DTMF Detection :	YES
Enable MFR1 :	NO
Enable MFR2 Forward :	NO
Enable MFR2 Backward :	NO
Enable Line Signaling :	NO
Enable Call Progress :	YES
Enable User Define Tone Detection :	NO
DTMF Volume :	-11
DTMF Transport Type :	RFC2833Relay DTMF
MF Trasport Type :	3

Figure 6-23: Jitter Buffer Settings Screen



Jitter Buffer Settings

Channel :	2
Jitter Buffer Minimum Delay :	70
Jitter Buffer Opt Factor :	7

6.4.5.2 Message Log

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

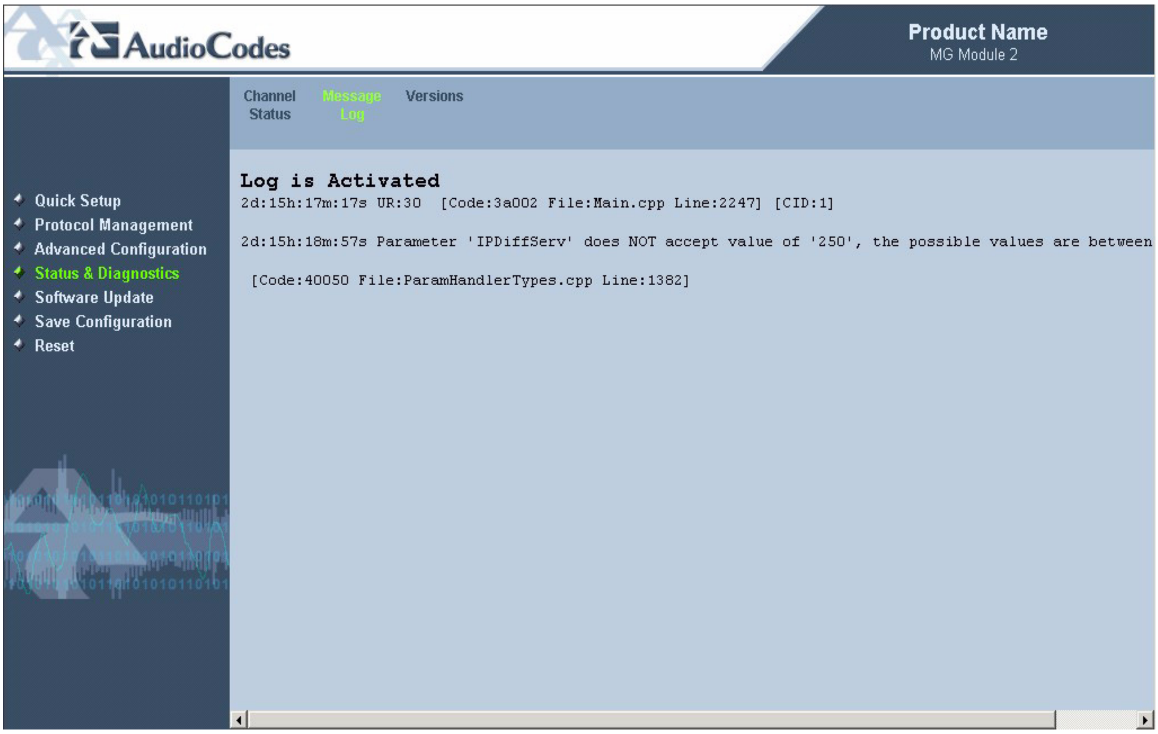
The Message Log serves the Web Server and is similar to a Syslog server. It displays debug messages useful when debugging. AudioCodes does not recommend using the Message Log screen for logging errors and warnings because errors can appear over a prolonged period of time, e.g., a device can display an error after running for a week. Similarly, AudioCodes does not recommend keeping a Message Log session open for a prolonged period (refer to the Note below). For logging of errors and warnings, refer to Section 8.1.1.4 on page 201.

➢ To activate the Message Log, take these 4 steps:

Step 1. From the main menu list on the left, click on the Status and Diagnostics link. The Status and Diagnostics screen appears.

Step 2. From the sub-menu bar on the top, click the Message Log link. The Message Log screen appears.

Figure 6-24: Message Log Screen



```
Log is Activated
2d:15h:17m:17s UR:30  [Code:3a002 File:Main.cpp Line:2247] [CID:1]

2d:15h:18m:57s Parameter 'IPDiffServ' does NOT accept value of '250', the possible values are between

 [Code:40050 File:ParamHandlerTypes.cpp Line:1382]
```

Step 3. Using the scroll bar, select the messages, copy them and paste them into a text editor such as Notepad. Send this ***txt*** file to AudioCodes Technical Support for diagnosis and troubleshooting as needed.

Step 4. To clear the screen of messages, click on the sub-menu Message Log. The screen is cleared. A new session is activated and new messages begin appearing.

Note: Do not keep the Message Log screen activated and minimized for a prolonged period as a prolonged session may cause the device to overload. As long as the screen is open (even if minimized), a session is in progress and messages are sent. Closing the window or moving to another link stops the messages and terminates the session.

6.4.5.3 Versions

The Versions screen displays hardware and software product information. This information can help you to expedite any troubleshooting process. Capture the screen and email it to AudioCodes' Technical Support personnel to ensure quick diagnosis and effective corrective action.

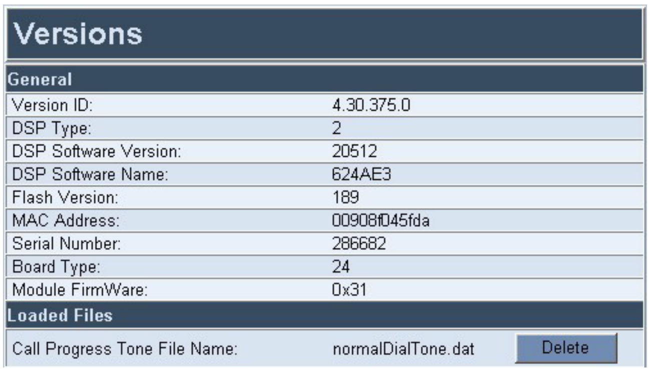
The screen also displays any loaded files in the device.

➢ To display the Versions screen, take these 2 steps:

Step 1. From the main menu list on the left, click on the Status and Diagnostics link. The Status and Diagnostics screen appears.

Step 2. From the sub-menu bar on the top, click the Versions link. The Versions screen appears.

Figure 6-25: Versions Screen



Versions

General		
Version ID:	4.30.375.0	
DSP Type:	2	
DSP Software Version:	20512	
DSP Software Name:	624AE3	
Flash Version:	189	
MAC Address:	00908f045fda	
Serial Number:	286682	
Board Type:	24	
Module FirmWare:	0x31	
Loaded Files		
Call Progress Tone File Name:	normalDialTone.dat	Delete

➢ To delete any loaded files, take these 5 steps:

Step 1. From the main menu list on the left, click on the Status and Diagnostics link. The Status and Diagnostics screen appears.

Step 2. From the sub-menu bar on the top, click the Versions link. The Versions screen appears.

Step 3. In the Versions table, click the **Delete** button. The file deletion takes effect only after a device reset is performed.

Step 4. In main menu to the left, click the **Reset** link. The Reset screen appears.

Step 5. Select the **Burn** option and click the **Reset** button to restart the device with the new settings. (Refer to Section 6.4.8 on page 143.)

6.4.6 Software Update

The Software Update screen offers two options for downloading current software update files: the Software Upgrade Wizard and Load Auxiliary Files screen. In addition, the License screen is provided for users to enter their updated license keys.

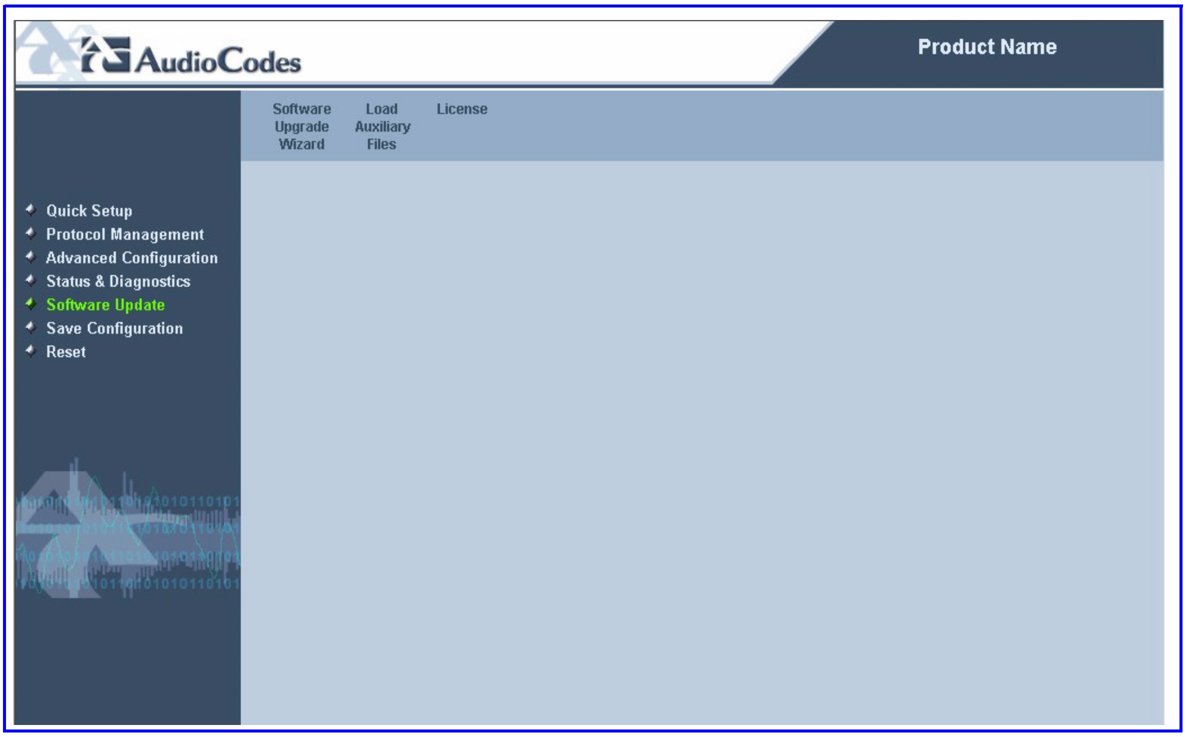
- **Software Upgrade Wizard** - Refer to Section 6.4.6.1 below
- **Load Auxiliary Files** - Refer to Section 6.4.6.2 on page 140
- **License** - Refer to Section 6.4.6.3 on page 141

6.4.6.1 Software Upgrade Wizard

The Software Upgrade Wizard allows the user to upgrade the device's software by loading a new ***.cmp** file together with a full suite of useful auxiliary files.

Loading a ***.cmp** file is mandatory in the Software Upgrade Wizard process. During the process, the user chooses from the auxiliary files provided to load as well. For each auxiliary file type, you can choose between reloading an existing file, loading a new file or not loading a file at all.

Figure 6-26: Start Software Upgrade Screen



➢ To use the Software Upgrade Wizard take these 11 steps:

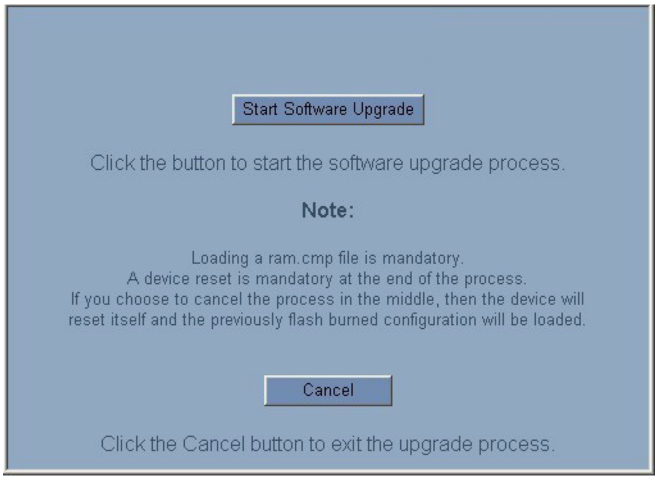
Note: The Software Upgrade Wizard requires the device to be reset at the end of the process which will disrupt any existing traffic on the device. To avoid disrupting traffic, disable all traffic on the device before initiating the Software Upgrade Wizard.

Step 1. Stop all traffic on the device (refer to the note above.)

Step 2. From the main menu list on the left, click on the Software Update link. The Software Upgrade screen with the sub-menu bar on the top is displayed.

Step 3. On the sub-menu bar on the top, click the Software Upgrade Wizard link. The Start Software Upgrade screen appears.

Figure 6-27: Start Software Upgrade Screen



Click the button to start the software upgrade process.

Note:

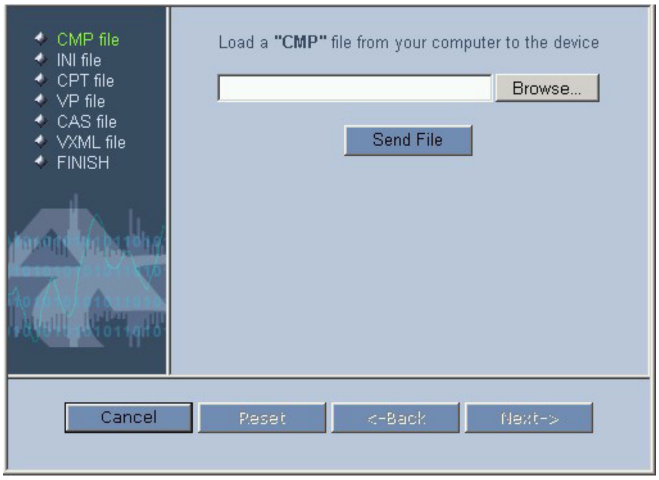
Loading a ram.cmp file is mandatory.
A device reset is mandatory at the end of the process.
If you choose to cancel the process in the middle, then the device will reset itself and the previously flash burned configuration will be loaded.

Click the Cancel button to exit the upgrade process.

Note: At this point you may cancel the Software Upgrade process with no consequence to the device by using the cancel button. If you continue with the Software Upgrade process by clicking the **Start Software Upgrade** button, the process must be followed through and completed with a device reset at the end of the process. If you use the **Cancel** button, in any of the subsequent screens, the Software Upgrade process causes the device to be reset.

Step 4. Click the **Start Software Upgrade** button to initiate the upgrade process. The File Loading screen appears displaying the ***cmp*** file information. The background Web screen is disabled. During the Software Upgrade process, the rest of the Web application is unavailable. After the Software Upgrade process is completed access to the full Web application is restored.

Figure 6-28: Load CMP File Dialog Screen



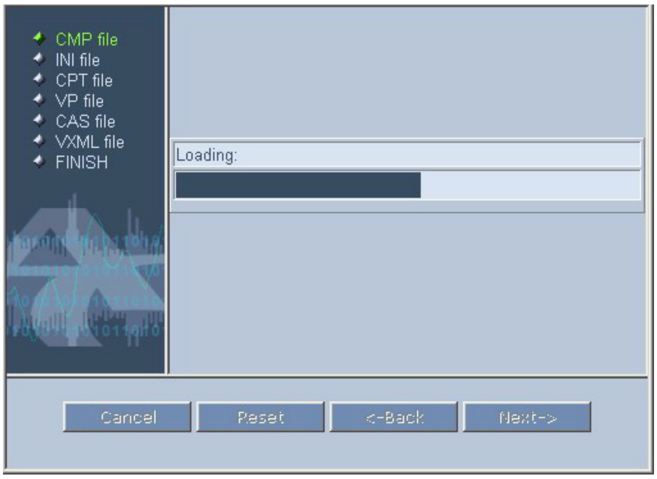
Load a **"CMP"** file from your computer to the device

Note the file type list in the left side of the screen. This list contains the relevant file types that can be loaded via the wizard for this device type (Analog, Digital, FXO, FXS, etc.). The highlighted file type in the file type list indicates which file type is being displayed in the main part of the screen. As you continue through the Software Upgrade process by clicking on the **Next** button, each of the relevant file type screens are presented, going down the list until the Finish screen appears.

Step 5. Click the **Browse** button and navigate to the location of the ***.cmp** file you wish to load. The path and file name appears in the field.

Step 6. Click the **Send File** button to send the file to the device. The File Loading screen appears with a progress bar indicating the loading period. When the loading is complete, a message is displayed indicated the file was successfully loaded into the device.

Figure 6-29: File Loading Dialog Screen



All four buttons (**Cancel**, **Reset**, **Back**, and **Next**) in the bottom portion of the screen are activated.

Step 7. You may choose between these options:

- **Loading Additional Auxiliary Files**
- **Completing the Software Upgrade Process**
- **Revert to the Previous Configuration Files**

Step 8. Loading Additional Auxiliary Files

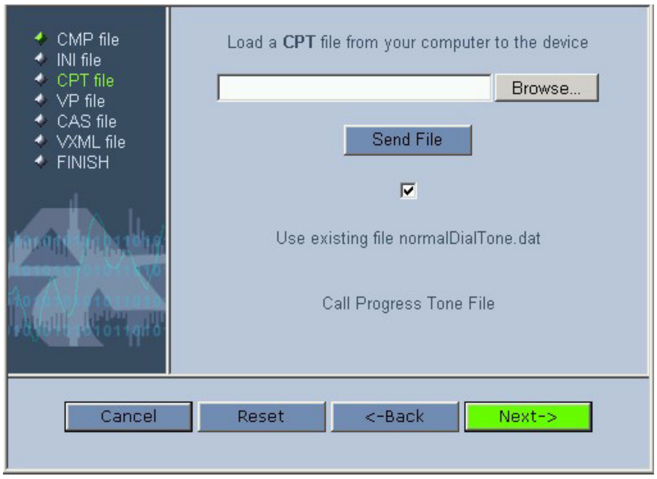
To move to the next file type on the list to the left, click the **Next** button. The File Loading screen appears with the next relevant file type highlighted.

For each file type the user has three options:

- Load a new auxiliary file to the device using the **Browse** and **Send File**' button as described above.
- Load the existing auxiliary file - A checkbox (checked as the default as shown in Figure 6-30) appears if relevant to the device.
- Avoid loading any file at all - Clear the **Load File** field and checkbox.

Continue through each of the file type screens by clicking **Next** and selecting a file option. As an example, Figure 6-30 displays the File Loading screen with the ***CPT*** file type selected.

Figure 6-30: File Loading Dialog Screen - CPT Type Displayed



Load a **CPT** file from your computer to the device

☑

Use existing file normalDialTone.dat

Call Progress Tone File

Step 9. Completing the Software Upgrade Process

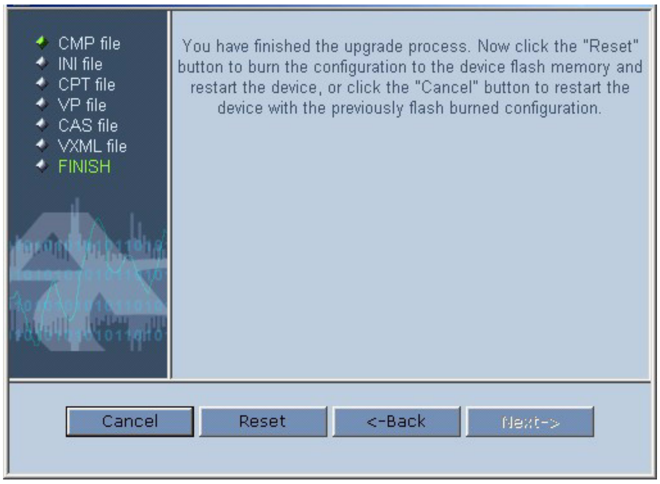
From any of the file type screens, you can complete the Software Upgrade process by clicking the **Reset** button. The device is reset utilizing the new files you have loaded up to that point.

Step 10. Revert to the Previous Configuration Files

From any of the file type screens, you can revert to the previous software files by clicking the **Cancel** button. The Software Upgrade process is terminated and the device is reset utilizing the previous configuration files. Similar screens as shown in Figure 6-31 and Figure 6-32 below are displayed.

Step 11. When continuing through the Software Upgrade process, you complete the process from the Finish screen by clicking the **Reset** button (the **Next** button is disabled.).

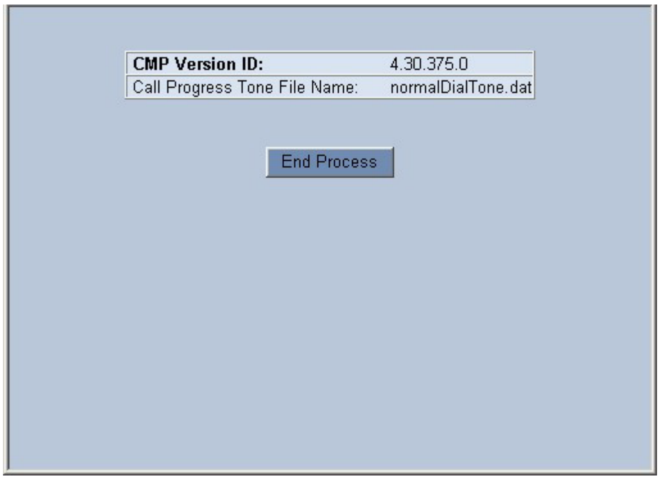
Figure 6-31: Finish Dialog Screen



You have finished the upgrade process. Now click the "Reset" button to burn the configuration to the device flash memory and restart the device, or click the "Cancel" button to restart the device with the previously flash burned configuration.

During the Reset process, the device ‘burns’ the newly loaded configuration to the flash memory. The File Burning screen appears displaying the File Burning to Flash Memory progress bar. When this is completed, the Reset Device screen appears displaying the Reset in progress bar. When this is complete, the End Of Process screen appears displaying the current configuration information.

Figure 6-32: End of Process Dialog Screen



CMP Version ID:	4.30.375.0
Call Progress Tone File Name:	normalDialTone.dat

Step 12. Click the **End Process** button. The Quick Setup screen appears and the full Web application is reactivated.

6.4.6.2 Auxiliary Files Download

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

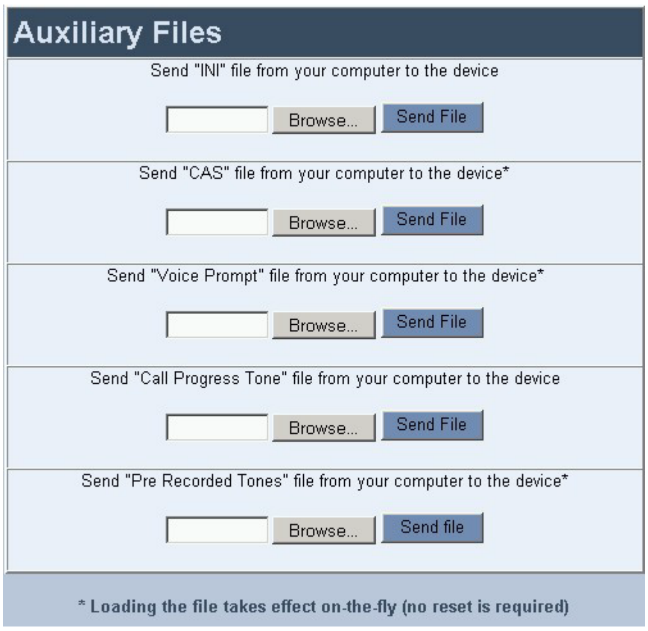
- ***ini*** - Download the file to provision the Mediant 2000 parameters. The Embedded Web Server enables practically full Mediant 2000 provisioning, but customers may occasionally require new feature configuration parameters, in which case this file is downloaded. Note that the ***ini*** file download only provisions parameters that are contained in the ***ini*** file.
- **Voice Prompt -** The ***dat*** Voice Prompts file is played by the device during the phone conversation on Call Agent request. Download if you have an application requiring Voice Prompts.
- **Call Progress Tone - *usa_tones.dat* -** This is a region-specific, telephone exchange-dependent file. Call Progress Tones provide call status/call progress to customers, operators, and connected equipment. Default Tones: U.S.A. ***usa_tones.ini* -** The ***ini*** file is the value of the different Call Progress Tones files (frequency, cadence, etc.). To convert the ***usa_tones.ini*** file, which is a ***txt*** file, to a **usa*_tones.dat*** file that can be downloaded into the Mediant 2000., use the Call Progress Tones file generator utility ***TPDMUtil.exe***.
- **Prerecorded Tones -** The *.dat PRT file enhances the devices capabilities to play a wide range of telephone exchange tones.
- **CAS -** ***cas.dat*** includes E1/T1 CAS signaling files, such as ***E_M_WinkTable.dat***, These files are not needed for ISDN protocols.

➢ To download an auxiliary file, take these 5 steps:

Step 1. From the main menu list on the left, click on the Software Download link. The Software Download screen is displayed.

Step 2. From the sub-menu bar on the top, click the Auxiliary Files Download link. The Auxiliary Files Download screen appears.

Figure 6-33: Auxiliary Files Download Screen



Auxiliary Files

Send "INI" file from your computer to the device

Send "CAS" file from your computer to the device*

Send "Voice Prompt" file from your computer to the device*

Send "Call Progress Tone" file from your computer to the device

Send "Pre Recorded Tones" file from your computer to the device*

* Loading the file takes effect on-the-fly (no reset is required)

Step 3. Use the **Browse** button to locate the appropriate file on your computer.

Step 4. Click the **Send File** button. The files are sent to the device.

Step 5. To commit the changes to the Flash memory, in the main menu on the left, click the **Save Configuration** button. In the **Save Configuration** screen that appears, click the **Save Configuration** button to reset the system.

Note: A device reset is required to activate a loaded CPT file, and may be required for the activation of certain ***ini*** file parameters. The **Burn** option must be selected. (Refer to Section 6.4.8 on page 143.)

6.4.6.3 License

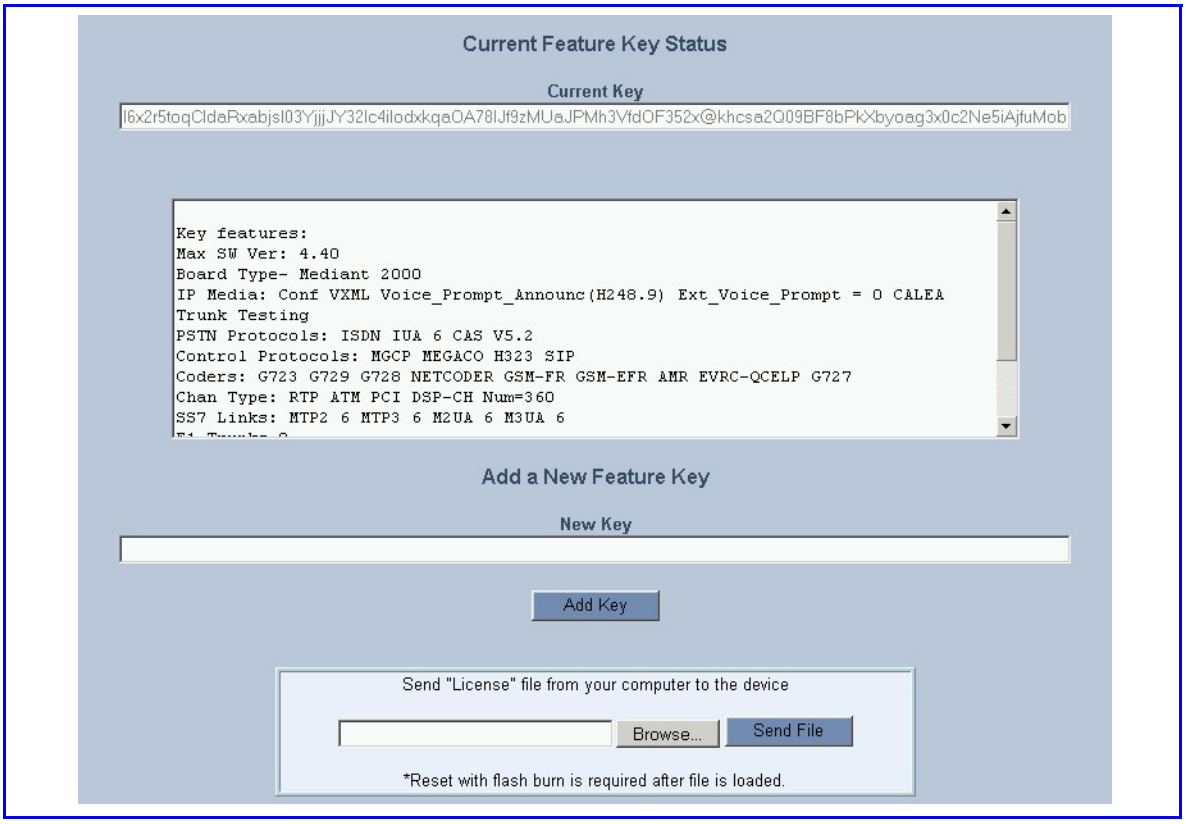
The License screen allows the user to enter an updated license key.

➢ To use the License screen, take these 7 steps:

Step 1. On the main menu to the left, click the Software Download link. The Software Download screen appears.

Step 2. On the sub-menu bar on the top, click the License link. The License screen appears.

Figure 6-34: License Screen


Current Feature Key Status
Current Key
Add a New Feature Key
New Key
Add Key
Send "License" file from your computer to the device
Browse...
Send File
*Reset with flash burn is required after file is loaded.

Current Key: l6x2r5toqCldaRxabjsl03YjjjJY32lc4ilodxkqaOA78lJf9zMUaJPMh3VfdOF352x@khcsa2Q09BF8bPkXbyoag3x0c2Ne5iAjfuMob

```
Key features:
Max SW Ver: 4.40
Board Type- Mediant 2000
IP Media: Conf VXML Voice_Prompt_Announc(H248.9) Ext_Voice_Prompt = 0 CALEA
Trunk Testing
PSTN Protocols: ISDN IUA 6 CAS V5.2
Control Protocols: MGCP MEGACO H323 SIP
Coders: G723 G729 G728 NETCODER GSM-FR GSM-EFR AMR EVRC-QCELP G727
Chan Type: RTP ATM PCI DSP-CH Num=360
SS7 Links: MTP2 6 MTP3 6 M2UA 6 M3UA 6
```

Step 3. In the **New Key** field, enter the updated license key supplied by AudioCodes.

Step 4. 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 5. To send a license key from your PC to the device, access the license key file using the **Browse** button and click the **Send File** button.

Step 6. To activate the license key, in the main menu on the left, click the **Reset** button. The Reset screen appears.

Step 7. Select the **Burn** option and click the **Reset** button. The device is reset

6.4.7 Save Configuration

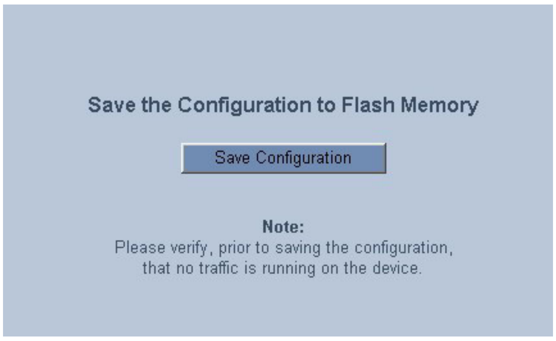
The Save Configuration screen allows users to save the current parameter configuration and the HTTP loaded files to the device flash memory.

> **Note:** If you perform a reset with the **Burn** option selected *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. From the main menu list on the left, click on the Save Configuration link. The Save Configuration screen is displayed.

Figure 6-35: Save Configuration Dialog Screen



Save the Configuration to Flash Memory

Note:
Please verify, prior to saving the configuration, that no traffic is running on the device.

Step 2. Click the **Save Configuration** button. The new/modified configuration and any HTTP loaded files are saved to the Flash memory on the device. A message informing you that it has been saved appears.

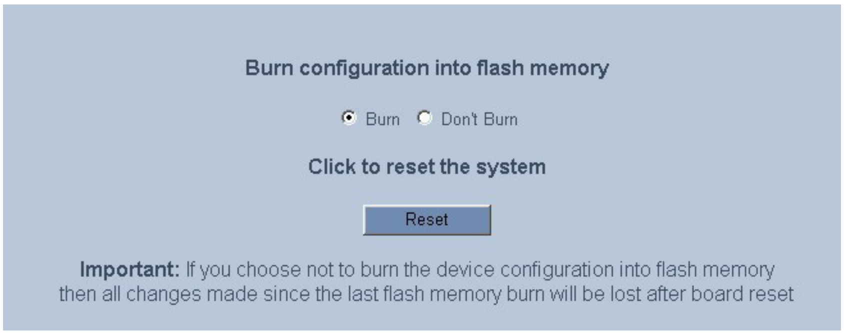
6.4.8 Reset Button

The Reset link in the main menu on the left allows the user to initiate a device reset following which the device utilizes the new configuration stored in the flash memory.

➢ To use the Reset Button screen, take these 3 steps:

Step 1. From the main menu list on the left, click on the **Reset** button. The Reset screen is displayed.

Figure 6-36: Reset Screen



Burn configuration into flash memory

● Burn ○ Don't Burn

Click to reset the system

Important: If you choose not to burn the device configuration into flash memory then all changes made since the last flash memory burn will be lost after board reset

Step 2. Select either of the following options:

- **Burn -** (Default setting) burns the current configuration to flash prior to reset
- **Don't Burn -** Resets the device without first burning the current configuration to flash (discards all modifications to the configuration)

Step 3. Click the **Restart** Button. If the **Burn** option is selected, all of the changes made to the configuration are saved to the Flash memory of the device. If the **Don't Burn** option is selected, all of the changes made to the configuration are discarded. The device is shut down and re-activated. A message about the waiting period is displayed. The screen is refreshed.

7 Configuration Parameters and Files

7.1 Configuration Files for All Call Control Protocols

AudioCodes Mediant 2000 is supplied to customers with factory-preconfigured parameter defaults stored in the device's flash memory. The first time a device is started up, it contains this default configuration. After first-time startup, the configuration can be uploaded from the device and saved to your PC hard disk as an ***ini*** file (using AudioCodes' Embedded Web Server. Refer to Section 6.3 on page 110). This ***ini*** file can be opened in any text file editor, modified and reloaded to the device using AudioCodes' BootP/TFTP Server (refer to Appendix A, on page 211).

You can also define a device's initialization parameters (MAC address, etc.) and the location in your PC of an ***ini*** file before starting up the device. This preconfigured ***ini*** file is then automatically loaded into the device from the BootP/TFTP Server upon startup and every time the device is reset.

The ***ini*** file contains:

- System Parameters (refer to Table 7-1 on page 146)
- Infrastructure Parameters (refer to Table 7-2 on page 147)
- Media Processing Parameters (refer to Table 7-3 on page 152)
- PSTN Parameters (refer to Table 7-4 on page 158)
- SS7 Parameters (refer to Table 7-5 on page 165)
- Common Control Protocols Parameters (refer to Table 7-6 on page 170)
- MGCP Specific Parameters (refer to Table 7-7 on page 173)
- MEGACO Specific Parameters (refer to Table 7-8 on page 175)
- Web Interface Parameters (refer to Table 7-9 on page 178)
- SNMP Parameters (refer to Table 7-10 on page 179)
- SCTP Parameters (refer to Table 7-11 on page 181)
- Names for optional configuration files (CAS signaling, Call Progress Tones and Voice Prompts files).

Users do not have to specify all (or any) of the parameters in the ***ini*** file. If a parameter is left unspecified in an ***ini*** file and the ***ini*** file is then loaded to the device, the device will be configured with that parameter's default value. Leaving all ***ini*** file parameters unspecified and loading the file to a device will thus result in the device being configured with its defaults (contained in the software image ***cmp*** file).

Note: To restore the device's default configuration parameters, use an empty ***ini*** file without any valid parameters or with a semicolon (;) preceding all lines in the file.

Array Parameters

Some parameters have array values. For each of these parameters listed in the parameter tables below, if the ***ini*** file field name is used as is, the parameter applies to all of its elements. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Information about the array value's elements is contained in the Description column.

7.1.1 System Parameters

Table 7-1 lists and describes the system parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 7-1: System Parameters

ini File Field Name	Host/Manual Default Value	Valid Range	Description
CallProgressTonesFilename	Null		Call progress tone filename
DisableWebConfig	0	0 or 1	Enables or disables Web Configuration 0 = Read & Write mode (default) 1 = Read Only mode
DIsableWebtask	0	0 or 1	Enables or disables Web Server Task 0 = Enable (default) 1 = Disable
DNSPriServerIP	0.0.0.0	Legal IP address	IP of primary DNS server IP
DNSSecServerIP	0.0.0.0	Legal IP address	IP of secondary DNS server IP.
EnableSyslog	0	0 or 1	Enable Syslog protocol log. 1 = Activate 0 = Deactivate = default.
NTPServerIP	0.0.0.0	Legal IP address	NTP Server IP address.
NTPServerUTCOffset	0	-43200 to + 43200 sec	NTP time to offset in seconds.
NTPUpdateInterval	86400 seconds	0 to 2592000 sec	NTPupdate interval in seconds. Default is 86400 (24 hours). Not recommended to be set beyond one month (2592000 sec).
SaveConfiguration	1	0 or 1	Determines if the device configuration will be saved in flash 1 = save configuration, 0 = don't save.
SyslogServerIP	0.0.0.0	Legal IP address	IP address in dotted format notation, e.g., 192.10.1.255
SystemOperationStateChangeProfile	0	Integer >0	System Operation state Change Profile: 0 = Disable,1 = Nortel AMS ATM. See acSystemOperationStateChangeProfile enum for possible values.

7.1.2 Infrastructure Parameters

Table 7-2 lists and describes the Infrastructure parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 7-2: Infrastructure Parameters (continues on page 147 to page 151)

ini **File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**
AuthorizedTPNCP Servers	0.0.0.0	Legal IP address	IP address of TPNCP authorized servers.
BaseUDPPort	4000	0 to 55000	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.
BootPRetries	3	1 to 14	Number of BootP Retries that the board sends during start-up. The board will stop issuing BootP requests when either a BootP reply is received or Number Of Retries is reached. This parameter is effective only from the next board reset
BootPSelectiveEnable	0	0 or 1	The Board will only accept BootP replies only from AudioCodes proprietary BootP/TFTP Software. 1 = Enable, 0 = Disable.
DisableNetRefOn TrunkFailure	0	0 or 1	1 = Disables the NetRef signal when PSTN reference trunk fails.
DisableTPNCPEvent	0	nn = TPNCP EventID do hide	This parameter disables Events Reporting for the selected event (refer to: enumacTEvent.
EnableDetectRemote MACChange	0	0 or 1	Allows for the detection of a MAC change of an incoming RTP stream. Used for board redundancy purposes. 0 = disable 1 = enable
EnableDiagnostics	0	0 to 2	The EnableDiagnostics checks the correct functionality of different hardware components on the board. Upon completion, the board sends EV_END_BIT which contains information on the results of the test of each hardware component. 0 = No diagnostics (default). 1 = Perform diagnostics (Full test of DSPs PCM Switch LAN Phy and Flash). 2 = Perform diagnostics (Full test of DSPs PCM Switch LAN Phy, but partial test of Flash, a quicker mode).
EnableICMP UnreachableReport	1	0 or 1	ICMP Unreachable report, reports on reception of ICMP unreachable packets. 0 = Disabled 1 = Enabled

Table 7-2: Infrastructure Parameters (continues on page 147 to page 151)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
EnableIPAddr Translation	1	0 or 1	Relevant to NAT. This parameter specifies the type of compare operation performed on the first packet that is received on a newly opened channel for the NAT feature. If set to one, the board compares the first incoming packet source IP address to the remote IP address stated in the opening of the channel. If the two do not match, the NAT operation takes place meaning both remote IP address and UDP port of the outgoing stream are replaced by the source IP address and UDP port of the first incoming packet.0 = disable 1 = enable.
EnableLANWatchdog	0	0 or 1	Mechanism for detection of LAN failures on the board. A LAN failure can result from a software or hardware malfunction. If a failure of the LAN is detected then board performs self reset (when not in PCI mode). 0 = disable 1 = enable.
EnableTPNCPSecurity	0	0 or 1	Enforces TPNCP to be secure - receiving only pre-determined servers via AuthorizedTPNCPServers parameter, 1 = Enabled, 0 = Disabled (default).
EnableUDPPort Translation	0	0 or 1	Same as EnableIpAddrTranslation only for UDP port. When set compare operation is done on the UDP ports. If set EnableIpAddrTranslation must also be set. 0 = disable 1 = enable.
EthernetPhy Configuration	4	0 to 4	Ethernet type. Auto-Negotiate falls back to Half-Duplex mode (HD) when the opposite port is not in Auto-Negotiate but the speed (10 Base-T or 100 Base-T) in this mode is always configured correctly. 0 = 10 Base-T half-duplex 1 = 10 Base-T full-duplex 2 = 100 Base-T half-duplex 3 = 100 Base-T full-duplex 4 = Auto-negotiate (= Default)
ExtBootPReqEnable	0	0 or 1	Enable extended information to be sent in the BootP request. The Device uses the vendor specific information in the BootP request to provide device related initial startup parameters like board type, current IP address, software version, geographical address, etc. Not available in DHCP.

Table 7-2: Infrastructure Parameters (continues on page 147 to page 151)

***ini* File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**
ForceExceptionDump	0	0 or 1	Force exception dump sends last software exception dump each time board restarts. 0 = Disable 1 = Enable
HeartbeatDestIP	0.0.0.0	Legal IP address	Sets the destination IP to which the heartbeat packets are sent. IP address in dotted notation
HeartbeatDestPort	0	0 to 64000	Sets the destination UDP port to which the heartbeat packets are sent.
HeartbeatIntervalmsec	0xFFFFFFFF	0x0 to 0xffffffff	Sets the time delay in msec between consecutive heartbeat packets
HeartbeatSecondary DestIP	0.0.0.0	Legal IP address	Sets the secondary destination IP to which the heartbeat packets are sent. IP address in dotted notation
ICMPUnreachable ReportInterval	5000	Any number	This parameter determines: (a) the time the board ignores incoming ICMP unreachable packets from the channel activation time, (b) the time it takes between the last ICMP Unreachable packet until the board reports ICMP Reachable
INIFileVersion	0	Any number	This optional field contains the ***ini*** file version number. This ***ini*** version number is reported in the acEV_BOARD_STARTED event
MIIRedundancyEnable	0	0 or1	For platforms other than TP-1610 that have 2 Ethernet ports. Determines whether to activate LAN redundancy or not. 0 = disable 1 = enable.
PCMLawSelect		1 to 3	Selects the type of PCM companding law in input/output TDM bus (TDM bus is defined using the TDMBusType parameter). 1 = A-law 3 = Mu-Law
SubnetBroadcastAfter ENetSOEnabled	0	0 or 1	1 = Enable Subnet Broadcast After Ethernet Switch Over.

Table 7-2: Infrastructure Parameters (continues on page 147 to page 151)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
TDMBusClockSource	1	1 to 14	Selects the clock source on which the board synchronizes. 1 = local oscillator 3 = MVIP 4 = PSTN Network 8 = H.110A 9 = H.110B 10 = NetRef1 11 = NetRef2 12 = SC2M 13 = SC4M 14 = SC8M
TDMBusEnable Fallback	0	0 or 1	Enable/Disable clock fallback. The Fall Back clock is determined by BSPTDMBusFallbackClock. 0 = Disable 1 = Enable
TDMBusFallbackClock	4	4 to 11	Selects the fall-back clock source on which board synchronizes in the event of clock failure. 4 = PSTN Network 8 = H.110A 9 = H.110B 10 = NetRef1 11 = NetRef2
TDMBusLocal Reference		0 to 7	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	0 to 2	Set SC/MVIP/H.100/H.110 to either: 0 = Slave mode (some other board in the system must supply the clock to the TDM bus) or Master mode (the board is the clock source for the TDM bus) or Secondary Master mode (for H100/H110 Bus only). 1 = H110A Master in Master mode. 2 = H.110B Master

Table 7-2: Infrastructure Parameters (continues on page 147 to page 151)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
TDMBusNetRef OutputMode	0	0 to 3	Selects the Netref Output functionality. 0 = Do not output any NetRef, 1 = generation of NetRef 1, 2 = generation of NetRef 2, 3 = generation of both.
TDMBusNetRefSpeed	0	0 to 2	Determine the NetRef frequency (for both generation and synchronization). 0 = 8 kHz 1 = 1.544 MHz 2 = 2.048 MHz
TDMBusOutputPort	0	0 to 15 for SC/ MVIP 0 to 31 for H.110	Defines the SC/MVIP/H.100/H.110 port to be used for board's channel #0. All other channels then occupy the next timeslots sequentially
TDMBusOutput StartingChannel	0	0 to 127	Defines the outgoing TDM Timeslot for board's channel #0. The remaining channels are organized sequentially.
TDMBusSpeed	3	0 to 3	Selects the TDM Bus Speed according to the Bus Type as follows: SC = 0/2/3 H.110/H.100 = 3 VIP = 0, where: 0 = 2048 kbps 2 = 4096 kbps 3 = 8192 kbps 4 = 16384 kbps
TDMBusType	2	0 to 4	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

7.1.3 Media Processing Parameters

Table 7-3 lists and describes the Media Processing parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 7-3: Media Processing Parameters (continues on page 152 to page 157)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
AMDDetection Direction	0	0 or 1	Answer machine detector (AMD) detection direction. 0 = detection from the TDM side. 1 = Detection from the Network side.
ATMG711DefaultLaw Select	0	0 or 1	ATM G711Default Law Select. 0 = A Law, 1 = Mu-Law
BasicRTPPacke tInterval	0	0 to 3	Selects the RTP packet rate for sample based coders (such as G.711, G.726, G.727). Also applicable for G.729, G.729E & G.728. 0 = Default (set internally), 1 = 5 msec, 2 = 10 msec, 3 = 20 msec
BrokenConnection EventActivationMode	0 = activate on first received RTP packet	0 to 1	Determines if Broken connection mechanism will be activated on RTP stream activation = 1 or first received RTP packet = 0. (acTActivateBrokenConnection)
BrokenConnection EventTimeout	3 (= 300 msec)	3 to 21474836 in units of 100 ms	Determines how long the RTP connection should be broken for the Broken Connection event to be issued. Unit is [100 msec]. (300 to 0x80000000 msec)
CallerIDTransportType	3	0 to 3	Type of CallerID Transport. 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. 0 = Disable 1 = Relay 3 = Mute
CallerIDType	0	0 to 18	The supported caller ID standard types. 0 = BellCore, 1 = ETSI, 2 = NTT, 16 = ETSI_ETS, 17 = Denmark, 18 = India
CallProgressDetector Enable	1	0 or 1	Enables or disables detection of Call Progress Tones. 0 = Disable, 1 = Enable

Table 7-3: Media Processing Parameters (continues on page 152 to page 157)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
CASTransportType	0	0 or 1	Controls the ABCD Signaling transport type over IP. 0 = No Relay over the network. 1 = Enable CAS relay RFC 2833
CNGDetectorMode	0	0 to 2	FAX CNG tone Detector mode. 0 = Disable 1 = Relay 2 = Events Only.
Connection Establishement NotificationMode	0	0 or 1	RTP connection establishment event (acEV_CONNECTION_ESTABLISHED) mode. 0 = specify if to notify only after broken connection event. 1 = also upon first RTP packet reception
DisableNAT	1	0 or 1	The User can choose to disable the NAT feature by setting this parameter, 0 = do not disable NAT 1 = disable NAT
DisableRTCP Randomize	0	0 or 1	Controls whether RTCP reports intervals are randomized or each exactly RTCPMeanTxInterval msec, 0 = Randomize 1 = don't Randomize
DJBufMinDelay	150	0 to 150	Dynamic Jitter Buffer Minimum Delay (msec)
DJBufOptFactor	7	0 to 12	Dynamic Jitter Buffer frame error/delay optimization.
DSPVersionTemplate Number	0	0 to 255	Selects the DSP load number. Each load has different Coder list Channel capacity and features supported
DTMFDetctor Sensitivity	0 - Most sensitive. SNR value according to ITU.T Q.24 recommendation	0 to 3	Sensitivity of DTMF detector. 0 = Most Sensitive, to 3 = Least sensitive, SNR value according to ITU.T Q.24 recommendation
DTMFDetectorEnable	1	0 or 1	Enables or disables detection of DTMF signaling.
DTMFTransportType	3	0 to 3	Type of DTMF Transport. 0 = erase DTMFs from voice transport not relayed to remote. 2 = DTMFs not erased not relayed to remote. 3 = DTMFs are muted from the voice stream & relayed according to RFC 2833.
DTMFVolume	-11dBm	-31 to 0 dBm	DTMF generation Volume Control

Table 7-3: Media Processing Parameters (continues on page 152 to page 157)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
ECHybridLoss	0 = 6 dBm	0 to 3	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. 0 = 6 dBm = default 1 = 9 dBm 2 = 0 dBm 3 = 3 dBm
EnableContinuity Tones	0	0 or 1	Enable or Disable Continuity Test tone detection and generation according to the ITU-T Q.724 recommendation. 0 = Disable, 1 = Enable
EnableDSPIPM Detectors	0	0 or 1	Enable/Disable DSP IPmedia Detectors. Enabling this parameter might reduce the board channels capacity. 0 = Disable, =1 = Enable
EnableEchoCanceller	1	0 or 1	Controls the activation of the Echo Canceller 0 = Disable, 1 = Enable
EnablePatternDetector	0	0 or 1	Activates the Pattern Detector. 0 = Disable, 1 = Enable
EnableRFC2658 Interleaving	0	0 or 1	When enabled RTP packets include interleaving byte for VBR coders. 0 = Disable, 1 = Enable
EnableSilence Compression	0	0 to 2	Silence Suppression mode. 0 = Disable. 1 = Enable. 2 = Enable without adaptation.
EnableStandardSID PayloadType	0	0 or 1	If set to 1 the SID packets are sent with the RTP SID type (RFC 3389) 0 = Disable, 1 = Enable
EnableTrunkTesting Tones	0	0 or 1	Enables = 1, Disables = 0, trunk testing tones
FaxBypassPayload Type	102	0 to 127	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) then a TP_SETUP_PARAMETER_INVALID_ERROR is issued and the Payload type is set to the default value (102). It is the User's responsibility to avoid congestion with other payload types.
FaxModemBypas DJBufMinDelay	40 msec	0 to 150 msec	Determines the Jitter Buffer constant delay during Fax & Modem bypass session.

Table 7-3: Media Processing Parameters (continues on page 152 to page 157)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
FaxModemBypassBasicRTPPacketInterval	0	0 to 3	Sets the basic Fax / Modem bypass RTP packet rate, 1 = 5 msec, 2 = 10 msec, 3 = 20 msec (0 = Default (set internally)
FaxModemBypassCoderType	0	0 to 64	The User can set the Fax/Modem bypass coder (according to acTCoders). 0 = G.711 A-law
FaxModemBypassM	1	1 or 2	Number of basic frames to generating one RTP fax/modem bypass packet, 1 or 2
FaxModemRelayVolume	-12	-18 to -3	Fax gain control. The range -18 to -3 relates to -18.5 dBm to -3.5 dBm in 1 dBm steps.
FaxRelayECMEnable	1	0 or 1	Enables or disables the using of ECM mode during Fax Relay. 0 = Disable, 1 = Enable
FaxRelayEnhancedRedundancyDepth	4	0 to 4	Number of repetitions to be applied to control packets when using T.38 standard. 4 = Maximum redundancy
FaxRelayMaxRate	5	0 to 5	Limits the maximum rate at which fax messages are transmitted. 0 = 2400 bps 1 = 4800 bps 2 = 7200 bps 3 = 9600 bps 4 = 12000 bps 5 = 14400 bps
FaxRelayRedundancyDepth	0	0 to 2	Amount of redundancy for fax relay packets 0 = No redundancy
FaxTransportMode	1	0 to 3	Sets the Fax over IP transport method. 0 =Transparent 1 = relay = default 2 = bypass 3 = Transparent with Events.
FlashhookPeriod	400	>0 msec	Flash hook period (detection & generation) in msec
IBSDetectionRedirection	0	0 or 1	Determines the IBS Detection Direction. 0 = PCM, 1 = Network
IdleABCDPattern	NA	0x0 to 0xF	ABCD (CAS) Pattern to be applied on the signaling bus before changed by the user or PSTN protocol. This is only relevant when using PSTN interface with CAS protocols.
IdlePCMPattern	0xFF if Mu-Law,: 0x55 if A-law.	0x00 to 0xFF	PCM Pattern applied to E1/T1 timeslot (B-channel) when channel is idle. 0xFF if PCMLawSelect is Mu-Law, 0x55 if PCMLawSelect is A-law.

Table 7-3: Media Processing Parameters (continues on page 152 to page 157)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
InputGain	0 = No Gain	-32 to +31	PCM input gain. The range is -32 dB to +31 dB in 1 dB steps.
LowDSPResources EventHyst	0	0 to max DSP channels	Number of DSPs between Low and High watermarks of Low DSP Resource Notifications
LowDSPResources EventThreshold	0	0 to max DSP channels	This value determines when Low number of DSP Resource Notification is issued.
MaxDTMFDigitsInCID String	26	0 to 26	Max DTMF Digits in DTMF based Caller ID String
MaxEchoCanceller Length	0	according to acTECLength Lib enumerator	The maximum Echo Canceler Length capability. This parameter may reduce the channel capacity of the product. 0 = decision is made internally to reach maximum channel capacity
MFSS5DetectorEnable	0	0 or 1	Enables or disables detection of MFSS5 signaling. 0 = Disable, 1 = Enable
MFTransportType	3	0 to 3	Type of MF Transport. 0 = erase MFs from voice transport not relayed to remote. 2 = MFs not erased not relayed to remote. 3 = MFs are muted from the voice stream & relayed according to RFC 2833.
MinDTMFDigitsInCID String	0	0 to 26	Min DTMF Digits in DTMF based Caller ID String
ModemBypass PayloadType	103	0 to 127	User can modify the Modem 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) then a TP_SETUP_PARAMETER_INVALID_ERROR is issued and the Payload type is set to the default value (103). It is the User responsibility to avoid congestion with other payload types.
NSEMode	0	0 or 1	Enabling Cisco's NSE Fax/Modem automatic pass-through mode. 0 = Disable, 1 = Enable
NSEPayloadType	105	96 to 127	User can modify the NSE packets RTP Payload type by setting this parameter
PDPattern	NA	0 to 0xFF	Defines the patterns that can be detected by the Pattern Detector
PDThreshold	5	0 to 31	Defines the number of consecutive patterns to trigger the pattern detection event
PrerecordedTonesFile Name	NA	string of ASCII characters	The name (and path) of the file containing the Prerecorded Tones
RFC2198PayloadType	104	96 to 127	Sets the RFC 2198 (RTP Redundancy) packet's RTP Payload Type field
RFC2833RxPayload Type	96	96 to 127	Controls the RFC 2833 Rx Relay RTP Payload type

Table 7-3: Media Processing Parameters (continues on page 152 to page 157)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
RFC2833TxPayload Type	96	96 to 127	Controls the RFC 2833 Tx Relay RTP Payload type
RTPRedundancy Depth	0	0 or 1	Enables or disables generation of RFC 2198 redundancy packets. 0 = Disable, 1 = Enable
SITDetectorEnable	0	0 or 1	Enable or Disable SIT detection according to the ITU-T recommendation E.180/Q.35. 0 = Disable, 1 = Enable
TestMode	3	0 to 3	Testing mode applied. 0 = Coder Loopback performs an encoder/ decoder loopback inside DSP device. 1 = PCMLoopback loops back incoming PCM to the outgoing PCM. 2 = ToneInjection generates a 1000 Hz tone to outgoing PCM. 3 = NoLoopback sets the channel to work in normal mode.
UserDefinedTone DetectorEnable	0	0 or 1	Enables or disables detection of User Defined Tones. 0 = Disable, 1 = Enable
V22ModemTransport Type	2	0 to 3	Sets the V.22 modem transport method. 0 = Transparent 2 = bypass = default 3 = Transparent with Events
V23ModemTransport Type	2	0 to 3	Sets the V.23 modem transport method. 0 = Transparent 2 = bypass = default 3 = Transparent with Events
V32ModemTransport Type	2	0 to 3	Sets the V.32 modem transport method. 0 = Transparent 2 = bypass = default 3 = Transparent with Events
V34ModemTransport Type	2	0 to 3	Sets the V.34 modem transport method. 0 = Transparent 2 = bypass = default 3 = Transparent with Events
VoicePayloadFormat	0	0 to 1	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.
VoicePromptsFile Name	NA	string of ASCII characters	The name (and path) of the file containing the voice prompts
VoiceVolume	0 = No Gain	-32 to +31	Voice output gain control. The range is -32 dB to +31 dB in 1 dB steps. -32 = mute.

7.1.4 PSTN Parameters

Table 7-4 lists and describes the PSTN parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 7-4: PSTN Parameters (continues on page 158 to page 165)

ini **File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**
CASFileName		CASFileName_ 0 to CASFileName_ 7	CASFileName_0 to CASFileName_7 (or CASFileName) are the names (and path names) of the CAS protocol configuration files.
CASTableIndex_xx	0	Less than PSTNCAS TableNum parameter	An array value. The number of elements equals the number of Trunks. Indicates the CAS protocol file to use in the specific Trunk. The index value is according to the number in the field CASFileName_X.
CASTablesNum	0	0 to 8	Indicates how many CAS protocol configuration files are loaded, when 0 is no CAS table. It should match the number of ?CASFILENAME_X? fields.
ClockMaster_xx	0	0 or 1	An array value. The number of elements equals the number of Trunks. Selects the trunk clock source (OFF = clock recovered from the line, ON = the trunk clock source is provided by the internal/TDM bus clock source depending on the TDMBusClockSource parameter.). 0 = acCLOCK_MASTER_OFF 1 = acCLOCK_MASTER_ON.
DCHConfig	0	0 to 2	An array value. The number of elements equals the number of Trunks. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Defines D-channel configuration. This setting is only applicable to ISDN PRI protocols that support NFAS and/or D-channel backup procedures. 0 = D-channel is Primary = default 1 = Backup 2 = NFAS

Table 7-4: PSTN Parameters (continues on page 158 to page 165)

***ini* File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**
DisableTrunkAfter Reset	0	0 or 1	An array value. The number of elements equals the number of Trunks. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Change the Transmit state of the PSTN physical device. Enable, Disable (tri state) or send Blue alarm. 0 = Trunk Enabled 1 = Trunk Disabled.
FramingMethod	0 = default according to protocol type.E1 = E1 CRC4 MultiFrame Format extended G.706B (as c). T1= T1 Extended SuperFrame with CRC6	0, 1, a, b, c, A, C, F	An array value. The number of elements equals the number of Trunks. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. 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 A = T1 4-Frame multiframe C = T1 Extended SuperFrame without CRC6 F = J1 Extended SuperFrame with CRC6 (Japan)
ISDNDuplicate Q931BuffMode	0	See Manual LTRT-84001	Controls the activation/De-activation of Delivering Raw Q.931 messages. Refer to the sub-section 'ISDN Flexible Behavior' in the VoPLib Reference Library User's Manual, LTRT-84001.

Table 7-4: PSTN Parameters (continues on page 158 to page 165)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
ISDNGeneral CCBehavior	0	See Manual LTRT-84001	An array value. The number of elements equals the number of Trunks. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. This is the bit-field used to determine several general ISDN behavior options. Refer to the sub-section 'ISDN Flexible Behavior' in the VoPLib Reference Library User's Manual, LTRT-84001.
ISDNIBehavior	0	See Manual LTRT-84001	An array value. The number of elements equals the number of Trunks. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Bit-field used to determine several behavior options, which influence how the Q.931 protocol behaves. Refer to the sub-section 'ISDN Flexible Behavior' in the VoPLib Reference Library User's Manual, LTRT-84001.
ISDNInCalls Behavior	0	See Manual LTRT-84001	An array value. The number of elements equals the number of Trunks. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. This is the bit-field used to determine several behavior options that influence how the ISDN Stack INCOMING calls behave. Refer to the sub-section 'ISDN Flexible Behavior' in the VoPLib Reference Library User's Manual, LTRT-84001.
ISDNNFASInterface ID	-1	0 to 255	An array value. The number of elements equals the number of Trunks. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Defines the Interface ID works with NS_EXPLICIT_INTERFACE_ID bit. Refer to the sub-section 'ISDN Flexible Behavior' in the VoPLib Reference Library User's Manual, LTRT-84001.

Table 7-4: PSTN Parameters (continues on page 158 to page 165)

ini **File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**
ISDNOutCalls Behavior	0	See Manual LTRT-84001	An array value. The number of elements equals the number of Trunks. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. This is the bit-field used to determine several behavior options that influence how the ISDN Stack OUTGOING calls behave. Refer to the sub-section 'ISDN Flexible Behavior' in the VoPLib Reference Library User's Manual, LTRT-84001.
IUAInterfaceID	-1		An array value. The number of elements equals the number of Trunks. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. IUA trunk interface ID value
LineBuildOut. LOSS	0	0 to 3	An array value. The number of elements equals the number of Trunks. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Selects the line build out loss to be used for this trunk. 0 = 0 dB 1 = 7.5 dB 2 = 15 dB 3 = 22.5 dB
LineBuildOut. OVERWRITE	0	0 or 1	An array value. The number of elements equals the number of Trunks. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Overwrite the framer's XPM registers values (these registers control the line pulse shape). 0 = No over write 1 = Over write

Table 7-4: PSTN Parameters (continues on page 158 to page 165)

ini **File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**
LineBuildOut. XPM0	0	0 to 255	An array value. The number of elements equals the number of Trunks. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Controls the Framer's XPM0 register value (line pulse shape control). Applicable only when TrunkConfig.LineBuildOut.Overwrite=1. Should be used only by expert Users.
LineBuildOut. XPM1	0	0 to 255	An array value. The number of elements equals the number of Trunks. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Controls the Framer's XPM1 register value (line pulse shape control). Applicable only when TrunkConfig.LineBuildOut.Overwrite=1. Should be used only by expert Users.
LineBuildOut. XPM2	0	0 to 255	An array value. The number of elements equals the number of Trunks. Controls the Framer's XPM2 register value (line pulse shape control). Applicable only when TrunkConfig.LineBuildOut.Overwrite=1.Should be used only by expert Users.
LineCode_xx	0	0 to 2	An array value. The number of elements equals the number of Trunks. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. 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 (for T1 or E1 trunks). 2 = use HDB3 line code (for E1 trunks only).

Table 7-4: PSTN Parameters (continues on page 158 to page 165)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
NFASGroup Number	0	0 to 4	An array value. The number of elements equals the number of Trunks. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. 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).

ProtocolType (continued below)	0	0 to 36	An array value.
			The number of elements equals the number of Trunks.
			To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name.
			Sets the PSTN protocol to be used for this trunk. Relevant only when TDMBusType=acFRAMERS Japan Nippon TT Australian Telecom European Hong Kong - HKT Korean Operator European Hong Kong - HKT over T1 QSIG over E1 Telecom New Zealand Flag QSIG over T1:
			NONE = 0
			E1_EURO_ISDN = 1
			T1_CAS = 2
			T1_RAW_CAS = 3
			T1_TRANSPARENT = 4
			E1_TRANSPARENT_31 = 5
			E1_TRANSPARENT_30 = 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

Table 7-4: PSTN Parameters (continues on page 158 to page 165)

ini File Field Name	Host/Manual Default Value	Valid Range	Description

ProtocolType (continued from above)			E1_AUSTEL_ISDN = 17
			E1_HKT_ISDN = 18
			E1_KOR_ISDN = 19
			T1_HKT_ISDN = 20
			E1_QSIG = 21
			E1_TNZ_ISDN = 22
			T1_QSIG = 23
			V5_2_AN = 26
			T1_IUA = 28
			E1_IUA = 29
			T1_EURO_ISDN = 34
			T1_DMS100_MERIDIAN_ISDN = 35
			T1_NI1_ISDN = 36

Q931RelayMode	0	0, ActivateLAPDmessaging, Q931_RELAY_TO_HOST , Layer3_IS_IUA	ISDN level 3 mode
TDMBusPSTNAutoClockEnable	0	0 or 1	Enable or disable the PSTN trunk Auto fallback clock feature. 0 = PSTN_Auto_Clock_Disable 1 = PSTN_Auto_Clock_Enable.
TerminationSide	0	0 or 1	An array value. The number of elements equals the number of Trunks. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. For ISDN only. select the ISDN Termination side. User side = 0 = default Network side = 1

Table 7-4: PSTN Parameters (continues on page 158 to page 165)

ini **File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**

TraceLevel	0	0 to 15	An array value.
			The number of elements equals the number of Trunks.
			To specify each element individually, add **_*xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name.
			Defines the Trace level. acNO_TRACE = 0
			acFULL_ISDN_TRACE = 1
			acLAYER3_ISDN_TRACE = 2
			acONLY_ISDN_Q931_MSGS_TRACE = 3
			acLAYER3_ISDN_TRACE_NO_DUPLICATION = 4
			acFULL_ISDN_TRACE_WITH_DUPLICATION = 5
			acISDN_Q931_RAW_DATA_TRACE = 6
			acISDN_Q921_RAW_DATA_TRACE = 7
			acISDN_Q931_Q921_RAW_DATA_TRACE = 8
			acSS7_MTP2 = 10
			acSS7_MTP2_AND_APPLI = 11
			acSS7_MTP2_SL_L3_NO_MSU = 12
			acSS7_AAL = 15

7.1.5 SS7 Parameters

Table 7-5 lists and describes the SS7 parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 7-5: SS7 Parameters (continues on page 165 to page 169)

ini **File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**
SS7_M2UA_Interface_ID	-1	-1 to 4294967295	An array value. The number of elements equals the number of SS7 links. To specify each element individually, add **_*xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. M2UA Interface ID

Table 7-5: SS7 Parameters (continues on page 165 to page 169)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
SS7_MTP2_Param_ AERM_TIE	1	0 to 10	An array value. The number of elements equals the number of MTP parameter tables. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Alignment Error Rate Monitor emergency error rate threshold
SS7_MTP2_Param_ AERM_TIN	4	0 to 20	An array value. The number of elements equals the number of MTP parameter tables. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. SS7 Alignment Error Rate Monitor (AERM) Normal error rate threshold
SS7_MTP2_Param_ Error_Correction_ Method	0	0, B, P	An array value. The number of elements equals the number of MTP parameter tables. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Basic or Preventive cyclic retransmission
SS7_MTP2_Param_ IAC_CP	5	0 to 10	An array value. The number of elements equals the number of MTP parameter tables. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Number of aborted proving attempts before sending out-of-service to MTP-3
SS7_MTP2_Param_ Link_Rate	0	0, A to D	An array value. The number of elements equals the number of MTP parameter tables. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. SS7 SLI Link Rate A = 64 kbps, D = 56 kbps
SS7_MTP2_Param_ LSSU_Length_xx	1	1 to 2	An array value. The number of elements equals the number of MTP parameter tables. SS7 MTP2 LSSU length 1 or 2 (bytes)

Table 7-5: SS7 Parameters (continues on page 165 to page 169)

ini **File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**
SS7_MTP2_Param_ Octet_Counting_xx	16	0 to 256	An array value. The number of elements equals the number of MTP parameter tables. Octet counting N octets (# of Octets received while in octet counting mode)
SS7_MTP2_Param_ Reserved1	0	0 to 255	SS7 MTP2 PARAM reserved 1
SS7_MTP2_Param_ Reserved2	0	0 to 255	SS7 MTP2 PARAM reserved 2
SS7_MTP2_Param_ SUERM_SU_D_xx	0	0 to 256	An array value. The number of elements equals the number of MTP parameter tables. SS7 Signal Unit Error Rate Monitor: D threshold
SS7_MTP2_Param_ SUERM_T_xx	64	0 to 256	An array value. The number of elements equals the number of MTP parameter tables. SS7 Signal Unit Error Rate Monitor (SUERM) error rate, T threshold
SS7_MTP2_Param_ Timer_T1_xx	50000	0 to 100000	An array value. The number of elements equals the number of MTP parameter tables. SS7 MTP2 T1 alignment ready timer (msec)
SS7_MTP2_Param_ Timer_T2_xx	150000	0 to 200000	An array value. The number of elements equals the number of MTP parameter tables. SS7 MTP2 T2 timer not aligned (msec)
SS7_MTP2_Param_ Timer_T3	2000	0 to 20000	An array value. The number of elements equals the number of MTP parameter tables. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. SS7 MTP2 T3 timer aligned (msec)
SS7_MTP2_Param_ Timer_T4E	500	0 to 5000	An array value. The number of elements equals the number of MTP parameter tables. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. SS7 MTP2 T4e Emergency proving period timer (msec)

Table 7-5: SS7 Parameters (continues on page 165 to page 169)

***ini* File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**
SS7_MTP2_Param_ Timer_T4N	8200	0 to 15000	An array value. The number of elements equals the number of MTP parameter tables. To specify each element individually, add _***xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. SS7 MTP2 T4n Normal proving period timer (msec)
SS7_MTP2_Param_ Timer_T5	120	0 to 2400	An array value. The number of elements equals the number of MTP parameter tables. To specify each element individually, add _***xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. SS7 MTP2 Sending SIB timer (msec)
SS7_MTP2_Param_ Timer_T6	6000	0 to 10000	An array value. The number of elements equals the number of MTP parameter tables. To specify each element individually, add _***xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. SS7 MTP2 Remote Congestion timer (msec)
SS7_MTP2_Param_ Timer_T7	2000	0 to 5000	An array value. The number of elements equals the number of MTP parameter tables. To specify each element individually, add _***xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. SS7 MTP2 excessive delay of ack timer (msec)
SS7_Signaling_Link_ Number	-1	-1 to 7	An array value. The number of elements equals the number of SS7 links. To specify each element individually, add _***xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Unique number of SS7 Link
SS7_SLI_L3_Entity_ ID	1	1 or 3	An array value. The number of elements equals the number of SS7 links. To specify each element individually, add _***xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Identifier of the layer above MTP-2

Table 7-5: SS7 Parameters (continues on page 165 to page 169)

***ini* File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**
SS7_SLI_MTP2_ Param_Index	3	0 to 3	An array value. The number of elements equals the number of SS7 links. To specify each element individually, add **_*xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Index to table MTP-2_Param (default: 3)
SS7_SLI_Trace_ Level	0	1 to 15	An array value. The number of elements equals the number of SS7 links. To specify each element individually, add **_*xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. PSTN trace level for this SLI
SS7_SLI_Variant	0	0 to 1	An array value. The number of elements equals the number of SS7 links. To specify each element individually, add **_*xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. MTP2 variants
SS7_Time_Slot_Of_ SLI	-1	-1 to 31	An array value. The number of elements equals the number of SS7 links. To specify each element individually, add **_*xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Time Slot where the SS7 link is located
SS7_TrunkID_Of_ SLI	-1	-1 to 14	An array value. The number of elements equals the number of SS7 links. To specify each element individually, add **_*xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Trunk ID where the SS7 link is located
SS7M3BDebugFlags	0		Container parameter for SS7 debug flags. Controls SS7 Debug Outputs.

7.1.6 Parameters Common to All Control Protocols

Table 7-6 lists and describes the parameters, contained in the ***ini*** file, that are common to all call control protocols. Use this table as a reference when modifying ***ini*** file parameter values.

Table 7-6: Control Protocol Parameters (continues on page 170 to page 172)

ini **File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**
AdminState	2	0 or 2	Determines the gateway's initial administrative state. 0 = locked, 2 = unlocked
AdminStateLock Control	-1	-1, 0, > 0	Time left (in seconds) for the shutdown to complete. 0 is an immediate shutdown, (-1) will wait until all calls drop and positive number is the number of seconds to wait.
CallAgentDomain Name	""	String[63]	This parameter defines a domain name to be used for connecting with the Call Agent. The parameter takes precedence upon the Call agent IP and the provisioned call agent parameters.
CallWaitingTone Duration	12000	300 to 30000 msec	Changes the call waiting tones family duration in msec.
ControlDiffServ	0	0 to 63	Defines the value of the 'DiffServ' field in the IP header for the control session.
CPPlayAnnouncement ToNetworkSide	0	0 or 1	MGCP: While On (1), force all announcements to be played towards network side
CPTransportType	0	0 or 1	Defines the transport type for the control: 0 = UDP, 1 = TCP (SCTP on road map).
CPTrunkIdOffset	0	0 or 2	Sets the offset to the Trunk numbering. For example, CPTRUNKIDOFFSET=2 will cause the first trunk number to be 2.
DefaultPacketization Period	20 msec (for G.723 30)	5 to 80 msec	Default packetization period (Frame Size)
DigitMapTimeoutTimer	16 sec	1 to 65535 sec	Defines the timeout value (T symbol) in digit map. For MEGACO it is the start timer, and for the rest it is the end timer.
DTMFDigitLength	100	0 to 65535	Time to play DTMF in msec.
DTMFInterDigitInterval	100	0 to 65535	Time between DTMFs played in msec.
EnableCallerIDType Two	1	0 or 1	MGCP: When Off ,Caller ID type two will not be played if asked from Call Agent 0 = off, 1 = on
EndpointName	MGCP: default = "Acgw" MEGACO: default = 'line/' for analog board and '/c' for Trunking boards	String[19]	MGCP: Gateway's end point name. This is a prefix used to identify the endpoint, i.e., 'ACgw' in the following example: 'ACgw5@acl.com'. MEGACO: Prefix of the endpoint part of the termination name

Table 7-6: Control Protocol Parameters (continues on page 170 to page 172)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
EndpointPrefix	default =""	String[19]	MGCP: Together with 'TrunkName' field generates local endpoint name on trunk enabled gateways
GatewayName	MGCP: default = Audio-Codes.com MEGACO: default = NULL for analog board and 'tgw/' for Trunking boards	String[63]	MGCP:Gateway's identification name towards the MGCP Call Agent. If empty Gateway name holds the IP address of the board MEGACO: Prefix of the gateway part of the termination name
IPDiffServ	0	0 to 63	Defines the value of the 'DiffServ' field in the IP header for the media session.
IPPrecedence	0	0 to 7	Sets value of the IP precedence field in the IP header for all packets generated from this channel.
IPTOS	0	0 to 15	Sets value of IP Type Of Service field in IP header for all packets generated from this channel.
KeepAliveEnabled	0	0 or >0	This parameter can be used to enable a KeepAlive message (NOP ServiceChange). 0 = disable, >0 = enable
KeepAliveInterval	12 sec	1 to 300 sec	This parameter is used to define the interval in seconds of a KeepAlive message
MGControlProtocol Type	1	0 to 8	This parameter defines the control protocol type 0 = none, 1 = MGCP, 2 = MEGACO, 4 = H323, 8 = SIP (For H.323 and SIP contact AudioCodes)
MGCPCommunication LayerTimeout	30	>0 sec	This parameter defines the maximal time to wait for a response before declaring a disconnection.
MGCPCompatibility Profile	1	Integer >0	Use this parameter to control MGCP/ MEGACO behavior for specific compatibility. See user manual or mgTMGCPProfile enum for possible values.
MGCPDefaultCoder	“G.711”		The User can set a default coder for channel opening. Valid coder name according to the list in the user manual.
MGCPDefault PacketizationPeriod	See Default-Packetization-Period	See Default-Packetization-Period	Default packetization period (Frame Size)
MGCPDTMF DetectionPoint	1	0 or 1	Defines if the detection of DTMF events is notified at the end of DTMF or at the start of DTMF. 0 = at start of DTMF 1 = at the end of DTMF = default.

Table 7-6: Control Protocol Parameters (continues on page 170 to page 172)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
MGCPRetransmission Timeout	200 msec	0 to 10000 msec	Sets initial time for first retransmission. The Retransmission intervals thereafter increase exponentially.
ProvisionedCall Agents	0	Legal IP Address	List of up to 10 (MGCP) or 5 (MEGACO) legal IP addresses separated by “,” or by “;” for the ServiceChange command.
ProvisionedCall AgentsPorts	2944	0 to 65535	List of up to 10 (MGCP) or 5 (MEGACO) Call Agent separated by “,” or by “;”
Randomize TransactionID	1	0 or 1	Defines if the transactions produced by the board start with a fixed number or random number. Refer also to 'TransactionIdRange' and 'TransactionIdBase'
RedundantCallAgent DomainName	""	String[63]	Redundant MGCP Call Agent Domain name
RTCPInterval	5000 (= 5 sec)	0 to 65535 msec	Time interval between the adjacent RTCP reports. Set in msec.
SingleSIDPacketWith SCEG729	0	0 or 1	If parameter is set to '1' 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	2000	> 0	Defines the minimum number for transaction ID.
TransactionIDRange	999999999	> 0	Defines the range for the transaction ID.
TransparentCoder PayloadType	116	0 to 127	Alternative payload type to use as transparent coder.
TrunkName	MGCP: default = "" MEGACO: default = "" for analog boards and = 's' for Trunking boards	String[19]	MGCP: Together with 'EndpointPrefix' field generates local endpoint name on trunk enabled gateways MEGACO: Prefix of the trunk part of the termination name
UseTransparentCoder WithHBR	0	0 or 1	If parameter is On (1 = use), and connection is using high bit rate coders. DTMF transport type is set to Transparent.Coders list: G711Alaw_64,G711Mulaw,G726_16,G726_24,G726_32,G726_40,G727_16,G727_24_16,G727_24,G727_32_16,G727_32_24,G727_32,G727_40_16,G727_40_24,G727_40_32. 0 = do not use 1 = use

7.1.7 MGCP-Specific Parameters

Table 7-7 lists and describes the MGCP-specific parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 7-7: MGCP-Specific Parameters (continues on page 173 to page 174)

ini **File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**
ActivateallChannels OnBoardInit	0	0 or 1	Activate (= 1) all DSP's on board init. Used in order to perform signals/events operations perior to CRCX. Default = 0 (do not activate).
CallAgentIP		Legal IP address	The Call Agent IP address in dotted notation. to be used for the initial restart in progress message (RSIP). Set to 0.0.0.0 to avoid sending RSIP. This parameter overrides the Bootp's Call Agent IP address, if provided.
CallAgentPort	2427	0 to 65534	The Call Agent port number. Defaults to the MGCP default port number, 2427.
ClearRequestBuffer	1	0 or 1	If this parameter is OFF (= 0) only empty R: will clear the event list and only empty S: will clear and stop the current signals list. If ON (= 1 = default) if encapsulated identifier (X:) is present in the command all TO signals and all events will be cleared
ConnectionIDBase	20	> 0	Defines the lowest number for the Connection ID values assigned by the gateway.
ConnectionIDRange	999999999	> 0	Defines the range for the Connection ID values assigned by the gateway.
DefaultSecretKey Method	0	0 or 1	Set RTP RTCP default method of key generation, base64 = 0 (default), clear = 1
GatewayMGCPPort	2427	0 to 65535	If User wants to force Media Gateway to listen to another UDP port instead of the original 2427 as defined in RFC 2705.
LongDurationEvent Time	3600	0 & > 0.	Default time to trigger the long duration event (in second)
MGCPEndPoint NumberingOffset	0	> 0	User can add offset to endpoints. This functions only with End point naming configuration. Use this parameter With Trunk Naming configuration is forbidden
MGCPPersistent Events	MPs: 'L/hd,L/hu,L/hf' Others: Empty.	String [127]	A list of events which shall be used as persistent events. Format is same as the requested event parameter (R:), such as "D/x(N)".
MGCPSendMACWith RSIP	0	0 or 1	When this parameter exists in the ***ini*** file generated RSIPs have the MGs MAC address in addition to the usual parameters. This parameter is sent as an MGCP extension parameter. 1 = include MG's MAC address 0 = doesn't include MAC

Table 7-7: MGCP-Specific Parameters (continues on page 173 to page 174)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
MGCPVersion	"MGCP 1.0"	String[39]	MGCP protocol version
MGHistoryBufferTimeLim	30	0 & > 0.	The time a transaction is kept in the history buffer
QuarantineModeState	0	0 or 1	Sets the default quarantine handling state. When set, the quarantine handeling state will be set to LockStep, otherwise it will be set to loop and discard. 0 = loop/discard(default) 1 = lockstep
RedundantAgentIP		IP address in dotted format notation	Redundant Call Agent IP address to be used for the initial restart in progress message (RSIP). Set to 0.0.0.0 to avoid sending RSIP.
RedundantAgentPort	2427	0 to 65534	The redundant Call Agent port number. Defaults to the MGCP default port number, 2427.
RSIPOnNetwork Disconnection	1	0 or 1	This flag specifies whether to send or don't send RSIP when LAN is re-connected. 0 = Do not send 1 = Send RSIP
UseBracketsWith GatewayName	1	0 or 1	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] 0 = Off, 1 = On
UseNewFormatCoder Negotiation	1	0 or 1	Disable the response of all coders description in CRCX\MDCX without coder and SDP as described in CoderNegotiation on 2705 0 = Do not use new format 1 = use new format
UseWildCardWithRSIP	1	0 or 1	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 than 64 RSIP message sent for each Endpoint. 0 = Do not use, 1 = Use

7.1.8 MEGACO-Specific Parameters

Table 7-8 lists and describes the MEGACO-specific parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 7-8: MEGACO-Specific Parameters (continues on page 175 to page 177)

ini **File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**
AASPackagesProfile	0	0 or 1	Profile for choice between AAS Package spec. standard. 0 = TD-51 standard spec. 1 = H.248.9 standard spec.
ATM_Bit_Field_ Size	0	0 to 30	An array value. The number of elements is 2. To specify each element individually, add _***xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. For Binary MEGACO: Bit field size for each ATM termination name level
ATM_Num	0	Any positive number	An array value. The number of elements is 2. To specify each element individually, add _***xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Starting number for the ATM termination name
DialToneDuration	16 sec	1 to 65535 sec	This parameter overrides default 180 sec time out defined for the dial tone signal.
DigitMapName	NULL	String[10]	Name of the provisioned digit map.
DigitMapping	NULL	String[151]	The digit map patterns separated by a vertical bar (|), as defined in the MEGACO RFC.
EP_Bit_Field_Size	0	0 to 30	An array value. The number of elements is 5. To specify each element individually, add _***xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. For Binary MEGACO: Bit field size for each name level (level 0 is the left one, i.e., Trunk number). The total binary name is 32 bits.
EP_Max	0	0 & > 0	Maximal number for each name level (level 0 is the left one, i.e Trunk number)
EP_Min	0	0 & > 0	Minimal number for each name level (level 0 is the left one, i.e Trunk number)

Table 7-8: MEGACO-Specific Parameters (continues on page 175 to page 177)

ini **File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**
EP_Num	0	Any positive number	An array value. The number of elements is 5. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Starting number for each name level (level 0 is the left one, when looking at the "PhysTermNamePattern" parameter. So to start the trunk numbering from 1, EP_NUM_0 will be set to 1.
LogicalATMTerm Pattern	NULL	String [30]	Defines the name pattern of an ATM termination
LogicalRTPTerm Pattern	NULL	String [30]	Defines the name pattern of an RTP termination. For example: "gw/rtp/*". The "*" sign stands for the actual number of the RTP termination.
MEGACO_MID	NULL	String[64]	Gateway's MID (See protocol spec) towards the MGCP/MEGACO Call Agent. If empty or illegal, MID holds the IP address of the board
MEGACOASN1Profile	1	Integer >0 See user manual.	Used for profiling of the binary ASN.1 encoding. See user manual for possible values.
MEGACOCheck LegalityOfMGC	0	0 or 1	This parameter specified if the MEGACO rejects commands from an MGC not in the provisioned list (1 = Reject)
MEGACOContextID Offset	0	0 to 4294967295	Offset for the context ID generated by the gateway. E.g., offset = 100 causes the first context to be 101.
MEGACOEncoding	0	0 or 1	Sets the MEGACO Coding method. 1 = support the binary ASN.1 format of the MEGACO protocol, 0 = Text mode.
MEGACOTermination IDOffset	0	0 to 4294967295	Offset for the ephemeral terminations IDs in the gateway. E.g., offset = 100 causes the first ephemeral termination ID to be 101. Note: This parameter was replaced by the parameter "RTP_Num".
MEGACOTrunkID Offset	0	0 to 4294967295	Sets the offset to the Trunk numbering. E.g., Offset = 2 causes the first trunk number to be 2. Note: This parameter was replaced by the parameter "EP_NUM".
MGCExecutionTime	100 msec	0 to 2000 msec	Estimated execution time of MGC (in msec)
MGCProvisional ResponseTime	100 msec	0 to 20000 msec	Provisional response timer for MGC (in msec)
MGExecutionTime	100 msec	0 to 2000 msec	Estimated execution time of Media Gateway (in msec)

Table 7-8: MEGACO-Specific Parameters (continues on page 175 to page 177)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
MGProvisional ResponseTime	100 msec	0 to 20000 msec	Provisional response timer for Media Gateway (in msec)
PhysTermName Pattern	NULL	String [30]	Defines the name pattern of a physical termination. Example: "tgw/t*/c*". The "*" sign stands for the actual numbers of the trunk and B-channel.
RTP_BIT_Field_ Size	0	0 to 30	An array value. The number of elements is 2. To specify each element individually, add **_*xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. For Binary MEGACO: Bit field size for the RTP termination name
RTP_Num	0	Any positive number	An array value. The number of elements is 2. To specify each element individually, add **_*xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Starting number for the RTP termination.

7.1.9 Web Interface Parameters

Table 7-9 lists and describes the Web Interface parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 7-9: Web Interface Parameters

ini **File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**
BKGImageFIleName	NULL	String[47]	GIF/JPEG image file name to replace the background appearing in upper bar of the device Web interface pages. (Note: 1 = background height should be 85 pixels 2 = background image is duplicated alongside to fit screen width).
HTTPPort	80	1 to 65535 (other restrictions may apply in this range)	Determine the local HTTP port of the device.
LogoFileName	NULL	String[47]	GIF/JPEG image file name to replace AudioCodes Web logo image appearing in upper left hand corner of the device Web interface pages. (Note: Image height should be 85 pixels).
LogoWidth	441	Up to 9999 pixels	The logo image (upper left hand corner of Web interface pages) width in pixel units.
UseProductName	0	0,1	If set to 1 it will enable the userProductName string to override any AudioCodes defaults. If set to 0 userProductName string will have no effect on the product name.
UseRProductName	NULL	String[29]	A string of characters to replace the default AudioCodes product name appearing in the upper right hand corner of the device Web interface pages.
UseWeblogo	0	0,1	If set to 1 it will enable the webLogoText string to override any loaded logo image file (and AudioCodes default logo image). If set to 0 webLogoText string will have no effect on the logo image.
WebLogoText	NULL	String[15]	A string of characters to replace the default AudioCodes logo image appearing in the upper left hand corner of the device Web interface pages. (Note: 1 = this string will also replace the AudioCodes name in the title bar).

7.1.10 SNMP Parameters

Table 7-10 lists and describes the SNMP parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 7-10: SNMP Parameters (continues on page 179 to page 180)

ini **File Field Name**	**Host/Manual Default Value**	**Valid Range**	**Description**
ActiveAlarmTableMaxSize	10	10 to 20	Determines the maximum number of rows in the Active Alarm Table.
AlarmHistoryTableMaxSize	500	50 to 1000	Determines the maximum number of rows in the Alarm History Table. The parameter is controllable via the nlmConfigGlobalEntryLimit MIB parameter (located in the notificationLogMIB).
DisableSNMP	0	0 or 1	If the field name exists then disable SNMP access. 0 = No, 1 = Yes
RemoteResetControl	1	1 to 3	Determine the remote reset action. 1 = burn staged load image, if any, and config to flash, perform a reset, and load from flash 2 = perform a reset, and load from flash (staged load image, if any, and config is not burned to flash) 3 = perform a reset, and load from BootP/ TFTP (staged load image, if any, and config burn is not needed) * = an error is returned for any other value Note: The staged load image is an image of a cmp file that was previously loaded to an area in RAM via an HTTP upload.
SetCommunityString	private	String[19]	User determined community string with access limited to ***ini*** file entered values only. This parameter is the singular version of the readWriteCommunityStrings, and corresponds to: readWriteCommunityStrings_0
SNMPManagerIP		string[15]	IP address of default SNMP manager in dotted notation format: xxx.xxx.xxx.xxx. SNMP traps are sent to this manager.
SNMPManagerTrapPort	162	100 to 3999	An array value. The number of elements is 5. To specify each element individually, add _***xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Sets the trap ports to be used by the different managers. This parameter is the tabular version of the SNMPTrapPort parameter.
SNMPManagerTableIP	0.0.0.0	String[15]	SNMP manager server IP address. This is the tabular version of the SNMPManagerIP parameter.

Table 7-10: SNMP Parameters (continues on page 179 to page 180)

ini File Field Name	Host/Manual Default Value	Valid Range	Description
SNMPManagerTrap SendingEnable	1	0 or 1	An array value. The number of elements is 5. To specify each element individually, add ***_xx*** (***xx*** equals the element number) to the end of the ***ini*** file field name. Enable the SNMP Manager's IP to have traps sent to them.
SNMPPort	161	100 to 3999	This parameter specifies the port number for SNMP requests and responses. Usually should not be specified, use default.
SNMPReadOnly CommunityString	public'	String[19]	Read-only community string
SNMPReadWrite CommunityString	private'	String[19]	Read-write community string
SNMPTrap CommunityString		String[19]	Community string used in traps
SNMPTrapPort	162	100 to 3999	This parameter specifies the port number for SNMP Traps. Usually should not be specified, use default.
SNMPTrustedMGR	0.0.0.0	String[15]	IP address of trusted SNMP manager

7.1.11 SCTP Parameters

Table 7-11 lists and describes the SCTP parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 7-11: SCTP Parameters

ini **File Field Name**	**Class Type**	**Host/Manual Default Value**	**Valid Range**	**Description**
SCTPAssociationsNum	Value	3	1 to 8	Maximum number of SCTP associations that could be opened
SCTPDNetNum	Value	3	1 to 3	Maximum number of association transport addresses that could be active
SCTPHBInterval	Value	30	1 to 3600	SCTP heartbeat interval
SCTPISTRMNum	Value	200	1 to 200	Maximum number of incoming streams
SCTPMaxAssocInitAttempts	Value	5000	5 to 10000	Maximum number of SCTP associations initialization attempts
SCTPMaxAssocRet	Value	10	5 to 20	Maximum number of SCTP associations retrans attempts
SCTPMaxDataChunkSize		500	50 to 1450	Maximum length of SCTP data chunk
SCTPOSTRMNum	Value	200	1 to 200	Maximum number of outgoing streams
SCTPOutChunksNum	Value	630	50 to 630	Maximum number of outgoing chunks
SCTPPortsNum	Value	5	1 to 5	Maximum number of SCTP endpoints that could be opened
SCTPT4SAckTimer	Value	3	1 to 5	SCTP 4 stack timer interval

7.2 Initialization File (*ini* files)

7.2.1 *ini* File Structure

The ***ini*** file can contain any number of parameters. The ***ini*** file structure supports both **Parameter = Value** (refer to Section 7.2.1.1 on page 182) and **Tables of Parameter Values** Constructs (refer to Section 7.2.1.2 on page 183). The Figure 7-1 illustrates the general structure of the ***ini*** file for both the Parameter = Value and Tables of Parameter Values Constructs.

Figure 7-1: *ini* File Structure

```
[Sub Section Name]
Parameter_Name = Parameter_Value
Parameter_Name = Parameter_Value
.
..

; REMARK

 [Sub Section Name]
...

; Tables Format Rules:
[Table_Name]
; Fields declaration
Format Index_Name_1 ... Index_Name_N = Param_Name_1 ... Param_Name_M
; Table's Lines (repeat for each line)
Table_Name Index_1_val ... Index_N_val = Param_Val_1 ... Param_Val_M
[\Table_Name]
```

7.2.1.1 Parameter = Value Rules in the *ini* File Structure

- Lines beginning with a semi-colon ';' (as the first character) are ignored.
- An Enter must be the final character of each line.
- The 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 incorrect values).
- Sub-section names are optional.
- String parameters, representing file names, for example, CallProgressTonesFileName, must be placed between two inverted commas ('…').
- The parameter name is NOT case sensitive; the parameter value is not case sensitive except for coder names.
- Numeric parameter values should be entered only in decimal format.
- The ***ini*** file should be ended with one or more Enters.

7.2.1.2 About Tables of Parameter Values

Tables of Parameter Values group related parameters of a given entity and handle them together. Tables are composed of lines and columns. The columns represent parameters types, while each line represents an entity. The parameters in each line are called the line attributes. Lines in table may represent (for example) a trunk, SS7 Link, list of timers for a given application, etc.

Table 7-12 and Table 7-13 below provide useful examples for reference.

Note: Table 7-12 and Table 7-13 are provided as examples for the purpose of illustration only and are NOT actually implemented in AudioCodes products.

Table 7-12: Table of Parameter Values Example - Remote Management Connections

Index Fields:

1. Connection Number

Connection Number	User Name	User Password	Time Connected (msec)	Permissions
0	Admin	Yellow9	0	All
1	Gillian	Red5	1266656	Read Only
2	David	Orange6	0	Read Write

Table 7-13: Table of Parameter Values Example - Port-to-Port Connections

Index Fields:

1. Source Ports

2. Destination IP

3. Destination Port

Source Port	Destination IP	Destination Port	Connection Name	Application Type
2020	10.4.1.50	2020	ATM_TEST_EQ	LAB_EQ
2314	212.199.201.20	4050	ATM_ITROP_LOOP	LAB_EQ
6010	10.3.3.41	6010	REMOTE_MGMT	MGMT

7.2.1.2.1 Table Indices

Each line in a table must be unique. For this reason, each table defines one or more Index fields. The combination of the Index fields determines the 'line-tag'. Each line-tag may appear only once.

In the example provided in Table 7-12 on page 183, there is only one index field. This is the simplest way to mark lines.

In the example provided in Table 7-13 on page 183, there are three Index fields. This more complicated method is a result of the application it represents.

7.2.1.2.2 Table Permissions

Each field in a line has a 'permission' attribute which determines if and when the user may modify the field.

There are several types of permissions:

- **Read -** The user may read the value of a field (true for all fields).
- **Write -** The user may modify the value of the field at any time.
- **Create -** The user must provide a value for the field at creation time.

The default values set for all fields already determine the initial values.

- **Maintenance write** - The user may modify the value of the field only when the entity represented by the line is in maintenance state.

Each table includes rules to determine when it is in a maintenance state.

In the example in Table 7-12 on page 183, ‘User Name’ and ‘User Password’ fields have Read-Create permissions. The ‘Time Connected’ field has Read-Only permission, and the ‘Permissions’ field has a Read-Create-Maintenance_write permission.

7.2.1.3 Tables of Parameter Value Rules in the *ini* File Structure

The ***ini*** file allows you to add/modify parameters in tables. When using tables, Read-Only parameters are not uploaded, since they cause an error when trying to download the uploaded file. Therefore read-only parameters should not be included in tables in the ***ini*** file. Consequently, tables are uploaded with all parameters having at least one of the following permissions:

- Write
- Create
- Maintenance write

The ‘format-line’ rule defines which fields of the table are to be modified by the given ***ini*** file (this may vary among ***ini*** files for the same table). The ‘format-line’ must only include fields which can be modified (which are all parameters that are not specified as read-only).

One exception is the index-fields which are ALWAYS mandatory fields. In the example provided in Table 7-12, all fields except the ‘Time Connected’ field are uploaded.

7.2.1.3.1 Tables Structure Rules

Tables are composed of four elements:

- **Table-Title -** The Table's string name in square brackets (e.g. [MY_TABLE_NAME]).
- **Format Line -** This line specifies the table's fields by their string names.
  - The first word MUST be "FORMAT", followed by indices field names, and after '=' sign, all data fields names should be listed.
  - Items must be separated by ',' sign.
  - The Format Line must end with ';' sign.
- **Data Line(s) -** The actual values for parameters are specified in each Data line. The values are interpreted according to the format line. The first word must be the table’s string name.
  - Items must be separated by a ',' sign.
  - A Data Line must end with a ';' sign.
- End-of-Table-Mark: Marks the end of a table. Same as Table title, but string name is preceded by '\'.

Figure 7-2, on page 185 displays an example of the Table structure in an ***ini*** file.

Figure 7-2: Structure of a Table in an *ini* File

```
; Table: Items Table.
; Fields: Item_Name, Item_Serial_Number, Item_Color, Item_weight.
; NOTE: Item_Color is not specified. It will be given default value.
[Items_Table]
; Fields declaration
Format Item_Index =  Item_Name, Item_Serial_Number, Item_weight;
Items_Table 0 = Computer, 678678, 6;
Items_Table 6 = Computer-screen, 127979, 9;
Items_Table 2 = Computer-pad, 111111, $$;
[\Items_Table]
```

- Indices (in both the Format line and the Data lines) must all appear in order, as determined by the table's specific documentation. The Index field must NOT be omitted.
- Data fields in the Format line may use a sub-set of all of the configurable fields in a table only. In this case, all other fields are assigned with the pre-defined default value for each configured line.
- The order of the Data fields in the Format line is not significant (unlike the Index-fields). Field values in Data lines are interpreted according to the order specified in the Format line.
- The sign '$$' in the Data line means that the user wants the pre-defined default value assigned to the field for the given line.
- The order of Data lines is insignificant.
- Data lines must match the Format line, i.e., it must contain exactly the same number of Indices and Data fields and should be in exactly the same order.
- A line in a table is identified by its table-name and its indices. Each such line may appear only once in the ***ini*** file.
- Tables' dependencies:

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

7.2.1.3.2 Dynamic Tables versus Static Tables

Static Table

The Static table type does not support adding new lines or removing (deleting) an existing line. All lines in a Static table are pre-configured with default values. Users may modify values in existing lines. After reset, all lines in a Static table are available.

Dynamic Table

The Dynamic table type supports adding and removing lines. It is always initialized as an empty table, with no lines. Users should add lines to the Dynamic table via the ***ini*** file or at run-time.

Note: Certain Dynamic tables may initialize a line (or more) at start-up time. If so, it is documented in the table's specific section.

7.2.1.3.3 Tables in the Uploaded *ini* File

Tables are grouped according to the applications they configure. For example, several tables are required to configure SS7, and other tables are required to configure ATM.

When uploading the ***ini*** file, the policy is to include only tables that belong to applications which have been configured (Dynamic tables of other applications are empty, but static tables are not). The trigger for uploading tables is further documented in the applications' specific sections.

7.2.1.4 *ini* File Example

Figure 7-3 shows an example of an ***ini*** file.

Figure 7-3: Example of an *ini* File for MGCP

```
[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'
VoicePromptsFilename = 'tpdemo_723.dat'
CasFilename = 'E_M_WinkTable.con'
Trace Level = 0
```

Figure 7-4 shows an example of an ***ini*** file for MEGACO.

Figure 7-4: *ini* File Example for MEGACO

```
[MEGACO]


; List of Call agents, separated by ','. The default is the loading
computer.

PROVISIONEDCALLAGENTS = 10.2.1.254

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

PROVISIONEDCALLAGENTSPORTS = 2944

; IP of the LOG server

LOGSERVERIP = 10.2.1.254

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

; 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 in 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 = 2
```

Note: Before loading an ***ini*** file to a device, make sure that the extension of the ***ini*** file saved on your PC is correct: Verify that the checkbox **Hide extension for known file types** (**My Computer>Tools>Folder Options>View**) is unchecked. Then, verify that the ***ini*** file name extension is ***xxx.ini*** and NOT erroneously ***xxx.ini.ini*** or ***xxx~.ini***.

7.2.2 Secured Configuration File Downloading

The configuration file contains sensitive information required for appropriate functioning of the device. The file is uploaded to the device or downloaded from the gateway using TFTP or HTTP protocols. These protocols are unsecured (and thus vulnerable to a potential hacker). Conversely, if the configuration file is encoded, the file would be significantly less vulnerable to outside harm.

7.2.2.1 Encoding Mechanism

The configuration file to be loaded and retrieved could be available as is or encoded. When an encoded configuration file is downloaded to the device, it is also retrieved encoded from the device as well. When a decoded file is downloaded to the device, it is also retrieved decoded from the device as well.

In order to create an encoded configuration file, the user must first create a configuration file and then apply the **DConvert** utility to it in order to encode it. (Refer to Section E.2.6 on page 246 for detailed instruction on ***ini*** file encoding.)

In order to decode an encoded configuration file retrieved from the device, the user must retrieve an encoded configuration file from the device using the Web server (refer to Section 7.3.1 on page 188) and then use the **DConvert** utility in order to decode it. (Refer to Section E.2.6 on page 246 for detailed instruction on decoding the ***ini*** file.)

Downloading the configuration file as is or encoded may be performed by utilizing either TFTP and HTTP.

7.3 Downloading and Formating Configuration Files

7.3.1 Downloading Configuration Files

In the ***ini*** file, you can specify the Voice Prompt/Call Progress Tone table files to be loaded to the Mediant 2000 (using TFTP). In the BootP/TFTP Server, you can specify that these files be stored in the device's non-volatile flash memory when loaded so that it won't be necessary to perform a TFTP process every time the device boots up / is reset.

The following ***ini*** file fields are related to this operation:

- "VoicePromptsFileName" - The name (and path) of the file containing the voice prompts. This file should be constructed using the "TrunkPack Conversion Utility" supplied as part of the software package on the CD accompanying the device. The Voice Prompt buffer size in the board is 10 Mbyte. The Voice Prompt buffer size is also controlled by the feature key. For more information contact an AudioCodes representative.
- "CallProgressTonesFilename" - The name (and path) of the file containing the Call Progress and User-Defined Tones definition.
- "PrerecordedTonesFileName" - The name (and path) of the file containing the Prerecorded Tones. This file should be constructed using the "TrunkPack Conversion Utility" supplied as part of the software package on the CD accompanying the device
- CASFileName_0"…"CASFileName_7" (or "CASFileName") - The names (and path names) of the files containing the CAS protocol configuration. It is possible to use 1 to 8 files. The "CASFileName" name is still supported and can be used instead of the enumerated names when using only one CAS protocol file.
- "CASTablesNum" - Indicates how many CAS protocol configuration files are loaded. Its range is 1-8. It should match the number of "CASFileName_X" fields.
- CASTableIndex_TrunkNum (TrunkNum should be an integer) - This field is a CAS protocol file index. It indicates the CAS protocol file to use in a specific Trunk. The index value corresponds to the number in the field "CASFileName_X".

- SaveConfiguration - (default = 1 = enabled) This parameter replaces the following parameters: BlastCallProgressSetupFile, BlastCASProtocolSetupFile, BlastVoicePromptsFile. When enabled, all configuration and downloadable files are stored in non-volatile memory.

7.3.2 Call Progress Tones and User-Defined Tones

The Call Progress Tones and User-Defined Tones Configuration file contains the definitions of the Call Progress Tones and User-Defined Tones to be detected/generated by the Mediant 2000. The Call Progress Tones are mostly used for Telephony In-Band Signaling applications (e.g. Ring Back tone) while the User-Defined Tones are general purpose tones to be defined by the user. Users can use either one of the Mediant 2000 configuration files supplied by AudioCodes, or construct their own file.

The Call Progress Tones and User-Defined Tones file used by the Mediant 2000 is a binary file with the extension ***tone.dat***. Only this binary ***tone.dat*** file can be loaded to a device. Users can generate their own ***tone.dat*** file by opening the modifiyable ***tone.ini*** file (supplied with the ***tone.dat*** file as part of the software package on the CD accompanying the device) in any text editor, modify it, and convert the modified ***tone.ini*** back into a binary ***tones.dat*** file using AudioCodes' "TrunkPack Downloadable Conversion Utility" supplied with the Mediant 2000 software package. (Refer to Section 7.3.1 on page 188 for a description of the procedure for generating and downloading the Call Progress Tone file using this utility.)

To load the Call Progress Tones and User-Defined Tones configuration file to the Mediant 2000, correctly define their parameters in the device's ***ini*** file. (Refer to Section 7.2.1 on page 182 for the ***ini*** file structure rules and ***ini*** file example.)

7.3.2.1 Format of the Call Progress Tones Section

The Call Progress Tones section of the ***ini*** file format 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 0 and not exceeding the number of Call Progress Tones -1 defined in the first section) using the following keys:

- **Tone Type** - Call Progress Tone type

Basic Tone Type Indices

1	Dial Tone
2	Ringback Tone
3	Busy Tone
4	Congestion Tone
5	N/A
6	Warning Tone
7	Reorder Tone
8	Confirmation Tone
9	Call Waiting Tone

For a full tone indices list, refer to enum definition in AudioCodes' “VoPLib API Reference Manual”, Document #: LTRT-84001.

- **Low Freq [Hz]** - Frequency in Hertz of the lower tone component for a dual frequency tone, or the frequency of the tone for a single tone.
- **High Freq [Hz]** - Frequency in Hertz of the higher tone component for of a dual frequency tone, or zero (0) for a single tone.
- **Low Freq Level [-dBm]** - Generation level 0 dBm to -31 dBm.
- **High Freq Level** - Generation level. 0 to -31 dBm. The value should be zero (0) for 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.
- **Default Duration [msec]** - The default duration (in 1 msec units) of the generated tone.

Using this configuration file, you can create up to 16 different Call Progress Tones using up to 15 different frequencies (in the range of 300 Hz to 1980 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 the tone cadence. This is specified by 2 sets of ON/OFF periods, but you 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 high frequency field should be set to zero.

For a continuous tone (such as a 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, the tone is detected after 3 seconds (300 x 10 msec). The minimum detection time is 100 msec.

Table 7-14 shows the Output Tone Levels for each DSP code.

Note 1: When defining the same frequencies for both a continuous tone and a cadance tone, the Signal On Time parameter of the continues tone should have a value that is greater than the Signal On Time parameter of the cadance tone. Otherwise the continues tone is detected instead of the cadance tone.

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

Table 7-14: Call Progress Tones Output Tone Levels

Code	Output Level	Code	Output Level	Code	Output Level
0	Mute	11	-21 dBm	22	-10 dBm
1	-31 dBm	12	-20 dBm	23	-9 dBm
2	-30 dBm	13	-19 dBm	24	-8 dBm
3	-29 dBm	14	-18 dBm	25	-7 dBm
4	-28 dBm	15	-17 dBm	26	-6 dBm
5	-27 dBm	16	-16 dBm	27	-5 dBm
6	-26 dBm	17	-15 dBm	28	-4 dBm
7	-25 dBm	18	-14 dBm	29	-3 dBm
8	-24 dBm	19	-13 dBm	30	-2 dBm
9	-23 dBm	20	-12 dBm	31	-1 dBm
10	-22 dBm	21	-11 dBm	32	0 dBm

7.3.2.2 Format of the User-Defined Tones Section

The User Defined Tones section of the ***ini*** file format starts from the following string:

[NUMBER OF USER DEFINED TONES] - Containing only the following key:

- **Number of User Defined Tones** - Defines the number of User defined Tones to be defined in the file.

[USER DEFINED TONE #X] - containing the Xth tone definition (starting from 0 and not exceeding the number of (User Defined Tones - 1) defined in the first section) using the following keys:

- **Tone Type** - refer to the acTToneType enum definition of the AudioCodes' "VoPLib API Reference Manual", Document #: LTRT-84401
- **Low Freq [Hz]** - Frequency in Hertz of the lower tone component when a dual frequency tone or the frequency of the tone when a single tone.
- **High Freq [Hz]** - Frequency in Hertz of the higher tone component when dual frequency tone or zero (0) when a single tone.
- **Low Freq Level [-dBm]** - Generation level in dBm of the lower tone component when a dual frequency tone or the generation level of the tone when a single tone.
- **High Freq Level [-dBm]** - Generation level in dBm of the higher tone component when a dual frequency tone, or zero (0) when a single tone.
- **Default Duration [msec]** - The default duration (in 1 msec units) of the generated tone.

Using this configuration file, you can create up to 16 different User-Defined Tones using up to 15 different frequencies (in the range of 500 Hz to 3940 Hz). Each of the User-Defined Tones is specified by the tone frequency (either single or dual frequencies are supported). When the tone is made up of a single frequency, the high frequency field should be set to zero.

The Detection time of the User-Defined Tones is faster than that of the Call Progress Tones'. The maximum detection time is 30 msec.

Table 7-15 on page 192 shows the Output Tone Levels for each DSP code.

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

Table 7-15: User-Defined Tones Output Tone Levels

Code	Output Level	Code	Output Level	Code	Output Level
0	Mute	11	-21 dBm	22	-10 dBm
1	-31 dBm	12	-20 dBm	23	-9 dBm
2	-30 dBm	13	-19 dBm	24	-8 dBm
3	-29 dBm	14	-18 dBm	25	-7 dBm
4	-28 dBm	15	-17 dBm	26	-6 dBm
5	-27 dBm	16	-16 dBm	27	-5 dBm
6	-26 dBm	17	-15 dBm	28	-4 dBm
7	-25 dBm	18	-14 dBm	29	-3 dBm
8	-24 dBm	19	-13 dBm	30	-2 dBm
9	-23 dBm	20	-12 dBm	31	-1 dBm
10	-22 dBm	21	-11 dBm	32	0 dBm

7.3.2.3 Default Template for Call Progress Tones

The Mediant 2000 is initialized with the default Call Progress Tones configuration. 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 Table 7-16 on page 193 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]=22 (-10dBm)

High Freq Level [-dBm]=0

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 specific tone is according to the first definition of the specific tone. For example, the user 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 one of the two tones is detected.

Table 7-16: Call Progress Tones (continues on page 191 to page 195)

[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 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

Table 7-16: Call Progress Tones (continues on page 191 to page 195)

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

Table 7-16: Call Progress Tones (continues on page 191 to page 195)

[NUMBER OF CALL PROGRESS TONES] Number of Call Progress Tones=9	
#Call Waiting Tone [CALL PROGRESS TONE #8]	Tone Type=9 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

7.4 Modifying the Call Progress Tones & Sending it to the Device

AudioCodes supplies customers with a modifiable Call Progress Tone ***ini*** file and a non-modifiable Call Progress Tone ***dat*** binary file in the software package.

Only the binary ***dat*** file can be sent to the device.

In the ***ini*** file, customers can modify Call Progress Tone levels, Call Progress Tone frequencies to be detected/generated by the device, to suit customer-specific requirements. An example of a Call Progress Tone ***ini*** file name is ***usa_tones.ini***. Note that the word 'tones' is defined in the Call Progress Tone ***ini*** file name, to differentiate it from the device's ***ini*** file.

➢ **To modify the Call Progress Tones *ini* file and send the *dat* file to the device, take these 4 steps:**

Step 1. Open the Call Progress Tone ***ini*** file (it opens in **Notepad** or in a customer-defined text file editor.)

Step 2. Modify the file in the text file editor according to your specific requirements.

Step 3. Save your modifications and close the file.

Step 4. Convert the file with AudioCodes' Download Conversion Utility into a binary ***dat*** file (refer to Section 7.4.1).

7.4.1 Converting a Modified CPT *ini* File to a *dat* File with AudioCodes' Download Conversion Utility

After modifying the original Call Progress Tone *ini* file (supplied with the device's software package), use AudioCodes' Download Conversion Utility to convert the modified file into a ***dat*** binary file. You can only send the ***dat*** file to the device. The *ini* file cannot be sent.

To convert a modified CPT ***ini*** file to a binary ***dat*** file, Run the executable Download Conversion Utility file, ***DConvert240.exe***. For more information, refer to Appendix E on page 237.

After making the ***dat*** file, send it to the device using either:

- AudioCodes' Embedded Web Server GUI's Auxiliary Files. (Refer to Section 6.4.6.2 on page 140.)

  or

- AudioCodes' BootP/TFTP Server to send to the device the device's ***ini*** file (which simultaneously downloads the Call Progress Tone ***dat*** file, provided that the device's ***ini*** file parameter CallProgressTonesFilename is defined and provided that both files are located in the same directory.) (Refer to Appendix A, on page 211).

  or

- For PCI/cPCI boards, refer to the appropriate section in the VoPLib User's Manual, Document #: LTRT-84401.

7.4.2 Playing Prerecorded Tones (PRT)

The Call Progress Tones and the User-Defined Tones mechanisms have several limitations such as limited number of predefined tones, or limited number of frequency integrations in one tone. To solve these problems and provide a more flexible tone generation capability, prerecorded tones and play can be downloaded to the device and be played using regular tones generation commands.

7.4.2.1 PRT File Configuration

The PRT file that should be downloaded to the device is a binary dat file which was created using AudioCodes' DConvert utility. The tones should be recorded (or create using a Signaling Editor) if the user intends to download them in separate PCM files. The PCM files should be with the following characteristics:

- Coder: G711 A-law, G711 µ -law or Linear PCM.
- Rate: 8 kHz
- Resolution: 8-bit
- Channels: mono

The PRT module plays the recorded tone repeatedly. This provides the ability to record only part of the tone, while still playing it for a full duration. For example, if a tone has a cadence of 2 seconds on and 4 seconds off, the recorded file should contain only the 6 seconds of the cadence. The PRT module repeatedly plays this cadence for the configured duration. In the same manner, a continues tone can be played by repeating only part of it.

After the PCM files are properly prepared, these files should be converted into one ***dat*** file using AudioCodes' DConvert utility For more information regarding the DConvert utility, and how to make a ***dat*** PRT file, refer to Appendix E, on page 237.

Note: The maximum prerecorded tones that can be stored in one ***dat*** file is 40.

7.4.2.2 Downloading the PRT *dat* File

Downloading the PRT ***dat*** file into the device can be done via HTTP, TFTP or VoIPLib API. For HTTP and TFTP download, refer to Section 6.4.6 on page 136. For VoIPLib API download, refer to the Playing Prerecorded Tones (PRT) section of the VoPLib User's Manual, Document #: LTRT-84401.

Note 1: The maximum PRT buffer size is 1MB.

Note 2: If the same tone type was defined as PRT and as Call Progress Tone or User-Defined Tone, the device plays it using the PRT module.

Reader's Notes

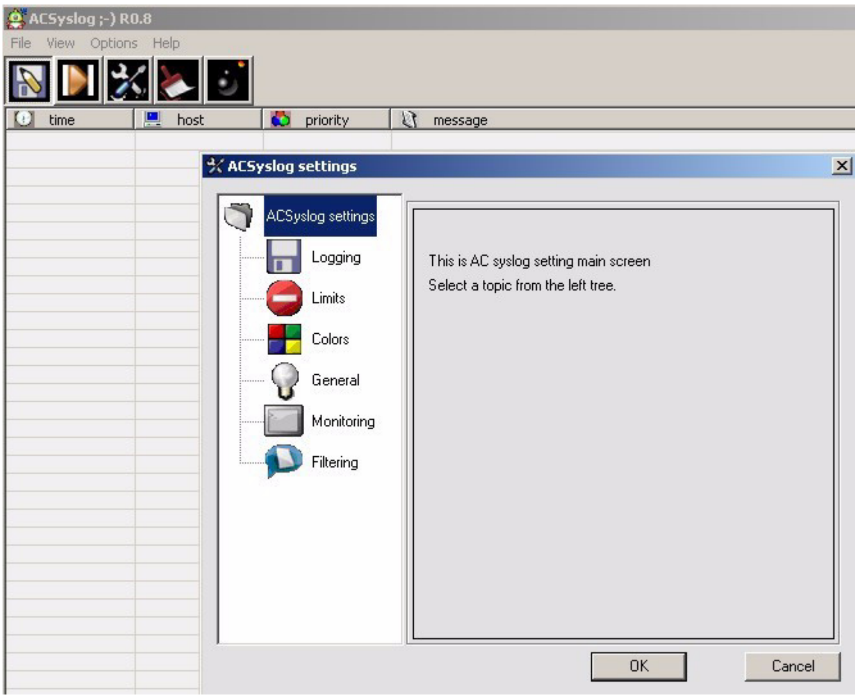
8 Diagnostics & Troubleshooting

8.1 Syslog

AudioCodes' Syslog server (refer to Figure 8-1, on page 199), now available with version 4.4 of the VoIPerfect platform, enables filtering of messages according to priority, IP sender address, time, date, etc. Customers can alternatively choose to download and use the following examples of the many Syslog servers available as shareware on the Internet:

- Kiwi Enterprises: http://www.kiwi-enterprises.com/software_downloads.htm
- The US CMS Server: http://uscms.fnal.gov/hanlon/uscms_server/
- TriAction Software: http://www.triaction.nl/Products/SyslogDaemon.asp
- Netal SL4NT 2.1 Syslog Daemon: http://www.netal.com

Figure 8-1: AudioCodes' Syslog Server Main Screen and Settings Screen



This is AC syslog setting main screen
Select a topic from the left tree.

Syslog protocol is an event notification protocol that allows a machine to send event notification messages across IP networks to event message collectors- also known as Syslog servers. Syslog protocol is defined in the IETF's RFC 3164 standard.

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

Syslog uses User Datagram Protocol (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:

<37> Oct 11 16:00:15 mymachine su: 'su root' failed for lonvick on /dev/pts/8

8.1.1 Operating the Syslog Server

8.1.1.1 Sending the Syslog Messages

AudioCodes' Syslog client, embedded in the firmware of the device, sends error reports/ events generated by the device application to a Syslog server, using IP/UDP protocol.

There are presently five error levels reported by the Syslog client:

- **Emergency level message:**

<128>sctp socket setsockopt error 0xf0 [File:sctp.cpp Line:453]

- **Warning level message**

<132>Release contains no h.225 Reason neither q.931 Cause information stateMode:1 [File: Line:-1];

- **Notice level message:**

<133>(lgr_flow)(2546) | #0:ON_HOOK_EV

- **Info level message:**

<134>document http://ab.pisem.net/RadAAIP.txt was not found in documents table [File:vxml_handleDB.cpp Line:2348]

- **Debug level message:**

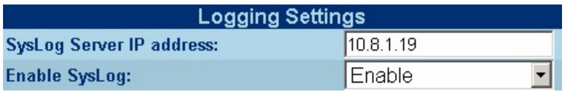
<135>SCTP port 2905 was initialized [File:csAPI.cpp Line:150] [CID:0]

8.1.1.2 Setting the Syslog Server IP Address

➢ To set the address of the Syslog server:

- Use AudioCodes' Embedded Web Server GUI (Advanced Configuration>Network Settings - screen section Logging Settings). (Refer to Section 6.3 on page 110 and to Figure 8-2)

Figure 8-2: Setting the Syslog Server IP Address



Logging Settings	
SysLog Server IP address:	10.8.1.19
Enable SysLog:	Enable

- Alternately, use AudioCodes' Embedded Web Server GUI or AudioCodes' BootP/ TFTP Server to send the ***ini*** configuration file containing the address parameter SyslogServerIP to the device. Before sending the ***ini*** file to the device, specify the address parameter. For detailed information on AudioCodes' BootP/TFTP Server, refer to Appendix A, on page 211. For an ***ini*** file example showing this parameter, refer to Section 8.1.1.4 on page 201 and to Figure 8-3, on page 201.

8.1.1.3 Activating the Syslog Client

➢ **To activate the Syslog client:**

- Use AudioCodes' Embedded Web Server GUI (Advanced Configuration>Network Settings - screen section Logging Settings). (Refer to Section 6.4.4 on page 119 and to Figure 8-2, on page 200.
- Alternately, use AudioCodes' Embedded Web Server GUI or AudioCodes' BootP/TFTP Server to send the ***ini*** configuration file containing the parameter EnableSyslog to the device. For detailed information on AudioCodes' BootP/TFTP Server, refer to Appendix A, on page 211. For an ***ini*** file example showing this parameter, refer to Section 8.1.1.4 on page 201 and to Figure 8-3, on page 201.

8.1.1.4 Setting Syslog Server IP Address, Enabling Syslog, in an *ini* File (Example)

Figure 8-3 shows an ***ini*** file section with an example configuration for the address parameter SyslogServerIP and an example configuration for the client activation parameter EnableSyslog.

Figure 8-3: Setting Syslog Server IP Address, Enabling Syslog, in an *ini* File (Example)

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

8.2 The Embedded Web Server's 'Message Log' (Integral Syslog)

The 'Message Log' screen in AudioCodes' Embedded Web Server GUI, similar to a Syslog server only integral to the web server, displays debug messages useful for debugging. For detailed information, refer to Section 6.4.5.2 on page 134. The Message Log screen is not recommended for logging of errors and warnings because errors can appear over a prolonged period of time, e.g., a device can display an error after running for a week, and it is not recommended to prolong a 'Message Log' session. For logging of errors and warnings, refer to Section 8-1 on page 199.

8.3 Control Protocol Reports

8.3.1 TPNCP Error Report

When working with AudioCodes’ proprietary TPNCP (TrunkPack Network Control Protocol), the TP-1610 reports all events using a TPNCP log event report mechanism (using error/debug events), through the network interface. For a list of events, refer to the section, “Board Originated Error Codes,” in AudioCodes’ “VoPLib API Reference Manual”, Document #: LTRT-84401.

Examples of using the Log Event Report Mechanism are also shown in AudioCodes’ “VoPLib API Reference Manual”, Document #: LTRT-84401.

8.3.2 MGCP/MEGACO Error Conditions

When working with MGCP/MEGACO, the TP-1610 reports error conditions via AudioCodes' Call Manager (or via a Call Manager of the customer's choice) using the standard MGCP/MEGACO facilities, through the network interface. For more information on MGCP/MEGACO error conditions, refer to the IETF web site at URL http://www.ietf.org/rfc/ and refer to RFC 2705 for MGCP and RFC 3015 for MEGACO.

8.3.3 SNMP Traps

AudioCodes boards support various SNMP traps via the SNMP Agent running on the board. Among these traps are Trunk MIB traps, acBoardStarted and acResetingBoard traps. Refer to Section 6.1 on page 97 for more details on all SNMP traps available on the board.

8.4 TP-1610 Self-test

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

Rapid self-test mode - Used each time the TP-1610 completes the initialization process. 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.

Detailed self-test mode - 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 or via PCI.) In this mode, the TP-1610 tests all the hardware components (memory, DSP, etc.) When EnableDiagnostics is set to **1**, flash is tested thoroughly, and when EnableDiagnostics is set to 2, flash is partially tested only. Test results are reported in the status output. The board sends EV_END_BIT containing information on the test results for each hardware component. To continue operational running, reset the board but this time with the EnableDiagnostics parameter set to **0**.

8.5 Solutions to Possible Problems

8.5.1 Possible Common Problems

Solutions to possible common problems are described in Table 8-1.

Table 8-1: Solutions to Possible Common Problems

Problem	Possible Cause	Solutions
No communication	Software does not function in the device	Try to "ping" the board/module. If ping fails, check for network problems/definitions and try to reset the board/module
	Network problem	Check the cables.
	Network definitions	Check if the default gateway can reach the IP of the board/ module.
		Check if the board/module got the correct IP.
		Check the validity of the IP address, subnet and default gateway. If the default gateway is not used, enter 0.0.0.0
	BootP did not reply to the board/module	Check if the BootP server replied to the board/module at restart by viewing the log of the BootP server.
		Try to restart the BootP server.
		Check the MAC address of the board/module in the BootP server.
***ini* file was not loaded**	TFTP server down	Check if the TFTP server is working.
	TFTP server didn't get the request	Check the log of the TFTP server.
	TP-1610 didn't request the file from your TFTP	Check that the TFTP server's IP address is the one that the TP-1610 is trying to use by viewing the Syslog.
	TFTP server bug	Try to restart the TFTP server.
	BootP sent to a board with the wrong TFTP server address	Check the IP address of the TFTP server being used.
	ini file does not exists in the default directory of the TFTP server	Check the default directory of the TFTP server and check that the ***ini*** file exists there.
	Wrong ***ini*** file name	Verify in Windows Explorer that file extensions are displayed and the ***ini*** file is not ***XXX.ini.ini*** by mistake. Also verify that the extension ***ini*** is in lowercase letters.
	TFTP server's timeout is too short	Verify that the TFTP server settings are as follows: Timeout = 2 sec, # of retransmission = 10
Wrong *ini* file loaded	***ini*** file is not in the correct position	An old ***ini*** file was probably loaded. Check which ***ini*** file was loaded by using the Syslog server. The Gateway displays contents of ***ini*** file before it began.
	ini file corrupted	Check the ***ini*** file syntax
BootP reply from wrong BootP server	Other BootP servers contain the MAC address of the board/module	Check that only your BootP server contains the TP-1610's MAC address.

8.5.2 Possible Voice Problems

Solutions to possible voice problems are described in Table 8-2.

Table 8-2: Solutions to Possible Voice Problems

Problem	Possible Cause	Solutions
G.711 voice quality is bad (clicks)	Silence compression is not compatible (when working with different Gateway other than AudioCodes Gateway)	Disable it and check if the quality is better.
	The Packet size is not compatible (with G.711)	Check that the packet period in the 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 the codecs	Change the codec definition.

9 Functional Specifications

9.1 Mediant 2000 Selected Specifications

Table 9-1: Selected Specifications (continued on page 205 to page 209)

Item	Characteristic
Channel Capacity	
Network Ports/DSP Calls (independent digital voice, fax or data ports)	Up to 480 All media processing ports can be tied to IP-RTP, PSTN-DS0 Time Slots independently
DSP Channel Configuration Options	30, 60, 120, 240, 480 ports
Voice Messaging	
PCI-based Recording/Playback	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) Dependent on the supported coders in the board
	Up to full board channel capacity ports of record/playback - PCI bus based (Coder dependent) Playback to/Record from PSTN side
	PSTN / TDM to *.wav (per coders listed above)
	PSTN / TDM to LBR raw data
Playback from Local Storage	Prompts and announcements playback (10 MB integral memory for 20 min. of G.711 or 200 min. for G.723 recorded prompts)
HTTP Streaming	Playback and recording (streaming via HTTP) of raw or wav G.711 files stored on a remote Web server
Media Processing	
IP Transport	VoIP (RTP/ RTCP) per IETF RFC 3550 and RFC 3551
Packet interfaces	Integral NIC (IP) or PCI; Selectable per port
DTMF/MF Transport	DTMF/MF RTP Relay per RFC-2833, Mute, Transparent (transfer in coder as voice)

Table 9-1: Selected Specifications (continued on page 205 to page 209)

Item	Characteristic
Voice Processing	All voice processing features are supported simultaneously on all ports
	Dynamic Network Jitter Buffer with reordered RTP packets correction
	Call Progress Tones generation and detection
	Transcoding of a G.711 RTP stream to any Low Bit-Rate Coder RTP stream using one DSP channel resource
	Mediation between two IP endpoints of the same coder without using any DSP channel resource
	Media duplication (one source to many destinations) using the same coder without using additional DSP channel resources
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 Cancellation	G.165 & G.168-2000 compliant 32, 64, 128 msec echo tail (64 and 128 may reduce channel capacity)
Fax and Modem Transport	
Fax Relay and ByPass	Supported on all ports
	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
	Automatic Fax ByPass (pass-through) to G.711 or ADPCM
Modem ByPass	Automatic switching (pass-through) to PCM or ADPCM for modem signals (V.34 or V.90 modem detection)

Table 9-1: Selected Specifications (continued on page 205 to page 209)

Item	Characteristic
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
SS7	MTP-2 (ITU / ANSI) link termination
SIGTRAN	IUA (RFC 3057) over SCTP (RFC 2960) M2UA, (RFC 3331) over SCTP (RFC 2960) transferring MTP-2 payload M3UA, (RFC 3332) over SCTP (RFC 2960) transferring MTP-3 payload
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, CAS and R2 package, Basic announcements package, Conferencing
TPNCP	AudioCodes proprietary TPNCP (Via Ethernet of PCI/cPCI interface)
Processor	
Control Processor	Motorola PowerQUICC 8260
Control Processor Memory	SDRAM – 64 MB

Table 9-1: Selected Specifications (continued on page 205 to page 209)

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
PSTN	Either: 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 8 E1/T1/J1 ports. Or: 1, 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	
Supply Voltages and Power Consumption (typical)	40.7 W, 3 A at 5 V, 7.8 A at 3.3 V 24 W, 1.5 A at 5 V, 5 A at 3.3 V (120 channels)With 16 span 1610 board: 230 VAC 62 watts -48 VDC 60 watts
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 boards Redundant Power Supplies provide protection but are non Hot Swappable
Host Interface	Via cPCI bus, using AudioCodes' libraries
	Via Packet interface using AudioCodes' proprietary TPNCP 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) 2 rear mounting flanges - Optional
Type Approvals	
Telecommunication Standards (Chassis and hosted Telecom boards)	IC CS03, FCC part 68 CTR4, CTR12 , CTR13, JATE, TS.016, TSO, Anatel, Mexico Telecom
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 (approved) For DC powered version Complies with ETS 300019-1, -2, -3 (T1.1, T2.3 & T3.2)

Table 9-1: Selected Specifications (continued on page 205 to page 209)

Item	Characteristic
LED Indicators	
LED Indications on Front Panel	Power, Ready/Fail T1/E1/J1 per trunk status, LAN statusSwap ready indication
Connectors and Switches	
Rear Panel	Power: (model dependent) - Standard AC power socket or - DC power - MSTB2.5/2-STF (5.08mm) of Phoenix Contact E1/T1 Trunk and Ethernet connectors
Front Panel	Hardware Reset button
Diagnostics	
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 AudioCodes BootP/TFTP Server

A.1 Introduction

AudioCodes **BootP/TFTP Server** enables easy configuration and provisioning for AudioCodes boards and Media Gateways. The BootP and TFTP servers contain specific adaptations as per AudioCodes' requirements. The latest version of the BootP/TFTP application is 2.3.0.5.

A.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
- 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
- Display of AudioCodes vendor specific information parameters
- Support for AudioCodes' Selective BootP feature (The BootP server inserts AudioCodes specific vendor information that include the text **AUDC**)
- Compatible with Windows™ 98, Windows™ NT, Windows™ 2000, Windows™ XP

A.1.2 Specifications

BootP standards: RFC 951 and RFC 1542

TFTP standards: RFC 1350 and RFC 906

Operating System: Windows™ 98, Windows™ NT and Windows™ 2000, 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
- Call Agent IP
- New MAC (optional)

Screens:

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

A.1.3 BootP/TFTP Server Installation

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

➢ To install the BootP/TFTP Server, take these 3 steps:

Step 1. Unzip the ***TPxx.exe*** file and navigate to the BootP ***zip*** file under .\Utilities\BootP & TFTP server.

Step 2. Double click on the BootP ***zip*** file and run ***setup.exe***. The installation procedure begins. After completing the procedure, open **Start>Programs>BootP.** The BootP/TFTP Server main screen is displayed.

At first run, the user is requested to fill in the fields displayed on the Preferences screen.

Step 3. To open the Preferences screen, from the main screen, select **Edit>Preference.** Follow the directions detailed in Section A.2.2 on page 216 to configure the screen.

A.1.4 Logging Screen

The AudioCodes BootP/TFTP Server main screen (Figure A-1, on page 214) 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

- Client specific Information - contains AudioCodes' vendor specific information, which includes: Board type, last IP, bootload version, flash ***cmp*** version. In order to access the board information, add **-be 1** to the ***ini*** file selection in the BootP application. With this initial setting, even after deleting **-be 1**, the board continues to report its internal data.

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

Right clicking a Log line opens up a menu.

Selecting **Reset** causes a soft reset of the board. Reset is available only for client MACs that are configured on the BootP server. The second option on the menu is View Client, which produces the same display as when clicking on the Log line.

A.1.5 Preferences Screen

The Preferences screen (Figure A-2, on page 216) 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 AudioCodes device (e.g., **TP-1610** board/module) is installed behind the firewall that blocks BootP broadcast requests.

A.1.6 Client Configuration Screen

The Client Configuration screen (Figure A-3, on page 217) shows:

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

With 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.

A.1.7 Template Screen

The Template screen (Figure A-1, on page 214) enables the user to add, modify, or delete templates.

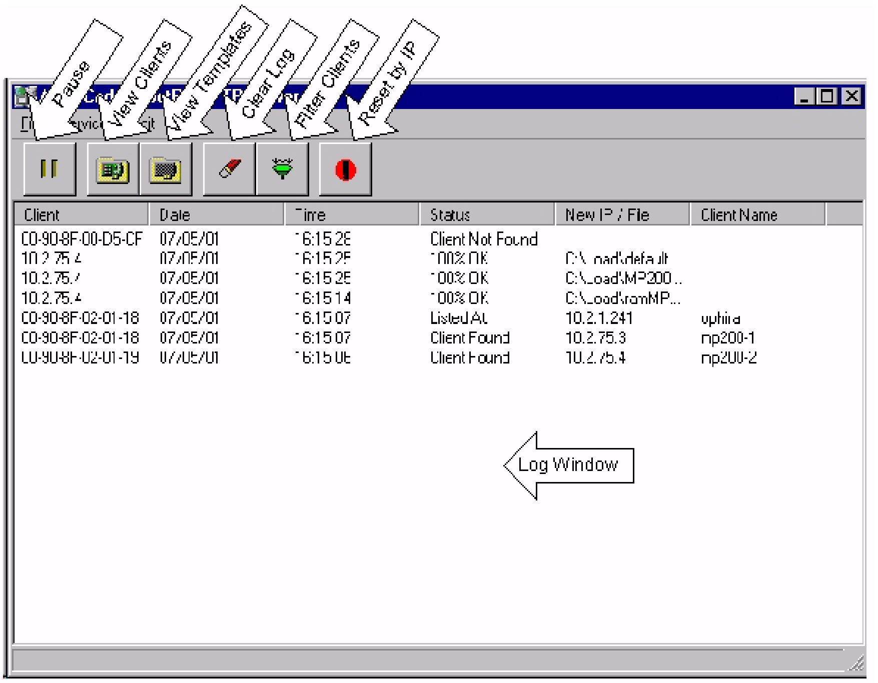
The template includes:

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

A.2 Screen Details

A.2.1 Main Screen

Figure A-1: Main Screen



Client	Date	Time	Status	New IP / File	Client Name
C0-90-8F-00-D5-CF	07/05/01	16:15:28	Client Not Found		
10.2.75.4	07/05/01	16:15:25	100% OK	C:\Load\default	
10.2.75.4	07/05/01	16:15:25	100% OK	C:\Load\MP200...	
10.2.75.4	07/05/01	16:15:14	100% OK	C:\Load\ramMP...	
C0-90-8F-02-01-18	07/05/01	16:15:07	Listed At	10.2.1.241	uphira
C0-90-8F-02-01-18	07/05/01	16:15:07	Client Found	10.2.75.3	mp200-1
C0-90-8F-02-01-19	07/05/01	16:15:06	Client Found	10.2.75.4	mp200-2

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

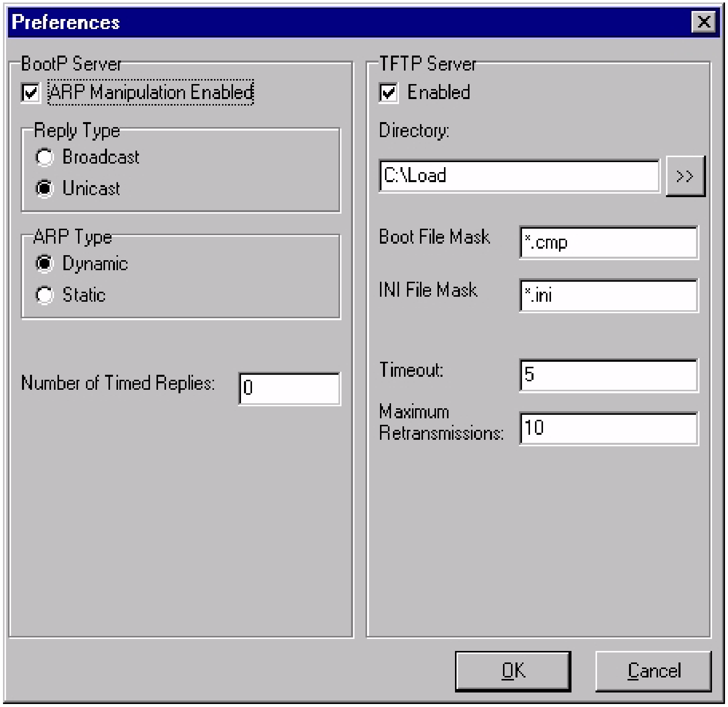
- **Program State -** Pauses the program. When the program is paused, no replies to BootP requests are sent.
- **View Clients button -** Opens up the Clients Configuration screen.
- **View Templates button -** Opens up the Templates Configuration screen.
- **Clear Log button - C**lears the log.
- **Filter Unknown Clients button -** Filters all BootP requests that are not listed in the Client Configuration screen.

- **Reset 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 -** Opens the Preferences screen for defining BootP and TFTP parameters.
- **Log Screen -** Displays all BootP requests and TFTP sessions, 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 -** Right-clicking on a line in the log screen displays the pop-up menu. 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 does not have to know the current IP address of the client. To perform this option, the user must have administrator privileges, otherwise 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 remaining fields require fill in.

A.2.2 Preferences Screen

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

Figure A-2: Preferences Screen



Preferences

BootP Server

☑ ARP Manipulation Enabled

Reply Type

○ Broadcast

● Unicast

ARP Type

● Dynamic

○ Static

Number of Timed Replies: 0

TFTP Server

☑ Enabled

Directory: C:\Load

Boot File Mask: *.cmp

INI File Mask: *.ini

Timeout: 5

Maximum Retransmissions: 10

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 AudioCodes device 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 AudioCodes device immediately after the remote reset command.

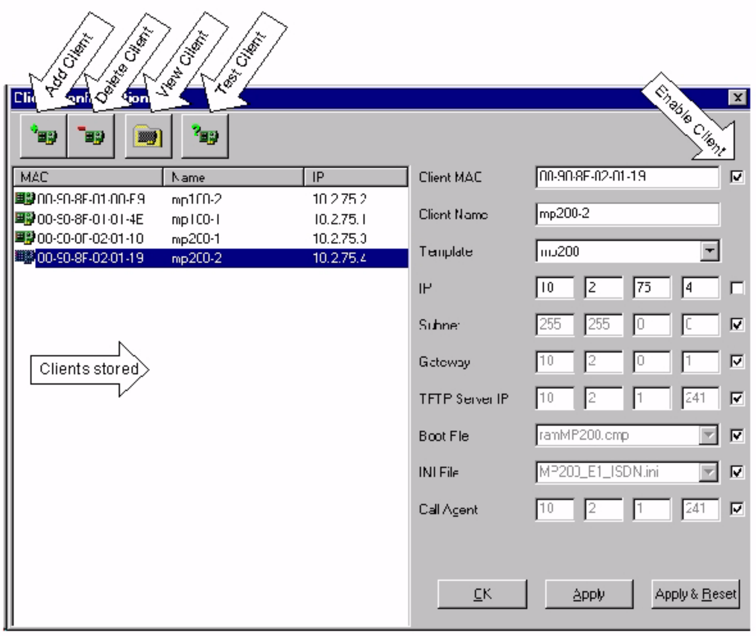
For the TFTP server, the user can configure a TFTP directory and a value for TFTP Timeout and Maximum Retransmissions. Set these values to **2** and **10** as shown above.

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

A.2.3 Client Configuration Screen

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

Figure A-3: Client Configuration Screen



MAC	Name	IP
00-90-8F-01-00-F9	mp100-2	10.2.75.2
00-90-8F-01-01-4E	mp100-1	10.2.75.1
00-90-8F-02-01-10	mp200-1	10.2.75.3
00-90-8F-02-01-19	mp200-2	10.2.75.4

Client MAC	00-90-8F-02-01-19				☑
Client Name	mp200-2				
Template	mp200				
IP	10	2	75	4	□
Subnet	255	255	0	0	☑
Gateway	10	2	0	1	☑
TFTP Server IP	10	2	1	241	☑
Boot File	ramMP200.cmp				☑
INI File	MP200_E1_ISDN.ini				☑
Call Agent	10	2	1	241	☑

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. This causes 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** - A text field for entering the 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 address of the TFTP Server.
- **Boot File, *ini* File -** The files to request from the TFTP server.

Note the seven check boxes to the right of the parameters. These enable the user to assign only the fields from the template, which have adjacent marked checkboxes. The rest can be unique for each client. When the field is assigned a value from the selected template, the field is grayed (and unmodifiable).

To save them after performing changes, click **Apply**. By clicking **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 A-2, on page 216) 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.

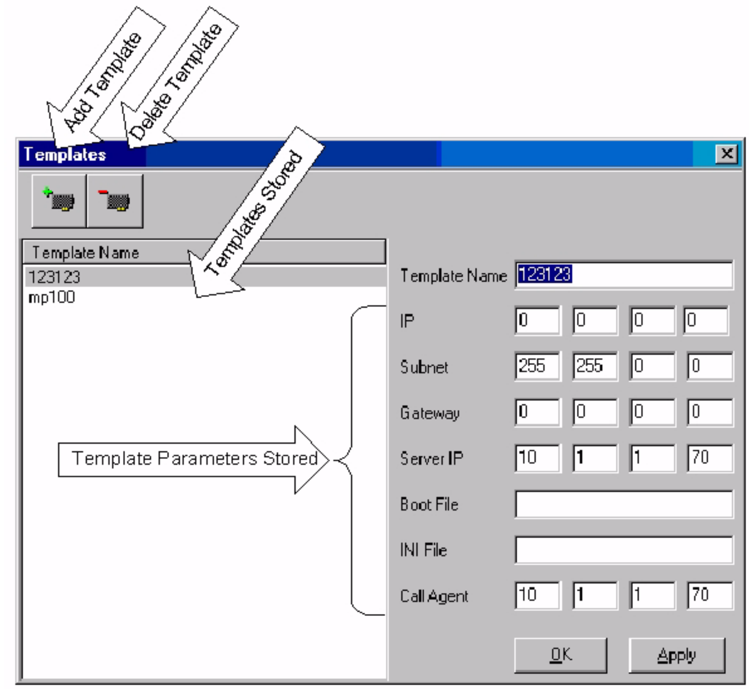
When adding a new client, click **Add Client**. A client dialog box with blank parameters is displayed. After filling out the parameters, click **Apply**. The client is added.

To find out if there is another BootP server on the net that contains a client with the same MAC address, click **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.

A.2.4 Templates Screen

Figure A-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 A-4: AudioCodes BootP/TFTP Templates Screen



Template Name: 123123, mp100

Template Name	123123			
IP	0	0	0	0
Subnet	255	255	0	0
Gateway	0	0	0	0
Server IP	10	1	1	70
Boot File				
INI File				
Call Agent	10	1	1	70

Appendix B RTP/RTCP Payload Types

Latest RTP Payload Types are defined in RFC 3551. For coders that should have dynamic Payload types, AudioCodes has defined proprietary default values out of the dynamic Payload type range. These defaults are appropriate when working with AudioCodes products only. However, it is recommended to set a dynamic Payload type for them, which is usually done by higher applications during call setup. Be sure not to overload dynamic Payload types.

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

B.1 Payload Types Defined in RFC 3551

Table B-1: Payload Types Defined in RFC 3551

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
35	G.726-16	20
36	G.726-24	20
38	G.726-40	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	

B.2 Payload Types Not Defined in RFC 3551

Table B-2: Payload Types Not Defined in RFC 3551

Payload Type	Description	Basic Packet Rate [msec]
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
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
72	Vox ADPCM	20
90	Linear PCM	20

B.3 Default Dynamic Payload Types Which are Not Voice Coders

Table B-3: Payload Types Not Defined in RFC 3551

Payload Type	Description
96	RFC 2833
102	Fax Bypass
103	Modem Bypass
104	RFC 2198
105	NSE

B.4 Mediant 2000 Default RTP/RTCP/T.38 Port Allocation

The following table shows the Mediant 2000 default RTP/RTCP/T.38 port allocation.

Table B-4: 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 C DTMF, Fax and Modem Transport Modes

C.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 four 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= 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 relayed to the remote side using the RFC 2833 Relay syntax. RFC 2833 digit packets that are received from the remote side are muted on the audio stream.

C.2 Fax/Modem Settings

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

- **fax relay -** demodulation / 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. Therefore, the channel is in **Answer Tone** mode until a distinction is determined. The packets being sent to the network at this stage are fax relay T.38 packets.

C.3 Configuring Fax Relay Mode

When FaxTransportType= 1 (relay mode), upon detection of fax, the channel automatically switches from the current voice coder to answer tone mode, and then to fax T.38 relay mode.

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

The fax rate can be limited by using the FaxRelayMaxRate parameter and the ECM Fax Mode can be enabled/disabled using the FaxRelayECMEnable parameter settings.

When using T.38 Relay mode, the user can select between two protection strategies - redundancy packets or forward error correction (FEC). This selection is made through the (proprietary) redundancy mode that was specially designed to improve protection against packet loss through the EnhancedFaxRelayRedundancyDepth parameter. Although this is a proprietary redundancy scheme, it is compatible with other T.38 decoders. The depth of the redundancy (that is, the number of repetitions) is defined by the FaxRelayredundancyDeoth configuration parameter.

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

C.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 in the FaxModemBypassCoderType configuration parameter.

If relay is enabled for one of the modes (Fax/Modem), 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 user can also configure the basic frame size by through the FaxModemBypassBasicRTPPacketInterval configuration parameter. The network packets generated and received during the bypass period are regular RTP voice packets (as 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 performed.

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

C.5 Configuring Modem Bypass NSE mode

Setting the NSEMode to **1** configures the answering Modem channel to send NSE packets to the calling Modem channel to switch to Bypass. Using the NSEPayloadType parameter, the user can control the NSE RTP packet's Payload type (default = 105). Note that the value of this parameter should be within the RTP Dynamic Payload Type range (96 to 127).

C.6 Supporting V.34 Faxes

Unlike the T.30 fax machines, the V.34 fax machines have no relay standard to transmit the data over IP to the remote side. Therefore AudioCodes provides the following operation modes for transporting the V.34 fax data over the IP.

Note: For all the setups described below, the CNG tone detector is disabled.

C.6.1 Using Bypass Mechanism for V.34 Fax Transmission:

Configuration:

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

Expected events for V.34Fax to V.34Fax - Bypass Mode are shown in Table C-1.

Table C-1: V.34Fax to V.34Fax - Bypass Mode

Calling	Answering
	EV_DETECT_MODEM (2100 AM + Reversal)
EV_DETECT_MODEM	
	EV_DETECT_FAX
EV_DETECT_FAX (Refer to Note 1 below)	
EV_END_FAX	EV_END_FAX

Note: The board changes 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 fax relay benefits for the T.30 fax machines and, in parallel, are a variable when using a V.34 fax with its full rate. Therefore, AudioCodes recommends this setup. 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.

C.6.2 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 to V.34Fax - Events Only Mode are shown in Table C-2.

Table C-2: V.34Fax to V.34Fax - Events Only Mode

Calling	Answering
	EV_DETECT_ANSWER_TONE
	EV_DETECT_FAX

C.6.3 Using Relay Mode for Various Fax Machines (T.30 and V.34)

The user can force the V.34 fax machines to revert 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 to V.34Fax - Relay Mode are shown in Table C-3.

Table C-3: V.34Fax to V.34Fax - Relay Mode

Calling	Answering
	EV_DETECT_ANSWER_TONE
	EV_DETECT_FAX
EV_DETECT_FAX	
EV_END_FAX	EV_END_FAX

Appendix D CAS Protocol Table

D.1 Constructing a CAS Protocol Table

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

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

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

Step 1. Learn the protocol text file rules (rules detailed in this manual and their syntax are based on C pre-processor commands).

Step 2. Get the AudioCodes example.

Step 3. Build the specific protocol/script text (***xxx.txt***) file and its related numerical value h file (***xxx.h***).

Step 4. Compile the ***xxx.txt*** with the "TrunkPack Downloadable conversion utility" to produce the ***xxx.dat*** file. Refer to Section E on page 237 ("API Demonstration Utilities") for a detailed description of the utility usage.

Step 5. Download the ***User_protocol.dat*** file to the board via acOpenBoard() command at initialization phase.

D.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:

D.2.1 INIT variables

INIT variables - Numeric values in ***UserProt_defines_xxx.h***, defined by the user. 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.

D.2.2 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 can be found in the ***CASSetup.h*** file.

D.2.3 Functions

Functions - Define a certain procedure that can be activated in any state or in the transition from one state to another. The available functions include, for example, SET_TIMER (timer number, timeout in ms.) SEND_CAS (AB value, CD value). A full list of the possible pre-defined functions can be found in the ***CASSetup.h*** file.

D.2.4 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 D-1 was taken from an E&M wink start table protocol file:

Table D-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 be triggered: either timer 2 expiration or end of dial event. When timer 2 expires, the protocol table executes SEND_DEST_NUM function and remains in the same state (NEXT_STATE=NO_STATE). When the dial event ends, the protocol table sets timer 4 and moves to ST_DIAL_ENDED written in the 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. The two states are ST_INIT and ST_IDLE.

- **ST_INIT -** When channels initialization is selected, the table enters the 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 were required 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 maximum value as defined in the DIAL_PLAN Parameter #1 and #2 for destination and ANI numbers respectively
- A pre-defined time-out value defined in the 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 minimum/maximum 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 the 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** - Enables or Disables 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).
- **NUM_OF_EVENT_IN_STATE** - Inserted for detection on TOTAL_NUMBER_OF_EVENTS_IN_STATE (***CASSetup.h***).
- **INIT_MGCP_REPORT** - Enables the event for MGCP. These tables are specifics and relevant for MGCP only. Do not use it otherwise.
- **INIT_GLOBAL_TIMERS** - Initiates specific timers, is used with Parameter#1 for metering pulse timer duration.
- **INIT_VERSION** - Defines the version number. The version number is related to the release version number.
- **INIT_SIZE_OF_TABLE_PARAM** - Users must insert the definition of TOTAL_NUMBER_OF_EVENTS_IN_STATE fron the ***CASSetup.h***.

- **ST_IDLE -** When no active call is established or in the process of being established, the table resides in Idle state, allowing it to start the process of incoming or outgoing calls. When the call is cleared, the state machine table returns to its idle state.

D.3 Reserved Words

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

D.4 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

D.5 Action/Event

Action/event is the name of the table's events that are the possible triggers for the entire 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. They events are functions that are unconditionally performed when the table reaches the state. These actions are labeled FUNCTION0 to FUNCTION3.

The 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()
- **EV_USER_BLOCK_COMND** - When using acCASBlockChannel(), this event handled for blocking or unblocking the channel.
- **EV_MAKE_METERING_PULSE** - When using acCASMeteringPulse it triggers the start of the metering pulse while using the SET_PULSE_TIMER function to start the timer to get the Off event (refer to the EV_METERING_TIMER_PULSE_OFF event.).
- **EV_METERING_TIMER_PULSE_OFF** - event after timer (SET_PULSE_TIMER function) expires. (Refer to EV_MAKE_METERING_PULSE.)
- **EV_MAKE_FLASH_HOOK** - When using acCASFlashHook, a flash hook is triggered.

2 **CAS Change Oriented:**

- **EV_CAS_1_1** - a new CAS A,B bits is received (A=1, B=1, was stable for the bouncing period).
- **EV_CAS_1_0** - a new CAS A,B bits is received (A=1, B=0, was stable for the bouncing period).
- **EV_CAS_0_1** - a new CAS A,B bits is received (A=0, B=1, was stable for the bouncing period).
- **EV_CAS_0_0** - a new CAS A,B bits is 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 has expired.
- **EV_TIMER_EXPIRED2** - timer 2 that was previously set by table has expired.
- **EV_TIMER_EXPIRED3** - timer 3 that was previously set by table has expired.
- **EV_TIMER_EXPIRED4** - timer 4 that was previously set by table has expired.
- **EV_TIMER_EXPIRED5** - timer 5 that was previously set by table has expired.
- **EV_TIMER_EXPIRED6** - timer 6 that was previously set by table has expired.
- **EV_TIMER_EXPIRED7** - timer 7 that was previously set by table has expired.
- **EV_TIMER_EXPIRED8** - timer 8 that was previously set by table has expired.

4 Counter Oriented:

- **EV_COUNTER1_EXPIRED** - counter 1 value has reached 0.
- **EV_COUNTER2_EXPIRED** - counter 2 value has reached 0.

5 IBS oriented:

- **EV_RB_TONE_STARTED** - Ring back tone as defined in the call progress ***ini*** file (type and index) is detected.
- **EV_RB_TONE_STOPPED** - Ring back tone as defined in the call progress ***ini*** file (type and index) is stopped after it has been previously detected.
- **EV_DIAL_TONE_DETECTED** - Dial tone as defined in the call progress ***ini*** file (type and index) is detected.
- **EV_DIAL_TONE_STOPPED** - Dial tone as defined in the call progress ***ini*** file (type and index) was stopped after it has been previously detected.

6 MF Oriented (MFCR2 protocol related):

- **EV_MFRn_1** - MF digit 1 is detected.
- **EV_MFRn_2** - MF digit 2 is detected.
- **EV_MFRn_3** - MF digit 3 is detected.
- **EV_MFRn_4** - MF digit 4 is detected.
- **EV_MFRn_5** - MF digit 5 is detected.
- **EV_MFRn_6** - MF digit 6 is detected.
- **EV_MFRn_7** - MF digit 7 is detected.
- **EV_MFRn_8** - MF digit 8 is detected.
- **EV_MFRn_9** - MF digit 9 is detected.
- **EV_MFRn_10** - MF digit 10 is detected.
- **EV_MFRn_11** - MF digit 11 is detected.
- **EV_MFRn_12** - MF digit 12 is detected.
- **EV_MFRn_13** - MF digit 13 is detected.
- **EV_MFRn_14** - MF digit 14 is detected.
- **EV_MFRn_15** - MF digit 15 is detected.
- **EV_MFRn_1_STOPPED** - MF digit 1 previously detected, is now stopped.
- **EV_MFRn_2_ STOPPED** - MF digit 2 previously detected, is now stopped.
- **EV_MFRn_3_ STOPPED** - MF digit 3 previously detected, is now stopped.
- **EV_MFRn_4_ STOPPED** - MF digit 4 previously detected, is now stopped.

- **EV_MFRn_5_ STOPPED** - MF digit 5 previously detected, is now stopped.
- **EV_MFRn_6_ STOPPED** - MF digit 6 previously detected, is now stopped.
- **EV_MFRn_7_ STOPPED** - MF digit 7 previously detected, is now stopped.
- **EV_MFRn_8_ STOPPED** - MF digit 8 previously detected, is now stopped.
- **EV_MFRn_9_ STOPPED** - MF digit 9 previously detected, is now stopped.
- **EV_MFRn_10_ STOPPED** - MF digit 10 previously detected, is now stopped.
- **EV_MFRn_11_ STOPPED** - MF digit 11 previously detected, is now stopped.
- **EV_MFRn_12_ STOPPED** - MF digit 12 previously detected, is now stopped.
- **EV_MFRn_13_ STOPPED** - MF digit 13 previously detected, is now stopped.
- **EV_MFRn_14_ STOPPED** - MF digit 14 previously detected, is now stopped.
- **EV_MFRn_15_ STOPPED** - MF digit 15 previously detected, is now stopped.
- **EV_END_OF_MF_DIGIT** - When using DialMF() and no more dialed number digits are available. (They already have been sent. For example, the far side requests the next ANI digit, but all digits already have been 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 type 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 is 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 is 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 the 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 the SEND_PROG_TON function to stop the dial tone.

D.6 Function

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

D.7 Parameters

Table D-2: CAS Parameters

Parameter #1	These columns are used as the function's parameters. The list of global parameters can be found in ***CasSetup.h***. If a parameter is not essential, the parameter is marked **None**.
Parameter #2	
Parameter #3	

List of available user-functions and their parameters:

- **SET_TIMER (timer number, timeout) -** 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 msec).
- **SEND_CAS (AB value, CD value) -** ABCD bits are sent as line signaling for the specific channel when the call is setup.
- **SEND_EVENT (event type, cause) -** The specific event type is sent to the host/user and retrieved by applying acGetEvent().
- **SEND_DEST_NUM -** Enbloc dialing: Enbloc dialing: refers to the digits string located in the acPSTNPlaceCall function. Three types are available: (1) DestPhoneNum (2) InterExchangePrefixNum (3) SourcePhoneNum.
- **DEL_TIMER (timer number) -** Delete specific or all timers (0 for all) for the B-channel.
- **START_COLLECT -** Initiates the collection of address information i.e., the dialed (destination) number for incoming calls where appropriate according to the protocol. At the time between START_COLLECT and STOP_COLLECT, no digit is reported to the user (EV_DIGIT is blocked) and 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) -** Set counters that are managed per B-channel and their expiration triggers the state machine. The counter initialization value should be a non-negative number. To delete all timers, perform this function with 0 in the counter number field.
- **DEC_COUNTER (counter number) -** Decreases counter value by 1. When the 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 cell (the types are defined in ***CAS-Setup.h*** file):

- **ADDRESS** - Sends the digit from the address vector (destination number) according to the index requested. (Refer to the Index definition).
- **ANI** - Sends the digit from the ANI vector (source number) according to the index requested.
- **SPECIFIC** - Sends the MF digit specified in the Parameter #2 rubric.
- **SOURCE_CATEGORY** - Sends the 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 the 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 -** Specifies the Offset of the next digit to be sent from the vector (ADDRESS or ANI types described above):

- **Index 1** - Used to send the next digit in the vector.
- **Index -n** - Used to send the last **n** digit. Underflow can occur if n is greater than the number of digits sent so far.
- **Index 0** - Used to send the last sent digit.
- **Index SEND_FIRST_DIGIT** - Used to start sending the digits vector from the beginning. (Refer to ***CASSetup.h***.)

3 **MF Send Time** - This send time parameter specifies the maximum transmission time of the MF.

- **STOP_SEND_MF** - Stops sending the current MF.
- **SEND_PROG_TON** - Operation, Tone or NONE.

  - Two operations are available.

  a Sends the Call Progress Tone specified in the Parameter #2 rubric (The second parameter can be taken from ***CASsetup.h***).

  b Stops 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 the MF digit of the requested type. The type can be one of the following:
- **ADDRESS** - The user is waiting for the reception of address digits.
- **ANI** - The user is waiting for the reception of ANI digits.
- **SOURCE_CATEGORY** - The user is waiting for the reception of the source category.
- **TRANSFER_CAPABILITY** - The user is waiting for the reception of the source transfer capability (line category).

D.8 Next State

The Next State column contains the next state the table moves to after executing the function for that action/event line. When 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 current state, the unconditional actions (FUNCTION0) at the beginning of the state are performed. In contrast, NO_STATE skips these functions and waits for another action to 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.

D.9 Changing the Script File

D.9.1 General

- CAS bouncing is filtered globally for each received CAS for each channel. The User defines the time for the filtering criteria in the protocol table file (see INIT_DEBOUNCE), and this is above the bouncing in the DSP detection of 30 msec.
- ANI/CLI is enabled using ST_INIT ANI parameter with 'YES'. ANI/CLI is supported using EV_ANI_NUM_DETECTED as the table action for collecting the ANI number in an incoming call. For outgoing calls, the table's function SEND_DEST_NUM with ANI parameter I initiates ANI dialing. The ANI number is provided by the User in the Source phone number parameter of acPSTNPlaceCall().

D.9.2 MFC-R2 Protocol

- Use the SEND_MF script function to generate the outgoing call destination number. In this case, the first parameter should be ADDRESS (or ANI for source phone number) and the second parameter -3 to 1 (+1), indicating which digit is sent out of the number that the string conveyed by the User in the acPSTNPlaceCall().
- (+1) implies sending of next digit, 0 implies repeat of last digit, and -1 implies last but one digit. This parameter actually changes the pointer to the phone number string of digits. Thus, a one-to-one mapping with the MF backward signals of the R2 protocol exists.
- Using the parameter SEND_FIRST_DIGIT initiates resending the string from the beginning, (change the pointer back to first digit and then proceed as above). This parameter is defined in CASSetup.h.
- When MFC-R2 protocol is used, the two detectors (opened by default) are the Call Progress Tones and MFC-R2 Forward MF. When the User invokes outgoing call via acPSTNPlaceCall(), MFC-R2 Forward MF detector is replaced with MFC-R2 Backward MF detector, since only two detectors per DSP channel are permitted to operate simultaneously.
- The correct MF is automatically generated according to the call direction - Forward for outgoing calls and Backward for incoming calls.
- MFC-R2 protocol fault could cause a channel block. In this case, the script file provided by AudioCodes releases the call to enable the User to free the call resources and be notified about being in blocking state.
- START_COLLECT and STOP_COLLECT must be used in the script file for MF collecting both in outgoing and incoming calls. Warning: If this script function isn't used, the script gets stuck and forward\backward MF are not detected.

- The Ringback Call Progress Tone is translated to a unique event acEV_PSTN_ALERTING, since the Ringback tone is actually used in all AudioCodes protocols' state machines. All other Call Progress Tones are conveyed via the acEV_TONE_DETECTED, and retrieved by the User according to their type and index (note that the Ringback tone should be defined in the Call Progress Tones table with the relevant type in order to get this event).
- When the tone detection event is received, Users can perform any action. For example, if the event is received with BUSY tone indication, Users can invoke acPSTNDisconnectCall() to end the call.
- The MFR2 destination number is collected using: EXPECTED_NUM_OF_DIGITS_MINUS_1 parameter for SET_COUNTER that the User defines at UserProt_defines_R2_MF.h. The counter function is used to trigger the script file for the last-but-one received, after receiving the last digit, the script file (acting as the outgoing register) initiates the A6/A3 FWD MF. Normally variant supports end of digit information (MF15) or silence at the end of the dialing (when MF15 is not used), a short pulse of MF3 (A3) is sent to indicate that the entire string of digits (according to Q442, 476) is received.
- Sending Group B digit by an incoming register requires invoking acCASAcceptCall() with a certain reason parameter. Six reason parameters are available:

  1. **CALLED_IDLE** - Subscribers line is free. Continue the call sequence. Usually should be followed by accept or reject.
  2. **CALLED_BUSY** - Subscriber line is busy. Perform disconnect procedures.
  3. **CALLED_CONGESTION** - Congestion encountered. Perform disconnect procedures.
  4. **CALLED_UNALLOCATED** - Dial number was not allocated. Perform disconnect procedures.
  5. **CALLED_RESERVE1** - Reserved for additional group B (user additional requirments).
  6. **CALLED_RESERVE2** - Reserved for additional group B (user additional requirments).

  Each reason generates a specific action, defined by the User, who modifies the script file. The action is then used to generate/respond with a group B MF (free, busy, etc.).
- Transfer Capability - This parameter of acPSTNPlaceCAll() function is used by the outgoing register to generate the service nature of the originating equipment. In most variants (countries) this is the same as the Calling Subscriber Categories but in some countries it is different, such as in R2 china protocol, it is referred to as the KD (Group II) digit.

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

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

Note: Applicable only to MFC-R2 protocol type.

Appendix E Utilities

This section describes the functionality and operation of a list of utilities supplied with the TrunkPack software package:

E.1 API Demonstration Utilities

LOCATION:

.\TPVoIPDemo\Debug\Gui.exe, .\VoIPSh\Debug\voipsh.exe

DESCRIPTION:

These utilities (supplied in source form) are designed to serve both as a reference for using the VoPLib/PSTNLib and as a demo applications which the user can run immediately after installing the board/module.

OPERATION:

The TPVoIPDemo application - This Windows NT based MFC application can exercise all VoPLib/PSTNLib API through a Graphical User Interface (GUI). The operation of this application is self explanatory.

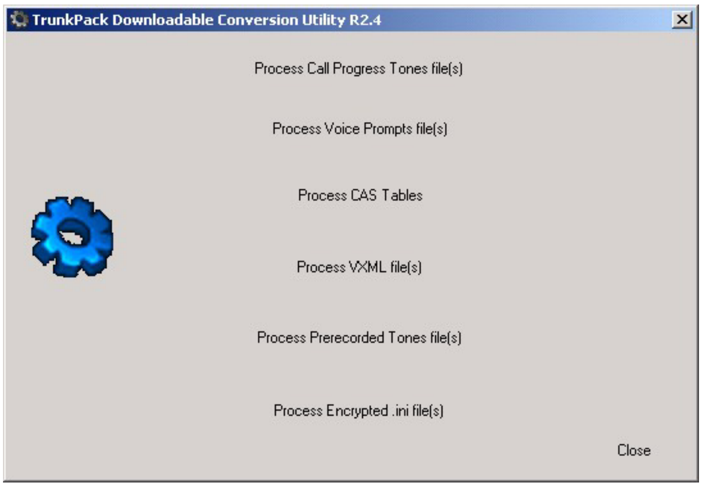
The voipsh application - This script-based application was designed for cross-platform use (it can be compiled both for Windows NT, Solaris, and Linux OS). Basically, it provides a wrapper VoPLib/PSTNLib API. The user can run any one of the API functions either through the console-like application or by using a script file.

E.2 TrunkPack Downloadable Conversion Utility

LOCATION:

.\Utilities\ Downloadables Construction Utility\DConvert240.exe

Figure E-1: Downloadable Conversion Utility Opening Screen



TrunkPack Downloadable Conversion Utility R2.4
Process Call Progress Tones file(s)
Process Voice Prompts file(s)
Process CAS Tables
Process VXML file(s)
Process Prerecorded Tones file(s)
Process Encrypted .ini file(s)
Close

This utility is used to generate the following:

- **Call Progress Tones configuration files**
- **Voice Prompts files**
- **CAS protocol configuration files**
- **VXML files (Applicable to IPmedia 2000 and IPmedia 3000 only)**
- **Prerecorded Tones files**
- **Encrypted *ini* files**

The above files can be used when:

- Configuring the board using the VoPLib function acOpenBoard().
- Using an ***ini*** file during BootP/DHCP session
- Using the Web Interface.

Some files may have usage restrictions as described under their usage information.

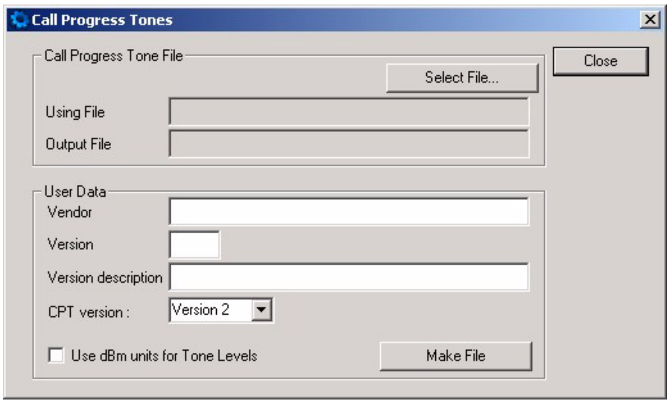
E.2.1 Generating the Call Progress Tones Configuration Files

➢ To produce a Call Progress Tones configuration file, take these 6 steps:

Step 1. Construct a **CPT** file according to the instructions in the section on “Configuration Files Description” in AudioCodes’ “VoPLib API Reference Manual”, Document #: LTRT-84001.

Step 2. Execute ***DConvert240.exe*** and click the **Process Call Progress Tones file(s)** button. The Call Progress Tones dialog appears.

Figure E-2: Call Progress Tones Dialog



Step 3. Click the **Select File** button and navigate to the location of the appropriate file created in the previous step. Click **Open**. (This automatically designates the output file to be the same name and path, but with the ***dat*** extension).

Step 4. Fill in the vendor, version and version description fields.

Step 5. If the **CPT** file used dBm units when setting tone levels, click a checkmark in the **use dBm units for tone levels** checkbox.

Step 6. Click the **Make File** button. The last action generates the ***dat*** file in the same directory and with the same name as the **CPT** file.

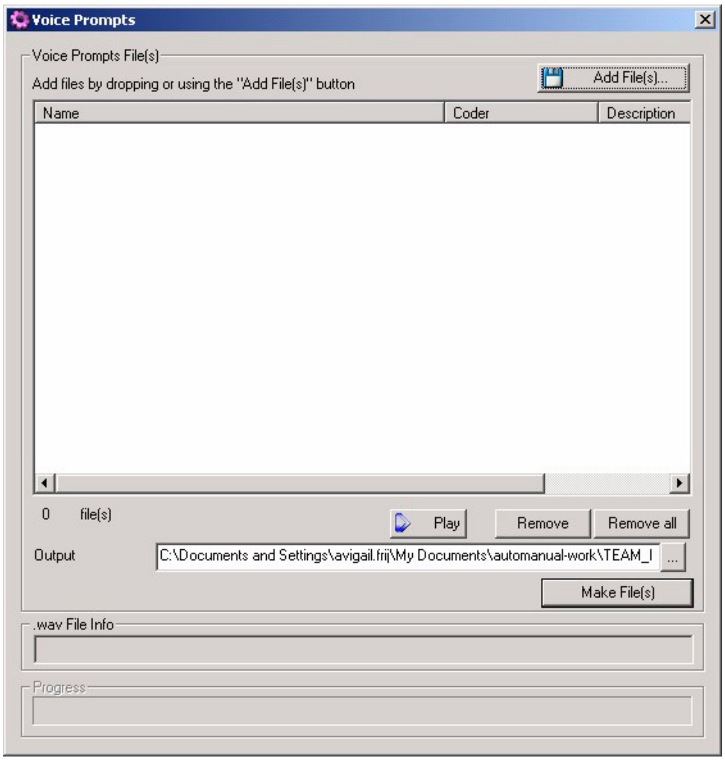
E.2.2 Generating Voice Prompts Files

➢ To generate a Voice Prompts file, take these 12 steps:

Step 1. Create raw Voice Prompt files according to the instructions in the section on “Relaying DTMF/MF Digits” in AudioCodes’ “VoPLib User’s Manual”, Document #: LTRT-84401). From version 2.2, **DConvert** supports ***wav*** files as well.

Step 2. Execute ***DConvert240.exe*** and press the **Process Voice Prompts file(s)** button. The Voice Prompts window appears.

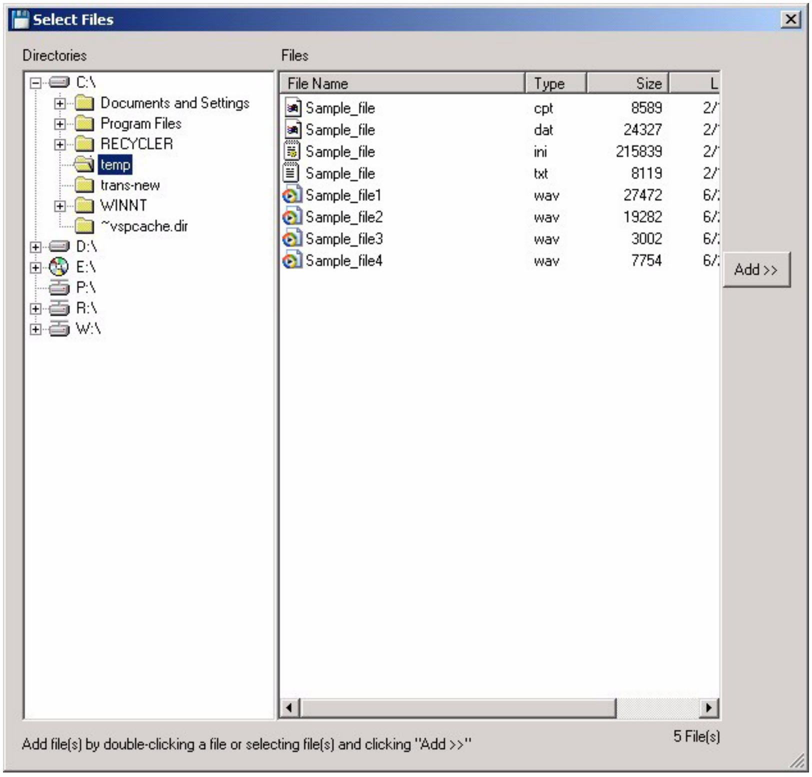
Figure E-3: Voice Prompts Window



Output: C:\Documents and Settings\avigail.frij\My Documents\automanual-work\TEAM_I

Step 3. Click the **Add Files** button in the upper right corner. The Add Files window appears.

Figure E-4: Select Files Window



Select Files

Directories

- C:\
  - Documents and Settings
  - Program Files
  - RECYCLER
  - temp
  - trans-new
  - WINNT
  - ~vspcache.dir
- D:\
- E:\
- P:\
- R:\
- W:\

Files

File Name	Type	Size	L
Sample_file	cpt	8589	2/
Sample_file	dat	24327	2/
Sample_file	ini	215839	2/
Sample_file	txt	8119	2/
Sample_file1	wav	27472	6/:
Sample_file2	wav	19282	6/:
Sample_file3	wav	3002	6/:
Sample_file4	wav	7754	6/:

Add file(s) by double-clicking a file or selecting file(s) and clicking "Add >>"

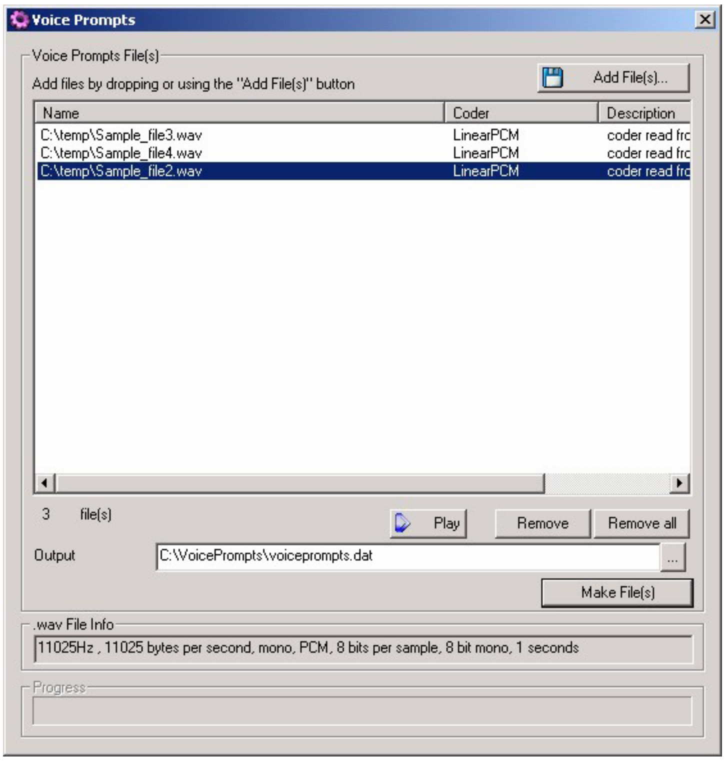
5 File(s)

Step 4. Select the files from the previous step either by one of these actions:

a. Click the **Add Files** button. The Add Files window appears.. Navigate to the appropriate file and click the **Add>>** button. (To close the Add Files window, click the ☒ Exit button.)

b. Drag-drop to the Voice Prompts window.

Figure E-5: Voice Prompts Window with *wav* Files



Add files by dropping or using the "Add File(s)" button

Name	Coder	Description
C:\temp\Sample_file3.wav	LinearPCM	coder read frc
C:\temp\Sample_file4.wav	LinearPCM	coder read frc
C:\temp\Sample_file2.wav	LinearPCM	coder read frc

3 file(s)

Output C:\VoicePrompts\voiceprompts.dat

.wav File Info

11025Hz , 11025 bytes per second, mono, PCM, 8 bits per sample, 8 bit mono, 1 seconds

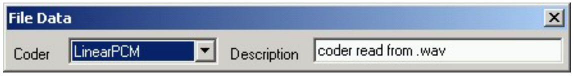
Step 5. Arrange the files as desired by dragging and dropping them from one location in the list to another location.

Note: The sequence of files in the Add Files window defines the Voice Prompt ID and the coder type (to be used by the acPlayVoicePrompt() function).

Step 6. Use the **Play** button to preview the sound of the wav file. Use the **Remove** and **Remove all** buttons to remove files in the list as needed.

Step 7. Select a coder for each file by first selecting the file (or files) and then double-clicking or right-clicking on it. The File Data window appears.

Figure E-6: File Data Window



Coder LinearPCM Description coder read from .wav

Step 8. From the Coder Dropdown list, select a coder type (Linear PCM is the default).

Step 9. Press **Enter**.

Step 10. In the Description field, enter a description (optional).

Note: For ***wav*** files, a coder is auto selected from the ***wav*** file header.

Step 11. Using the ... Browse button, select the appropriate output file name and location. (The default file name is ***voiceprompts.dat***).

Step 12. Click the **Make File(s)** button to generate the voice prompts file. The Progress bar at the bottom of the window is activated.

The generated file can be used only for downloading using the ***ini*** file facility or using acOpenRemoteBoard() in full configuration operation mode. When using the acAddVoicePrompt(), use the single raw voice prompt files.

E.2.3 Generating CAS Protocol Configuration Files

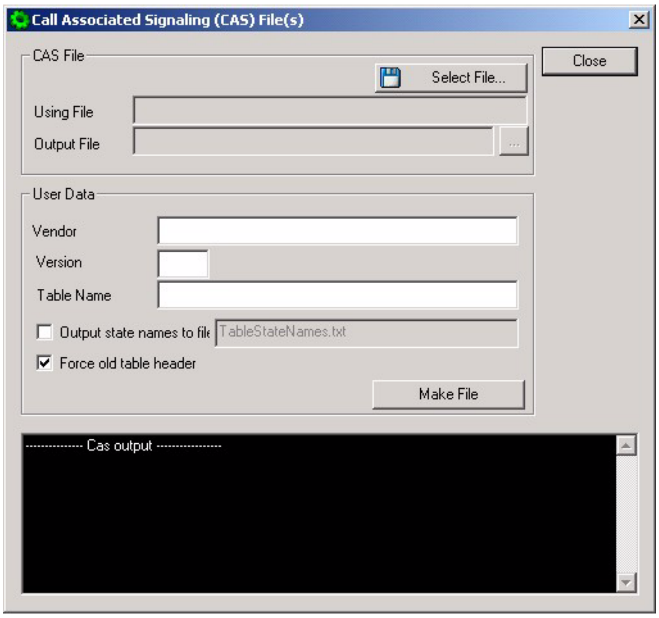
➢ To produce CAS Protocol configuration file, take these 8 steps:

Step 1. Construct the CAS protocol ***xxx.txt*** and ***xxx.h*** files according to the instructions in the sections on “Caller ID Support” and “CAS Protocol Table” in AudioCodes’ “VoPLib User’s Manual”, Document #: LTRT-84401.

Step 2. Copy the files generated in the previous step (or at least the ***xxx.h*** file) to the same directory in which ***DConvert240.exe*** is located and make sure that the two following files, ***CASSetup.h*** and ***CPP.exe***, are located in the same directory.

Step 3. Execute ***DConvert240.exe*** and click the **Process CAS Tables** button. The Call Associated Signaling (CAS) Window appears.

Figure E-7: Call Associated Signaling (CAS) Window



□ Output state names to file TableStateNames.txt

☑ Force old table header

```
--------------- Cas output ---------------
```

Step 4. Click the **Select File** button. A Browse window appears.

Step 5. Navigate to the desired location and select the file to be converted. (This automatically designates the output file as the same name and path, but with the ***dat*** extension).

Step 6. Fill in the vendor name and version number fields. If appropriate, alter the table name that is auto-generated with the selected table file name.

Step 7. If the file to be converted uses the new table header, un-check the **Force old table header** checkbox.

Step 8. Click the **Make File** button. The last action should generate the ***dat*** file in the same directory and with the same name as the ***txt*** file.

E.2.4 Generating VXML Files

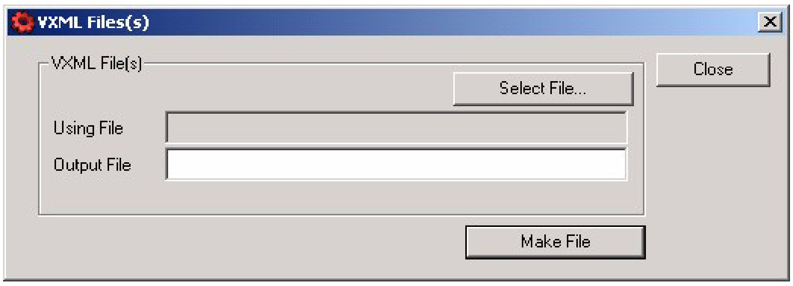
Note: The VXML utility is applicable to IPmedia 2000 and IPmedia 3000 only.

➢ To produce a VXML file, take these 5 steps:

Step 1. Prior to the conversion process the user should prepare a valid VXML file.

Step 2. Execute ***DConvert240.exe*** and click the **Process VXML file(s)** button. The VXML file(s) window appears.

Figure E-8: VXML Window



Step 3. Click the **Select File** button. A Browse window appears.

Step 4. Navigate to the desired location and select the file to be converted. (This automatically designates the output file as the same name and path, but with the ***dat*** extension).

Step 5. Click the **Make File** button. The VXML file is generated.

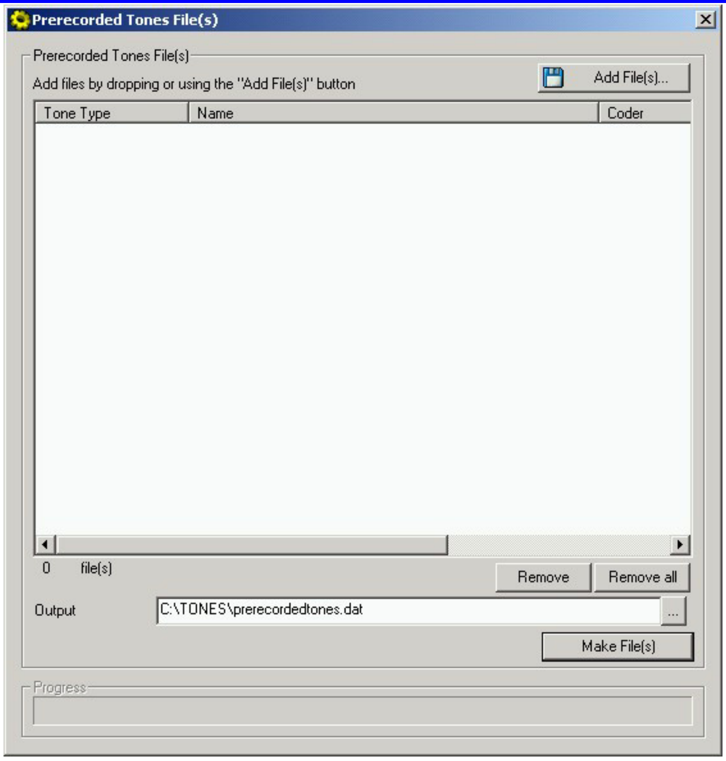
E.2.5 Generating Prerecorded Tones Files

➢ To generate a Prerecorded Tones file, take these 11 steps:

Step 1. Prior to the conversion process, the user should prepare the appropriate prerecorded tones file(s).

Step 2. Execute ***DConvert240.exe*** and press the **Process Prerecorded Tones file(s)** button. The Prerecorded Tones file(s) Window appears.

Figure E-9: Prerecorded Tones file(s Window



Add files by dropping or using the "Add File(s)" button

0 file(s)

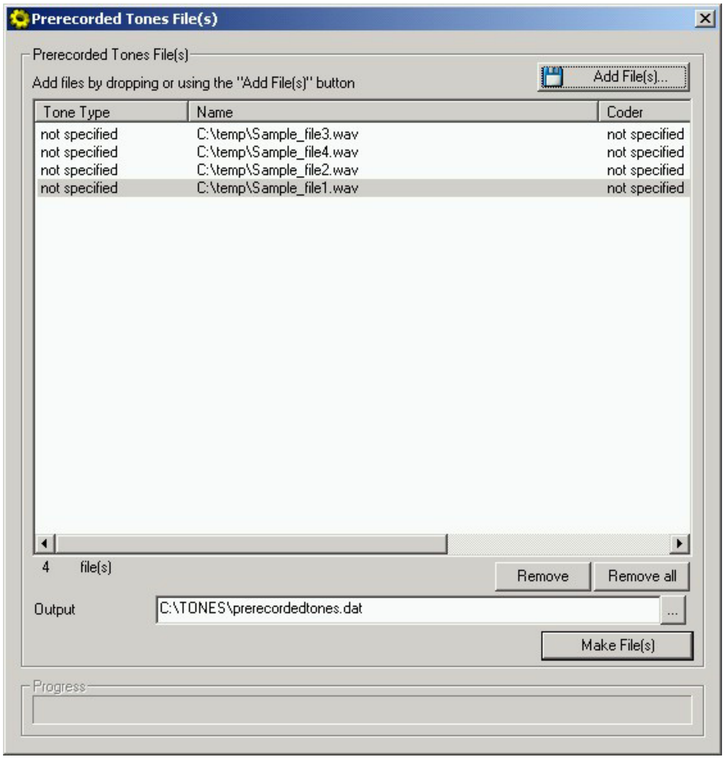
Output C:\TONES\prerecordedtones.dat

Step 3. Select the files from the previous step either by one of these actions:

a. Click the **Add Files** button. The Add Files window appears (refer to Figure E-4, on page 240). Navigate to the appropriate file and click the **Add>>** button. (To close the Add Files window, click the ☒ Exit button.)

b. Drag-drop to the Voice Prompts window.

Figure E-10: Prerecorded Tones file(s) Window with *wav* Files



Prerecorded Tones File(s)

Add files by dropping or using the "Add File(s)" button

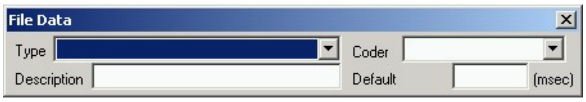
Tone Type	Name	Coder
not specified	C:\temp\Sample_file3.wav	not specified
not specified	C:\temp\Sample_file4.wav	not specified
not specified	C:\temp\Sample_file2.wav	not specified
not specified	C:\temp\Sample_file1.wav	not specified

4 file(s)

Output C:\TONES\prerecordedtones.dat

Step 4. Select a tone type, coder and default duration for each file (select the file(s)) then double click or right click on it. The File Date window appears.

Figure E-11: File Data Window



Step 5. From the Type Dropdown list, select a Ring parameter type.

Step 6. From the Coder Dropdown list, select a coder type (G.711 ALaw_64, G.711 MuLaw, or Linear PCM).

Step 7. In the Description field, enter a description An optional (optional).

Step 8. In the Default field, enter the number of miliseconds.

Step 9. Press **Enter**.

Step 10. Select the appropriate output file (it defaults to ***prerecordedtones.dat)***.

Step 11. Click **Make File(s)** to generate the prerecorded tones file.

E.2.6 Generating Encrypted *ini* File(s)/Decrypting *ini* files

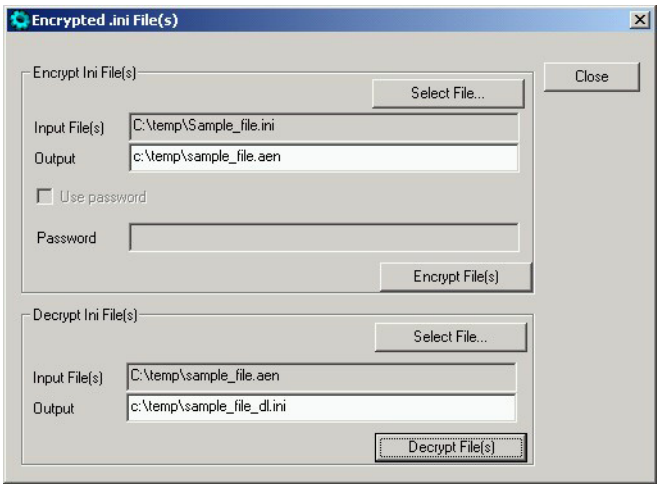
The ***ini*** file can be both encrypted and decrypted using **DConvert**. Encryption usually takes place before downloading an ***ini*** file to the board while decryption usually takes place after uploading an ***ini*** file from the board.

➢ To Encrypt an *ini* file, take these 5 steps:

Step 1. Prior to the encryption process, the user should prepare the appropriate ***ini*** file either by uploading from the board or by constructing one (refer to Section 6 on page 97).

Step 2. Execute ***DConvert240.exe*** and click the **Process Encrypted *ini* file(s)** button. The Encrypted *ini* file(s) window appears.

Figure E-12: Encrypted *ini* file(s) window



Input File(s): C:\temp\Sample_file.ini
Output: c:\temp\sample_file.aen
□ Use password
Password:

Input File(s): C:\temp\sample_file.aen
Output: c:\temp\sample_file_dl.ini

Step 3. In the **Encrypt Ini File(s)** area, press the **Select File…** Button. A Browse window appears.

Step 4. Select the ***ini*** file to be encrypted. An output file is suggested using the same path and an extension ***aen***.

Step 5. Press **Encrypt File(s)** to create the encrypted file. The encrypted is created.

➢ To Decrypt an *ini* file follow these 4 steps:

Step 1. Prior to the decryption process the user should prepare the appropriate encrypted ***ini*** file either by uploading from the board or by using the encryption process on an exiting ***ini*** file.

Step 2. Execute ***DConvert240.exe*** and click the **Process Encrypted *ini* file(s)** button.

Step 3. In the **Decrypt Ini File(s)** area, press **Select File(s)** and select the ***aen*** file to be decrypted. An output file is suggested using the same path and the extension ***_dl.ini***.

Step 4. Press "**Decrypt File(s)**" to create the decrypted file.

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

Encrypted ***ini*** file can be loaded using the regular ***ini*** file procedure. To upload a file from a device, use the Web Interface (refer to Section 6.4.6 on page 136).

E.3 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 ISDN/SS7/ATM Protocols

➢ To generate a readable text file out of the binary trace file when using ISDN/SS7/ATM 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.

E.4 Enabling PSTN Trace via the Web

This section contains direction on starting and collecting the PSTN trace via the Web. (Refer to Figure E-13 below for a view of the Trunk Traces). Also, note if the PSTN trace is of a ISDN/SS7/ATM or CAS collection based on the physical device 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.

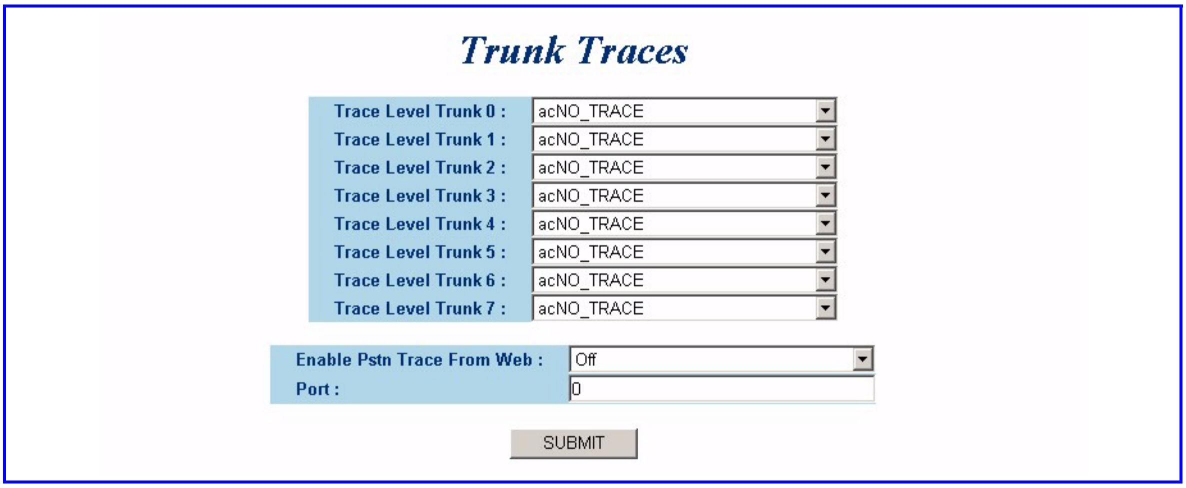
Step 9. Determine the UDP port (the same as in step 3).

Step 10. Click the SUBMIT button. The board starts to send the trace messages.

Step 11. In the UDP2File utility (Refer to Figure E-14 below) you should see the number in the packets counter increasing.

Step 12. In the UDP2File utility, click the **Stop** button. The file created is a bianary file. To

Figure E-13: Trunk Traces

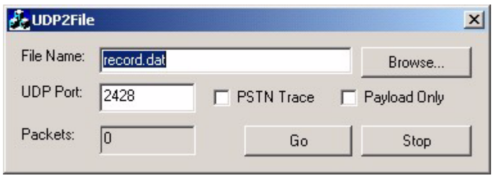


Trunk Traces

Trace Level Trunk 0 :	acNO_TRACE
Trace Level Trunk 1 :	acNO_TRACE
Trace Level Trunk 2 :	acNO_TRACE
Trace Level Trunk 3 :	acNO_TRACE
Trace Level Trunk 4 :	acNO_TRACE
Trace Level Trunk 5 :	acNO_TRACE
Trace Level Trunk 6 :	acNO_TRACE
Trace Level Trunk 7 :	acNO_TRACE

Enable Pstn Trace From Web :	Off
Port :	0

Figure E-14: UDP2File Utility



UDP2File

File Name: record.dat

UDP Port: 2428 □ PSTN Trace □ Payload Only

Packets: 0

E.5 MGCP Tester Utility

LOCATION:

.\Utilities\MGCPTester

DESCRIPTION:

This utility serves as a simulation for the MGCP Call Agent. It can send any MGCP command, as well as run complicated scripts. This utility can be used to exercise the MGCP clients embedded in TrunkPack series boards and modules.

OPERATION:

The MGCP tester demo application operation is self-explanatory.

E.6 MEGACO Tester Utility

LOCATION:

.\Utilities\MEGACOTester

DESCRIPTION:

This utility serves as a simulation for the MEGACO Call Agent. It can send any MEGACO command, as well as run complicated scripts. This utility can be used to exercise the MAGACO clients embedded in TrunkPack series boards and modules.

OPERATION:

The MEGACO tester demo application operation is self-explanatory.

Reader's Notes

Appendix F MGCP Compliance

The MGCP Compliance Matrix Table F-1 below summarizes the supported MGCP features respectively. The Reference column in Table F-1 refers to IETF RFC 3435 from January 2003 (which replaced RFC 2705).

F.1 MGCP Compliance Matrix

Table F-1: MGCP Compliance Matrix (continues on page 251 to page 263)

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
1	Case Sensitivity			Page 62
2	Command Verb	Yes		
3	Parameters	Yes		
4	EndPoint identifier			2.1.2 Page 15
5	"*" Wild-carding	Yes		
6	"$" Wild-carding	Yes		
7	Domain name for Call Agent	No	IP address is used to identify Call Agent	Pages 23, 96
8	Digit Maps	Yes	12 Digit Maps Such as: R: [0 -9](D) R: D/X(D) D: xxxx | 88# | 7xx|xxxT 5x.T	2.1.5 Page 24
9	Timer indication - T	Yes	Interdigit timer Fixed Timer of 4 sec is used	6.1.2 Pages 27, 112
10	Digits and Letters			Page 27
11	#	Yes		
12	X	Yes		
13	X.	Yes	X. - Arbitrary number of X Occurrences	Page 26
14	*	Yes		
15	[1-7]	Yes	For digit maps	Page 25
16	A,B,C,D	Yes		
17	Event names	Yes		Pages 26, 2.1.6 page 37, 3.2.2.7 page 73
18	Wildcard notations (X, $, *,all)	Yes		Page 27

Table F-1: MGCP Compliance Matrix (continues on page 251 to page 263)

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
19	Optional connection ID (G/rt@A3F58)	No		Page 28
20	Signals on/off			Page 29
21	On/Off (OO)	Yes		
22	Time out (TO)	Yes		
23	Brief (BR)	Yes		
24	Using "+", "-" to turn on/off the "OO" Signal	Yes	for DTMF digits	
25	Connection modes			Pages 31, 3.2.2.6 page 73
26	Inactive	Yes		Page 31
27	Send only	Yes		Pages 21,31
28	Receive only	Yes		Page 31
29	Send/receive	Yes		
30	Conference	Yes		
31	Data	No	3 participants only	
32	Loopback	Yes		
33	Continuity test	Yes		
34	Network loop back	No		
35	Network continuity (netwtest)	No		
36	Endpoint Configuration command	No		2.3.1 page 32, 44
37	Notification Request command			2.3.2 page 33
38	Endpoint ID	Yes		2.1.2 page 15
39	Notified Entity	Yes		Pages 24, 38
40	RequestedEvents (with associate actions)	Yes	If not specified, notifications is send to command originator	Pages 34, 3.2.2.8 74
41	RequestIdentifier	Yes		2.3.2 page 33
42	DigitMap	Yes	Limited to 8 digits	2.1.5 page 24
43	Defined explicitly or through a previous command	Yes		Page 35
44	SignalRequests	Yes		3.2.2.9 page 76

Table F-1: MGCP Compliance Matrix (continues on page 251 to page 263)

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
45	QuarantineHandling			36, 3.2.2.12 page 77
46	Discard	No		
47	Process loop	No	Events are always processed	
48	Process	Yes	Events are always processed	
49	Loop	No		
50	Process step by step			
51	Requested events	Yes	Empty buffer	
52	Digit map	Yes	Empty buffer	
53	DetectEvents	Yes	Empty buffer	37, 3.2.2.13 page 77
54	Encapsulated Endpoint Configuration	Yes		37
55	Event associated actions			Pages 34, 76
56	Notify event immediately with all accumulated events	Yes		35
57	Swap audio	No		34
58	Accumulate event in buffer, but do not notify yet	Yes		
59	Accumulate according to digit map	Yes		33
60	Keep signal active	No		34
61	Process Embedded Notification Request	Yes		Pages 34, 36, 75
62	Ignore the event	Yes		34
63	Supporting two or more actions, hf(S,N)	Yes	Combining actions	Pages 35, 75
64	Persisted events	Yes	Notify off-hook / on-hook	Page 34, 53
65	Number of active connection on an endpoint	1 or 2		

Table F-1: MGCP Compliance Matrix (continues on page 251 to page 263)

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
66	Synchronization of Signalrequest action with detected event	Yes	TO (Timeout) signals stop when one of the requested events is detected Example 1: Ringing stops if off-hook event was detected Example 2: Dial tone stops if DTMF was detected	Page 36
67	Notification request with empty signal list for stopping tone generation	Yes		Page 36
68	Detection of events on Connections	No		Page 37
69	Notifications			Page 37
70	EndpointID	Yes		
71	NotifiedEntity	Yes		
72	RequestIDentifier	Yes		
73	ObservedEvents	Yes		3.2.2.10 page 73, 37, 52, 65
74	Create Connection command			3.2.2.2 Page 87
75	CallID	Yes		Page 39
76	Endpoint	Yes		Page 39
77	NotifiedEntity	Yes		Page 39
78	Multiple connections per endpoint	Yes	- Only two connections - Only one of them can be in send/send receive mode	Page 39
79	LocalConnection Options			Pages 39, 40
80	Encoding method	Yes	One value List of values not supported	Page 41
81	Packetization period	Yes	SDP "ptime" parameter Supports only 20 msec	Pages 40, 41
82	Bandwidth	No	Single value or range	Page 40
83	Type of Service (TOS)	Yes	2 Hex digits	
84	Echo cancelation	Yes		
85	Silence suppression	Yes	-32 to +31 value	
86	Gain control	Yes	32..31 value	Page 41
87	Reservation service	No		Page 38

Table F-1: MGCP Compliance Matrix (continues on page 251 to page 263)

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
88	RTP security	No	Providing Key as per RFC 2327	Page 39
89	Type of network (IN, Local, ATM)	Yes		
90	Vendor specific extensions	No		Page 40
91	Mode	Yes		Pages 41, 42
92	RemoteConnection Descriptor	Yes		Page 42
93	SecondEndpointID	Yes		Page 42
94	Encapsulated Notification Request			Page 77
95	R:	Yes		
96	S:	Yes		
97	Encapsulated Endpoint Configuration	Yes		
98	Create Connection return parameters			
99	ConnectionID	Yes		
100	SpecificEndpointID ("Z")	Yes		Page 40
101	LocalConnection Descriptor	Yes		
102	SecondEndpointID	Yes		Pages 39, 42
103	Secondconnection ID	Yes		
104				
105	ModifyConnection			2.3.4 page 46
106	CallID	Yes		
107	Endpoint	Yes		
108	Connection ID	Yes		
109	NotifiedEntity	Yes		
110	LocalConnection Options	Yes	CreateConnectionCmnd refers	
111	Mode	Yes		Page 42
112	RemoteConnection Descriptor	Yes		Pages 42, 43

Table F-1: MGCP Compliance Matrix (continues on page 251 to page 263)

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
113	Encapsulated Notification Request			Pages 43, 45, 46, 77
114	R:	Yes		
115	S:	Yes		
116	Encapsulated Endpoint Configuration	No		
117	Modify Connection return parameters			
118	LocalConnection Descriptor	Yes	Returns if local connection parameters were modified	Page 44
119	Delete Connection (from Call Agent)			2.3.5 page 46
120	CallID	Yes		
121	EndpointID	Yes		
122	ConnectionID	Yes		
123	Encapsulated Notification Request			
124	R:	Yes		
125	S:	Yes		
126	Encapsulated Endpoint Configuration	No		
127	DeleteConnection return Parameters			
128	Connection Parameters			Page 50
129	Number of packets send	Yes		Pages 47, 49
130	Number of octets send	Yes	Such as on hook or off hook state per endpoint	Pages 47, 49
131	Number of packets received	Yes		Pages 47, 49
132	Number of octets received	Yes		Pages 47, 49
133	Number of packets lost	Yes		Pages 48, 49
134	Inter-packet arrival jitter	Yes		Pages 48 - 50
135	Average transmission delay - latency	Yes		Pages 48 - 50

Table F-1: MGCP Compliance Matrix (continues on page 251 to page 263)

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
136	Delete Connection (from gateway)	No		2.3.6 page 51, 30
137	CallID			
138	EndPointID			
139	ConnectionID			
140	ReasonCode			
141	Connection Parameters			
142	DeleteConnection (multiple connections)	Yes		2.3.7 page 51
143	CallID	Yes		
144	EnpointID	Yes		
145	Audit Endpoint	Partially		2.3.8 page 52
146	EndpointID			
147	RequestedInfo	Yes		3.2.2.11 pages 76, 53
148	Wildcard convention * ("all of")	Yes		Pages 27, 52
149	AuditEndpoint Return parameters			
150	Endpoint ID list, "Z='	Yes	If nothing was requested, return positive ack if connection exists	
151	RequestedEvents	Yes		Page 53
152	Including actions associated with the events	Yes		
153	DigitMap	Yes		Page 53
154	SignalRequests TO signals currently active On/Off signals currently ON Pending Brief signals	Yes		Page 53
155	RequestIDentifier	Yes		Page 53
156	NotifiedEntity	Yes		Page 53
157	Connection Identifiers	Yes		Page 53

Table F-1: MGCP Compliance Matrix (continues on page 251 to page 263)

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
158	DetectEvents	Yes	See connection parameters under delete connection	Pages 53,77
159	ObservedEvents	Yes	Call agent IP is defined in BootP server or ***ini*** file	Page 53
160	EventStates	Yes		Page 54
161	BearerInformation	No	"*" Is reported only in MP-200 for all its endpoints	Page 54
162	RestartReason	Yes		Page 54
163	RestartDelay	Yes		Page 54
164	ReasonCode	No		Page 54
165	Capabilities			3.2.2.3 pages 70,71,72, 54
166	List of supported codecs	Yes		Page 54
167	PacketizationPeriod	No		Page 54
168	Bandwidth	No		Page 54
169	Echo Cancelation	No		Page 54
170	Silence Suppression	No		Page 54
171	Gain Control	No		
172	Type of Service	No		Page 54
173	Resource reservation	No	OK response	
174	Encryption key	No	Connection was Deleted	
175	Type of network	Yes	Transient error in transactions	
176	Supported Event Packages	Yes	Phone is already off hook	
177	Connection Modes	Yes	Phone is already on hook	Page 55
178	Audit Connection	Yes	Unknown endpoint	2.3.9 page 55
179	ConnectionID	Yes	Endpoint not ready	Page 55
180	RequestedInfo	Yes	Insufficient resource	Page 57
181	Audit Connection Return Parameters		Protocol error	
182	CallID	Yes	Unrecognized extension	4.1 Page 119
183	Notified Entity	Yes	Cannot detect event	Page 56
184	LocalConnection Options	Yes	Cannot generate signal	Pages 40, 41, 56
185	Mode	Yes	Cannot send announcement	Page 56

Table F-1: MGCP Compliance Matrix (continues on page 251 to page 263)

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
186	RemoteConnection Descriptor	No	Incorrect transaction id	Page 56
187	LocalConnection Descriptor	No	Unknown call id	Page 56
188	Connection Parameters	Yes		2.3.5 pages 48, 56
189	Restart in Progress (RSIP)		Internal inconsistency	2.3.10 page 56
190	EndpointID			Page 56
191	"All of" wildcard (*)	Yes		Page 57
192	Restart Method	Partially		Page 57
193	Graceful	No		Page 57
194	Forced	Yes		Page 57
195	Restart	Yes		Page 57
196	Disconnected	Yes	Not supporting Domain name ID, instead use IP address	Page 57
197	Cancel-graceful	No	Not supporting Domain name ID, instead use IP address	Page 57
198	Restart Delay	No	UDP port definition is not supported	Page 57
199	ReasonCode	No	K: 6257 K: 6234-6255	Page 57
200	Restart in progress return parameters (notified entity & return code)	No		Page 58
201	Return Codes and Error Codes	Partially		2.4 page 58
202	100	No	The transaction is currently being executed An actual completion message will follow later	Page 59
203	200	Yes	The requested transaction was executed normally	Page 59
204	250	Yes	The connection was deleted	Page 59
205	400	Yes	The transaction couldn't be executed due to a transient error	Page 59
206	401	Yes	The phone is already off hook	Page 59
207	402	Yes	The phone is already on hook	Page 59

Table F-1: MGCP Compliance Matrix (continues on page 251 to page 263)

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
208	405	Yes	The transaction could not be executed, because the endpoint is "restarting".	
209	500	Yes	The transaction could not be executed because the endpoint is unknown	Page 59
210	501	Yes	The transaction could not be executed because the endpoint is not ready	Page 59
211	502	Yes	The transaction could not be executed because the endpoint does not have sufficient resources	Page 59
212	503	No	"All of" wildcard not fully supported The transaction contained an "all of" wildcard, however NotificationRequests non-empty	
213	504	Yes	Unknown or unsupported command.	
214	505	Yes	Unsupported RemotedConnectionDescriptor	
215	506	No	Unable to satisfy both LocalConnectionOptions and RemoteConnection Descriptor	
216	507	Yes	Unsupported functionality	Page 71
217	510	Yes	The transaction could not be executed because a protocol error was detected	Page 59
218	511	No	The transaction could not be executed because of the command contained an unrecognized extension	Page 59
219	512	No	The transaction could not be executed because the gateway is not equipped to detect one of the requested events	Page 59
220	513	Yes	The transaction could not be executed because the gateway is not equipped to generate one of the requested signals	Page 60
221	514	No	The transaction could not be executed because the gateway cannot send the specified announcement	Page 60
222	515	Yes	The transaction refers to an incorrect connection ID	Page 60

Table F-1: MGCP Compliance Matrix (continues on page 251 to page 263)

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
223	516	Yes	The Transaction refers to an unknown call ID	Page 60
224	517	Yes	Unsupported or invalid mode	
225	518	No	Unsupported or unknown package	
226	519	Yes	Gateway does not have a digit map	
227	520	Yes	The transaction could not be executed because the GateWay is restarting	
228	521	Yes	Endpoint redirected to another Call Agent endpoint is restarting	
229	522	Yes	No such event or signal	
230	523	Yes	Unknown action or illegal combination of actions	
231	524	Yes	Internal inconsistency in localConnectionOptions	
232	525-531	No		
233	532	Yes	Unsupported value in LocalConnectionOptions	
234	533	Yes	Response too big	
235	534	Yes	Codec negotiation failure	
236	535	Yes	Packetization period not supported	
237	536	No	Unknown or unsupported RestartMethod	
238	537	No	Unknown or unsupported digit map extension	
239	538	Yes	Event or SIgnal error	
240	Reason Codes (900, 901, 902)	No		2.5 page 61
241	900			
242	901			
243	902			
244	MGCP Command Header		The transaction could not be executed because the GateWay is restarting	3.2 page 62
245	Endpoint identifier	Yes		3.2.1.3 page 64
246	Notified entity	Yes		Page 64

Table F-1: MGCP Compliance Matrix (continues on page 251 to page 263)

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
247	In notified entity, If port # is omitted, using default MGCP port (2427)	Yes		Page 65
248	Response Acknowledgement	Yes (receive side only)		3.2.2.1 Page 68
249	Encoding of Session Description - SDP			3.5 page 86
250	SDP parameters: v,c,m,a	Yes		
251	Using RTPMAP attribute to define encoding of dynamic audio formats	No		Page 87
252	Optional Ptime attribute to define packet duration	No		Page 87
253	IP address of remote/local gateways	Yes		Page 87
254	Transmission over UDP			3.6 page 91
255	Transaction identifiers	Yes		Page 92
256	Receiving Duplicated transaction IDs	Yes		Page 63, 92
257	Retransmission timers	Yes		3.6.3 page 93
258	Piggy backing	Yes		3.6.4 Page 94
259	Provisional responses	No		3.6.5 Page 94
260	MultipleCall Agents and Call Agent Redundancy	Yes		
261	States, failover and race conditions	Partially		3.2 Page 79
262	Basic Assumptions			3.2.1.3 Page 81
263	Failover Assumptions and Highlights			4.1 Page 119
264	Call Agents DNS	No		4.1 Page 119
265	Notified Entity for endpoint	Yes		4.1 Page 119

Table F-1: MGCP Compliance Matrix (continues on page 251 to page 263)

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
266	Responses send to source address	Yes		
267	Backup Call Agent	Yes		
268	Security, Retransmission, Detection			4.2 page 96
269	Commands retransmission	Yes		
270	Checking presence of new CA	No		Page 99
271	Disconnecting endpoint/gateway	No		Page 99
272	Race Conditions			4.3 Page 99
273	Quarantine list	Yes		4.3.1 page 99
274	Explicit detection	Yes		4.3.2 page 103
275	Ordering of commands	Yes		Page 104
276	Restart avalanche	Yes		Page 105
277	Disconnected endpoints	No		Page 107
278	Security requirements			5. Page 108
279	MGCP IP security (RFC 1825)	No		Page 108

Reader's Notes

Appendix G MEGACO Compliance

The MEGACO Compliance Matrix Table G-1 below summarizes the supported MEGACO features. The Reference column in Table G-1 refers to IETF RFC 3015 from September 2002.

G.1 MEGACO Compliance Matrix

Table G-1: MEGACO Compliance Matrix (continues on page 265 to page 275)

Reference (in RFC3015)	Item	Support	Comments
7	**Commands 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	
	V.91	No	

Table G-1: MEGACO Compliance Matrix (continues on page 265 to page 275)

Reference (in RFC3015)	Item	Support	Comments
	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	
	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	

Table G-1: MEGACO Compliance Matrix (continues on page 265 to page 275)

Reference (in RFC3015)	Item	Support	Comments
	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	
	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	

Table G-1: MEGACO Compliance Matrix (continues on page 265 to page 275)

Reference (in RFC3015)	Item	Support	Comments
	ServiceChangeMGCId	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	
7.1.16	**Package Descriptor**		
		Yes	
7.1.17	**Observed Events Descriptor**		
	Request Identifier	Yes	
	Event	Yes	
	Detection Time	Yes	
7.1.18	**Topology Descriptor**		**Topology used only for conference**
	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	

Table G-1: MEGACO Compliance Matrix (continues on page 265 to page 275)

Reference (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	
7.2.2	**Modify**		
	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	
7.2.3	**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	**Audit Value**		
	TerminationID	Yes	
	Wildcard	Yes	
	AuditDescriptor	Yes	
	Media	Yes	
	Modem	No	

Table G-1: MEGACO Compliance Matrix (continues on page 265 to page 275)

Reference (in RFC3015)	Item	Support	Comments
	Mux	No	
	Event	Yes	
	Signal	Yes	
	DigitMap	Yes	
	ObservedEvents	Yes	
	EventBuffer	Yes	
	Statistics	Yes	
	Packages	Yes	
7.2.6	**Audit Capabilities**		
	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	
	Restart	Yes	
	Disconnected	Yes	
	Handoff	Yes	

Table G-1: MEGACO Compliance Matrix (continues on page 265 to page 275)

Reference (in RFC3015)	Item	Support	Comments
	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	
	ServiceChangeMgcId	Yes	
	TimeStamp	Yes	
7.2.9	**Manipulating and Auditing Context Attributes**		
		Yes	
7.2.10	**Generic Command Syntax**		
	Text Encoding	Yes	
	Binary Encoding	Yes	
7.3	**Command Error**		
	400 - Bad Request	Yes	
	401 - Protocol Error	Yes	

Table G-1: MEGACO Compliance Matrix (continues on page 265 to page 275)

Reference (in RFC3015)	Item	Support	Comments
	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	
	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	

Table G-1: MEGACO Compliance Matrix (continues on page 265 to page 275)

Reference (in RFC3015)	Item	Support	Comments
	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 - Unsupported Media Type	Yes	
	517 - Unsupported 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	
	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	

Table G-1: MEGACO Compliance Matrix (continues on page 265 to page 275)

Reference (in RFC3015)	Item	Support	Comments
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	
	normalMGEcecutionTime	Yes	
	normalMGCEcecutionTime	Yes	
8.3	**Messages**		
	Receive Messages	Yes	
	Send Messages	Yes	
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	
	Secondary Call Agents support	Yes	
	Cyclic check for Call Agent	Yes	

Table G-1: MEGACO Compliance Matrix (continues on page 265 to page 275)

Reference (in RFC3015)	Item	Support	Comments
11.3	**Negotiation of Protocol Version**		
		No	
11.4	**Failure of an MG**		
		No	
11.5	**Failure of an MGC**		
		Yes	

Reader's Notes

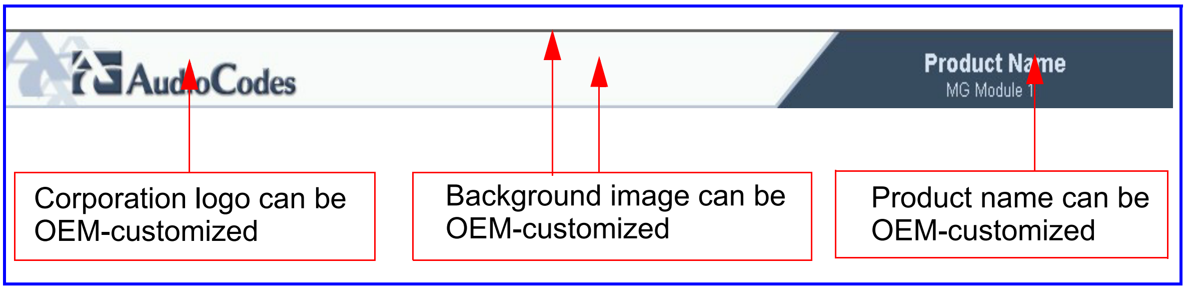
Appendix H Customizing the AudioCodes Web Interface

OEMs incorporating AudioCodes' devices into their portfolios can customize the device's Web interface to suit their specific corporate logo and product naming conventions.

OEMs can customize the Web interface's title bar (AudioCodes' title bar is shown in Figure H-1 and an example of a customized title bar is shown in Figure H-2).

Note: The product name appears according to the AudioCodes product utilized together with the AudioCodes Web Interface.

Figure H-1: Web Interface Title Bar

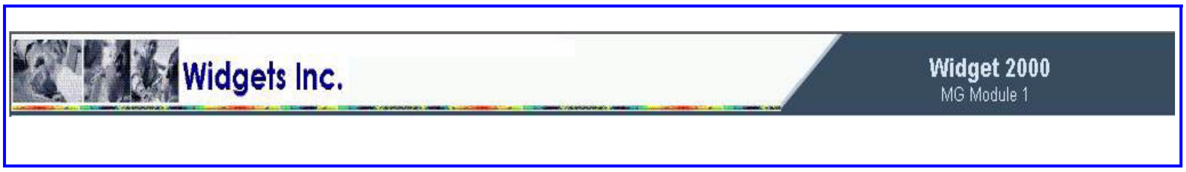


Corporation logo can be OEM-customized

Background image can be OEM-customized

Product name can be OEM-customized

Figure H-2: Customized Web Interface Title Bar



H.1 Company & Product Bar Components

The Title bar is composed of 3 components:

- **Replacing the main corporation logo** - refer to Section H.2 on page 278
- **Replacing the title bar's background image file -** refer to Section H.3 on page 280
- **Customizing the product's name -** Refer to Section H.4 on page 281

H.2 Replacing the Main Corporation Logo

The main corporation logo can be replaced either with a different logo image file (refer to Section H.2.1 below) or with a text string (refer to Appendix H.2.2 on page 279).

> **Note:** When the main corporation logo is replaced, the AudioCodes logo in the main menu bar on the left (refer to Figure 6-2, on page 114) and the AudioCodes logo in the Software Upgrade Wizard (refer to Figure 6-31, on page 139) disappear.

H.2.1 Replacing the Main Corporation Logo with an Image File

> **Note:** Use a gif, jpg or jpeg file for the logo image. It is important that the image file has a fixed height of 59 pixels (the width can be configured). The image file size is limited to 64 k bytes.

➢ **To replace AudioCodes' default logo with your own corporation image via the Web interface, take these 8 steps:**

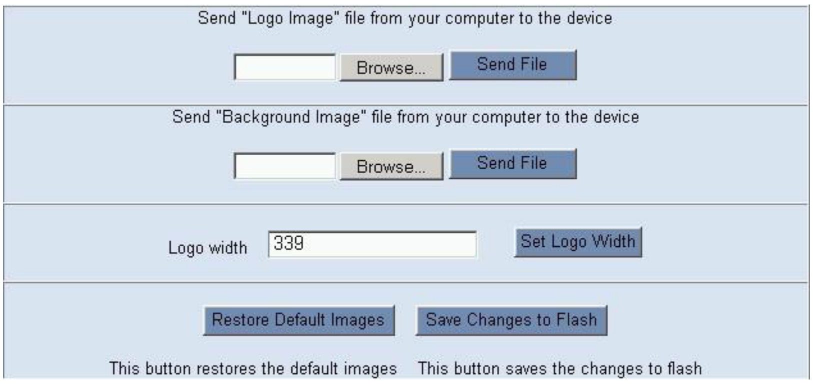
Step 1. Access the AudioCodes Embedded Web Server (refer to Section 6.3.4 on page 112).

Step 2. In the MS Windows Explorer Address field, enter the IP address of the location of the AudioCodes' Web Interface Application, followed by **/AdminPage**.

Step 3. If you have not accessed this page for a while, you are prompted for your user name and Password. Enter them and press **OK**.

Step 4. On the Main-menu bar to the left, click the **Image Download** option. The Image Download screen appears.

Figure H-3: Image Download Screen



Send "Logo Image" file from your computer to the device

Send "Background Image" file from your computer to the device

Logo width 339

This button restores the default images

This button saves the changes to flash

Step 5. Click the **Browse** button in the **Send Logo Image File from your computer to the device** box. Navigate to the folder that contains the logo image file you want to download.

Step 6. Click the **Send File** button. The file is sent to the device. When the download is complete, the screen is automatically refreshed and the new logo image is displayed.

Step 7. Check the appearance of the logo to verify that it appears as desired. If you want to modify the width of the logo (the default width is 339 pixels), in the **Logo Width** field, enter the new width (in pixels) and press the **Set Logo Width** button.

Step 8. Save the image to flash memory by clicking the **Save Changes to Flash** button. The new logo appears on all Web interface screens.

Note: If you encounter any problem during the loading of the files, or you want to restore the default images, click the **Restore Default Images** button.

➢ To replace AudioCodes' default logo with your own corporation image via the *ini* file, take these 2 steps:

Step 1. Place your corporate logo image file in the same folder in which the device's ***ini*** file is located (i.e., the same location defined in the BootP/TFTP server). For detailed information on the BootP/TFTP server, refer to Appendix A, on page 211.

Step 2. Add/modify the two ***ini*** file parameters in Table H-1 according to the procedure described in Section 6.4.6.1 on page 136.

Note: Loading the device's ***ini*** file via the 'Configuration File' screen in the Web interface does not load the corporate logo image files as well.

Table H-1: Customizable Logo *ini* File Parameters for the Image File

Parameter	Description
LogoFileName	The name of the image file containing your corporate logo. Use a gif, jpg or jpeg image file. The default is AudioCodes' logo file. **Note:** The length of the name of the image file is limited to 47 characters.
LogoWidth	Width (in pixels) of the logo image. **Note:** The optimal setting depends on the resolution settings. The default value is 339, which is the width of AudioCodes' displayed logo.

H.2.2 Replacing the Main Corporation Logo with a Text String

The main corporation logo can be replaced with a text string. To replace AudioCodes' default logo with a text string via the Web interface, modify the two ***ini*** file parameters in Table H-2 according to the procedure described in Section H.5 on page 282.

Table H-2: Customizable Logo *ini* File Parameters for the String Text

Parameter	Description
UseWebLogo	0 = Logo image is used (default value). 1 = Text string is used instead of a logo image.
WebLogoText	Text string that replaces the logo image. The string can be up to 15 characters.

H.3 Replacing the Background Image File

The background image file is repeated across the width of the screen. The number of times the image is repeated depends on the width of the background image and screen resolution. When choosing your background image, keep this in mind.

Note: Use a gif, jpg or jpeg file for the logo image. It is important that the image file has a fixed height of 59 pixels (the width can be configured). The image file size (for both logo and background) is limited to 64 k bytes.

➢ To replace the background image via the Web interface, take these 7 steps:

Note: If you intend to replace both the background image and the logo image, load the logo image first and then load the background image.

Step 1. Access the AudioCodes Embedded Web Server (refer to Section 6.3.4 on page 112).

Step 2. In the MS Windows Explorer Address field, enter the IP address of the location of the AudioCodes' Web Interface Application, followed by **/AdminPage**.

Step 3. If you have not accessed this page for a while, you are prompted for your user name and Password. Enter them and press **OK**.

Step 4. On the Main-menu bar to the left, click the **Image Download** option. The Image Download screen appears.(shown in Figure H-3, on page 278).

Step 5. Click the **Browse** button in the **Send Background Image File from your computer to gateway** box. Navigate to the folder that contains the background image file you want to download.

Step 6. Click the **Send File** button. The file is sent to the device. When the download is complete, the screen is automatically refreshed and the new background image is displayed.

Step 7. Save the image to the flash memory by clicking the **Save Changes to Flash** button. The new background appears on all Web interface screens.

Note: If you encounter any problem during the loading of the files, or you want to restore the default images, click the **Restore Default Images** button.

➢ To replace the background image via the *ini* file:

Step 1. Place your background image file in the same folder in which the device's ***ini*** file is located (i.e., the same location defined in the BootP/TFTP server). For detailed information on the BootP/TFTP server, refer to Appendix A, on page 211.

Step 2. Add/modify the ***ini*** file parameters in Table H-3 according to the procedure described in Section 6.4.6 on page 136.

Note that loading the device's ***ini*** file via the Configuration File screen in the Web interface doesn't load the logo image file as well.

Table H-3: Customizable Background *ini* File Parameters

Parameter	Description
BkgImageFileName	The name (and path) of the file containing the new background. Use a gif, jpg or jpeg image file. The default is AudioCodes background file. **Note:** The length of the name of the image file is limited to 47 characters.

H.4 Customizing the Product Name

The Product Name text string can be modified according to OEMs specific requirements.

- To replace AudioCodes' default product name with a text string via the Web interface, modify the two ***ini*** file parameters in Table H-4 according to the procedure described in Section H.5.
- To replace AudioCodes' default product name with a text string via the ***ini*** file, add/modify the two ***ini*** file parameters in Table H-4 according to the procedure described in Section 6.4.6 on page 136.

Table H-4: Customizable Product Name *ini* File Parameters

Parameter	Description
UseProductName	0 = Don't change the product name (default). 1 = Enable product name change.
UserProductName	Text string that replaces the product name. The default is "Mediant 2000". The string can be up to 29 characters.

H.5 Modifying *ini* File Parameters via the Web Interface's AdminPage

➢ To modify *ini* file parameters via the AdminPage, take these 7 steps:

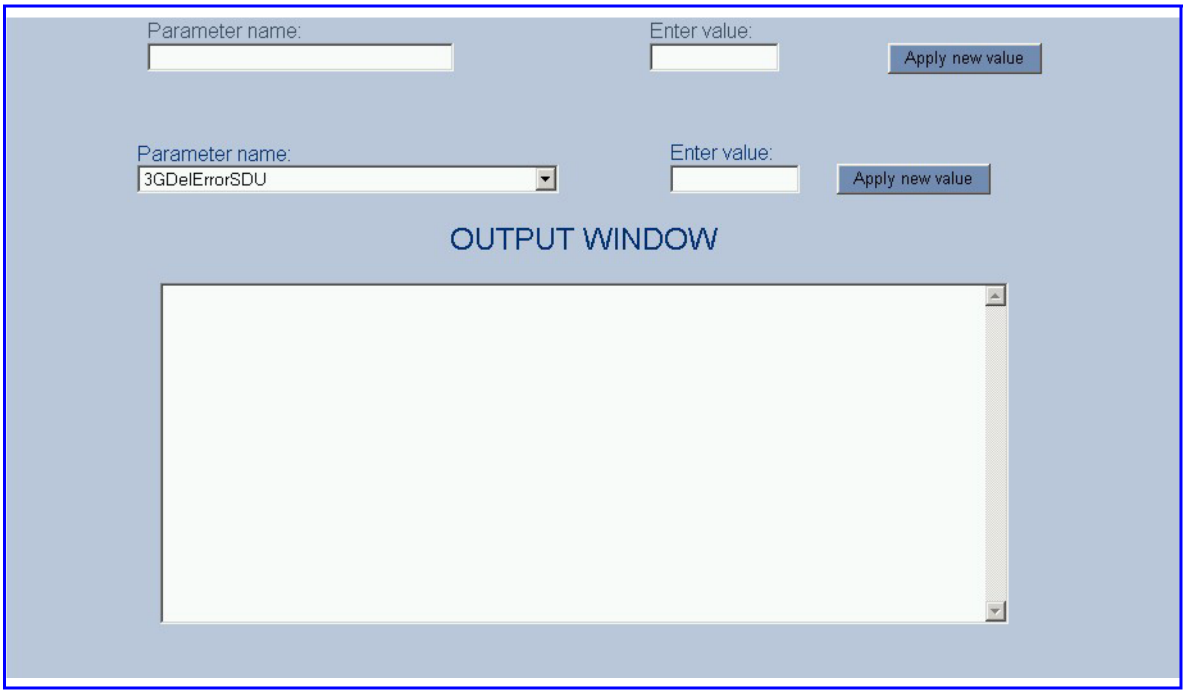
Step 1. Open AudioCodes' Web Interface Application, using the directions in the Device Management section of the accompanying AudioCodes product user's manual.

Step 2. In the MS Windows Explorer Address field, enter the IP address of the location of the AudioCodes' Web Interface Application, followed by **/AdminPage**.

Step 3. If you have not accessed this page for a while, you are prompted for your user name and Password. Enter them and press **OK**.

Step 4. Click the **INI Parameters** option, the ***ini*** Parameters screen is displayed.

Figure H-4: *ini* Parameters Screen



Step 5. In the **Parameter Name** dropdown list, select the required ***ini*** file parameter.

Step 6. In the **Enter Value** text box to the right, enter the parameter's new value.

Step 7. Click the **Apply new value** button to the right. The INI Parameters screen is refreshed, the parameter name with the new value appears in the fields at the top of the screen and the Output Window displays a log displaying information on the operation.

Note: You cannot load the image files (e.g., logo/background image files) to the device by choosing a file name parameter in this screen.

Appendix I List of Abbreviations

Table I-1: List of Abbreviations (continues on page 283 to page 286)

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 (Industry Standard)
COLP	Connected Line Identity Presentation
COLR	Connected Line Identity Restriction
DHCP	Dynamic Host Control Protocol
DID	Direct Inward Dial
DS1	1.544 Mbps USA Digital Transmission System (see E1 and T1)
DS3	44.736 Mbps USA Digital Transmission System, Encapsulates 28 T1 streams, Also called T3
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	A range of protocol standards for IP-based networks
H.323 Entity	Any H.323 Component

Table I-1: List of Abbreviations (continues on page 283 to page 286)

Abbreviation	Meaning
IE	Information Element (ISDN layer 3 protocol, basic building block)
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPmedia	AudioCodes series of VoIP Media Processing Boards
IPM-260	AudioCodes IPmedia PCI VoIP Media Processing Board, to 120 ports
IPM-1610	AudioCodes IPmedia cPCI VoIP Media Processing Board, to 240 ports
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
ITU	International Telecommunications Union
ITU-T	Telecommunications section of the ITU
IVR	Interactive Voice Response
Jitter	Variation of interpacket timing interval
kbps	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	AudioCodes series of Voice over Packet Media Gateways
Mediant for Broadband	AudioCodes series of Broadband Access Gateways, including Cable and V5.2 Access Gateways
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 Analog MediaPack Media Gateway
MP-104	AudioCodes 4-port Analog MediaPack Media Gateway
MP-108	AudioCodes 8-port Analog MediaPack Media Gateway
MP-124	AudioCodes 24-port Analog 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

Table I-1: List of Abbreviations (continues on page 283 to page 286)

Abbreviation	Meaning
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
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.
SB-1610	AudioCodes TrunkPack VoIP/ 1610 cPCI media streaming board, to 480 ports for Wireless systems
ScBus	Signal Computing Bus - part of SCSA
SCSA	Signal Computing System Architecture
SDK	Software Development Kit
SNMP	Simple Network Management Protocol
Stretto	AudioCodes series of Voice over Wireless Media Gateways
TCP	Transmission Control Protocol.
TCP/IP	Transmission Control Protocol/Internet Protocol.
TFTP	Trivial File Transfer Protocol.
TPNCP	AudioCodes TrunkPack Network Control Protocol.
TP-260	AudioCodes TrunkPack VoIP/260 Voice over IP PCI media streaming board, up to 128 ports
TP-2810	AudioCodes TrunkPack VoIP/2810 cPCI media streaming board, to 672 ports
TP-1610	AudioCodes TrunkPack VoIP/1610 cPCI media streaming board, to 480 ports
TPM-1100	AudioCodes TrunkPack Module
TrunkPack	AudioCodes series of voice compression boards
T1	1.544 Mbps USA Digital Transmission System (see E1 and DS1)
T3	45 Mbps USA Digital Transmission System, Encapsulates 28 T1 streams, Also called DS3

Table I-1: List of Abbreviations (continues on page 283 to page 286)

Abbreviation	Meaning
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)
µs or µsec	microsecond; a millionth part of a second

Appendix J Regulatory Information

Declaration of Conformity

Application of Council Directives:	73/23/EEC (including amendments), 89/336/EEC (including amendments), 1999/5/EC Annex-II of the Directive
Standards to which Conformity is Declared:	EN55022: 1998, Class B EN55024: 1998 ETSI EN 300 386 V1.3.1: 2001 EN61000-3-2: 1995 (including amendments A1: 1998, A2: 1998, A14: 2000) EN61000-3-3: 1995 EN60950-1: 2001 TBR-4: 1995 (including amendment A1: 1997) TBR-13: 1996 TBR-12: 1993 (including amendment 1: 1996)
Manufacturer's Name:	AudioCodes Ltd.
Manufacturer's Address:	4 HaHoresh Street, Yehud 56470, Israel.
Type of Equipment:	Digital VoIP System.
Model Numbers:	**IPmedia 2000**, **Mediant 2000**, **Stretto 2000**

I, the undersigned, hereby declare that the equipment specified above conforms to the above Directives and Standards.

	30th October, 2003	Yehud, Israel
Signature	*Date (Day/Month/Year)*	*Location*

I. Zusmanovich, Compliance Engineering Manager

Este producto està en conformidad con la directiva Europea 1999/5, 89/336/EEC, 72/23/EEC
Dette produkt er i overensstemmelse med Europæiske Direktiver 1999/5, 89/336/EEC, 72/23/EEC
Dieses Produkt ist konform mit der europäischen Richtlinie 1999/5, 89/336/EEC, 72/23/EEC
Ce produit est conforme aux exigences de la Directive européenne 1999/5, 89/336/EEC, 72/23/EEC
Questo prodotto è conforme con la normativa europea 1999/5, 89/336/EEC, 72/23/EEC
Este producto està em conformidade com as Diretrizes Européia 1999/5, 89/336/EEC, 72/23/EEC
Tuote on eurooppalaisen säännöstön mukainen 1999/5, 89/336/EEC, 72/23/EEC
Denna product följer europeiska direktiv 1999/5, 89/336/EEC, 72/23/EEC
Το προιόν, είναι συμφωνο με τους κανονιόμυς της Ευρωπαικης Κοιωτητας 1999/5, 89/336/EEC, 72/23/EEC
Tæki þetta er í samræmi við tilskipun Evrópusambandsins 1999/5, 89/336/EEC, 72/23/EEC
Dette produktet er i samhørighet med det Europeiske Direktiv 1999/5, 89/336/EEC, 72/23/EEC

.

Safety Notice

Installation and service of this gateway must only be performed by authorized, qualified service personnel.

The 2000 must be permanently connected to ground via the protective earth terminal.

.

Industry Canada Notice

This equipment meets the applicable Industry Canada Terminal Equipment technical specifications. This is confirmed by the registration numbers. The abbreviation, IC, before the registration number signifies that registration was performed based on a declaration of conformity indicating that Industry Canada technical specifications were met. It does not imply that Industry Canada approved the equipment.

Digital Device Warnings

This equipment complies with Part 68 of the FCC rules and the requirements adopted by ACTA. On the bottom of this equipment is a label that contains a product identifier in the format US:AC1ISNANMED2KDC. If requested this number must be provided to the telephone company.

The Telephone company may make changes in the facilities, equipment, operations or procedures that could affect the operation of the equipment. If this happens the telephone company will provide advance notice in order for you to make necessary modifications to maintain uninterrupted service. Should you experience trouble with this telephone equipment, contact: *AudioCodes Inc, San Jose, CA USA. Tel: 1 408 577 0488*

Do not attempt to repair this equipment!

Facility Interface Code:	**04DU9.BN, 04DU9.DN, 04DU9.1KN, 4DU9.ISN**
Service Order Code:	**6.0N**
USOC Jack Type:	**RJ21X or RJ48C**

It this gateway causes harm to the telephone network, the telephone company will notify you in advance that temporary discontinuance of service may be required. But if advance notice isn’t practical, the telephone company will notify the customer as soon as possible. Also you will be advised of your right to file complaint with the FCC if you believe it is necessary.

Network Information and Intent of Use

The product is for access to ISDN at 2048 kb/s and for access to G.703 Leased lines at 2048 kb/s.

Network Compatibility

The product supports all the Telecom networks in EU that comply with TBR4 and TBR13.

Telecommunication Safety

The safety status of each port is declared and detailed in the table below:

Ports	Safety Status
E1 or T1	TNV-1
Ethernet (100 Base-T)	SELV

TNV-1: Telecommunication network voltage circuits whose normal operating voltages do not exceed the limits for SELV under normal operating conditions and on which over voltages from telecommunication networks are possible.

SELV: Safety extra low voltage circuit.

FCC Notice to Users

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.

- Increase the separation between the equipment and receiver.

- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.

- Consult the dealer or an experienced radio/TV technician for help.

International Headquarters: AudioCodes Ltd, 1 Hayarden Street, Airport City, Lod, Israel 70151.
Tel: +972-3-976 4000 - Fax: +972-3-976 4040
US: AudioCodes Inc, 2890 Zanker Road, Suite # 200, San Jose, CA 95134.
Tel: 1-408-577-0488 - Fax: 1-408-577-0492

AudioCodes Offices Worldwide: Beijing, Boston, Chicago, London, Paris, Tokyo
www.audiocodes.com

Original printed on recycled paper and available on CD or Web site

CERTIFIED QMS
SI
ISO 9001: 2000
THE STANDARDS INSTITUTION OF ISRAEL