Product Reference Manual

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

MGCP & MEGACO

Version 5.6

October 2008

Document # LTRT-77002

AudioCodes

Table of Contents

List of Figures

List of Tables

Notices

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

Notice

This Product Reference Manual describes the common features and functions of the device.

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.

Before consulting this Manual always check the Release Notes for this version regarding feature preconditions and/or specific support. In cases where there are differences between this Manual and the Release Notes, the information in the Release Notes supersedes that in this Manual.

Updates to this document and other documents can be viewed by registered customers at www.audiocodes.com.

Date Published: October 26, 2008

Trademarks

AC logo, Ardito, AudioCoded, AudioCodes, AudioCodes logo, CTI², CTI Squared, InTouch, IPmedia, Mediant, MediaPack, MP-MLQ, NetCoder, Netrake, Nuera, Open Solutions Network, OSN, Stretto, 3GX, TrunkPack, VoicePacketizer, VoIPerfect, What's Inside Matters, Your Gateway To VoIP, are trademarks or registered trademarks of AudioCodes Limited.

All other products or trademarks are property of their respective owners.

WEEE EU Directive

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

Customer Support

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

Abbreviations and Terminology

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

Applicable Products

The following table contains a list of products to which this Product Reference Manual applies. Each product has an individual User's Manual with instructions on how to install and service the product.

Product		User's Manual Reference #
Mediant Media Gateways 3000 Series	Mediant 3000 with TP-6310 blade	LTRT-95206
	Mediant 3000 with TP-8410 blade	LTRT-80202
	TP-6310 blade	LTRT-81406
	TP-8410 blade	LTRT-36202
IPmedia Media Servers 3000 Series	IPmedia 3000 with IPM-6310 blade	LTRT-95109
	IPM-6310 blade	LTRT-81306
	IPM-8410 blade	LTRT-36402
Mediant 2000		LTRT-69808
Mediant 1000		LTRT-66406

Related Documentation

The documentation package contains the following five publications, available on the AudioCodes Web site:

- **Product Reference Manual** (this manual) - intended for (intended) users of MGCP or MEGACO Network Control Protocol, providing an extremely comprehensive description of initialization & configuration; using MGCP or MEGACO Network Control Protocols and their compliance; Management using SNMP, Web GUI, CLI, SS7, CAS, or ini File; and Security.
- **User's Manual** contains the Product overview; hardware description and installation; software package, startup and initialization; Web GUI-based management; Diagnostics and Product Specification.
- **VoPLib API Reference Manual, Document # LTRT-840xx** - intended for users, who wish to control the device via the AudioCodes VoPLib API (over PCI or TPNCP). This manual is a documentation browser in HTML or CHM formats (created from the VoPLib documented source files). It provides detailed descriptions of the VoPLib functions, events, structures, enumerators and error codes. The 'Reference Library' is an essential reference for developers, containing extended documentation, information and examples of the VoPLib.
- **VoPLib Application Developer's Manual, Document # LTRT-844xx** – describes AudioCodes proprietary TrunkPack Control Protocol (TPNCP) based on AudioCodes API. The manual details how TPNCP provides control of the device

enabling Users to easily develop their applications without having to implement complex, standard control protocols.

- **Release Notes, Document # LTRT-617xx** - describes for each new version the various new features and functionality, issues from the previous version that have been solved, and known constraints of this new software version.

1 Introduction

This Product Reference Manual provides you with supplementary information on the total range of AudioCodes Voice-over-IP (VoIP) Media Gateways and Media Servers, supporting MGCP or MEGACO Network Control Protocols (NCP). For ease of reading, the Series of products are referred to collectively as devices, or individually as a device. The information within this Product Reference Manual is complementary to the information provided by the device's User's Manual and includes, for example, detailed descriptions on various supported features, AudioCodes proprietary applications, advanced configuration methods, and so on.

This manual relates to the following AudioCodes VoIP devices:

- Mediant 3000 Series:
  - Media Gateway series:
    - Mediant 3000 gateway hosting a single or dual (High Availability) TP-8410 blade
  - Media Server series:
    - IPmedia 3000 media server hosting a single or dual (High Availability) IPM-6310 blade
- Mediant 2000 Series:
  - Media Gateway series:
    - Mediant 2000 gateway hosting a single TP-1610 cPCI blade
- Mediant 1000 Media Gateway
  - Analog media gateways
  - Digital media gateways

For information on how to fully configure any device, please refer to the device's User's Manual and Release Notes.

Reader’s Notes

2 Device Initialization & Configuration Files

This section describes the Initialization Procedures and Configuration Options for the device. It includes:

- Startup Process (see below)
- Configuration Parameters and Files (refer to Configuration Parameters and Files on page 29)
- BootP/DHCP (refer to Using BootP/DHCP on page 37)

2.1 Startup Process

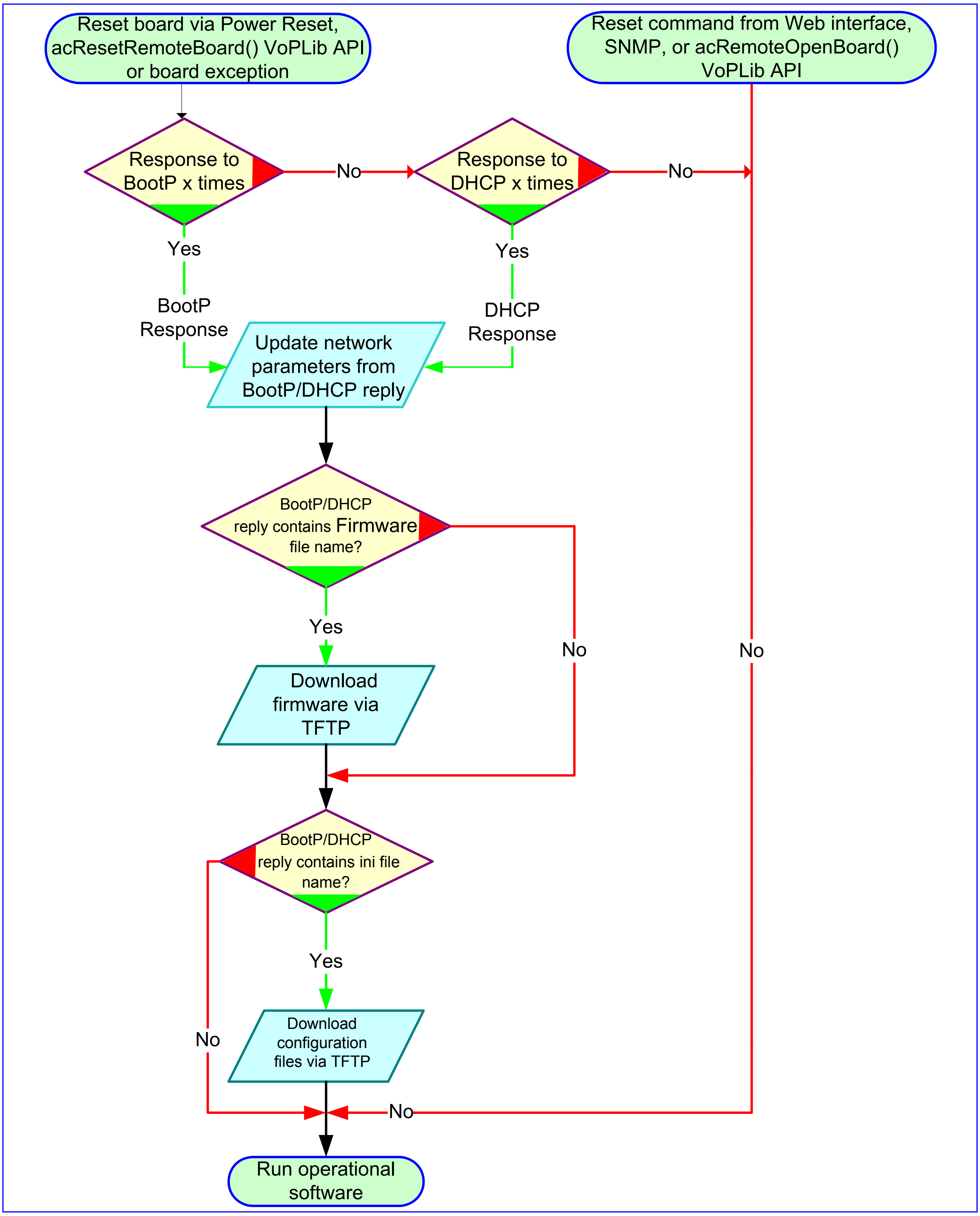
The device's startup process begins when it is reset. The startup process ends when the operational firmware is running. The startup process includes how the device obtains its IP parameters, firmware and configuration files.

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

1. The device is manually reset.
2. acOpenRemoteBoard() is called with RemoteOpenBoardOperationMode set to Full Configuration Mode (valid for VoPLib API users only).
3. acResetRemoteBoard() is called in the VoPLib API (valid for VoPLib API users only).
4. There is a device irregularity.
5. Users perform a reset in the Web Interface or SNMP manager.

The flowchart in the figure below illustrates the process that occurs in these scenarios.

Figure 2-1: Startup Process Diagram


Reset board via Power Reset, acResetRemoteBoard() VoPLib API or board exception
Reset command from Web interface, SNMP, or acRemoteOpenBoard() VoPLib API
Response to BootP x times
No
Response to DHCP x times
No
Yes
Yes
BootP Response
DHCP Response
Update network parameters from BootP/DHCP reply
BootP/DHCP reply contains Firmware file name?
Yes
No
No
Download firmware via TFTP
BootP/DHCP reply contains ini file name?
Yes
No
Download configuration files via TFTP
No
Run operational software

Note 1: The BootP/DHCP server should be defined with an ***ini*** file name 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 memory and must be loaded after every reset.

Note 2: 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 BootPRetries, both parameters can also be set using the Command Line Switches in the BootP reply packet.

Note 3: The ***ini*** file configuration parameters are stored in non-volatile 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). Refer to Using BootP/DHCP below.

2.2 Configuration Parameters and Files

The device's configuration is stored in two file groups.

- The Initialization file - an initialization (*ini*) text file containing configuration parameters of the device.
- The Auxiliary files - *dat* files containing the raw data used for various tasks such as Call Progress Tones, Voice Prompts, logo image, etc.

These files contain factory-pre-configured parameter defaults when supplied with the device and are stored in the device's non-volatile memory. The device is started up initially with this default configuration. Subsequently, these files can be modified and reloaded using either of the following methods:

- BootP/TFTP during the startup process (refer to 'Using BootP/DHCP' on page 37).
- Web Interface (refer to Configuration Using the Web Interface).
- Automatic Update facility (refer to Automatic Update Facility on page 35).

The modified auxiliary files are burned into the non-volatile memory so that the modified configuration is utilized with subsequent resets. The configuration file is always stored on the non-volatile memory. There is no need to repeatedly reload the modified files after reset.

Note 1: Users who configure the device with the Web interface do not require ***ini*** files to be downloaded and have no need to utilize a TFTP server.

Note 2: SNMP users configure the device via SNMP. Therefore a very small ***ini*** file is required which contains the IP address for the SNMP traps.

2.2.1 Initialization (ini) File

The ***ini*** file name must not include hyphens or spaces. Use underscores instead.

The ***ini*** file can contain a number of parameters. The ***ini*** file structure supports the following parameter value constructs:

- **Parameter = Value** (refer to 'Parameter = Value Constructs' on page 523). The lists of parameters are provided in the *ini* File Parameters chapter of the Product

Reference Manual.

- **Table** (refer to Table Structure on page 32). The lists of parameters are provided in Table Parameters on page 218.

The example below shows a sample of the general structure of the ***ini*** file for both the Parameter = Value and Tables of Parameter Value Constructs.

```
[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]
```

2.2.1.1.1 Parameter Value Structure

The following are the rules in the ***ini*** File structure for individual ***ini*** file parameters (Parameter = Value):

- 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 usually case sensitive.
- Numeric parameter values should be entered only in decimal format.
- The ***ini*** file should be ended with one or more empty lines.

***ini* File Examples**

The example below shows a sample ***ini*** file for MGCP.

```
[TDM BUS configuration]
; 1=aLaw 3=ulaw
PCMLawSelect = 1
BaseUDPPort = 4000
[Trunk Configuration]
```

```
;E1 trans 31
ProtocolType = 5
; 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.dat'
```

ini File Examples

The example below shows a sample ***ini*** file for the MediaPack.

```

[MGCP]
EndpointName = 'ACgw'
CallAgentIP = 192.1.10.3
CallAgentPort = 2427
BaseUDPPort = 4000

FlashHookPeriod = 700

[Channel Params]
DJBufferMinDelay = 75
RTPRedundancyDepth = 1

[Files]
CallProgressTonesFilename = 'CPUSA.dat'
VoicePromptsFilename = 'tpdemo_723.dat'
FXSLOOPCHARACTERISTICSFILENAME = 'coeff.dat'
```

The example below shows a sample ***ini*** file 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

; The next 2 fields are the termination names patterns.
; The first is the pattern for the physical termination, and the
; second is the pattern for the RTP termination. The '*' stands for
; a number.
PHYSTERMNAMEPATTERN    = gws*c*
LOGICALRTPTERMPATTERN = gwRTP/*
; This parameter activates MEGACO. If omitted, MGCP will be active
MGCONTROLPROTOCOLTYPE = 2

; The following disables the keep-alive mechanism if set to 0,
; else it is enabled. Note that the recommended KeepAlive method is
; the use of the inactivity timer package - 'it'.
KEEPALIVEENABLED = 1
;
; This parameter defines the profile used, and it is a bitmask
MGCPCOMPATIBILITYPROFILE = 2
```

Note: Before loading an ***ini*** file to the 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*.

The lists of individual ***ini*** file parameters are provided in the Individual ini File Parameters chapter of the Product Reference Manual.

2.2.1.2 Tables of Parameter Value Structure

Tables group the related parameters of a given entity. Tables are composed of rows and columns. The columns represent parameters types, while each row represents an entity. The parameters in each row are called the line attributes. Rows in tables may represent (for example) a trunk, SS7 Link, list of timers for a given application, etc.

Examples of the structure of the tables are provided below. For a list of supported tables please refer to the *ini* File Table Parameters section in the Product Reference Manual.

```
[ SS7 SIG INT ID TABLE ]
FORMAT SS7 SIG IF ID INDEX = SS7 SIG IF ID VALUE,
SS7 SIG IF ID NAME, SS7 SIG IF ID OWNER GROUP, SS7 SIG IF ID LAYER,
SS7 SIG IF ID NAI, SS7 SIG M3UA SPC;
SS7 SIG INT ID TABLE 1 = 101, AMSTERDAM1, 3, 3, 1, 4;
SS7 SIG INT ID TABLE 5 = 100, BELFAST12, 3, 3, 0, 11;

[ \SS7_SIG_INT_ID_TABLE ]
```

The table below is shown in document format for description purposes:

Table 2-1: Table Structure Example

IF ID Index	IF ID Value	SS7_SIG_IF_ID_NAME	SS7_SIG_IF_ID_OWNER_GROUP	F_ID_LAYER	SS7_SIG_IF_ID_NAI	SS7_SIG_M3UA_SPC
1	101	AMSTERDAM1	3	3	1	4
5	100	BELFAST12	3	3	0	11

2.2.1.2.1 Table Structure Rules

Tables are composed of four elements:

- **Table-Title -** The Table's string name in square brackets. In the example above, the Table Title is: [SS7_SIG_INT_ID_TABLE].
- **Format Line -** This line specifies the table's fields by their string names. In the example above, the format line is: FORMAT SS7_SIG_IF_ID_INDEX =

SS7_SIG_IF_ID_VALUE, SS7_SIG_IF_ID_NAME,
SS7_SIG_IF_ID_OWNER_GROUP, SS7_SIG_IF_ID_LAYER,
SS7_SIG_IF_ID_NAI, SS7_SIG_M3UA_SPC

  - The first word MUST be "FORMAT" (in capital letters), 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 comma (',' sign).
  - A Data line must end with a semicolon (';' sign).
  - 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. Each row 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 the table above, Table Structure Example', there is only one index field. This is the simplest way to mark rows.
  - 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.
  - Specifying '$$' in the Data line causes 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. 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.
- **End-of-Table-Mark**: Marks the end of a table. Same as Table title, but string name is preceded by '\'.

Below is an example of the table structure 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]
```

2.2.1.2.2 Tables in the Uploaded *ini* File

Tables are grouped according to the applications they configure.

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.

2.2.1.2.3 Secret Tables

A table is defined as a secret table if it contains at least one secret data field or if it depends on such a table. A secret data field is a field that must not be revealed to the user. An example of a secret field can be found in an IPSec application. The IPsec tables are defined as secret tables because the IKE table contains a pre-shared key field, which must not be revealed. The SPD table depends on the IKE table. Therefore, the SPD table is defined as a secret table.

There are two major differences between tables and secret tables:

- The secret field itself cannot be viewed via SNMP, Web Server or any other tool.
- ***ini*** File behavior: These tables are never uploaded in the ***ini*** File (e.g., 'Get INI-File from Web'). Instead, there is a commented title that states that the secret table is present at the blade, and is not to be revealed.

  Secret tables are always kept in the blade’s non-volatile memory, and may be over-written by new tables that should be provided in a new ***ini*** File. If a secret table appears in an ***ini*** File, it replaces the current table regardless of its content. The way to delete a secret table from a blade is, for example, to provide an empty table of that type (with no data lines) as part of a new ***ini*** File. The empty table replaces the previous table in the blade.

Note: When obtaining the ***ini*** file for the Mediant 1000, the current running hardware entities are added. This data is added to the ***ini*** file header in order to help the user analyze potential faulty situations.

2.2.1.3 Binary Configuration File Download

The ***ini*** file contains sensitive information required for appropriate functioning of the device. The ***ini*** 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 ***ini*** file is encoded, the ***ini*** file would be significantly less vulnerable to outside harm.

2.2.1.3.1 Encoding Mechanism

The ***ini*** file to be loaded and retrieved is available with or without encoding. When an encoded ***ini*** file is downloaded to the device, it is retrieved as encoded from the device. When a decoded file is downloaded to the device, it is retrieved as decoded from the device.

In order to create an encoded ***ini*** file, the user must first create an ***ini*** file and then apply the DConvert utility to it in order to encode it. (Refer to 'Utilities' on page 619 for detailed instruction on ***ini*** file encoding.)

In order to decode an encoded ***ini*** file retrieved from the device, the user must retrieve an encoded ***ini*** file from the device using the Web server (refer to "Downloading Auxiliary Files" below) and then use the DConvert utility in order to decode it. (Refer to the Utilities chapter in the Product Reference Manual for detailed instruction on decoding the ***ini*** file.)

Downloading the ***ini*** file with or without encoding may be performed by utilizing either TFTP or HTTP.

2.2.2 Automatic Update Facility

The device is capable of automatically downloading updates to the ***ini*** file, auxiliary files and firmware image. Any standard Web server, FTP server or NFS server may be used to host these files.

The Automatic Update processing is performed:

- Upon device start-up (after the device is operational)
- At a configurable time of day, e.g., 18:00 (disabled by default)
- At fixed intervals, e.g., every 60 minutes (disabled by default)
- If Secure Startup is enabled (refer to Secure Startup on page 40), upon start-up but before the device is operational.

The Automatic Update process is entirely controlled by configuration parameters in the ***ini*** file. During the Automatic Update process, the device contacts the external server and requests the latest version of a given set of URLs. An additional benefit of using HTTP (Web) servers is that configuration ***ini*** files would be downloaded only if they were modified since the last update.

The following is an example of an ***ini*** file activating the Automatic Update Facility.

```
# DNS is required for specifying domain names in URLs
DnsPriServerIP = 10.1.1.11

# Load extra configuration ini file using HTTP
INIFILEURL = 'http://webserver.corp.com/AudioCodes/inifile.ini'
# Load call progress tones using HTTPS
CPTFILEURL = 'https://10.31.2.17/usa tones.dat'
# Load voice prompts, using user "root" and password "wheel"
VPFILEURL = 'ftps://root:wheel@ftpserver.corp.com/vp.dat'

# Update every day at 03:00 AM
AutoUpdatePredefinedTime = '03:00'
```

Notes on Configuration URLs:

- Additional URLs may be specified, as described in the System *ini* File Parameters in the Product Reference Manual.
- Updates to non-***ini*** files are performed only once. To update a previously-loaded binary file, you must update the ***ini*** file containing the URL for the file.
- To provide differential configuration for each of the devices in a network, add the string "<MAC>" to the URL. This mnemonic is replaced with the hardware (MAC) address of the device.
- To update the firmware image using the Automatic Update facility, use the CMPFILEURL parameter to point to the image file. As a precaution (in order to protect the device from an accidental update), you must also set AUTOUPDATECMPFILE to 1.
- URLs may be as long as 255 characters.

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

➢ To utilize Automatic Updates for deploying the device with minimum manual configuration, take these 7 steps:

1. Set up a Web server (in this example it is **http://www.corp.com/**) where all the configuration files are to be stored.
2. On each device, pre-configure the following setting: (DHCP/DNS are assumed)

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

3. Create a file named *master_configuration.ini*, with the following text:

```
# Common configuration for all devices
# ------------------------------------
CptFileURL = 'http://www.corp.com/call progress.dat'
# Check for updates every 60 minutes
AutoUpdateFrequency = 60

# Additional configuration per device
# -----------------------------------
# Each device will load a file named after its MAC address,
# e.g. config 00908F033512.ini
IniFileTemplateURL = 'http://www.corp.com/config <MAC>.ini'

# Reset the device after configuration has been updated.
# The device will reset after all files were processed.
RESETNOW = 1
```

4. You can modify the *master_configuration.ini* file (or any of the *config_<MAC>.**ini*** files) at any time. The device queries for the latest version every 60 minutes, and applies the new settings immediately.
5. For additional security, usage of HTTPS and FTPS protocols is recommended. The device supports HTTPS (RFC 2818) and FTPS using the AUTH TLS method (RFC 4217) for the Automatic Update facility.
6. To download configuration files from an NFS server, the file system parameters should be defined in the configuration ***ini*** file. The following is an example of a configuration ***ini*** file for downloading files from NFS servers using NFS version 2:

```
# Define NFS servers for Automatic Update
[ NFSServers ]
FORMAT NFSServers Index = NFSServers HostOrIP, NFSServers RootPath,
NFSServers NfsVersion;
NFSServers 1 = 10.31.2.10, /usr/share, 2 ;
NFSServers 2 = 192.168.100.7, /d/shared, 2 ;
[ \NFSServers ]

CptFileUrl = 'file://10.31.2.10/usr/share/public/usa tones.dat'
VpFileUrl =
'file://192.168.100.7/d/shared/audiocodes/voiceprompt.dat'
```

2.3 Boot Firmware & Operational Firmware

The device runs two distinct software programs: Boot firmware and operational firmware.

- Boot firmware - Boot firmware (also known as flash software) resides in the device's non-volatile memory.

When the device is reset, Boot firmware is initialized and the operational software is loaded into the SDRAM from a TFTP server or integral non-volatile memory.

Note: Applicable to Mediant 1000, MediaPack, 6310 and 3000 Devices.

When the device is reset, Boot firmware is initialized and the operational software is loaded into the SDRAM from the PCI host, a TFTP server or integral non-volatile memory.

Boot firmware is also responsible for obtaining the device's IP parameters and ***ini*** file name (used to obtain the device's configuration parameters) via integral BootP or DHCP clients. The Boot firmware version can be viewed on the Embedded Web Server's GUI (refer to 'Embedded Web Server'). The last step the Boot firmware performs is to invoke the operational firmware.

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

2.4 Using BootP/DHCP

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

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

While BootP is always available, DHCP has to be specifically enabled in the device configuration, before it can be used.

A BootP/DHCP request is issued after a power reset or after a device exception.

Note: BootP is normally used to initially configure the device. Thereafter, BootP is no longer required as all parameters can be stored in the gateway's non-volatile memory and used when BootP is inaccessible. For example, BootP can be used again to change the IP address of the device.

2.4.1 BootP/DHCP Server Parameters

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

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

- **Syslog server IP address** - An optional parameter that is sent to the device only if configured in the BootP/DHCP server. This parameter is available only in DHCP.
- **Firmware file name** – An optional parameter that contains the name of the CMP firmware file to be loaded to the gateway via TFTP.
- ***ini* file name** - An optional parameter that contains the name of the ***ini*** file to be loaded to the gateway via TFTP. The ***ini*** file name shall be separated from the CMP file name using a semicolon.

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

- **Configuration (*ini*) file name** - The ***ini*** file is a proprietary configuration file with an ***ini*** extension, containing configuration parameters and tables. For more information on this file, refer to 'Configuration Parameters and Files' on page 29. When the device detects that this optional parameter field is defined in BootP, 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 memory. Whenever the device is reset and no BootP reply is sent to the blade or the ***ini*** file name is missing in the BootP reply, the device uses the previously stored ***ini*** file.

2.4.1.1 Command Line Switches

In the BootP/TFTP Server configuration, you can add command line switches in the Boot File field. Command line switches are used for various tasks, such as to determine if the firmware should be burned on the non-volatile memory or not. The table below describes the different command line switches.

➢ To use a command line switch, take these 4 steps:

1. In the **Boot File** field, leave the file name defined in the field as it is (e.g., *ramxxx.cmp*).
2. Place your cursor after *cmp*.
3. Press the space bar.
4. Type in the switch you require (refer to the table 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)

The table below lists and describes the available switches.

Table 2-2: Command Line Switch Descriptions

Switch	Description
-fb	Burn *ram.cmp* in non-volatile memory. Only the *cmp* file (the compressed firmware file) can be burned to the device's non-volatile memory.

Table 2-2: Command Line Switch Descriptions

Switch	Description
-em#	Use this switch to set Ethernet mode. 0 = 10 Base-T half-duplex 1 = 10 Base-T full-duplex 2 = 100 Base-TX half-duplex 3 = 100 Base-TX full-duplex 4 = auto-negotiate (default) 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-TX) 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 a BootP 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: The device ignores BootP replies where option 43 does not contain the name "AUDC". Refer to Selective BootP on page 40.
-be	Use -be 1 for the device to send client information back to the DHCP server. See the "Vendor Specific Information" section below for more information.

2.4.2 Host Name Support

If DHCP is selected, the device requests 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 to decimal representation). The DHCP server usually registers this Host Name on the DNS server. This feature allows users to configure the device via the Web Browser by providing the following URL: http://ACL_nnnnnnn (instead of using the device's IP address).

2.4.3 Selective BootP

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

- To activate this feature, add the command line switch **-bs 1** to the Firmware File Name field. When activated, the device accepts only BootP replies containing the text AUDC in the Vendor Specific Information field (option 43).
- To de-activate, use **-bs 0**.

2.4.4 Secure Startup

The TFTP protocol is not considered secure; some network operators block it using firewalls. It is possible to disable TFTP completely, using the ini file parameter EnableSecureStartup=1. This way, secure protocols such as HTTPS may be used to retrieve the device configuration.

➢ To work with HTTPS instead of TFTP, take the following steps:

- Prepare the device configuration file on an HTTPS serve, and obtain a URL to it. e.g., https://192.168.100.53/audiocodes.ini
- Enable DHCP if necessary
- Enable SSH and connect to it; refer to Command-line Interface on page 43 for instructions.
- Type the following commands in the CLI, to set IniFileURL to the URL of the configuration file, set EnableSecureStartup, and restart the device with the new configuration:

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

Once Secure Startup has been enabled, it can only be disabled using the reverse sequence, i.e. setting EnableSecureStartup to 0 via CLI. Loading a new ini file via BootP/TFTP will not be possible until EnableSecureStartup has been disabled.

For additional information about the Automatic Update facility and supported URL protocols, refer to Automatic Update Facility on page 35.

2.4.5 Vendor Specific Information

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

To enable / disable this feature, perform 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 reply as follows: **ramxxx.cmp -be 0** to disable, or **-be 1** to enable.

The table below details the Vendor Specific Information field for the device:

Table 2-3: Vendor Specific Information Field Tags

Tag #	Description	Value	Length (bytes)
220	Device Type	Numeric	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

The structure of the Vendor Specific Information field is demonstrated in the table below.

Table 2-4: Example of Vendor Specific Information Field Structure

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

Reader’s Notes

3 Management Functions

Two types of Management are detailed in this section:

- Command-line Interface - refer to Command-line Interface on page 43
- SNMP - refer to Using SNMP-based Management on page 64

3.1 Command-line Interface

The CLI is available via a Telnet or an SSH session to the management interface of the media gateway.

It provides a predefined set of commands with a choice of options that comprehensively cover the maintenance tasks required on the media gateway, including:

- Show status & configuration
- Modify configuration
- Debugging

3.1.1 Starting a CLI Management Session

➢ To start a CLI management session, take these 4 steps:

1. Enable the CLI (Telnet or SSH) using either the ini file, Web interface or SNMP.

ini file example for enabling CLI:

```
;
; This is an example INI file for enabling telnet and SSH
;
TelnetServerEnable = 1
SSHServerEnable = 1
```

To enable CLI using the Web interface, go to "Advanced Configuration" -> "Network Settings" -> "Application Settings", and enable Telnet or SSH using the appropriate configuration fields.

To enable CLI using SNMP, set the objects acSysTelnetSSHServerEnable and acSysTelnetServerEnable to "enable" (1).

Note: For security reasons, all CLI access is disabled by default.

2. A Telnet or SSH client application must run on the management PC. Most operating systems, including Microsoft Windows, include a built-in Telnet client, which can be activated from the command prompt. SSH, however, should usually be installed separately. See the following link for a discussion of available SSH client implementations:
http://en.wikipedia.org/wiki/Comparison_of_SSH_clients
3. Establish a Telnet or SSH session with the gateway's OAMP IP address using the system username and password.

```
Username: Admin
```

```
Password: Admin
```

Note: The username and password are case-sensitive.

Note: The CLI username and password can be altered by the media gateway administrator. Multiple users can be defined.

Note: If using RADIUS authentication when logging in to the CLI, an access level of 200 (Security Administrator) is required. Otherwise, the primary user account defined on the Web Interface (named "Admin" by default) may be used to log in.

4. Note the current directory (root), available commands (SHow, PING), available subdirectories and welcome message displayed in the CLI prompt.

```
login: Admin
password:

AudioCodes device ready. Type "exit" to close the connection.


MGmt/ CONFiguration/ IPNetworking/ TPApp/ BSP/
SHow PING
/>
```

3.1.2 CLI Navigation Concepts

Commands are arranged in subdirectories. When the CLI session is started, you are positioned in the "root" directory.

To access a subdirectory, type its name and press enter. To go back one directory, type ".." (two periods) and press <Enter>.

Alternatively, if you know the full path to a command inside one of the subdirectories, the short format may be used to run it directly.

3.1.3 Commands

The following table summarizes the CLI commands and their options.

Table 3-1: CLI Commands and their Options

Purpose	Commands	Description
Help	h	Shows the help for a specific command, action or parameter
Navigation	cd	Goes to another directory
	cd root	Goes to the root directory (/)
	..	Goes up one level.

Table 3-1: CLI Commands and their Options

Purpose	Commands	Description
	exit	Terminates the CLI session
Status	show	Shows the MG / MS operational status
Configuration	/conf/scp	Sets a value for the specific parameter
	/conf/rfs	Restores factory defaults
	/conf/sar	Restarts the device

3.1.3.1 General Commands

The following table summarizes the General commands and their options.

Table 3-2: General Commands

Command	Short Format	Arguments	Description
SHow	sh	info | mgcp | tdm | dsp | ip | log	Displays operational data. The individual sub-commands are documented below.
SHow INFO	sh info	-	Displays device hardware information, versions, uptime, temperature reading and the last reset reason.
SHow	sh hw	-	Displays system information: power status, High-Availability status, and fan information.
SHow MGCP	sh mgcp	conf | perf | ner | calls | detail | rsip | dur | err | cs	Displays data relating to MGCP. Refer to the following subsection 'MGCP/MEGACO Commands' for details.
SHow MEGACO	sh megaco	conf | perf | ner | calls | detail | dur	Displays data relating to MEGACO. Refer to the following subsection 'MGCP/MEGACO Commands' for details.
SHow TDM	sh tdm	status | perf | summary	Displays the alarm status and performance statistics for E1/T1 trunks.
SHow DSP	sh dsp	status | perf	Displays status and version for each DSP device, along with overall performance statistics.

Table 3-2: General Commands

Command	Short Format	Arguments	Description
SHow IP	sh ip	conf | perf | route	Displays IP interface status and configuration, along with performance statistics. Note: Display format may change according to actual configuration.
SHow LOG	sh log	[stop]	Displays (or stops displaying) Syslog messages inside the CLI session.

Example

```
/>sh ?

Usage:
   SHow INFO           Displays general device information
   SHow MGCP           Displays MGCP data
   SHow TDM            Displays PSTN-related information
   SHow DSP            Displays DSP resource information
   SHow IP             Displays information about IP interfaces
   SHow VOICEPROMPT    Displays information about Voice Prompt
table
   SHow TONES          Displays information about special tones

/>sh info

Board type: TrunkPack firmware version 5.20.000.017
Uptime: 0 days, 0 hours, 3 minutes, 54 seconds
Memory usage: 63%
Temperature reading: 39 C
Last reset reason:
Board was restarted due to issuing of a reset from Web interface
Reset Time : 7.1.2000 21.51.13

/>sh tdm status

Trunk 00:  Active
Trunk 01:  Active
Trunk 02:  Active
Trunk 03:  Active
Trunk 04:  Active
Trunk 05:  Active
Trunk 06:  Active
Trunk 07:  Active
Trunk 08:  Active
Trunk 09:  Active
Trunk 10:  Active
Trunk 11:  Active
Trunk 12:  Active
Trunk 13:  Active
Trunk 14:  Active
Trunk 15:  Not Configured
Trunk 16:  Not Configured
Trunk 17:  Not Configured
Trunk 18:  Not Configured
Trunk 19:  Not Configured
Trunk 20:  Not Configured
Trunk 21:  Not Configured

/>sh tdm perf

DS1 Trunk Statistics (statistics for 948 seconds):
Trunk #    B-Channel  Call count RTP packet RTP packet Activity
           utilizatio             Tx         Rx        Seconds
0          1          1          2865       0          57
1          0          0          0          0          0
2          20         20         149743     0          3017
3          0          0          0          0          0
4          0          0          0          0          0
5          0          0          0          0          0
6          0          0          0          0          0
7          0          0          0          0          0
8          0          0          0          0          0
9          0          0          0          0          0
10         0          0          0          0          0
11         0          0          0          0          0
12         0          0          0          0          0
13         0          0          0          0          0
14         0          0          0          0          0

/>sh dsp status

DSP firmware:491096AE8 Version:0540.03 - Used=0 Free=480 Total=480
DSP device  0:  Active    Used=16   Free= 0   Total=16
DSP device  1:  Active    Used=16   Free= 0   Total=16
DSP device  2:  Active    Used=16   Free= 0   Total=16
DSP device  3:  Active    Used=16   Free= 0   Total=16
DSP device  4:  Active    Used=16   Free= 0   Total=16
DSP device  5:  Active    Used=16   Free= 0   Total=16
DSP device  6:  Inactive
DSP device  7:  Inactive
DSP device  8:  Inactive
DSP device  9:  Inactive
DSP device 10:  Inactive
DSP device 11:  Inactive
DSP device 12:  Active    Used=16   Free= 0   Total=16
DSP device 13:  Active    Used=16   Free= 0   Total=16
DSP device 14:  Active    Used=16   Free= 0   Total=16
DSP device 15:  Active    Used=16   Free= 0   Total=16
DSP device 16:  Active    Used=16   Free= 0   Total=16
DSP device 17:  Active    Used=16   Free= 0   Total=16
DSP device 18:  Inactive
…

IPSEC - DSP firmware: AC491IPSEC Version: 0540.03

CONFERENCE - DSP firmware: AC491256C Version: 0540.03


/>sh dsp perf

DSP Statistics (statistics for 968 seconds):
Active DSP resources: 480
Total DSP resources: 480
DSP usage %: 100

/>sh ip perf


Networking Statistics (statistics for 979 seconds):
IP KBytes TX: 25
IP KBytes RX: 330
IP KBytes TX per second: 0
IP KBytes RX per second: 1
IP Packets TX: 1171
IP Packets RX: 5273
IP Packets TX per second: 3
IP Packets RX per second: 12
Peak KByte/s TX in this interval: 18
Peak KByte/s RX in this interval: 4
Discarded packets: 186
DHCP requests sent: 0
IPSec Security Associations: 0

/>/mg/perf reset

Done.

/>sh ip perf

Networking Statistics (statistics for 2 seconds):
IP KBytes TX: 2
IP KBytes RX: 4
IP KBytes TX per second: 0
IP KBytes RX per second: 1
IP Packets TX: 24
IP Packets RX: 71
IP Packets TX per second: 3
IP Packets RX per second: 12
Peak KByte/s TX in this interval: 18
Peak KByte/s RX in this interval: 4
Discarded packets: 0
DHCP requests sent: 0
IPSec Security Associations: 0

/>sh tones cpt

Call Progress Tone - General information:
-------------------------------------------
Num of Tones: 20, (20 loaded to dsp)
Num of Frequencies: 0
High Energy Threshold=0
Low Energy Threshold=35
Max Frequency Deviation=10
Total Energy Threshold=44
Twist=10
SNR=15

/>show ip conf

Multiple IPs Enabled, VLANs Disabled, 3 interfaces active;
    Physical Network Separation Disabled;

No IP Address       Pfx  Name
-- ---------------  ---  ----------------
0  10.50.166.200    16   OAM
1  10.51.166.200    16   MyOAM1
2  10.31.85.63      16   MyMedia1

* MAC address: 00-90-8f-0b-ce-fe



/>sh ip route

Destination      Mask             Gateway          Intf Flags
---------------- ---------------- ---------------- ---- ---------
0.0.0.0          0.0.0.0          10.4.0.1         OAM  A  S
10.4.0.0         255.255.0.0      10.4.64.13       OAM  A L
127.0.0.0        255.0.0.0        127.0.0.1             AR S
127.0.0.1        255.255.255.255  127.0.0.1             A L    H
```

```
Flag legend: A=Active R=Reject L=Local S=Static E=rEdirect
M=Multicast
             B=Broadcast H=Host I=Invalid
End of routing table, 4 entries displayed.

/>ping 10.31.2.10

Ping process started for address 10.31.2.10. Process ID - 27.

Reply from 10.31.2.10: bytes=0 time<0ms
Reply from 10.31.2.10: bytes=0 time<0ms
Reply from 10.31.2.10: bytes=0 time<0ms
Reply from 10.31.2.10: bytes=0 time<0ms

Ping statistics for 10.31.2.10:
Packets:Sent = 4, Received = 4, Lost 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

/>show voiceprompt numofentries

First Used VoicePrompt Index: 0  First Free VoicePrompt Index: 18

/>show voiceprompt entries 11

VP-00011 Coder: 36 ,Length  7245
VP-00012 Coder: 48 ,Length  12930
VP-00013 Coder: 50 ,Length  5488
VP-00014 Coder: 53 ,Length  7486
VP-00015 Coder: 57 ,Length  15939
VP-00016 Coder: 21 ,Length  9207
VP-00017 Coder: 43 ,Length  28320

/>show voiceprompt entries 9 4

VP-00009 Coder: 32 ,Length  3812
VP-00010 Coder: 34 ,Length  5324
VP-00011 Coder: 36 ,Length  7245
VP-00012 Coder: 48 ,Length  12930
```

3.1.3.2 MGCP/MEGACO Commands

The commands ‘SHow MGCP’ and 'SHow MEGACO' have the following sub-commands:

Table 3-3: Sub-commands of command ‘SHow MGCP’ / 'SHow MEGACO'

Sub-command	Description
conf	Displays the overall configuration of MGCP/MEGACO, including: 1. MGCP/MEGACO version string 2. Configured call-agent data 3. Endpoint-name / Termination-name pattern used by the call agent 4. Various feature flags 5. List of supported packages
perf	Displays performance statistics, including: 1. Current/Total number of voice calls

Table 3-3: Sub-commands of command ‘SHow MGCP’ / 'SHow MEGACO'

Sub-command	Description
	2. Average call length (calculated in 15-minute intervals) 3. Number of messages sent/received from the call agent 4. Number of successful/failed commands, per command type 6. Re-transmission counters Note: ‘sh mgcp perf’ / 'sh megaco perf' is identical to ‘/mg/perf control’.
ner	Calculates the Network Efficiency Rate for CRCX/ADD and MDCX/MODIFY commands, defined as the number of successful commands divided by the total number of commands.
calls	Displays a list of all active calls, with the following information per each call: 1. Endpoint/Termination name (Trunk/B-Channel) 2. Mode (recvonly / sendonly / sendrecv) 3. DSP device used for the call 4. Call duration in seconds For MGCP: 5. Call ID (value of "C:" parameter specified by the call agent) 6. Connection ID (value of "I:" parameter selected by the device) 7. RTP port numbers For MEGACO: 5. Context ID 6. CallID (Internal Channel Handler used for the call) 7. Local Address 8. Remote Address 9. CallType - The call type of the call (FAX, MODEM, VOICE). 10. MediaType - The media type of the call (Audio, Video, AudioVideo) If a trunk number is specified as an argument to ‘sh mgcp calls’ / 'sh megaco calls', only calls made on that trunk are displayed.
detail	Displays detailed information for a specific voice call, selected by endpoint/termination name (specify the full name as used by the call agent, e.g. "ds/Tr2/5") or by call ID (e.g. "C=1a2ff01"). The following information is displayed: 1. Trunk and B-channel used for call 2. Call duration in seconds 3. RTP information – on/off, vocoder used, ports and target IP address 4. DSP information – which device is used, echo cancelation length, etc. 5. RTCP information – number of bytes Tx/Rx, quality (jitter/delay), etc.
dur	Displays call duration averages for the past 48 hours. For every hour, the following items are displayed: 1. Number of calls completed during that hour. 2. Average call duration for completed calls (seconds).

Table 3-3: Sub-commands of command ‘SHow MGCP’ / 'SHow MEGACO'

Sub-command	Description
rsip	MGCP: Displays counters for RSIP messages sent to the Call Agent, including break-down by individual RSIP reasons (Restart, Forced, Graceful, etc.). This subcommand is not available for 'sh megaco'.
err	Displays a breakdown of the various error codes with which the gateway responded to ADD/MODIFY commands sent by the call agent. For each individual error code, the following data is displayed: Error description, as per the MGCP / H.248 standard. Counter for erroneous response to CRCX / ADD commands. Counter for erroneous response to MDCX / MODIFY commands. In addition, the following statistical data is available: 1. A counter per each type of failed response to a CRCX command (e.g., a 501 response counter, a 502 response counter …). 2. A counter per each type of failed response to a MDCX command (e.g., a 501 response counter, a 502 response counter …). 3. A counter per each type of reason code in a DLCX command sent by the gateway (e.g., a 901 reason code counter, a 905 reason code counter …)
cs	Displays Call Attempts Per Second (CAPS) statistics in 15-minute intervals. For every interval, the following data is displayed: Minimum CAPS (how many call attempts were in the least-busy second) Maximum CAPS (how many call attempts were in the most-busy second) Average CAPS Call Trials Statistics are also available. They can display the following information: 1. Number of call attempts made in the last second. (This value is updated every second). 2. Historical data regarding the last 3 time intervals of 15 minutes each. For each time interval the following data is presented: a. maximum number of call attempts per second in the interval. b. minimum number of call attempts per second in the interval. c. the average number of call attempts per second during the time interval.

Example

```
/>sh mgcp ?

Usage:
   SHow MGCP CONF             Displays MGCP configuration
   SHow MGCP PERF             Displays MGCP performance statistics
   SHow MGCP NER              Displays MGCP network efficiency rate
   SHow MGCP CALLS [Trunk#]   Displays currently active calls
   SHow MGCP DETAIL <C=id>|<endpoint>   Displays detailed data for
the specified call
   SHow MGCP DUR              Displays history of call duration
averages
   SHow MGCP RSIP             Displays MGCP RSIP counters
   SHow MGCP ERR             Displays MGCP failed responses per
error code
   SHow MGCP CS              Displays MGCP calls per second
statistics

/>sh mgcp ner

Network Efficiency Rate:
        CRCX success/total = 20/22 =  90%
        MDCX success/total = 0/0 = 100%

/>sh mgcp calls

Endpoint  CallID(C)  ConnID(I)  Time(T)  Port(P)  Mode(M)   DSP
ds/Tr0/1    C=56aa    I=21     T=262  P=4000,0    M=recvonly    0
ds/Tr0/2    C=56ab    I=22     T=261  P=4010,4000 M=recvonly    1
ds/Tr0/3    C=56ac    I=23     T=261  P=4020,0    M=recvonly    2
ds/Tr0/4    C=56ad    I=24     T=260  P=4030,0    M=recvonly    3
ds/Tr0/5    C=56ae    I=25     T=34  P=4040,0    M=recvonly    4
ds/Tr1/15    C=56af    I=26     T=26  P=4450,0    M=recvonly    5
ds/Tr1/16    C=56ba    I=27     T=25  P=4460,0    M=recvonly    0
ds/Tr1/17    C=56bb    I=28     T=24  P=4470,0    M=recvonly    1
ds/Tr5/1    C=56bc    I=29     T=9  P=5550,0    M=recvonly    2
ds/Tr5/2    C=56bd    I=30     T=9  P=5560,0    M=recvonly    3
ds/Tr5/3    C=56be    I=31     T=8  P=5570,0    M=recvonly    4
ds/Tr5/4    C=56bf    I=32     T=8  P=5580,0    M=recvonly    5
ds/Tr5/5    C=56ca    I=33     T=8  P=5590,0    M=recvonly    0
ds/Tr5/6    C=56cb    I=34     T=7  P=5600,0    M=recvonly    1
ds/Tr5/7    C=56cc    I=35     T=7  P=5610,0    M=recvonly    2
ds/Tr5/8    C=56cd    I=36     T=7  P=5620,0    M=recvonly    3
ds/Tr5/9    C=56ce    I=37     T=6  P=5630,0    M=recvonly    4
ds/Tr5/10    C=56cf    I=38     T=5  P=5640,0    M=recvonly    5
ds/Tr5/11    C=56da    I=39     T=5  P=5650,0    M=recvonly    1
ds/Tr5/12    C=56db    I=40     T=4  P=5660,0    M=recvonly    2

/>sh mgcp calls 1

Endpoint  CallID(C)  ConnID(I)  Time(T)  Port(P)  Mode(M)   DSP
ds/Tr1/15    C=56af    I=26     T=235  P=4450,0    M=recvonly    5
ds/Tr1/16    C=56ba    I=27     T=234  P=4460,0    M=recvonly    0
ds/Tr1/17    C=56bb    I=28     T=233  P=4470,0    M=recvonly    1

/>sh mgcp detail ds/Tr2/23

Trunk/BChannel:         2/23
DSP device:             48
RTP:                    Off
Coder:                  PCMU, packetization 20 ms
Echo Canceler:          On, length 128 ms
Silence Compression:    Off
High Pass Filter:       On
DTMF Detection:         On
Voice Volume:           0 dB
Input Gain:             0 dB
Call Duration:          53 seconds
Local RTP port:         4840
Remote RTP address:     10.4.64.13 port 4840
Fax transport type:     Disabled
Call type:              Voice

/>sh mgcp detail c=56ab

Trunk/BChannel:         0/2
DSP device:             1
RTP:                    On sendonly
Coder:                  PCMU, packetization 20 ms
Echo Canceler:          On, length 128 ms
```

```
Silence Compression:    Off
High Pass Filter:       On
DTMF Detection:         On
Voice Volume:           0 dB
Input Gain:             0 dB
Call Duration:          23 seconds
Local RTP port:         4010
Remote RTP address:     10.4.64.13 port 4000
Fax transport type:     Disabled
Call type:              Voice
Tx/Rx bytes:            185440/0
Tx/Rx packets:          1159/0
Jitter:                 0 ms
Packet Loss:            0
SSRC of sender:         493569092
```

3.1.3.3 Call Detail Reports (CDR) Commands

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

Table 3-4: Subcommands of Call Detail Reports (CDR) Command

Subcommand	
start [syslog | file | both]	Starts generating CDR records. If 'syslog' is specified, the records are sent to the Syslog. If 'file' is specified, the records are collected in a file which can be viewed in the CLI or transferred to an NFS host using the '/cp/cdr send' command. If 'both' is specified, the records are sent to both the Syslog and the file.
show	Displays the current CDR file (history of last calls). Note that in a high-load system, the file is overwritten relatively quickly as it can hold approximately 1000 CDRs (possibly less than a minute of activity). Using the '/cp/cdr show' command can yield unpredictable results.
send <nfs_location>	Sends the CDR file to an NFS host. The remote NFS file system must be pre-defined and mounted (for detailed information on NFS support, refer to the User's Manual). The argument to this command must be a URI (Uniform Resource Identifier) in the form: file://server-ip-address/path/filename Note that the URI is case-sensitive.
stop	Stops generation of CDR records and clears the CDR file.

3.1.3.4 Configuration Commands

The commands under the "CONFiguration" directory are used to query and modify the current device configuration. The following commands are available:

Table 3-5: Configuration Commands

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

Example

```
/>conf


SaveAndReset RestoreFactorySettings SetConfigParam
GetParameterDescription GetConfigParam CHangePassWord ConfigFile
AutoUPDate
/CONFiguration>gpd SyslogServerIP

SYSLOGSERVERIP = Defines the Syslog server IP address in dotted
format notation.
 e.g., 192.10.1.255

SaveAndReset RestoreFactorySettings SetConfigParam
GetParameterDescription GetConfigParam CHangePassWord ConfigFile
AutoUPDate
/CONFiguration>gcp syslogserverip

Result: SYSLOGSERVERIP = 10.31.4.51

SaveAndReset RestoreFactorySettings SetConfigParam
GetParameterDescription GetConfigParam CHangePassWord ConfigFile
AutoUPDate
/CONFiguration>scp syslogserverip 10.31.2.10

Old value: SYSLOGSERVERIP = 10.31.4.51
```

```
New value: SYSLOGSERVERIP = 10.31.2.10

SaveAndReset RestoreFactorySettings SetConfigParam
GetParameterDescription GetConfigParam CHangePassWord ConfigFile
AutoUPDate
/CONFiguration>cf set

Enter data below. Type a period (.) on an empty line to finish.
EnableSyslog = 1
SyslogServerIP = 10.31.2.10
.
INI File replaced.

SaveAndReset RestoreFactorySettings SetConfigParam
GetParameterDescription GetConfigParam CHangePassWord ConfigFile
AutoUPDate
/CONFiguration>..

MGmt/ CONFiguration/ IPNetworking/ TPApp/ BSP/
SHow PING
/>
```

3.1.3.5 Management Commands

The commands under the "MGmt" directory are used to display current performance values and fault information. The following commands are available:

Table 3-6: Management commands

Command	Short Format		Description
/MGmt/PERFormance	/mg/perf	basic | control | dsp | net | ds1 | ss7 | reset	Displays performance statistics. '/mg/perf reset' clears all statistics to zero.

Example

```
/>mg


FAult/
PERFormance
/MGmt>fa


ListHistory ListACtive
/MGmt/FAult>lac

  1. Board#1                         1 major      Board Config
Error: PSTN Trunk Validation Check Warning - TDMBusClockSource is
set to Netw
  2. Board#1/EthernetLink#0          9 major      Ethernet link
alarm. Redundant Link (Physical port #2) is down.


ListHistory ListACtive
/MGmt/FAult>lh
```

```
  1. Board#1                         1 major      Board Config
Error: PSTN Trunk Validation Check Warning - TDMBusClockSource is
set to Netw
  2. Board#1/EthernetLink#0          9 major      Ethernet link
alarm. Redundant Link (Physical port #2) is down.

ListHistory ListACtive
/MGmt/FAult>
```

3.1.3.6 PSTN Commands

The commands under the "PSTN" directory allow the user to perform various PSTN actions.

Table 3-7: PSTN Commands

Command	Short Format	Arguments	Description
PstnLoopCommands	PS/PH/PLC	<TrunkId> <LoopCode > <BChannel>	Activates a loopback on a specific trunk and BChannel For loop on all the trunk you need to set BChannel = (-1). LoopCode: 0-NO_LOOPS 1-REMOTE_LOOP (whole trunk only). 2-LINE_PAYLOAD_LOOP (whole trunk only). 3-LOCAL_ALL_CHANNELS_LOOP (whole trunk only). 4-LOCAL_SINGLE_CHANNEL_LOOP 10-PRBS_START (whole trunk only). 11-PRBS_STOP (whole trunk only).

Examples

```
MGmt/ PStn/ DebugRecording/ ControlProtocol/ CONFiguration/
IPNetworking/ TPApp/ BSP/
PING SHow
/>ps



CAS/ PHysical/ PstnCOmmon/

/PStn>ph
```

```
IsdnGetDChannelStatus PstnQueryTrunkStatus PstnSendAlarm
PstnLoopCommands PstnGetPerformanceMonitoring PstnSto
PPerformanceMonitoring PstnStarTPerformanceMonitoring
/PStn/PHysical>pqts 1



TrunkId 1       LOS 0   LOF 0   RAI 1   AIS 0   RAI CRC 0
TrunkStatus 0   LoopBackStatus 0

TrunkIndexAlarmRedundancyDB 3
TrkMtc.Alarm                            2
TrkMtc.AlarmBitMap              0x00000001
NoMultiframeAlignment 0



IsdnGetDChannelStatus PstnQueryTrunkStatus PstnSendAlarm
PstnLoopCommands PstnGetPerformanceMonitoring PstnSto
PPerformanceMonitoring PstnStarTPerformanceMonitoring
/PStn/PHysical>plc 22 1 10
Command sent to board. Use PstnQueryTrunkStatus to check the trunk
status.


IsdnGetDChannelStatus PstnQueryTrunkStatus PstnSendAlarm
PstnLoopCommands PstnGetPerformanceMonitoring PstnSto
PPerformanceMonitoring PstnStarTPerformanceMonitoring
/PStn/PHysical>psa 22 1
Command sent to board. Use PstnQueryTrunkStatus to check the trunk
status.


IsdnGetDChannelStatus PstnQueryTrunkStatus PstnSendAlarm
PstnLoopCommands PstnGetPerformanceMonitoring PstnSto
PPerformanceMonitoring PstnStarTPerformanceMonitoring
/PStn/PHysical>pstpm 1
Command sent to board. Use PstnGetPerformanceMonitoring to check
the trunk status.


IsdnGetDChannelStatus PstnQueryTrunkStatus PstnSendAlarm
PstnLoopCommands PstnGetPerformanceMonitoring PstnSto
PPerformanceMonitoring PstnStarTPerformanceMonitoring
/PStn/PHysical>psppm 1
Command sent to board



IsdnGetDChannelStatus PstnQueryTrunkStatus PstnSendAlarm
PstnLoopCommands PstnGetPerformanceMonitoring PstnSto
PPerformanceMonitoring PstnStarTPerformanceMonitoring
/PStn/PHysical>pgpm 0 0
TrunkId = 0
Interval = 0
AlarmIndicationSignal = 0
LossOfSignal = 0
LossOfFrame = 0
FramingErrorReceived = 0
RemoteAlarmReceived = 0
LostCRC4multiframeSync = 0
CRCErrorReceived = 0
EBitErrorDetected = 0
BitError = 0
LineCodeViolation = 0
ControlledSlip = 0
```

```
ErroredSeconds = 0
ControlledSlipSeconds = 0
SeverelyErroredFramingSeconds = 0
SeverelyErroredSeconds = 0
BurstyErroredSeconds = 0
UnAvailableSeconds = 0
PathCodingViolation = 0
LineErroredSeconds = 0
DegradedMinutes = 0
AssessedSeconds = 331


IsdnGetDChannelStatus PstnQueryTrunkStatus PstnSendAlarm
PstnLoopCommands PstnGetPerformanceMonitoring
PstnStoPPerformanceMonitoring PstnStarTPerformanceMonitoring
/PStn/PHysical> igdcs 0
TrunkId 0 DChannelStatus 0


IsdnGetDChannelStatus PstnQueryTrunkStatus PstnSendAlarm
PstnLoopCommands PstnGetPerformanceMonitoring PstnSto
PPerformanceMonitoring PstnStarTPerformanceMonitoring
/PStn/PHysical>..



CAS/ PHysical/ PstnCOmmon/

/PStn>pco



PstnQueryCallState PstnSetTraceLevel PstnRestartRequest
/PStn/PstnCOmmon>pstl 1 2 1

Command sent to board.


PstnQueryCallState PstnSetTraceLevel PstnRestartRequest
/PStn/PstnCOmmon>prr 1 2

Command sent to board.


PstnQueryCallState PstnSetTraceLevel PstnRestartRequest
/PStn/PstnCOmmon>..



CAS/ PHysical/ PstnCOmmon/

/PStn>cas



GenerateCasFlashHook CasBlockChannel
/PStn/CAS>cbc 1 2 1


Command sent to board.

GenerateCasFlashHook CasBlockChannel
/PStn/CAS>gcfh 1 0 2
```

3.1.4 Debug Recording (DR)

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

3.1.4.1 Collecting DR Messages

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

The default DR port is 925. This can be changed in Wireshark (Edit menu > Preferences > Protocols > ACDR). When loaded, the WireShark plugin dissects all packets on port 925 as DR packets.

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

3.1.4.2 Activating DR

Debug Recording activation is performed using the CLI interface under the DebugRecording directory. This section describes the basic procedures for quickly activating the DR and collecting the call traces. For a more detailed description of all the DR commands, refer to 'DR Command Reference' below.

➢ To activate the DR, take these 7 steps:

1. Start a CLI management session (refer to Starting a CLI Management Session).
2. At the prompt, type **DR** to access the DebugRecording directory.
3. At the prompt, type **STOP** to terminate all active recordings, if any.
4. At the prompt, type **RTR ALL** to remove all previous recording rules.
5. At the prompt, type **RT ALL** to remove all DR targets (i.e., client IP addresses) from the list.
6. At the prompt, type **AIT** <IP address of the target> to define the IP address of the PC (running Wireshark) to which the gateway sends its debug packets.
7. Continue with the procedures described below for capturing PSTN and/or DSP traces.

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

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

➢ To capture DSP traces (internal DSP packets, RTP, RTCP, T38, events and Syslog), take these 4 steps:

1. Setup the DR, as described at the beginning of this section.

2. At the prompt, type ANCT ALL-WITH-PCM 1 Dynamic; the next call on the gateway is recorded.
3. At the prompt, type **START**.
4. Start Wireshark, and then filter according to the UDP port (default is 925) to where debug packets are sent.

Notes: PSTN and DSP recording can be performed simultaneously.
All DR rules are deleted after the gateway is reset.

3.1.4.3 DR Command Reference

The below tables describe all the DR commands. You can also view the description of a DR command in the CLI interface, by simply typing the command name without any arguments.

Table 3-8: Client Setup Commands

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

Table 3-9: Trace Rules

Command	Parameters	Description

AddIPTrafficTrace	TracePoint PDUType SourcePort DestPort [SourceIP] [DestIP] [DebugTarget]	Record IP traffic.
		Trace Point:
		Net2Host = Inbound non-media traffic.
		Host2Net = outbound non-media traffic.
		PDUType:
		UDP = UDP traffic.
		TCP = TCP traffic.
		ICMP = ICMP traffic.
		IPType = Any other IP type (as

Table 3-9: Trace Rules

Command	Parameters	Description
		defined by http://www.iana.com). A = All traffic types. SourcePort: datagram's source port number (ALL for IP wildcard). DestPort: datagram's destination port number (ALL for IP wildcard). SourceIP (optional): datagram's source IP address (ALL for IP wildcard). DestIP (optional): datagram's source IP address (ALL for IP wildcard). DebugTarget (optional): debug target list index; if not specified, the default target is used.
AddIPControlTrace	TracePoint ControlType [DebugTarget]	Records an IP control. Trace Point: Net2Host = Inbound non-media traffic Host2Net = Outbound non-media traffic ControlType: MEGACO - MEGACO traffic MGCP - MGCP traffic TPNCP - TPNCP traffic DebugTarget (optional): debug target list index; if not specified, the default target is used.
AddPstnSignalingTrace	PacketType [DebugTarget]	Records PSTN signaling. Packet Type: CAS = CAS signaling. ISDN = ISDN signaling. SS7 = SS7 signaling. DebugTarget (optional): debug target list index; if not specified, the default target is used. Note: To record PSTN signaling, 'PSTN Trace Level' (TraceLevel ini file) must be set to 1.
AddNextCallTrace	PacketType NumOfCalls [TraceType] [DebugTarget]	Records the next media calls. Packet Type: ALL = all media related (internal DSP packets, RTP, RTCP, T38, events, and Syslog) of a certain call.

Table 3-9: Trace Rules

Command	Parameters	Description
		ALL-WITH-PCM = all media related plus PCM traffic of a certain call. NumOfCalls: amount of next media calls to record. (Note: Currently, only 1 call can be recorded.) Trace Type (optional): New (default) = the next new NumOfCalls calls to record. When these calls end, new calls are not recorded. Dynamic = the next new NumOfCalls calls to record. When these calls end, new calls are recorded until this trace is deleted. DebugTarget (optional): debug target list index; if not specified, the default target is used.
AddTrunkBchannelTrace	PacketType TRUNK [TO_TRUNK] [BCHANNEL] [TO_BCHANNEL][DebugTarget]	Records media calls according to trunk and B-channel. Packet Type: ALL = all media related (internal DSP packets, RTP, RTCP, T38, events and Syslog) of a certain call. ALL-WITH-PCM = all media related plus PCM traffic of a certain call. Trunk: start of range trunk number for recording. (Note: Currently, only 1 channel can be recorded.) To_Trunk (optional): end of range trunk number. BChannel (optional): start of range B-Channel number for recording. To_BChannel (optional): end of range B-Channel number for recording. DebugTarget (optional): debug target list index; if not specified, the default target is used.
AddChannelIdTrace	PacketType Channel-Id [To Channel-Id][DebugTarget]	Records media calls according to CID. Packet Type: ALL = all media related (internal DSP packets, RTP, RTCP, T38, events and Syslog) of a certain call. ALL-WITH-PCM = all media related plus PCM traffic of a certain call. Channel-Id: start of range channel ID number for recording. (Note: Currently, only 1 channel can be

Table 3-9: Trace Rules

Command	Parameters	Description
		recorded.) To Channel-Id (optional) = end of range channel ID number for recording. DebugTarget (optional): debug target list index; if not specified, the default target is used.
RemoveTraceRule	Index	Removes TraceRule from list. Index: rule index (as displayed via ListTraceRules). ALL for rule wildcard.
ListTraceRules	--	Displays added TraceRules.

Table 3-10: DR Activation

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

3.1.5 Changing the Network Parameters via CLI (for MediaPack and Mediant 1000)

The Command Line Interface (CLI) is available on RS-232 for configuring network parameters using serial communication software (e.g., HyperTerminalTM) connected to the device's RS-232 port.

3.1.5.1.1 Accessing the CLI

➢ To access the CLI via the RS-232 port, take these 2 steps:

1. Connect the device RS-232 port to either COM1 or COM2 RS-232 communication port on your PC.
2. Use a serial communication software (e.g., HyperTerminalTM) to connect to the device.
   Set your serial communication software to the following communications port settings:
   - Baud Rate: 115,200 bps (MP-124), 9,600 bps (MP-11x)
   - Data bits: 8
   - Parity: None
   - Stop bits: 1
   - Flow control: None

The CLI prompt is available immediately.

3.1.5.1.2 Assigning an IP Address

➢ **To assign an IP address via the CLI, take these 4 steps:**

1. At the prompt type ‘conf’ and press enter; the configuration folder is accessed.
2. To check the current network parameters, at the prompt, type ‘GCP IP’ and press enter; the current network settings are displayed.
3. Change the network settings by typing: ‘SCP IP [ip_address] [subnet_mask] [default_gateway]’ (e.g., ‘SCP IP 10.13.77.7 255.255.0.0 10.13.0.1’); the new settings take effect on-the-fly. Connectivity is active at the new IP address.

Note: This command requires you to enter all three network parameters (each separated by a space).

4. To save the configuration, at the prompt, type ‘SAR’ and press enter; the device restarts with the new network settings.

3.2 Using SNMP-based Management

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

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

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

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

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

3.2.1 SNMP Standards and Objects

Four types of SNMP messages are defined:

3.2.1.1 SNMP Message Standard

- Get - A request that returns the value of a named object.

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

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

- Get Request - Specific values can be fetched via the "get" request to determine the performance and state of the device. Typically, many different values and parameters can be determined via SNMP without the overhead associated with logging into the device, or establishing a TCP connection with the device.
- Get Next Request - Enables the SNMP standard network managers to "walk" through all SNMP values of a device (via the "get-next" request) to determine all names and values that a device supports.
- Get-Bulk – Extends the functionality of GETNEXT by allowing multiple values to be returned for selected items in the request.
- This is accomplished by beginning with the first SNMP object to be fetched, fetching the next name with a "get-next", and repeating this operation.
- Set Request - The SNMP standard provides a action method for a device (via the "set" request) to accomplish activities such as disabling interfaces, disconnecting users, clearing registers, etc. This provides a way of configuring and controlling network devices via SNMP.
- Trap Message - The SNMP standard furnishes a mechanism for a device to "reach out" to a network manager on their own (via the “trap" message) to notify or alert the manager of a problem with the device. This typically requires each device on the network to be configured to issue SNMP traps to one or more network devices that are awaiting these traps.

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

3.2.1.2 SNMP MIB Objects

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

- 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).

3.2.1.3 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.

3.2.2 Carrier-Grade Alarm System

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

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

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

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

3.2.2.1 Active Alarm Table

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

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

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

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

3.2.2.2 Alarm History

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

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

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

3.2.3 Cold Start Trap

device technology supports a cold start trap to indicate that the unit is starting. This allows the EMS to synchronize its view of the unit'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.

3.2.4 Performance Measurements

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 device system.

The device 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 device performance measurements are provided by several proprietary MIBs (located under the "acPerformance" sub tree:

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

The information supplied by the device is divided into time slices of 15 minutes, when the indexed 0 interval is the current one.

There are two formats of Performance Monitoring MIBs:

1. Older Format

Each MIB is made up of a list of single MIB objects, each relating to a separate attribute within a gauge or counter. All counters and gauges give the current time value only.

- **acPerfMediaGateway** - a generic-type of PM MIB that covers:
    - **Control protocol**
    - **RTP stream**
    - **System packets statistics**
- **acPerfMediaServices** - Media services devices specific performance MIB.

1. New Format - includes new MIBs.

   They all have an identical structure, which includes two major subtrees:

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

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

   The MIBs are:

   - **acPMMedia** - for media (voice) related monitoring such as RTP and DSP.
   - **acPMControl** - for Control Protocol related monitoring such as connections, commands.
   - **acPMAnalog** – Analog channels offhook state. (**Applicable to MediaPack only**)
   - **acPMPSTN** - for PSTN related monitoring such as channel use, trunk utilization. (**Not Applicable to MediaPack**)
   - **acPMSystem** - for general (system related) monitoring.
   - **acPMMediaServer** - for Media Server specific monitoring. (**Applicable to 3000/6310/8410**)

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

3.2.4.1 Total Counters

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

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

- PM-Analog - acPMAnalogConfigurationResetTotalCounters
- PM-Control - acPMControlConfigurationResetTotalCounters.
- PM-Media - acPMMediaConfigurationResetTotalCounters.
- PM-PSTN- acPMPSTNConfigurationResetTotalCounters.
- PM-System - acPMSystemConfigurationResetTotalCounters.

3.2.4.2 Reporting Congestion in Performance Monitoring

The Media Gateway should report the status of several important resources of the device to the MGC. The MGC should perform several actions as a result of the current status.

The resources that are to be managed are:

- General Resources
- DSP Resources
- IP Resources
- ATM Resources
- Extension Resources

A new MIB's sub-tree has been added, which displays the counters' current values: *acPMSystemCongestion*.

A table has been defined for each resource*:*

- *acPMCongestionGeneralResourcesTable*
- *acPMCongestionDSPresourcesTable*
- *acPMCongestionIPresourcesTable*
- *acPMCongestionATMresourcesTable*
- *acPMCongestionConferenceResourcesTable*

3.2.4.3 TrunkPack-VoP Series Supported MIBs

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

- **The Standard MIB (MIB-2)** - The various SNMP values in the standard MIB are defined in RFC 1213. The standard MIB includes various objects to measure and monitor IP activity, TCP activity, UDP activity, IP routes, TCP connections, interfaces, and general system description.
  - The standard icmpStatsTable and icmpMsgStatsTable under MIB-2 support ICMP statistics for both IPv4 and IPv6.
  - inetCidrRouteTable supports both IPv4 and IPv6. ipCidrRouteTable supports IPv4 only.
  - sysDescr - support was added to TP-8410 and IPM-8410.
  - entPhysicalTable – support was added to TP-8410 and IPM-8410.

Tip: An HTML format description for all supported MIBs can be found in the MIBs directory in the release package.

Note: **Applicable to TP-1610, Mediant 2000, TP-6310, Mediant 3000 & SB-1610**

In the ipCidrRouteIfIndex the IF MIB indices are not referenced. Instead, the index used is related to one of the IP interfaces in the device - 1 - OAMP, 2 - Media, 3 - Control. (When there is only one interface then the only index is 1 - OAMP. Refer to Getting Started with VLANs and IP Separation on page 237.

- **RTP MIB** - The RTP MIB is supported according to RFC 2959. It contains objects relevant to the RTP streams generated and terminated by the device and to the RTCP information related to these streams.

> **Note:** The inverse tables are NOT supported.

- **Notification Log MIB** - This standard MIB (RFC 3014 - iso.org.dod.internet.mgmt.mib-2) is supported as part of AudioCodes' implementation of Carrier Grade Alarms.
- **Alarm MIB** - This IETF MIB (RFC 3877) is supported as part of the implementation of Carrier Grade Alarms. This MIB is a new standard and therefore is under the audioCodes.acExperimental branch.
- **SNMP Target MIB** - This MIB (RFC 2273) allows for configuration of trap destinations and trusted managers.
- **SNMP MIB** – This MIB (RFC 3418) allows support of the coldStart and authenticationFailure traps.
- **SNMP Framework MIB** – (RFC 3411).
- **SNMP Usm MIB** – This MIB (RFC 3414) implements the user-based Security Model.
- **SNMP Vacm MIB** – This MIB (RFC 3415) implements the view-based Access Control Model.
- **SNMP Community MIB** – This MIB (RFC 3584) implements community string management.
- **RTCP-XR** – This MIB (RFC) implements the following partial support:
  - The rtcpXrCallQualityTable is fully supported.
  - In the rtcpXrHistoryTable, support of the RCQ objects is provided only with no more than 3 intervals, 15 minutes long each.
  - supports the rtcpXrVoipThresholdViolation trap.

> **Note:** SONET MIB is only applicable to **6310/3000**.

- **SONET MIB** – This MIB (RFC 3592) implements the following partial support:
  - In the SonetMediumTable, the following objects are supported:
    - SonetMediumType
    - SonetMediumLineCoding
    - SonetMediumLineType
    - SonetMediumCircuitIdentifier
    - sonetMediumLoopbackConfig
  - In the SonetSectionCurrentTable, the following objects are supported:
    - sonetSectionCurrentStatus
    - sonetSectionCurrentESs
    - sonetSectionCurrentSESs
    - sonetSectionCurrentSEFSs
    - sonetSectionCurrentCVs

- In the SonetLineCurrentTable, the following objects are supported:
  - sonetLineCurrentStatus
  - sonetLineCurrentESs
  - sonetLineCurrentSESs
  - sonetLineCurrentCVs
  - sonetLineCurrentUASs
- The following tables were added :
  - sonetSectionIntervalTable
  - sonetLineIntervalTable

The following proprietary MIB objects are associated with the SONET/SDH configuration:

- Traps (all defined in the AcBoard MIB):
  - acSonetSectionLOFAlarm
  - acSonetSectionLOSAlarm
  - acSonetLineAISAlarm
  - acSonetLineRDIAlarm
  - acSonetIfHwFailureAlarm

(Refer to the MIB for more details).

- in the acPSTN MIB:
  - acSonetSDHTable - currently has one entry - acSonetSDHFbrGrpMappingType - for selecting a low path mapping type. Relevant only for PSTN applications. (refer to the MIB for more details).
- in the acSystem MIB:
  - acSysTransmissionType - to set the transmission type to optical or DS3 (T3)

Note: ds1 MIB is not applicable to **MediaPack**.

- **ds1 MIB** - support for the following:
  - dsx1ConfigTable - partial supports following objects have SET and GET applied:
    - dsx1LineCoding
    - dsx1LoopbackConfig
    - dsx1LineStatusChangeTrapEnable
    - dsx1CircuitIdentifier

  All other objects in this table support GET only.

  - dsx1CurrentTable
  - dsx1IntervalTable
  - dsx1TotalTable
  - dsx1LineStatusChange trap

Note: ds3 MIB is not applicable to **6310/3000**.

- **ds3 MIB** - (RFC 3896) supports the following:
  - dsx3ConfigTable - refer to the MIB version supplied by AudioCodes for limits on specific objects.

The following objects have been added to the config table:

  - TimerElapsed
  - ValidIntervals

- dsx3LineStatusChange

The following tables (RFC 2496) are supported:

  - dsx3CurrentTable
  - dsx3IntervalTable
  - dsx3TotalTable

There are some proprietary MIB objects that are connected to the SONET/SDH configuration:

- in the acSystem MIB:
    - ♦ acSysTransmissionType - to set the transmission type to optical or DS3 (T3)
-
- **ipForward MIB** (RFC 2096) - fully supported

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

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

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

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

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

> **Note:** The AcBoard MIB is being phased out. It is still supported, but it is being replaced by an updated proprietary MIB.

The AcBoard MIB has the following Groups:

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

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

- **AcAnalog MIB** (Applicable to **MediaPack only**)
- **acControl MIB**
- **acMedia MIB**
  - acIPMediaChannelsresourcesTable - describes IPmedia channel information including Module ID and DSP Channels Reserved. For more information please refer to the IPmedia Channels (Mediant 1000 devices only) section in the VoPLib Application Developer's Manual.

- **acPSTN MIB** (Not Applicable to **MediaPack**)
- **acSystem MIB**
  - acSysInterfaceTable - supports the Networking multiple interfaces feature which allows the configuration of features like Vlan and IP address for each network interface. Refer to Working with VLANs and Multiple IPs on page 237 for more information.
  - acSysModuleTable – support was added for TP-8410 and IPM-8410.
  - acSysEthernetStatusTable -describes Ethernet relevant information including Duplex Mode, Port Speed, Active Port Number for Ethernet. For more information, refer to Working with VLANs and Multiple IPs on page 237.
- **acSS7 MIB**
- **AcAlarm** - This is a proprietary carrier-grade alarm MIB. It is a simpler implementation of the notificationLogMIB and the IETF suggested alarmMIB (both also supported in all AudioCodes devices).

  The acAlarm MIB has the following groups:

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

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

For **MediaPack**, the table size can be any value between 10 and 100 and the default is 100.

For **all other devices**, the table size can be any value between 10 and 1000 and the default is 500.

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

- A detailed explanation of each parameter can be viewed in the MIB Description field.
- Not all groups in the MIB are implemented. Refer to version release notes.
- MIB Objects which are marked as 'obsolete' are not implemented.
- When a parameter is SET to a new value via SNMP, the change may affect device functionality immediately or may require that the blade be soft reset for the change to take effect. This depends on the parameter type.

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

- **Traps**

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

Full proprietary trap definitions and trap Varbinds are found in AcBoard MIB and AcAlarm MIB. For a detailed inventory of traps, refer to 'SNMP Alarm Traps' on page 95.

The following proprietary traps are supported in the device:

- **acBoardFatalError -** Sent whenever a fatal device error occurs.
- **acBoardConfigurationError -** Sent when a device's settings are illegal - the trap contains a message stating/detailing/explaining the illegality of the setting. (**Not applicable to MediaPack**)
- **acBoardTemperatureAlarm -** Sent when a device exceeds its temperature limits. (**Not applicable to MediaPack and 260**)
- **acBoardEvResettingBoard -** Sent after a device is reset.
- **acBoardEvBoardstarted -** Sent after a device is successfully restored and initialized following reset.
- **acFeatureKeyError -** Development pending. Intended to relay Feature Key errors etc. (To be supported in the next applicable release)
- **acgwAdminStateChange -** Sent when Graceful Shutdown commences and ends.
- **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.
- **acAudioProvisioningAlarm** - Raised if the device is unable to provision its audio.
- **acOperationalStateChange** - Raised if the operational state of the node goes to disabled. Cleared when the operational state of the node goes to enabled.
- **acKeepAlive** – part of the NAT traversal mechanism. If the STUN application in the device detects a NAT then this trap is sent out on a regular time laps - 9/10 of the acSysSTUNBindingLifeTime object. The AdditionalInfo1 varbind has the MAC address of the device.
- **acNATTraversalAlarm** - When the NAT is placed in front a device, it is identified as a symmetric NAT - this alarm is raised. It is cleared when a non-symmetric NAT or no NAT replace the symmetric one.
- **acEnhancedBITStatus** - This trap is used to for the status of the BIT (Built In Test). The information in the trap contains device hardware elements being tested and their status. The information is presented in the additional info fields.
- **acPerformanceMonitoringThresholdCrossing** - This log trap is sent out for every time the threshold of a Performance Monitored object is crossed. The severity field is 'indeterminate' when the crossing is above the threshold and 'cleared' when it goes back under the threshold. The 'source' varbind in the trap indicates the object for which the threshold is being crossed.

Note: The following traps are applicable to the **3000 devices**.

- **acFanTrayAlarm** – fault in the fan tray or fan tray missing.
- **acPowerSupplyAlarm** - fault in one of the power supply modules or PS module missing.
- **acPEMAlarm** - fault in the one of the PEM modules or PEM module missing.
- **acSAMissingAlarm** – SA module missing or non operational.
- **acUserInputAlarm** – the alarm is raised when the input dry contact is short circuited and cleared when the circuit is reopened.
- **acHASystemFaultAlarm** – for High Availability (HA) system only - the HA system is faulty and therefore there is no HA.
- **acHASystemConfigMismatchAlarm** – for High Availability (HA) system only - configuration to the modules in the HA system us uneven causing instability.
- **acHASystemSwitchOverAlarm** – for High Availability (HA) system only - a switch over from the active to the redundant module has occurred.

Note: The following traps are applicable to **devices that support SS7**.

- **acSS7LinkStateChangeAlarm** - This alarm is raised if the operational state of the SS7 link becomes BUSY. The alarm is cleared when the operational state of the link becomes -SERVICE or OFFLINE.
- **acSS7LinkInhibitStateChangeAlarm** - This alarm is raised if the SS7 link becomes inhibited (local or remote). The alarm is cleared when the link becomes uninhibited - local AND remote. Note that this alarm is raised for any change in the remote or local inhibition status.
- **acSS7LinkBlockStateChangeAlarm** - This alarm is raised if the SS7 link becomes blocked (local or remote). The alarm is cleared when the link becomes unblocked - local AND remote. Note that this alarm is raised for any change in the remote or local blocking status.
- **acSS7LinkCongestionStateChangeAlarm** - This alarm is raised if the SS7 link becomes congested (local or remote). The alarm is cleared when the link becomes uncongested - local AND remote. Note that this alarm is raised for any change in the remote or local congestion status.
- **acSS7LinkSetStateChangeAlarm** - This alarm is raised if the operational state of the SS7 linkset becomes BUSY. The alarm is cleared when the operational state of the linkset becomes -SERVICE or OFFLINE.
- **acSS7RouteSetStateChangeAlarm** - This alarm is raised if the operational state of the SS7 routeset becomes BUSY. The alarm is cleared when the operational state of the routeset becomes -SERVICE or OFFLINE.
- **acSS7SNSetStateChangeAlarm** - This alarm is raised if the operational state of the SS7 node becomes BUSY. The alarm is cleared when the operational state of the node becomes IN-SERVICE or OFFLINE.

Note: The following trap is applicable to **all devices**.

- **acHTTPDownloadResult** – log trap for the success or failures of the HTTP Download action.

Note: **acDChannelStatus is NOTapplicable to MediaPack.**

- **acDChannelStatus** – Non-alarm trap sent at the establishment, re-establishment or release of LAPD link with its peer connection occurs. The trap is sent out with one of the following in the textual description:
  - D-channel synchronized
  - D-channel not-synchronized

Note: The following SONET and standard traps are only applicable to **6310/3000 devices**.

- **acSonetSectionLOFAlarm** - SONET section Loss of Frame alarm
- **acSonetSectionLOSAlarm** - SONET section Loss of Signal alarm.
- **acSonetLineAISAlarm** - SONET Line AIS alarm.
- **acSonetLineRDIAlarm** - SONET Line RDI alarm.

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

- **authenticationFailure**
- **coldStart**
- **linkDown**
- **linkup**
- **entConfigChange**
- **dsx1LineStatusChange** (Not applicable to **MediaPack**)
- **dsx3LineStatusChange** – supported as in the standard. (Applicable to **6310/3000 devices**)

3.2.5 Toplogy MIB - Objects

3.2.5.1 Physical Entity - RFC 2737

The following groups are supported:

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

3.2.5.2 IF-MIB - RFC 2863

The following interface types are being presented in the ifTable:

- ethernetCsmacd(6) - for all Ethernet-like interfaces, regardless of speed, as per RFC 3635 (Gigabit Ethernet).
- ds1(18) - DS1-MIB
- sonet(39) – SONET-MIB
- ds3(30) – DS3-MIB

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

For each interface type the following objects are supported:

Table 3-11: DS1 Digital Interfaces

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

Table 3-12: Gigabit Ethernet Interface

ifTable & ifXTable	Values
ifIndex	Constructed as defined in the device's Index format.
ifDescr	Ethernet interface.
ifType	ethernetCsmacd(6)
ifMtu	1500
ifSpeed	0 since it’s GBE – please refer to ifHighSpeed.
ifPhysAddress	00-90-8F + acSysIdSerialNumber in hex. Same for both dual ports.
ifAdminStatus	Always UP. [Read Only] - Write access is not required by the standard. Support for 'testing' is not required.
ifOperStatus	Up or Down corresponding to acAnalogFxsFxoType where Unknown is equal to Down.
ifLastChange	The value of sysUpTime at the time the interface entered its current operational state.
ifInOctets	The number of octets in valid MAC frames received on this interface, including the MAC header and FCS. This does include the number of octets in valid MAC Control frames received on this interface. See above section.
ifInUcastPkts	See above section.
ifInDiscards	As defined in IfMIB.
ifInErrors	The sum for this interface of dot3StatsAlignmentErrors, dot3StatsFCSErrors, dot3StatsFrameTooLongs, and dot3StatsInternalMacReceiveErrors.
ifInUnknownProtos	As defined in IfMIB.
ifOutOctets	The number of octets transmitted in valid MAC frames on this interface, including the MAC header and FCS. This does include the number of octets in valid MAC Control frames transmitted on this interface. See above section.
ifOutUcastPkts	See above section.
ifOutDiscards	As defined in IfMIB.
ifOutErrors	The sum for this interface of: dot3StatsSQETestErrors, dot3StatsLateCollisions, dot3StatsExcessiveCollisions, dot3StatsInternalMacTransmitErrors and dot3StatsCarrierSenseErrors.
ifName	GB-Ethernet Port no1 or 2.

Table 3-12: Gigabit Ethernet Interface

ifTable & ifXTable	Values
ifInMulticastPkts	As defined in IfMIB.
ifInBroadcastPkts	As defined in IfMIB.
ifOutMulticastPkts	As defined in IfMIB.
ifOutBroadcastPkts	As defined in IfMIB.
ifHCInOctets ifHCOutOctets	64-bit versions of counters. Required for ethernet-like interfaces that are capable of operating at 20 Mb/s or faster, even if the interface is currently operating at less than 20 Mb/s.
ifHCInUcastPkts ifHCInMulticastPkts ifHCInBroadcastPkts ifHCOutUcastPkts ifHCOutMulticastPkts ifHCOutBroadcastPkts	64-bit versions of packet counters. Required for ethernet-like interfaces that are capable of operating at 640 Mb/s or faster, even if the interface is currently operating at less than 640 Mb/s. Therefore will be constant zero.
ifLinkUpDownTrapEnable	Refer to [RFC 2863]. Default is 'enabled'
ifHighSpeed	1000.
ifPromiscuousMode	Constant False. [R/O]
ifConnectorPresent	Constant True.
ifAlias	An 'alias' name for the interface as specified by a network manager. (NVM)

Table 3-12: Gigabit Ethernet Interface

ifTable & ifXTable	Values
ifCounterDiscontinuityTime	As defined in IfMIB.

SONET/SDH Interfaces	
ifTable & ifXTable	Values
ifDescr	SONET/SDH interface. Module #n Port #n.
ifType	sonet(39).
ifMtu	Constant zero.
ifSpeed	155520000
ifPhysAddress	The value of the Circuit Identifier. If no Circuit Identifier has been assigned this object should have an octet string with zero length.
ifAdminStatus	Read-only access – always up.
ifOperStatus	The value testing(3) is not used. This object assumes the value down(2), if the objects sonetSectionCurrentStatus and sonetLineCurrentStatus have any other value than sonetSectionNoDefect(1) and sonetLineNoDefect(1), respectively.
ifLastChange	The value of sysUpTime at the time the interface entered its current operational state.
ifName	SONET-SDH Port no. n
ifLinkUpDownTrapEnable	Set to enabled(1)
ifHighSpeed	Speed of line in Mega-bits per second: 155
ifConnectorPresent	Set to true(1) normally
ifCounterDiscontinuityTime	Always zero.

DS3 Interfaces

ifTable	Values
ifDescr	DS3 interface, Module no.#d, Port no.#d
ifType	Ds3(30).
ifMtu	Constant zero.

DS3 Interfaces

ifTable	Values
ifSpeed	44736000
ifPhysAddress	The value of the Circuit Identifier. If no Circuit Identifier has been assigned this object should have an octet string with zero length.
ifAdminStatus	Read-only access – always up.
ifOperStatus	The value testing(3) is not used. This object assumes the value down(2), if the objects dsx3LineStatus has any other value than dsx3NoAlarm(1).
ifLastChange	The value of sysUpTime at the time the interface entered its current operational state.
ifXTable	Values
ifName	DS3 Port no.n
ifLinkUpDownTrapEnable	Set to enabled(1)
ifHighSpeed	Speed of line in Mega-bits per second: 45
ifConnectorPresent	Set to true(1).
ifCounterDiscontinuityTime	Always zero.

3.2.6 SNMP Interface Details

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

There are several alternatives for SNMP security:

1. Use SNMPv2c community strings
2. Use SNMPv3 User-based Security Model (USM) users
3. Use SNMP encoded over IPSec. For more details, refer to Security on page 491
4. Use some combinations of the above

For ***ini*** file encoding, refer to Utilities on page 619.

3.2.6.1 SNMP Community Names

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

Up to 5 read-only community strings and up to 5 read-write community strings, and a single trap community string can be configured.

Each community string must be associated with one of the following pre-defined groups.

Table 3-13: SNMP Predefined Groups

Group	Get Access?	Set Access?	Can Send Traps?
ReadGroup	Yes	No	Yes
ReadWriteGroup	Yes	Yes	Yes
TrapGroup	No	No	Yes

3.2.6.1.1 Configuring 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.

3.2.6.1.2 Configuring Community Strings via SNMP

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

Note: For versions 4.4 to 4.6, the srSnmpCommunityTable in the proprietary srCommunityMIB is used. For versions 4.8 and higher, the snmpCommunityTable in the standard snmpCommunityMIB is used.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

3.2.6.2 SNMPv3 USM Users

It is possible to configure up to 10 SNMPv3 USM users. Each user can be configured for one of the following security levels:

Table 3-14: SNMPv3 Security Levels

Security Levels	Authentication	Privacy
noAuthNoPriv(1)	none	none
authNoPriv(2)	MD5 or SHA-1	none

Table 3-14: SNMPv3 Security Levels

Security Levels	Authentication	Privacy
authPriv(3)	MD5 or SHA-1	DES, 3DES, AES128, AES192, or AES256

Each SNMPv3 user must be associated with one of the pre-defined groups listed in the following table.

Table 3-15: SNMPv3 Predefined Groups

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

3.2.6.3 Configuration of SNMPv3 users via the ini File

Use the SNMPUsers INI table to add, modify and delete SNMPv3 users.

The SNMPUsers INI table is a hidden parameter. Therefore, when you do a “Get INI file” operation on the web interface, the table will not be included in the generated file.

The table columns are described below.

Table 3-16: SNMPv3 Table Columns Description

Parameter	Description/ Modification	Default	Note
Row number	This is the table index. Its valid range is 0 to 9.	n/a	
SNMPUsers_Username	Name of the v3 user. Must be unique. The maximum length is 32 characters.	n/a	
SNMPUsers_AuthProtocol	Authentication protocol to be used for this user. Possible values are 0 (none), 1 (MD5), 2 (SHA-1)	0	
SNMPUsers_PrivProtocol	Privacy protocol to be used for this user. Possible values are 0 (none), 1 (DES), 2 (3DES), 3 (AES128), 4 (AES192), 5 (AES256)	0	

Table 3-16: SNMPv3 Table Columns Description

Parameter	Description/ Modification	Default	Note
SNMPUsers_AuthKey	Authentication key.	“”	
SNMPUsers_PrivKey	Privacy key.	“”	
SNMPUsers_Group	The group that this user is associated with. Possible values are 0 (read-only group), 1 (read-write group), and 2 (trap group). The actual group will be ReadGroup<sl>, ReadWriteGroup<sl> or TrapGroup<sl> where <sl> is the SecurityLevel (1=noAuthNoPriv, 2=authNoPriv, 3=authPriv)	0	

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

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

The following sample configuration creates 3 SNMPv3 USM users.

```
[ SNMPUsers ]
FORMAT SNMPUsers Index = SNMPUsers Username,
SNMPUsers AuthProtocol, SNMPUsers PrivProtocol, SNMPUsers AuthKey,
SNMPUsers PrivKey, SNMPUsers Group;
SNMPUsers 0 = v3user, 0, 0, -, -, 0;
SNMPUsers 1 = v3admin1, 1, 0, myauthkey, -, 1;
SNMPUsers 2 = v3admin2, 2, 1, myauthkey, myprivkey, 1;
[ \SNMPUsers ]
```

The user v3user is set up for a security level of noAuthNoPriv(1) and will be associated with ReadGroup1.

The user v3admin1 is setup for a security level of authNoPriv(2), with authentication protocol MD5. The authentication text password is “myauthkey” and the user will be associated with ReadWriteGroup2.

The user v3admin2 is setup for a security level of authPriv(3), with authentication protocol SHA-1 and privacy protocol DES. The authentication text password is “myauthkey”, the privacy text password is “myprivkey”, and the user will be associated with ReadWriteGroup3.

3.2.6.4 Configuration of SNMPv3 users via SNMP

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

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

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

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

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

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

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

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

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

➢ To delete the read-write, authPriv SNMPv3 user, v3admin1, take these 3 steps:

1. If v3admin1 is associated with a trap destination, follow the procedure for associating a different user to that trap destination. (See below.)
2. Delete the vacmSecurityToGroupTable row for SecurityName v3admin1, GroupName ReadWriteGroup1 and SecurityModel usm.
3. Delete the row in the usmUserTable for v3admin1

3.2.6.5 Trusted Managers

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

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

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

- There is no way to configure trusted managers for only a SNMPv3 user. Trusted managers are relevant only for SNMPv2c users. SNMPv2c users are applicable along side with SNMPv3 users ONLY when the community string is not the default string ('public'/'private').

3.2.6.5.1 Configuring 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.

3.2.6.5.2 Configuring Trusted Managers via SNMP

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

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

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

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

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

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

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

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

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

This 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 snmpCommunityTable are currently set to MGR. This procedure must be done from one of the existing trusted managers, but not the one that is being deleted.

- Remove the appropriate row from the snmpTargetAddrTable.

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

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

This 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 snmpCommunityTable are currently set to MGR. This procedure must be done from the final Trusted Manager.

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

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

3.2.6.6 SNMP Ports

The SNMP Request Port is 161 and Trap Port is 162.

These ports can be changed by setting parameters in the device ***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 whenever possible.

3.2.6.7 Multiple SNMP Trap Destinations

An agent can send traps to up to five managers. For each manager the user needs to set the manager IP and trap receiving port along with enabling the sending to that manager.

The user also has the option of associating a trap destination with a specific SNMPv3 USM user. Traps will be sent to that trap destination using the SNMPv3 format and the authentication and privacy protocol configured for that user.

3.2.6.7.1 Configuring Trap Manager via Host Name

A trap manager can be set using the manager's host name. This is currently supported via ***ini*** file only, using the parameter name, SNMPTrapManagerHostName.

When this parameter value is set for this trap, the device at start up tries to resolve the host name. Once the name is resolved (IP is found) the bottom entry in the trap manager's table (and also in the snmpTargetAddrTable in the snmpTargetMIB) is updated with the IP.

The port is 162 unless specified otherwise. The row is marked as 'used' and sending is 'enabled').

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

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

3.2.6.7.2 Configuring via the *ini* File

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

SNMPMANAGERTRAPUSER_<x> = ‘’ indicates to send an SNMPv2 trap using the trap user community string configured with the SNMPTRAPCOMMUNITYSTRING parameter. The user may instead specify a SNMPv3 user name.

Below is an example of entries in the device ***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 device maintains a table of trap destinations containing 5
; rows. The rows are numbered 0..4. Each block of 5 items below
; applies to a row in the table.
;
; To configure one of the rows, un-comment all 5 lines in that
; block. Supply an IP address and if necessary, change the port
; number.
;
; To delete a trap destination, set ISUSED to 0.
;
;
;SNMPMANAGERTABLEIP 0=
;SNMPMANAGERTRAPPORT 0=162
;SNMPMANAGERISUSED 0=1
;SNMPMANAGERTRAPSENDINGENABLE 0=1
;SNMPMANAGERTRAPUSER 0=''
;
;SNMPMANAGERTABLEIP 1=
;SNMPMANAGERTRAPPORT 1=162
;SNMPMANAGERISUSED 1=1
;SNMPMANAGERTRAPSENDINGENABLE 1=1
;SNMPMANAGERTRAPUSER 1=''
;
;SNMPMANAGERTABLEIP 2=
;SNMPMANAGERTRAPPORT 2=162
;SNMPMANAGERISUSED 2=1
;SNMPMANAGERTRAPSENDINGENABLE 2=1
;SNMPMANAGERTRAPUSER 2=''
;
;SNMPMANAGERTABLEIP 3=
;SNMPMANAGERTRAPPORT 3=162
;SNMPMANAGERISUSED 3=1
;SNMPMANAGERTRAPSENDINGENABLE 3=1
;SNMPMANAGERTRAPUSER 3=''
;
;SNMPMANAGERTABLEIP 4=
;SNMPMANAGERTRAPPORT 4=162
;SNMPMANAGERISUSED 4=1
;SNMPMANAGERTRAPSENDINGENABLE 4=1
;SNMPMANAGERTRAPUSER_4=''
```

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

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

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

3.2.6.7.3 Configuring via SNMP

There are two MIB interfaces for the trap managers. The first is via the acBoard MIB that has become obsolete and is to be removed from the code in the next applicable release. The second is via the standard snmpTargetMIB.

1. **Using the acBoard MIB:**

   The following parameters, which are defined in the snmpManagersTable:

   a. snmpTrapManagerSending
   b. snmpManagerIsUsed
   c. snmpManagerTrapPort
   d. snmpManagerIP

Note: Currently, any trap destinations created via SNMP are associated with the trap community string and are sent in the SNMPv2 format.

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 is to be set to 162. This is to be revised 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.

2. **Using the SNMPTargetMIB:**

➢ To add a SNMPv2 trap destination, take this step:

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

All changes to the trap destination configuration take effect immediately.

➢ To add a SNMPv3 trap destination, take these 2 steps:

1. Add a row to the snmpTargetAddrTable with these values: Name=trapN, TagList=AC_TRAP, Params=usm<user>, where N is an unused number between 0 and 4, and <user> is the name of the SNMPv3 that this user is associated with.
2. If a row does not already exist for this combination of user and SecurityLevel, add a row to the snmpTargetParamsTable with this values: Name=usm<user>, MPModel=3(SNMPv3), SecurityModel=3 (usm), SecurityName=<user>, SecurityLevel=M, where M is either 1(noAuthNoPriv), 2(authNoPriv) or 3(authPriv).

All changes to the trap destination configuration take effect immediately.

➢ To delete a trap destination, take these 2 steps:

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

➢ To modify a trap destination, take this step:

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

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

➢ To disable a trap destination, take this step:

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

➢ To enable a trap destination, take this step:

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

3.2.6.7.4 SNMP Manager Backward Compatibility

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

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

3.2.7 Dual Module Interface

Note: Dual Mode interface is only applicable to **1610/2000 devices**.

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

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

MIB object acSysIdFirstSerialNumber always returns the serial number of the first module.

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

3.2.8 SNMP NAT Traversal

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

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

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

  The Trap is instigated in three ways:

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

Note: The two latter options require the STUN client be enabled (***ini*** file parameter – EnableSTUN).

Also, once the acKeepAlive trap is instigated it does not stop.

- The manager can see the NAT type in the MIB: audioCodes(5003).acProducts(9).acBoardMibs(10).acSystem(10).acSystemStatus(2).acSysNetwork(6).acSysNAT(2).acSysNATType(1)
- The manger also has access to the STUN client configuration: audioCodes(5003).acProducts(9).acBoardMibs(10).acSystem(10).acSystemConfiguration(1).acSysNetworkConfig(3).acSysNATTraversal(6).acSysSTUN(21)
- **acNATTraversalAlarm** - When the NAT is placed in front a device is identified as a symmetric NAT - this alarm is raised. It is cleared when a non-symmetric NAT

or no NAT replace the symmetric one.

3.2.9 SNMP for AMS

Note: This section is only applicable to the IPmedia product line.

3.2.9.1 Media Server Configuration

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

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

Configuration parameters in the acBoardMIB specific to services on the media server are:

- **amsNumOfconferencePorts** - Number of conference ports
- **amsNumOfTestTrunkPorts** - Number of test trunk ports
- **amsNumOfLawfulInterceptPorts** - Number of Bearer Channel Tandeming ports
- **amsNumOfAnnouncementPorts** - Number of announcement ports
- **amsApsIpAddress** - The IP address of the audio provisioning server
- **amsApsPort** - The port number to use for the audio provisioning server
- **amsPrimaryLanguage** - The primary language used for audio variables
- **amsSecondaryLanguage** - The secondary language used for audio variables

3.2.9.2 Systems

Note: Systems is only applicable to the **3000** devices.

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

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

The above tables are complemented by the following alarm traps (as defined in the acBoard MIB. For more details, refer to SNMP Alarm Traps on page 95):

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

3.2.10 High Availability Systems

Note: High Availability Systems is only applicable to the **3000** devices.

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

Complementing the above are the following alarm traps (as defined in the acBoard MIB and further detailed in the appendix):

- **acHASystemFaultAlarm** – the High Availability system is faulty and therefore there is no HA.
- **acHASystemConfigMismatchAlarm** – configuration to the modules in the HA system us uneven causing instability.
- **acHASystemSwitchOverAlarm** – a switch over from the active to the redundant module has occurred.

3.2.11 Administrative State Control

3.2.11.1 Node Maintenance

Node maintenance for the device is provided via an SNMP interface. The acBoardMIB provides two parameters for graceful and forced shutdowns of the device. (Refer to the note in "Graceful Shutdown" 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.

3.2.11.2 Graceful Shutdown

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

The possible values received on a get request are:

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

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

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

Before setting acgwAdminState to perform a lock, acgwAdminStateLockControl 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.

For additional information about Cancelling Graceful Shutdown in MEGACO, refer to Cancelling a Graceful Shutdown in MEGACO on page 362.

3.3 SNMP Traps

Note: The sub-section on SNMP Traps is NOT applicable to **260/UNI**.

This section provides information regarding proprietary traps currently supported in the device. Note that traps whose purposes are alarms are different from traps whose purposes are not alarms, e.g., logs.

Currently, all traps have the same structure, which is made up of the same 11 varbinds. An example is: 1.3.6.1.4.1.5003.9.10.1.21.1

The source varbind is made up of a string that details the component from which the trap is being sent, forwarded by the hierarchy in which it resides. For example, an alarm from an SS7 link has the following string in its source varbind:
acBoard#1/SS7#0/SS7Link#6

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

3.3.1 Alarm Traps

The following provides information relating to those alarms that are raised as the result of a generated SNMP trap. The component name described within each of the following section headings refers to the string that is provided in the acBoardTrapGlobalsSource trap varbind. In all the following discussions, to clear a generated alarm the same notification type is sent but with the severity set to 'cleared'.

3.3.1.1 Component: Board#<n> (Devices other than 3000)

The source varbind text for all the alarms under the component below is Board# <n> where n is the slot number. **(Not applicable to 3000 devices.)**

3.3.1.2 Component: System#<n> (3000 only)

The source varbind text for all the alarms under the component below is System#0. **(Applicable to 3000 devices only.)**

<n> is the slot number when the blade resides in a chassis.

1 = the slot number when the blade resides in a chassis. **(Applicable to all devices except 2000/3000.)**

Table 3-17: acBoardFatalError Alarm Trap

Alarm:	acBoardFatalError
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.1
Default Severity	Critical
Event Type:	equipmentAlarm
Probable Cause:	underlyingResourceUnavailable (56)
Alarm Text:	Board Fatal Error: <text>
Status Changes:	
Condition:	Any fatal error
Alarm status:	Critical
<text> value:	A run-time specific string describing the fatal error
Condition:	After fatal error
Alarm status:	Status stays critical until reboot. A clear trap is not sent.
Corrective Action:	Capture the alarm information and the Syslog closes, if active. Contact AudioCodes support who will likely want to collect additional data from the device and then perform a reset.

Table 3-18: acBoardConfigurationError Alarm Trap (Applicable to TP, Mediant and SB only)

Alarm:	acBoardConfigurationError
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.2
Default Severity	critical
Event Type:	equipmentAlarm

Table 3-18: acBoardConfigurationError Alarm Trap (Applicable to TP, Mediant and SB only)

Alarm:	acBoardConfigurationError
Probable Cause:	underlyingResourceUnavailable (56)
Alarm Text:	Board Config Error: <text>
Status Changes:	
Condition:	A configuration error was detected
Alarm status:	critical
<text> value:	A run-time specific string describing the configuration error.
Condition:	After configuration error
Alarm status:	Status stays critical until reboot. A clear trap is not sent.
Corrective Action:	Inspect the run-time specific string to determine the nature of the configuration error. Fix the configuration error using the appropriate tool: web interface, EMS, or $_{\text{ini}}$ file. Save the configuration and if necessary reset the device.

Note: (Applicable to **3000** devices only.)

The alarm trap below does not apply to the High Availability Mode.

Table 3-19: acBoardTemperatureAlarm Alarm Trap (Applicable to 3000 devices only.)

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

Table 3-20: acBoardEvResettingBoard Alarm Trap

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

Table 3-21: acFeatureKeyError Alarm Trap (Not Applicable to MediaPack)

Alarm:	acFeatureKeyError
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.6
Default Severity	critical
Event Type:	processingErrorAlarm
Probable Cause:	configurationOrCustomizationError (7)
Alarm Text:	Feature key error
Status Changes:	
Condition:	This alarm's support is pending
Alarm status:	
Note:	**This alarm’s support is pending**

Table 3-22: acgwAdminStateChange Alarm Trap

Alarm:	acgwAdminStateChange

Table 3-22: acgwAdminStateChange Alarm Trap

Alarm:	acgwAdminStateChange
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.7
Default Severity	Major
Event Type:	processingErrorAlarm
Probable Cause:	outOfService (71)
Alarm Text:	Network element admin state change alarm Gateway is <text>
Status Changes:	
Condition:	Admin state changed to shutting down
Alarm status:	Major
<text> value:	shutting down. No time limit.
Condition:	Admin state changed to locked
Alarm status:	Major
<text> value:	locked
Condition:	Admin state changed to unlocked
Alarm status:	cleared
Corrective Action:	A network administrator has taken an action to lock the device. No corrective action is required.

Table 3-23: acOperationalStateChange Alarm Trap

Alarm:	acOperationalStateChange
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.15
Default Severity	Major
Event Type:	processingErrorAlarm
Probable Cause:	outOfService (71)
Alarm Text:	Network element operational state change alarm. Operational state is disabled.
Note:	**This alarm is raised if the operational state of the node goes to disabled. The alarm is cleared when the operational state of the node goes to enabled.**
Status Changes:	
Condition:	Operational state changed to disabled

Table 3-23: acOperationalStateChange Alarm Trap

Alarm:	acOperationalStateChange
Alarm status:	Major
Condition:	Operational state changed to enabled
Alarm status:	cleared
Note:	In IP systems, the operational state of the node is disabled if the device fails to properly initialize.
Corrective Action:	In IP systems, check for initialization errors. Look for other alarms and Syslogs that might provide additional information about the error.

Table 3-24: acH248LostConnectionWithCA Alarm Trap

Alarm:	acH248LostConnectionWithCA
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.44
Default Severity	Major
Event Type:	communicationsAlarm
Probable Cause:	outOfService (71)
Alarm Text:	H.248 lost connection with call agent: <text>
Status Changes:	
Condition:	Connection to call agent is lost and disconnect behavior is set to disable trunks.
Alarm status:	Major
<text> value:	<call agents IP>. Note: Trunks with signaling will be blocked.
Condition:	Connection to call agent is lost and disconnect behavior is set to reset.
Alarm status:	Major
<text> value:	<call agents IP>. Active calls were detected, Device will reset.
Condition:	Connection re-established
Alarm status:	cleared

Table 3-24: acH248LostConnectionWithCA Alarm Trap

Alarm:	acH248LostConnectionWithCA
Corrective Action:	Ensure Ethernet IF is well connected.

3.3.2 Component: AlarmManager#0

The source varbind text for all the alarms under the component below is Board#<n>/AlarmManager#0 where n is the slot number. **(Not applicable to 3000 devices.)**

The source varbind text for all the alarms under the component below is System#0/AlarmManager#0. **(Applicable to 3000 devices only.)**

Table 3-25: acActiveAlarmTableOverflow Alarm Trap

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

3.3.3 Component: EthernetLink#0

The source varbind text for all the alarms under the component below is Board#<n>/EthernetLink#0 where n is the slot number.

This trap is related to the Ethernet Link Module (the #0 numbering does not apply on the physical Ethernet link). **(Not applicable to 3000 devices.)**

Note: The acBoardTemperatureAlarm alarm trap below does not apply to the High Availability Mode. **(Applicable to 3000 devices only.)**

Table 3-26: acBoardEthernetLinkAlarm Alarm Trap (Not Applicable to 3000 devices)

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

3.3.4 Component: Chassis#0/TimingManager#0

Note: The following alarm traps in this component are only a**pplicable to the Mediant 3000.**

Table 3-27: acIPv6ErrorAlarm

Alarm:	acIPv6ErrorAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.53
Default Severity	Critical
Event Type:	operationalViolation
Probable Cause:	communicationsProtocolError
Alarm Text:	IP interface alarm. <text>
Status Changes:	
Condition:	Bad IPv6 address (already exists)
Alarm status:	Critical
<text> value:	IPv6 Configuration failed, IPv6 will be disabled.
Condition:	After alarm raise
Alarm status:	Status stays critical until reboot. A clear trap is not sent.
Corrective Action:	Find new IPV6 address and reboot.

Table 3-28: acTMInconsistentRemoteAndLocalPLLStatus

Alarm:	acTMInconsistentRemoteAndLocalPLLStatus
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.56
Default Severity	major
Event Type:	equipmentAlarm
Probable Cause:	underlyingResourceUnavailable
Alarm Text:	Timing Manger Alarm. <text>
Status Changes:	
Condition:	alarm is triggered when system is in 1+1 status and redundant board PLL status is deferent than active board PLL status
Alarm status:	major
<text> value:	Timing Manger Alarm.Local and Remote PLLs status is different
Condition:	
Alarm status:	Status stays major until reboot. A clear trap is not sent.
Corrective Action:	Synchronize the timing module.
Probable Cause:	underlyingResourceUnavailable
Alarm Text:	Timing Manger Alarm. <text>
Status Changes:	while primary and secondary clock references are down more than 24 hours alarm will be escelated to critical
Condition:	alarm is triggered when primary reference or secondary reference or both are down.
Alarm status:	major
<text> value:	Timing Manger Alarm.PRIMARY REFERENCE DOWN/SECONDARY REFERENCE DOWN/ALL REFERENCES ARE DOWN/

Condition:	
Alarm status:	Status stays major until reboot. A clear trap is not sent.
Corrective Action:	Synchronize the timing module.

Table 3-29: acTMReferenceChange

Alarm:	acTMReferenceChange
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.58
Default Severity	indeterminate
Event Type:	
Probable Cause:	
Alarm Text:	Timing Manager
Status Changes:	
Condition:	log is send on PLL status change.
Alarm status:	
<text> value:	
Condition:	
Alarm status:	
Corrective Action:	

3.3.5 Component: AudioStaging#0

The source varbind text for all the alarms under this component is Board#<n>/AudioStaging#0 where n is the slot number. **(Applicable to IPM-6310.)**

The source varbind text for all the alarms under the component below is System#0/AudioStaging#0. **(Applicable to IPmedia 3000.)**

Table 3-30: acAudioProvisioningAlarm Alarm Trap

Alarm:	acAudioProvisioningAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.14
Default Severity	critical

Table 3-30: acAudioProvisioningAlarm Alarm Trap

Alarm:	acAudioProvisioningAlarm
Event Type:	processingErrorAlarm
Probable Cause:	configurationOrCustomizationError (7)
Alarm Text:	Unable to provision audio
Status Changes:	
Condition:	Media server times out waiting for a successful audio distribution from the APS (Audio Provisioning Server)
Alarm status:	critical
Condition:	After raise, media server is successfully provisioned with audio from the APS
Alarm status:	cleared
Corrective Action:	From the APS (Audio Provisioning Server) GUI ensure that the device is properly configured with audio and that the device has been enabled. Ensure that the IP address for the APS has been properly specified on the device. Ensure that both the APS server and application are in-service. To get more information regarding the problem, view the Syslog from the device as well as the APS manager logs.

3.3.5.1 Component: SS7#0 (Devices other than MediaPack and 3000)

The source varbind text for all the alarms under the component below is Board#<n>/SS7#0/SS7Link#<m> where n is the slot number and m is the link number. **(Not Applicable to MediaPack and 3000 devices.)**

The source varbind text for all the alarms under the component below is System#0/SS7#0/SS7Link#<m> where m is the link number. **(Applicable to 3000 devices.)**

Table 3-31: acSS7LinkStateChangeAlarm Trap

Alarm:	acSS7LinkStateChangeAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.19
Default Severity	Major
Event Type:	communicationsAlarm
Probable Cause:	other
Alarm Text:	*** SS7 *** Link %i is %s $s
Status Changes:	

Table 3-31: acSS7LinkStateChangeAlarm Trap

Condition:	Operational state of the SS7 link becomes 'BUSY'.
Alarm status:	Major
<text> value:	%i - <Link number> %s - <state name>: { "OFFLINE", "BUSY", "INSERVICE"} %s – IF link has MTP3 layer, then this string equals: (SP %i linkset %i slc %i) Where: %i - <SP number> %i - <Link-Set number> %i - <SLC number> Otherwise there is NO additional text.
Additional Info1 varbind	BUSY
Condition:	Operational state of the link becomes 'IN-SERVICE' or 'OFFLINE'.
Alarm status:	cleared
Corrective Action:	For full details see the SS7 section and SS7 MTP2 and MTP3 relevant standards.

Table 3-32: acSS7LinkInhibitStateChangeAlarm Trap

Alarm:	acSS7LinkInhibitStateChangeAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.20
Default Severity	Major
Event Type:	communicationsAlarm
Probable Cause:	other
Alarm Text:	*** SS7 *** Link %i (SP %i linkset %i slc %i) is %s
Status Changes:	
Condition:	SS7 link becomes inhibited (local or remote).
Alarm status:	Major
<text> value:	%i - <Link number> %i - <SP number> %i - <Link-Set number> %i - <SLC number> %s - <congestion state>: { "UNINHIBITED", "INHIBITED" }

Table 3-32: acSS7LinkInhibitStateChangeAlarm Trap

Additional Info1 varbind	INHIBITED
Condition:	Link becomes uninhibited - local AND remote
Alarm status:	cleared
Corrective Action:	Make sure the link is uninhibited – on both local and remote sides
Note:	**This alarm is raised for any change in the remote or local inhibition status.**

Table 3-33: acSS7LinkBlockStateChangeAlarm

Alarm:	acSS7LinkBlockStateChangeAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.21
No	Major
Event Type:	communicationsAlarm
Probable Cause:	other
Note:	**Support pending**

Table 3-34: acSS7LinkCongestionStateChangeAlarmTrap

Alarm:	acSS7LinkCongestionStateChangeAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.22
Default Severity	Major
Event Type:	communicationsAlarm
Probable Cause:	other
Alarm Text:	*** SS7 *** Link %i is %s %s
Status Changes:	
Condition:	SS7 link becomes congested (local or remote).
Alarm status:	Major
<text> value:	%i - <Link number> %s – IF link has MTP3 layer, then this string equals: (SP %i linkset %i slc %i) Where:

Table 3-34: acSS7LinkCongestionStateChangeAlarmTrap

Alarm:	acSS7LinkCongestionStateChangeAlarm
	%i - <SP number> %i - <Link-Set number> %i - <SLC number> Otherwise there is NO additional text. %s - <congestion state>: { "UNCONGESTED", "CONGESTED" }
Additional Info1 varbind	CONGESTED
Condition:	Link becomes un-congested - local AND remote.
Alarm status:	cleared
Corrective Action:	Reduce SS7 traffic on that link.
Note :	**This alarm is raised for any change in the remote or local congestion status.**

The source varbind text for all the alarms under the component below is Board#<n>/SS7#0/SS7LinkSet#<m> where n is the slot number and m is the link set number. **(Not Applicable to 3000 devices.)**

The source varbind text for all the alarms under the component below is System#0/SS7#0/ SS7LinkSet#<m> where m is the link set number. **(Applicable to 3000 devices.)**

Table 3-35: acSS7LinkSetStateChangeAlarm Trap

Alarm:	acSS7LinkSetStateChangeAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.23
Default Severity	Major
Event Type:	communicationsAlarm
Probable Cause:	other
Alarm Text:	*** SS7 *** Linkset %i on SP %i is %s
Status Changes:	
Condition:	Operational state of the SS7 link-set becomes BUSY.
Alarm status:	Major
<text> value:	%i - <Link-Set number> %i - <SP number> %s - <state name: { "OFFLINE", "BUSY", "INSERVICE"}
Additional Info1 varbind	BUSY

Table 3-35: acSS7LinkSetStateChangeAlarm Trap

Condition:	Operational state of the link-set becomes IN-SERVICE or OFFLINE
Alarm status:	cleared
Corrective Action:	For full details see the SS7 section and SS7 MTP3 relevant standards

The source varbind text for all the alarms under the component below is Board#<n>/SS7#0/SS7RouteSet#<m> where n is the slot number and m is the route set number. **(Not Applicable to 3000 devices.)**

The source varbind text for all the alarms under the component below is System#0/SS7#0/ SS7RouteSet#<m> where m is the route set number. **(Applicable to 3000 devices.)**

Table 3-36: acSS7RouteSetStateChangeAlarm Trap

Alarm:	acSS7RouteSetStateChangeAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.24
Default Severity	Major
Event Type:	communicationsAlarm
Probable Cause:	other
Alarm Text:	*** SS7 *** Routeset %i on SP %i is %s
Status Changes:	
Condition:	Operational state of the SS7 route-set becomes BUSY
Alarm status:	Major
<text> value:	%i - <Route-Set number> %i - <SP number> %s - <state name: { "OFFLINE", "BUSY", "INSERVICE"}
Additional Info:	BUSY
Condition:	Operational state of the route-set becomes IN-SERVICE or OFFLINE
Alarm status:	cleared
Corrective Action:	For full details see the SS7 section and SS7 MTP3 relevant standards

The source varbind text for all the alarms under the component below is Board#<n>/SS7#0/SS7SN#<m> where n is the slot number and m is the SN (signaling node) number. **(Not Applicable to 3000 devices.)**

The source varbind text for all the alarms under the component below is System#0/SS7#0/ SS7SN#<m> where m is the (signaling node) number. **(Applicable to 3000 devices.)**

Table 3-37: acSS7SNSetStateChangeAlarmTrap

Alarm:	acSS7SNSetStateChangeAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.25
Default Severity	Major
Event Type:	communicationsAlarm
Probable Cause:	other
Alarm Text:	*** SS7 *** SP %i is %s
Status Changes:	
Condition:	Operational state of the SS7 node becomes BUSY
Alarm status:	Major
<text> value:	%i - <SP number> %s - <state name: { "OFFLINE", "BUSY", "INSERVICE"}
Additional Info1 varbind	BUSY
Condition:	Cleared when the operational state of the node becomes IN-SERVICE or OFFLINE
Alarm status:	cleared
Corrective Action:	Signaling Node must complete its MTP3 restart procedure and become un-isolated For full details see the SS7 section and SS7 MTP3 relevant standards

The source varbind text for all the alarms under the component below is Board#<n>/SS7#0/SS7Redundancy#0 where n is the slot number. **(Not Applicable to 3000 devices.)**

The source varbind text for all the alarms under the component below is System#0/SS7#0/SS7Redundancy#0. **(Applicable to 3000 devices.)**

Table 3-38: acSS7RedundancyAlarm

Alarm:	acSS7RedundancyAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.26
No	Major
Event Type:	communicationsAlarm
Probable Cause:	other

Table 3-38: acSS7RedundancyAlarm

Note:	**Support pending**

3.3.6 analogports#0 (Applicable to MediaPack and Mediant 1000)

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

Table 3-39: acAnalogPortSPIOutOfService Trap

Alarm:	acAnalogPortSPIOutOfService
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.46
Default Severity	Major
Event Type:	physicalViolation
Probable Cause:	equipmentMalfunction
Alarm Text:	Analog Port SPI out of service.
Status Changes:	
Condition:	Analog port has gone out of service
Alarm status:	Major
Condition:	Analog port is back in service.
Alarm status:	Cleared
Corrective Action:	none

Table 3-40: acAnalogPortHighTemperature Trap

Alarm:	acAnalogPortHighTemperature
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.47
Default Severity	Major
Event Type:	physicalViolation
Probable Cause:	equipmentMalfunction
Alarm Text:	Analog Port High Temperature.
Status Changes:	
Condition:	Analog device has reached critical temperature. Device gets disconnected automatically.
Alarm status:	Major
Condition:	Temperature back to normal - analog port is back in service.

Table 3-40: acAnalogPortHighTemperature Trap

Alarm status:	Cleared
Corrective Action:	none
Note:	Relevant to FXS only.

3.3.6.1 Component: Chassis#0

Note: The following is only applicable to **3000** devices.

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

Table 3-41: acFanTrayAlarm Alarm Trap

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

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

Table 3-42: acPowerSupplyAlarm Alarm Trap

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

The source varbind text for the alarm under this component is Chassis#0/PemCard#<m> where m is the power entry module's slot number.

Table 3-43: acPEMAlarm Alarm Trap

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

Table 3-43: acPEMAlarm Alarm Trap

Alarm status:	Cleared

The source varbind text for the alarm under this component is Chassis#0/SA#<m> where m is the shelf Alarm module's slot number.

Table 3-44: acSAMissingAlarm Alarm Trap

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

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

Table 3-45: acUserInputAlarm Alarm Trap

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

Table 3-45: acUserInputAlarm Alarm Trap

Condition:	Input dry contact circuit is reopened.
Alarm status:	Cleared

Note: The following **four alarm traps** are only applicable to **Mediant 1000**.

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

Table 3-46: acFanTrayAlarm Alarm Trap

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

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

Table 3-47: acPowerSupplyAlarm Alarm Trap

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

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

Table 3-48: acUserInputAlarm Alarm Trap

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

The source varbind text for the alarm under this component is Chassis#0/module#<m> where m is the module's number.

Table 3-49: acHwFailureAlarm Alarm Trap

Alarm:	acHwFailureAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.43
Default Severity	critical
Event Type:	equipmentAlarm
Probable Cause:	equipmentMalfunction
Alarm Text:	Module Alarm: <text>
Status Changes:	
Condition:	The module is faulty or has been removed incorrectly.
Alarm status:	critical
<text> value:	Faulty IF-Module.
	There is no clear on this alarm. The device must be restarted to overcome this issue.
Condition:	Module mismatch
Alarm status:	major
<text> value:	IF-Module Mismatch

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

Note: The following is only applicable to **3000** devices.

Note: The alarm traps discussed in this section applies to the device in **High Availability Mode** ONLY.

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

Table 3-50: acHASystemFaultAlarm Alarm Trap

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

Table 3-50: acHASystemFaultAlarm Alarm Trap

Alarm:	acHASystemFaultAlarm
	hasn't connected yet.
Alarm status:	Minor
<text> value:	Waiting for redundant to connect
Condition:	HA system is active.
Alarm status:	Cleared

Table 3-51: acHASystemConfigMismatchAlarm Alarm Trap

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

Table 3-52: acHASystemSwitchOverAlarm Alarm Trap

Alarm:	acHASystemSwitchOverAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.35
Default Severity	Critical
Event Type:	qualityOfServiceAlarm

Table 3-52: acHASystemSwitchOverAlarm Alarm Trap

Alarm:	acHASystemSwitchOverAlarm
Probable Cause:	outOfService
Alarm Text:	Switch-over:
Status Changes:	
Condition:	Switch over has taken place.
Alarm status:	Critical
<text> value:	see the acHASystemFaultAlarm table above.
Condition:	10 seconds have passed since the switch over.
Alarm status:	cleared

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

Table 3-53: acBoardEthernetLinkAlarm Alarm Trap

Alarm:	acBoardEthernetLinkAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.10
Default Severity	Critical
Event Type:	equipmentAlarm
Probable Cause:	underlyingResourceUnavailable (56)
Alarm Text:	Ethernet link alarm: <text>
Status Changes:	
Condition:	Fault on single interface of the Active module.
Alarm status:	Major
<text> value:	Redundant link (physical link n) is down
Condition:	Fault on both interfaces
Alarm status:	critical
<text> value:	No Ethernet link
Condition:	Fault on single interface of the Redundant module.

Table 3-53: acBoardEthernetLinkAlarm Alarm Trap

Alarm:	acBoardEthernetLinkAlarm
Alarm status:	Major
<text> value:	Redundant link in the redundant module (physical link n) is down
Condition:	Both interfaces are operational
Alarm status:	cleared
Corrective Action:	Ensure that both Ethernet cables are plugged into the back of the system. Inspect the system's Ethernet link lights to determine which interface is failing. Reconnect the cable or fix the network problem
Note:	The alarm behaves differently when coming from the redundant or the active modules of an HA system. The alarm form the redundant will be raised when there is an operational HA configuration in the system. There is no critical severity for the redundant module losing both its Ethernet Links as that is conveyed in the noHA alarm that follows such a case.

Table 3-54: acIPv6ErrorAlarm Alarm Trap

Alarm:	acBoardFatalError
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.53
Default Severity	Critical
Event Type:	operationalViolation
Probable Cause:	communicationsProtocolError
Alarm Text:	IP interface alarm. <text>
Status Changes:	
Condition:	Bad IPv6 address (already exists)
Alarm status:	Critical
<text> value:	IPv6 Configuration failed, IPv6 will be disabled.
Condition:	After alarm raise
Alarm status:	Status stays critical until reboot. A clear trap is not sent.
Corrective Action:	Find new IPV6 address and reboot.

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

Note: The following traps are only applicable to **6310/3000** devices.

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

Table 3-55: AcSonetSectionLOFAlarm Alarm Trap

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

Table 3-56: AcSonetSectionLOSAlarm Alarm Trap

Alarm:	acSonetSectionLOSAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.39
Default Severity	critical
Event Type:	communicationsAlarm
Probable Cause:	lossOfSignal

Table 3-56: AcSonetSectionLOSAlarm Alarm Trap

Alarm Text:	SONET-Section LOS.
Status Changes:	
Condition:	LOS condition is present on SONET no #n
Alarm status:	Critical
<text> value:	LOS
Note:	The sonetSectionCurrentStatus field in the sonetSectionCurrentTable will have a value sonetSectionLOS (2).
Condition:	AIS condition is present (LOS condition is not present)
Alarm status:	Critical
<text> value:	
Note:	
Condition:	LOS condition is not present.
Alarm status:	cleared

Table 3-57: AcSonetLineAISAlarm Alarm Trap

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

Table 3-57: AcSonetLineAISAlarm Alarm Trap

Table 3-58: AcSonetLineRDIAlarm Alarm Trap

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

3.3.6.4 Component: Interfaces#0/trunk#<m> (Not MediaPack)

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

Table 3-59: acTrunksAlarmNearEndLOS Alarm Trap

Alarm:	acTrunksAlarmNearEndLOS
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.49
Default Severity	critical
Event Type:	communicationsAlarm
Probable Cause:	lossOfSignal
Alarm Text:	Trunk LOS Alarm.

Table 3-59: acTrunksAlarmNearEndLOS Alarm Trap

Status Changes:	
Condition:	Near end LOS
Alarm status:	Critical
Condition:	End of LOS
Alarm status:	cleared
Corrective action:	Ensure trunk is properly connected.

Table 3-60: acTrunksAlarmNearEndLOF Alarm Trap

Alarm:	acTrunksAlarmNearEndLOF
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.50
Default Severity	critical
Event Type:	communicationsAlarm
Probable Cause:	lossOfFrame
Alarm Text:	Trunk LOF Alarm.
Status Changes:	
Condition:	Near end LOF
Alarm status:	Critical
Condition:	End of LOF
Alarm status:	cleared
Corrective action:	Ensure trunk is connected to a proper follow up device. Ensure correct clocking set up.

Table 3-61: acTrunksAlarmRcvAIS Alarm Trap

Alarm:	acTrunksAlarmRcvAIS
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.51
Default Severity	critical
Event Type:	communicationsAlarm
Probable Cause:	receiveFailure
Alarm Text:	Trunk AIS Alarm.
Status Changes:	
Condition:	Receive AIS.
Alarm status:	Critical
Condition:	End of AIS

Table 3-61: acTrunksAlarmRcvAIS Alarm Trap

Alarm status:	cleared
Corrective action:	None.

Table 3-62: acTrunksAlarmFarEndLOF Alarm Trap

Alarm:	acTrunksAlarmFarEndLOF
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.52
Default Severity	critical
Event Type:	communicationsAlarm
Probable Cause:	transmitFailure
Alarm Text:	Trunk RAI Alarm.
Status Changes:	
Condition:	RAI
Alarm status:	Critical
Condition:	End of RAI
Alarm status:	cleared
Corrective action:	Ensure correct transmit.

3.3.7 Log Traps (Notifications)

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

Table 3-63: acKeepAlive Log Trap

Trap	acKeepAlive
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.16
Default Severity	Indeterminate
Event Type:	other (0)
Probable Cause:	other (0)
Trap Text:	Keep alive trap
Status Changes:	
Condition:	The STUN client in the device is enabled and has either identified a NAT or is not finding the STUN server The ***ini*** file contains the following line: 'SendKeepAliveTrap=1'

Table 3-63: acKeepAlive Log Trap

Trap	acKeepAlive
Trap status:	Trap is sent
Note:	**Keep-alive is sent out every x second.x =0. 9 of the time defined in the NatBindingDefaultTimeout parameter**

Table 3-64: acPerformanceMonitoringThresholdCrossing Log Trap

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

Table 3-65: acHTTPDownloadResult Log Trap

Trap:	acHTTPDownloadResult
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.28
Default Severity	Indeterminate
Event Type:	processingErrorAlarm (3) for failures and other (0) for success.
Probable Cause:	other (0)
Status Changes:	
Condition:	Successful HTTP download.
Trap text:	HTTP Download successful
Condition:	Failed download.

Table 3-65: acHTTPDownloadResult Log Trap

Trap:	acHTTPDownloadResult
Trap text:	HTTP download failed, a network error occurred.
NOTE:	There are other possible textual messages describing NFS failures or success, FTP failure or success.

3.3.8 Other Traps

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

Table 3-66: coldStart Trap

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

Table 3-67: authenticationFailure Trap

Trap Name:	authenticationFailure
OID:	1.3.6.1.6.3.1.1.5.5
MIB	SNMPv2-MIB

Table 3-68: acBoardEvBoardStarted Trap

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

Note: The following trap is not applicable to **MediaPack**.

Table 3-69: AcDChannelStatus Trap

Trap Name:	acDChannelStatus
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.37
MIB	AcBoard
Severity	minor
Event Type:	communicationsAlarm
Probable Cause:	communicationsProtocolError
Alarm Text:	D-Channel Trap.
Source:	Trunk no.<m> where m is the trunk number (from 0 up).
Status Changes:	
Condition:	D-Channel un-established.
Trap status:	Trap is sent with the severity of Minor.
Condition:	D-Channel established.
Trap status:	Trap is sent with the severity of Cleared.

3.3.9 Trap Varbinds

Every AudioCodes Enterprise trap described above provides the following fields (known as ‘varbinds’). Refer to the AcBoard MIB for additional details on these varbinds.

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

Note that acBoardTrapGlobalsName is actually a number. The value of this varbind is ‘X’ minus 1, where ‘X’ is the last number in the trap OID. For example, the ‘name’ of acBoardEthernetLinkAlarm is ‘9’. The OID for acBoardEthernetLinkAlarm is 1.3.6.1.4.1.5003. 9.10.1.21.2.0.10.

3.4 Voice Menu

> **Note:** Voice Menu is only applicable to **MediaPack** and **Mediant 1000**.

Initial configuration of the device may be performed using a standard touch-tone telephone connected to one of the FXS analog ports. The voice menu may also be used to query and modify basic configuration parameters.

➢ To configure networking parameters for the device, take these 6 steps:

1. Connect a telephone to one of the FXS ports. Lift the handset and dial ***12345 (three stars followed by the digits 1, 2, 3, 4, 5).
2. Wait for the 'configuration menu' voice prompt to be played.
3. To change the IP address, press 1 followed by the # key.
   - The current IP address of the device will be played. Press # to change it.
   - Dial the new IP address; use the star (*) key instead of dots ("."), e.g. 192*168*0*4 and press # to finish.
   - Review the new IP address, and press 1 to save.
4. To change the subnet mask, press 2 followed by the # key.
   - The current subnet mask of the device will be played. Press # to change it.
   - Dial the new subnet mask; e.g. 255*255*0*0 and press # to finish.
   - Review the new subnet mask, and press 1 to save.
5. To change the default gateway address, press 3 followed by the # key.
   - The current default gateway address of the device will be played. Press # to change it.
   - Dial the new default gateway address; e.g. 192*168*0*1 and press # to finish.
   - Review the new default gateway address, and press 1 to save.
6. Hang up the handset. Using a web browser, connect to the device's web interface to complete the device configuration and save it to non-volatile memory. Alternatively, the initial configuration can be performed using an HTTP server, as discussed in Automatic Update Facility on page 35. The Voice Menu can be used to specify the configuration URL.

➢ To set a configuration URL, take these steps:

1. Obtain the IP address of the configuration HTTP server, e.g., 36.44.0.6.
2. Connect a telephone to one of the FXS ports. Lift the handset and dial ***12345 (three stars followed by the digits 1, 2, 3, 4, 5).
3. Wait for the 'configuration menu' voice prompt to be played.
4. Dial 31 followed by the # key.
   - The current configuration IP address will be played. Press # to change it.
   - Dial the configuration server's IP address; use the star (*) key instead of dots

(“.”), e.g. 36*44*0*6 and press # to finish.

5. Dial 32 followed by the # key.
   - Press # to change the configuration file name pattern.
   - Select one of the patterns below (aa.bb.cc.dd denotes the IP address of the configuration server):

#	Pattern	Notes
1	http://aa.bb.cc.dd/config.ini	Standard config.ini
2	https://aa.bb.cc.dd/config.ini	Secure HTTP
3	http://aa.bb.cc.dd/audiocodes/<MAC>.ini	The device MAC address will be appended to the file name, e.g., http://36.44.0.6/audiocodes/00908f012300.ini
4	http://aa.bb.cc.dd:8080/config.ini	HTTP on port 8080
5	http://aa.bb.cc.dd:1400/config.ini	HTTP on port 1400
6	http://aa.bb.cc.dd/cgi-bin/acconfig.cgi?mac=<MAC>&ip=<IP>	Generating configuration per IP/MAC address dynamically, using a CGI script. See perl example below.

   - Press the selected pattern code, and press '#' to finish.

1. Press 1 to save, and hang up the handset. The device will fetch the configuration from the HTTP server.
   The following is an example of a perl CGI script, suitable for most Apache-based HTTP servers, for generating configuration dynamically per pattern #6 above.
   Copy this script to /var/www/cgi-bin/acconfig.cgi on your Apache server, and edit it as required:

```
#!/usr/bin/perl
use CGI;
$query = new CGI;
$mac = $query->param('mac');
$ip = $query->param('ip');

print "Content-type: text/plain\n\n";
print "; INI file generator CGI\n";
print "; Request for MAC=$mac IP=$ip\n\n";
print <<"EOF";

SyslogServerIP = 36.44.0.15
EnableSyslog = 1
SSHServerEnable = 1

EOF
```

The following configuration parameters may be queried or modified via the voice menu:

Table 3-70: Configuration Parameters

Item Number at Menu Prompt	Description
1	IP address
2	Subnet mask

Table 3-70: Configuration Parameters

Item Number at Menu Prompt	Description
3	Default gateway address
4	Primary DNS server address
7	DHCP enable/disable
11	MGCP call agent IP address
12	MGCP call agent port number
31	Configuration server IP address
32	Configuration file name pattern
99	Voice Menu password (initially 12345). Note that unless the password is changed from the default, the Voice Menu will only be available for initial configuration. As soon as the web password is changed, Voice Menu access will be disabled.

3.5 Individual ini File Parameters

The individual parameters contained in the ***ini*** file are provided in the following parameter group tables:

- System Parameters (refer to System Parameters on page 134)
- PSTN Parameters (refer to 'PSTN Parameters' on page 168)
- Infrastructure Parameters (refer to 'Infrastructure Parameters' on page 142)
- Media Processing Parameters (refer to Media Processing Parameters on page 155)
- SS7 Parameters (refer to 'SS7 Parameters' on page 188)
- Common Control Protocols Parameters (refer to 'Common Control Protocols Parameters')
- MGCP Specific Parameters (refer to 'MGCP Specific Parameters' on page 197)
- MEGACO Specific Parameters (refer to 'MEGACO Specific Parameters' on page 202)
- SNMP Parameters (refer to 'SNMP Parameters' on page 209)
- Web Interface Parameters (refer to 'Web Interface Parameters' on page 206)
- Voice Streaming Parameters (refer to 'Voice Streaming Parameters' on page 212)
- SCTP Parameters (refer to 'SCTP Parameters' on page 214)
- Advanced Audio Server Parameters (refer to 'Advanced Audio Server Parameters' on page 215)
- Names for optional configuration files (CAS signaling, Call Progress Tones and Voice Prompts files).
- IPsec Parameters (refer to IPsec Parameters on page 196)

- Video Parameter (refer to Video Parameters on page 217)
- MRCP Parameters (refer to MRCP Parameters on page 205)
- NFS Parameters (refer to NFS Parameters on page 196)
- Analog Parameters (refer to Analog Parameters on page 184) (Applicable to **MediaPack** only)

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 is configured with that parameter's default value. Leaving all ***ini*** file parameters unspecified and loading the file to the device is 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.

3.5.1 System Parameters

The table below lists and describes the system parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 3-71: System Parameters - ALL

Parameter Name	Description	Default	Range
ActivityListToLog	This parameter defines what activities are to be reported by the device in the form of a log message. Parameter format is x,y,z-... where x,y,z are activity codes to be reported by the device. **Supported activity codes:** PVC - Parameter Value Changes. AFL - Auxiliary Files Loading. DR - Device Reset. FB - Flash Burning. SWU - SoftWareUpdate. ARD - Access to Restricted Domains. NAA - Non Authorized Access. SPC - Sensitive Parameters Changes.	Empty string	Refer to: Supported activity codes in the Description column.
AUPDCheckIfIniChanged	With this parameter, AutomaticUpdate performs CRC checking to determine if the INI file has changed prior to processing. Possible values are: 0 - Do not check CRC. The INI file will	0	0 to 2

Table 3-71: System Parameters - ALL

Parameter Name	Description	Default	Range
	be loaded whenever the server provides it. 1 - Check CRC for the entire file. Any change, including line order, will cause the INI file to be re-processed. 2 - Check CRC for individual lines. Use this option when the HTTP server scrambles the order of lines in the provided INI file.		
AUPDVerifyCertificates	This parameter configures the AutoUpdate facility to verify server certificates when using HTTPS.	0	0 or 1
AutoUpdateCmpFile	Enables / disables the automatic update mechanism for the cmp file. 0 = The automatic update mechanism doesn't apply to the cmp file (default). 1 = The automatic update mechanism includes the cmp file.	0	0 or 1
AutoUpdateFrequency	Determines the number of minutes the gateway waits between automatic updates.	0 (update at fixed intervals mechanism is disabled)	Any number
Any number AutoUpdatePredefinedTime	Schedules an automatic update to a predefined time of the day.	NULL	'HH:MM' (24-hour format)
BehaviorUponRadiusTimeout	This parameter defines device behavior upon a RADIUS timeout. 0 = Deny access 1 = Check password locally	1	0,1
CmpFileURL	This parameter provides a link to a software image (CMP file) to be downloaded from a remote server.	NULL	See Descr.
CoderTableFileUrl	Provides a link to a coder table (CTBL) file that is to be downloaded from a remote server.		See Descr.
CptFileUrl	Provides a link to a Call Progress Tones (CPT) file to be downloaded from a remote server.	NULL	http://server_name/file, https://server_name/file
DefaultAccessLevel	This parameter defines the default access level for the device. Default value is 'Security Administrator' (= 200).	200	0 to 255
DialPlanFileName	This parameter is used to indicate name of the file containing the Dial Plan.	NULL	See Descr.

Table 3-71: System Parameters - ALL

Parameter Name	Description	Default	Range
DialPlanFileUrl	URL for downloading a Dial Plan file using the Automatic Update facility.	NULL	See Descr.
DisableWebConfig	Enables or disables Web Configuration. 0 = Read & Write mode (default) 1 = Read Only mode	0	0,1
DisableWebTask	Enables or disables Web Server Tasks. 0 = Enable (default) 1 = Disable	0	0,1
DNSPriServerIP	This parameter defines the DNS primary server's IP address.	0.0.0.0	Legal IP address
DNSSecServerIP	This parameter defines the DNS secondary server's IP address.	0.0.0.0	Legal IP address
ENABLEPARAMETERSMONITORING	This parameter is used to enable monitoring of on-the-fly parameter changes via Syslog messages. 1 = Activate; 0 = Deactivate.	0	0 or 1
EnableSecureStartup	Enables or disables secure startup mode. In this mode, downloading of the *.*ini* file is restricted to a URL provided in prior configuration (see parameter IniFileURL) or via DHCP.	0	0 or 1
ENABLESTUN	This parameter is used to enable the STUN module, used for NAT traversal of UDP packets.	0	0 or 1
EnableSyslog	This parameter is used to enable the Syslog protocol log. 1 = Activate; 0 = Deactivate	0	0 or 1
ENABLETLSHW	Used to enable hardware acceleration for TLS (SIPS/HTTPS.). Note: enabling this parameter may result in channel capacity degradation (same as IPsec)	0	1 = enable 0 = disable
ETHERDISCOVERMODE	Controls EtherDiscover mode of operation. 0 = Always disable EtherDiscover 1 = Enable EtherDiscover if unconfigured; but allow changes to IP configuration (default) 2 = Always enable EtherDiscover, but do NOT allow changes.	1	0 to 2
IniFileTemplateUrl	Provides a link to an *.*ini* file to be downloaded from a remote server, in addition to IniFileUrl.	NULL	http://server_name/file, https://server_name/file

Table 3-71: System Parameters - ALL

Parameter Name	Description	Default	Range
IniFileURL	This parameter provides a link to an **.ini* file to be downloaded from a remote server.	NULL	See Descr.
InitialShellCommand	A Command Shell command to be executed during initialization. Several commands can be entered (each separated by a semicolon).	NULL	-
NATBINDINGDEFAULTTIMEOUT	This parameter is used to define the NAT binding lifetime, in seconds. STUN refreshes the binding information after this time expires.	30	0 to 2592000
NTPServerIP	This parameter is used to define the NTP server's IP address.	0.0.0.0	Legal IP address
NTPServerUTCOffset	This parameter is used to define the NTP time to offset, in seconds.	0	-43200 to +43200 seconds
NTPUpdateInterval	This parameter defines the NTP update interval, in seconds. It is inadvisable to set it exceeding 1 month (2592000 sec)	86400 seconds	0 to 2592000
OcspDefaultResponse	Determines default OCSP behavior when the server cannot be contacted. 0 = Reject peer certificate 1 = Allow peer certificate	0	0 or 1
OcspEnable	Enables or disables certificate checking via OCSP. 0 – Disable 1 – Enable	0	0 or 1
OcspServerIP	This parameter defines the OCSP server's IP address.	0.0.0.0	Legal IP address
OcspServerPort	This parameter defines the OCSP server's TCP port number.	2560	1 to 32767
PrtFileUrl	Provides a link to a prerecorded tones dat file, to be downloaded from a remote server.	NULL	http://server_name/file, https://server_name/file
RadiusLocalCacheMode	This parameter defines the ability to reset the expiry of the local Radius password cache: 0 = Expiry can't be reset 1 = Expiry resets on each successful access to device	1	0 or 1
RadiusLocalCacheTimeout	Expiry time [sec] of locally stored RADIUS password cache. -1 = No Expiry; 0 = No Cache	300 seconds	-1 or 0

Table 3-71: System Parameters - ALL

Parameter Name	Description	Default	Range
SaveConfiguration	Determines if the device configuration (and the loadable file) is saved in flash. Choose either: 0 = Don't save 1 = Save configuration file (the Call Progress Tones, PRT and/or coefficient file) in non-volatile memory	1	0 or 1
SNMPSysLocation	Defines the physical location of the node, to be returned in the sysLocation object of MIB-2. By convention, this is the physical location of this node (e.g., `telephone closet, 3rd floor').	NULL	See Descr.
SNMPSysName	Defines the sysName as described in MIB-2. This is an administratively-assigned name for this managed node. By convention, this is the node's fully-qualified domain name.	NULL	See Descr.
SSHAdminKey	This parameter holds an RSA public key for strong authentication to the SSH interface (if enabled). The value should be a base64-encoded string; see the Security chapter for additional information.	NULL	See Descr.
SSHRequirePublicKey	Enables or disables RSA public keys in SSH. When set to 0, RSA public keys are optional (if SSHAdminKey is set). When set to 1, RSA public keys are mandatory.	0	0 or 1
SSHServerEnable	Enables or disables the embedded SSH server. 0 = Disable; 1= Enable	0	0 or 1
SSHServerPort	Defines the port number for the embedded SSH server.	23	Valid port number
StunServerDomainName	Defines the STUN Server's domain name. The STUN module finds all the servers under this domain using DNS SRV queries. Max of 64 bytes.	0.0.0.0	String[64]
STUNSERVERPRIMARYIP	Defines the primary STUN Server IP address.	0.0.0.0	Legal IP address
STUNSERVERSECONDARYIP	Defines the secondary STUN server IP address.	0.0.0.0	Legal IP address
SyslogServerIP	This parameter defines the IP address in dotted format notation. e.g., 192.10.1.255	0.0.0.0	Legal IP address
SyslogServerPort	Defines Port number of Syslog Server.	514	Legal Port Number

Table 3-71: System Parameters - ALL

Parameter Name	Description	Default	Range
TelnetServerEnable	Enables or disables the embedded Telnet server. Telnet is disabled by default for security reasons. 0 = Disable; 1= Enable 2 = SSL mode (if available - requires an SSL-aware Telnet client software) SSL mode is NOT available on the MP-108 / MP-124 media gateways	0	0 to 2
TelnetServerIdleDisconnect	This parameter is used to set the timeout for disconnection of an idle Telnet session (minutes). When set to zero, idle sessions are not disconnected.	0	Any number
TELNETSERVERPORT	Defines the port number for the embedded Telnet server.	23	Valid port number
TelnetServerVerifyPeerCertificate	Determines whether to enable the verification of peer (client) certificates by the embedded Telnet server in SSL mode. This parameter is applicable only when the TelnetServerEnable parameter is equal to 2. Possible values: 0 = Do not verify client certificates 1= Require client certificates and verify them For more information on client certificates, refer to the Security Chapter in this manual.	0	0 or 1
TLSCertFileUrl	URL for downloading a TLS certificate file using the Automatic Update facility.	NULL	See Descr.
TLSPkeyFileUrl	URL for downloading a TLS private key file using the Automatic Update facility.	NULL	See Descr.
TLSRootFileUrl	URL for downloading a TLS trusted root certificate file using the Automatic Update facility.	NULL	See Descr.
TLSVersion	This parameter defines the supported versions of SSL/TLS. When set to 0, SSL/TLS handshakes always start with SSL 2.0 and switch to TLS 1.0 if both peers support it. When set to 1, TLS 1.0 is the only version supported; clients attempting to contact device using SSL 2.0 will be rejected. 0 = SSL 2.0, SSL 3.0, and TLS 1.0 are supported (default)	0	0 or 1

Table 3-71: System Parameters - ALL

Parameter Name	Description	Default	Range
	1 = TLS 1.0 will always be used		
VideoFontFileUrl	Indicates the URL for downloading a logo file for the Web interface using the Automatic Update Facility.	NULL	See Descr.
VpFileUrl	Provides a link to a Voice Prompts file to be downloaded from a remote server.	NULL	http://server_name/file, https://server_name/file
WebLogoFileUrl	URL for downloading a logo file for the web interface using the Automatic Update facility.	NULL	See Descr.

Table 3-72: System Parameters - 6310

Parameter Name	Description	Default	Range
ActiveBoardIPAddress	Defines the IP addresses of the active blades in a High Availability configuration, from which the redundant blade receives state DB packets.		Valid IP address
ActiveBoardPort	Defines the port number for the High Availability service.	BSP_TPNCP_UDP_CONTROL_PORT	Valid port number
M3KGlobalIpAddr	Defines the M3K global IP address to be used by the active module in HA system. (Dotted format notation)	0.0.0.0	Legal IP address
M3KHASwUpgradeMode	Defines the type of SW upgrade in M3K HA system: Hitless upgrade only or system reset upgrade if hitless is not supported. If HITLESS_UPGRADE = 1 then user allowed only hitless S/W upgrade. If SYSTEM_RESET_UPGRADE = 2 then user also accepts system reset if hitless is not supported . This parameter can be set by **.ini* file and snmp.	HITLESS_UPGRADE	1 or 2
RedundantBoardIPAddress	Defines IP address of redundant blade, to which state DB packets are sent.	NULL	Any IP address

Table 3-73: System Parameters - IPM

Parameter Name	Description	Default	Range
APSSegmentsFileUrl	Provides a link to an XML segments file, to be downloaded from a remote	Not	Not

Table 3-73: System Parameters - IPM

Parameter Name	Description	Default	Range
	server. See the chapter 'Automatic Update Facility' for supported URL options.	applicable	applicable

Table 3-74: System Parameters - TP

Parameter Name	Description	Default	Range
CasFileUrl	Provides a link to a Channel Associated Signaling (CAS) file to be downloaded from a remote server. Range = http://server_name/file, https://server_name/file	NULL	http://server_name/file, https://server_name/file
PM_EnableThresholdAlarms	This parameter enables sending SNMP traps and Syslog messages when performance of the device is degraded (according to the configured thresholds).	0	0 or 1
ResetNow	Invokes an immediate restart of the gateway. This option can be used to activate offline (NOT on-the-fly) parameters that are loaded via IniFileUrl. 0 = The immediate restart mechanism is disabled (default). 1 = The gateway immediately restarts after an *.ini* file with this parameter set to 1 is loaded.	0	0 or 1
SystemOperationStateChangeProfile	This parameter defines the System Operation State Change Profile. 0 = Disable 1 = Nortel AMS ATM Refer to the enumerator acSystemOperationStateChangeProfile enum for the possible values.	0	Integer >0
TrunkingToAnalogFunctionalityProfile	This parameter defines the Trunking to Analog Functionality Profile. 0 = Disable; 1 = Enable MelCAS/LoopStart/GroundStart to Analog Functionality Refer to the enumerator acTrunkingToAnalogFunctionalityProfile enum for the possible values.	0	Integer >0

Table 3-75: System Parameters - MediaPack & Mediant 1000

Parameter Name	Description	Default	Range
DisableRS232	Enables or disables the RS-232 port. 0 = Enable; 1 = Disable	0	0 or 1
FXOCoeffFileUrl	Link to an FXO coefficients file, to be downloaded from a remote server.	NULL	http://server_name/file, https://server_name/file
FXSCoeffFileUrl	Link to an FXS coefficients file, to be downloaded from a remote server.	NULL	See Descr.
vmEnableWhenRTPActive	This parameter is used to enable the voice menu even when RTP is active (mid-call). 0 = Disable; 1 = Enable	0	0 or 1
VoiceMenuPassword	Password for the voice menu, used for configuration and status. To activate the menu, connect an analog telephone and dial *** (3 stars) followed by password.	12345	See Descr.

3.5.2 Infrastructure Parameters

The table below lists and describes the Infrastructure parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 3-76: Infrastructure Parameters

Parameter Name	Description	Default	Range
AuthorizedTPNCPServers	Sets the IP address of TPNCP authorized servers. Range = IP address	0.0.0.0	xxx.xxx.xxx.xxx
BaseUDPPort	Defines the lower boundary of UDP ports to be used by the device. The upper boundary is calculated on the basis of BoardBaseUDPPort + 10 * (Number of Channels). This parameter value must be a multiple of 10.	4000	See Descr.
BootPDelay	Defines the delay that occurs from time the device is reset until first BootP request is issued by the device. The parameter takes effect only from the time device is next reset.	1	1 to 5
BootPRetries	Defines number of BootP retries the device sends during start-up. The device stops issuing BootP requests when either an AA122BootP reply is received or Number Of Retries is	3	1 to 7 & 15

Table 3-76: Infrastructure Parameters

Parameter Name	Description	Default	Range
	reached. Parameter effective after next device reset. 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.		
BootPSelectiveEnable	Configures the device so that it will only accept BootP replies, from AudioCodes proprietary BootP-TFTP Software. 1 = Enable; 0 = Disable	0	0 or 1
BspDebugLevel	Sets the output level of BSP debug messages sent by the Gateway. Possible values: 0 = Deny; 1 = Show	0	0 or 1
bspTiming_E1_Line_Build_Out	Sets the transmission power between the timing module on the SAT and the E1 external reference clock (ohms). This parameter is only relevant for the Mediant 3000.This parameter is enabled when bspTimingModuleCfgTimingMode is set to TM_External_MODE(1). Possible values: 1 - E_75_OHM_normal(0),E_120_OHM_normal 4 - E_75_OHM_with_high_return_loss 5 -E_120_OHM_with_high_return_loss 6 - E_75_OHM_normal_PLUS_enable_transmit_and_receive_gapped_clock 7 - E_120_OHM_normal_PLUS_enable_transmit_and_receive_gapped_clock	1	0,1,4,5,6,7
bspTiming_T1_Line_Build_Out	Sets the transmission power between the timing module on the SAT and the T1 external reference clock. This parameter is only relevant for the Mediant 3000. Possible values: 0 - DSX_1_0_to_133_feet_0dB_CSU, 1 - DSX_1_133_to_266_feet 2 - DSX_1_266_to_399_feet 3 - DSX_1_399_to_533_feet	1	0,1,2,3,4,7

Table 3-76: Infrastructure Parameters

Parameter Name	Description	Default	Range
	4 - DSX_1_533_to_655_feet 7 - DSX_1_0_to_133ft_0dB_CSU_pluse_enable_transmit_and_receive_gapped_clock		
bspTimingModuleCfgTimingMode	Synchronizes the Gateway with one of the PSTN interfaces. Possible values: 0 - TM_Standalone_MODE - Non-synchronized mode - each blade or TPM is synchronized internally from one of the PSTN interfaces without using the SAT timing module. 2 - TM_LineSync_MODE - Distributed Line Timing mode (for the 1610 blade) without using the SAT timing module.	0	0 to 2
DHCPEnable	Enables/disables DHCP support. 0 = Disable; 1 = Enable When gateway powered, it attempts to communicate with a BootP server. If no response and if DHCP is enabled, gateway attempts to obtain its IP address & network parameters from DHCP server. Note that during DHCP procedure, the BootP/TFTP application must be deactivated. If not, gateway receives response from the BootP server instead of the DHCP server. For additional information on DHCP, refer to the product documentation. Note: DHCPEnable is a special 'Hidden' parameter. Once defined and saved in flash memory, its assigned value doesn't revert to its default even if parameter doesn't appear in the *.ini file.	0	0 or 1
DHCPSpeedFactor	Controls the DHCP renewal speed. When set to 0, the DHCP lease renewal is disabled. Otherwise, the renewal time is divided by this factor. Some DHCP-enabled routers perform better when set to 4. 0 = Disable DHCP;1 = Normal; 2 to 10 = Fast	1	0 to 10
DisableTPNCPEvent	Disables Events Reporting. For the selected event, refer to enumerator acTEvent. Range = nn = TPNCP EventID do hide.	1	nn

Table 3-76: Infrastructure Parameters

Parameter Name	Description	Default	Range
EnableDetectRemoteMACChange	Allows for the detection of an incoming RTP stream from a changed remote MAC address. Used for device redundancy purposes. 0 = Disable 1 = Enable (trigger by media) 2 = Enable (trigger by GARP) 3 = Enable (trigger by media or GARP)	3	0 to 3
EnableDHCPLeaseRenewal	Enables/disables DHCP renewal support. 0 = Disable; 1 = Enable Parameter effective if DHCPEnable = 0. When gateway is powered up, it attempts to communicate with a BootP server. If no response and if DHCP is disabled, the gateway boots from flash. It then attempt to communicate with DHCP server to renew the lease. Note that throughout DHCP procedure, BootP/TFTP application must be deactivated. If not, gateway receives a response from the BootP server instead of t DHCP server. For additional information on DHCP, refer to the product documentation. For cases where booting up the device via DHCP is not desirable, but renewing DHCP leasing is. if DHCPEnable = 1, this parameter has no effect.	0	0 or 1
EnableDiagnostics	Checks the correct functionality of the different hardware components on the device. On completion of the check, the device sends an EV_END_BIT value, which contains information on the test results 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)	0	0 to 2
EnableICMPUnreachableReport	Reports receipt of unreachable ICMP packets. 0 = Disabled; 1 = Enabled	1	0 or 1
EnableIPAddrTranslation	Specifies type of compare operation performed on first packet received on a newly opened channel for Network Address Translation (NAT) feature. If set to 1, the device compares the first incoming packet's source IP address, to the remote IP address stated in the opening of the channel. If the two IP addresses do not match, NAT operation	1	0 to 3

Table 3-76: Infrastructure Parameters

Parameter Name	Description	Default	Range
	takes place. Consequently, the remote IP address and the 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 for RTP, RTCP, T38 2 = Enable for Aggregation 3 = Enable for ALL		
EnableLANWatchdog	Detects LAN failures on the device. A LAN failure can result from a software or hardware malfunction. If a LAN failure is detected, the device performs a self reset (when not in PCI mode). 0 = Disable; 1 = Enable	0	0 or 1
ENABLENETWORKPHYSICALSEPARATION	Enables Network Physical Separation. Allows the user to have separate port for each Network. Requires suitable hardware. 0 = Disabled; 1 = Enabled	0	0 or 1
EnableTPNCPSecurity	Secures the TrunkPack Network Control Protocol (TPNCP) by accepting only pre-determined servers via the parameter defining authorized TPNCP servers. 1 = Enabled; 0 = Disabled	0	0 or 1
EnableUDPPortTranslation	Specifies the type of compare operation performed on the UDP ports. When set, the compare operation is performed on the UDP ports. If this parameter is set, EnableIpAddrTranslation must also be set. 0 = Disable; 1 = Enable	0	0 or 1
EthernetPhyConfiguration	Controls Ethernet connection mode type. Auto-negotiate falls back to Half-Duplex mode (HD) when the opposite port is not in Auto-negotiate mode. The speed (10 Base-T or 100 Base-TX) in this mode is always configured correctly. 0 = 10 Base-T half-duplex 1 = 10 Base-T full-duplex 2 = 100 Base-TX half-duplex 3 = 100 Base-TX full-duplex 4 = Auto-negotiate	4	0 to 4
ExtBootPReqEnable	Enables 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 such as device type, current IP address, software version, geographical address,	0	0 or 1

Table 3-76: Infrastructure Parameters

Parameter Name	Description	Default	Range
	etc. This is not available in DHCP.		
ForceExceptionDump	Forces an exception dump that is sent every time the device restarts. The last SW exception dump would be sent each time the device restarts. 0 = Disable; 1 = Enable	0	0 or 1
HeartbeatDestIP	Sets the destination UDP port to which the Heartbeat Packets are sent. Range = IP address in dotted notation	0.0.0.0	xxx.xxx.xxx.xxx
HeartbeatDestPort	Sets the destination UDP port to which the heartbeat packets are sent.	0	0 to 64000
HeartbeatIntervalmsec	Sets the time delay in msec between consecutive heartbeat packets. Use multiples of 10.	0xFFFFFFFF	0x0 to 0xffffffff
HeartbeatSecondaryDestIP	Sets secondary destination IP address to which heartbeat packets are sent. Range = IP address in dotted notation	0.0.0.0	xxx.xxx.xxx.xxx
ICMPUnreachableReportInterval	Determines: (a) The time the device ignores incoming ICMP unreachable packets from the channel activation time. (b) The time it takes from the last ICMP unreachable packet until the device reports ICMP Reachable.	5000	unsigned long
INIFileVersion	Contains the .ini file version number that is reported in the acEV_BOARD_STARTED event.	0	Long integer value
IPv6PrefixAndInterfaceIdMode	0 = Interface-ID derived from MAC; global-prefix from Router Discovery 1 = Interface-ID set manually; global-prefix from Router Discovery 2 = Interface-ID set to a random value; global-prefix from Router Discovery 3 = Interface-ID derived from MAC; global-prefix set manually 4 = Interface-ID set manually; global-prefix set manually 5 = Interface-ID set to a random value; global-prefix set manually	0	0 to 4
NewRtpStreamPackets	Defines the number continuous RTP packets for New RTP stream decision (the protection against multiple RTP streams is not active at all when this parameter is set to 0)	10	0 to 20
RoutingTableDestinationMasksColu	Comprises the destination masks	NULL	Legal IP

Table 3-76: Infrastructure Parameters

Parameter Name	Description	Default	Range
mn	column of the static routing rules that users can add to.		address
RoutingTableDestinationPrefixLensColumn	The prefix length value of the destination masks column of the static routing rules. Users can add static routing rules data to this column.	0	Legal IP address
RoutingTableDestinationsColumn	Comprises the Destination column of the static routing rules that users can add to.	NULL	Legal IP address
RoutingTableGatewaysColumn	Comprises the gateways column of the static routing rules that users can add.	NULL	Legal IP address
RoutingTableHopsCountColumn	Comprises the Hops count column of static routing rules that users can add.	20	0 to 255
RoutingTableInterfacesColumn	Comprises the interfaces column of the static routing rules that users can add.	0	0 to 2
SctpIPAddress	Set the source IP address for the SCTP traffic. Default = 0.0.0.0 (the main source IP will be used in that case) When working in multiple IP mode, the control/OAMP IP will be used (according to parameter: EnableSCTPasControl).	0.0.0.0	xxx.xxx.xxx.xxx
StreamingCacheDecisionIntreval	Defines the streaming cache decision interval in minutes. If -1 is set, decision will be made upon each cache request.	4	-1 to 0xFFFF
StreamingCacheNumOfDescriptors	Defines the number of monitored descriptors in the streaming cache.	5000	0 to 10000
StreamingCacheRefreshTime	Defines the streaming cache data refresh time in minutes. If -1 is set, refresh is off.	-1	-1 to 0xFFFF
StreamingCacheSize	Sets the streaming cache size in MB. The remaining size (out of 32 MB) is used as VoicePrompt storage.	0	0 to 32
TPNCPConnectionTimeout	Defines the TPNCP KeepAlive timeout (in seconds). If TPNCP is being used as a control protocol, this parameter indicates the amount of seconds after which, in case of inactivity (and 'KeepAlive' probes), the TPNCP <-> Lib connection will be timed out. 0 = Disable KeepAlive function 10 sec is the minimum value; any smaller value will be counted as if the user configured 10 sec.	0	Any value ≥ 10 sec
TpncpNatTraversalMode	This parameter indicates that the device should initiate the connection to the	0	0 or 1

Table 3-76: Infrastructure Parameters

Parameter Name	Description	Default	Range
	TPNCP host.		
TpncpNatTraversalPassword	Selects a password for authentication with the TPNCP host.	Rumble	Any string
vlanSendNonTaggedOnNative	Specify whether to send non-tagged packets on the native VLAN.	0. Priority-tagged packets (vlanId=0) are sent.	0 or 1

Table 3-77: Infrastructure Parameters - IPM

Parameter Name	Description	Default	Range
TDMBusH100Termination Enable	Enables or Disables H.100 TDM Bus Termination. 0 = Disable; 1 = Enable	0	0 or 1

Table 3-78: Infrastructure Parameters – MediaPack & Mediant 1000

Parameter Name	Description	Default	Range
SerialData	Changes the serial data bit for the Simplified Message Desk Interface (SMDI). 7 = 7 Bit; 8 = 8 Bit	8	7 or 8
SerialFlowControl	Changes the serial flow control for the Simplified Message Desk Interface (SMDI). 0 = None; 1 = Hardware	0	0 or 1
SerialParity	Changes the serial parity for the Simplified Message Desk Interface (SMDI). 0 = None; 1 = Odd; 2 = Even	0	0 to 2
SerialStop	Changes the serial stop for the Simplified Message Desk Interface (SMDI). 1 = 1 Bit; 2 = 2 Bit	1	1 or 2
SMDI	Enables the Simplified Message Desk Interface (SMDI). SMDI defines a method whereby telephony systems can provide voice-messaging systems with data required by those telephony systems to process incoming calls intelligently. Whenever the phone system routes a call, it sends a SMDI message through an EIA/TIA-232 connection to the voice-messaging system. It tells it: the line that it is using; the type of call that it is forwarding; and information about the source and destination of the call.	0	0 to 3

Table 3-78: Infrastructure Parameters – MediaPack & Mediant 1000

Parameter Name	Description	Default	Range
	0 = Normal Serial 1 = Serial SMDI 2 = Ericsson flavor of SMDI 3 = NEC Ics flavor of SMDI		
SMDIInternalNumberLen	Defines length of PBX internal number. relevant for Ericsson SMDI only.	0	2 to 10
SMDILineIdLen	Defines the line identification string length. Use 7 (default) for Bellcore SMDI, or between 2 and 5 for Ericsson SMDI.	7	2 to 5, or 7
SMDIMWIMinInterval	Minimum time interval (milliseconds) between sending subsequent MWI messages over SMDI.	250 msec	0 to 10,000 msec
SMDIMWIQueueSize	Queue size (number of entries) for throttling outgoing MWI messages over SMDI.	100	0 to 100

Table 3-79: Infrastructure Parameters – TP

Parameter Name	Description	Default	Range
BRONZESERVICECLASSDIFFSERV	Sets the DiffServ for the Bronze service class content.	10	0 to 63
DisableH100ClocksOnTrunkFailure	Disables H.100 clock's output when PSTN reference trunk fails. 0 = Disable; 1 = NETREF; 2 = A\B; 3 = All	0	0 to 3
DisableNetRefOnTrunkFailure	Disables the NETREF signal when the PSTN reference trunk fails. 1 = Disables the NETREF signal when PSTN reference trunk fails.	0	0 or 1
EnableBitTask	Enables the bit task. 0 = Disabled; 1 = Enabled	1	0 or 1
EnableDNSasOAM	Sets location of DNS. If parameter is set & machine is functioning in multiple IPs mode, DNS is on OAMP interface = 0. If not, DNS is on control interface = 1	1	0 or 1
EnableMediaUDPChecksum	For UDP streams carrying media content (both Audio and Video), the device supports the ability to insert a non-zero UDP layer checksum on the outgoing packets. 0 = Disabled and the outgoing UDP packets will carry the value of 0x00 in the UDP checksum field of the UDP header. 1 = Enabled Note: IPv6 mandates the use of non-zero UDP checksum fields. For IPv6 streams,	0	0 or 1

Table 3-79: Infrastructure Parameters – TP

Parameter Name	Description	Default	Range
	the proper value of the UDP checksum will be inserted regardless of the value of the INI file parameter. **Applicable to: Mediant 3000, Mediant 3000 1+1, IPmedia 3000, TP-6310, IPM-6310, TP-8410 and IPM-8410.**		
EnableMultipleIPs	Enables the multiple IPs feature. 0 = Disable; 1 = Enable	0	0 or 1
EnableNTPasOAM	Sets location of Network Time Protocol (NTP). If parameter is set & machine is functioning in multiple IPs mode, NTP is on OAMP interface = 0. If not, NTP is on control interface = 1	1	0 or 1
EnableSCTPasControl	Sets location of SCTP (Stream Control Transmission Protocol). If parameter is set & machine is functioning in multiple IPs mode, SCTP is located on control network. If not, SCTP is located on OAMP network. 0 = Default; 1 = Enable	1	0 or 1
EnableTPNCPasOAM	Sets TPNCP location on Operation, Administration and Management (OAMP) network. If parameter is set & machine is working in multiple IPs mode, TPNCP is located on the OAMP network. If not, SCTP is located on OAMP network. 0 = TPNCP on Control network 1 = TPNCP on OAMP network	1	0 or 1
EnableVoicePathBITTest	Enables the voice path bit test. 0 = Disable; 1 = Enabled	0	0 or 1
GOLDSERVICECLASSDIFFSERV	Sets the DiffServ for the Gold service class content.	26	0 to 63
IPv6LocalMediaDefaultGW	Defines the IPv6 Default Gateway address of the Media. Default = ::	::	Legal IPv6 address
IPv6LocalMediaIPAddress	Defines the IPv6 IP address of the Media.	::	Legal IPv6 address
LocalControlDefaultGW	Defines default gateway of the Control when operating in a multiple IP mode.	0.0.0.0	Legal IP address
LocalControlIPAddress	Defines the IP address of the Control when operating in a multiple IP mode.	0.0.0.0	Legal Subnet
LocalControlSubnetMask	Defines the Subnet Mask of the Control when operating in a multiple IP mode.	0.0.0.0	Legal Subnet
LocalMediaDefaultGW	Defines the default gateway for the media interface, when operating in a multiple IP mode.	0.0.0.0	Legal IP address
LocalMediaIPAddress	Defines the IP address of the Media when operating in multiple IP mode.	0.0.0.0	Legal IP address

Table 3-79: Infrastructure Parameters – TP

Parameter Name	Description	Default	Range
LocalMediaSubnetMask	Defines the Subnet Mask for the media interface when operating in a multiple IP mode.	0.0.0.0	Legal Subnet
LocalOAMDefaultGW	Sets Default gateway for OAMP interface when operating in multiple IPs mode.	0.0.0.0	Legal IP address in subnet
LocalOAMIPAddress	Sets the IP address of the OAMP (Operation, Administration, Management and Provisioning) when operating in multiple IPs mode.	0.0.0.0	Legal IP address
LocalOAMSubnetMask	Sets Subnet Mask for OAMP interface, when operating in multiple IPs mode.	0.0.0.0	Legal Subnet
MIIRedundancyEnable	Determines whether or not to activate LAN redundancy, for TP-260/UNI and IPM-260/UNI with two Ethernet ports. 0 = Disable; 1 = Enable	0	0 or 1
NETWORKSERVICECLASSDIFFSERV	Parameter is used to set the DiffServ for Network service class content.	48	0 to 63
PCMLawSelect	Selects the type of PCM Companding law in input/output TDM bus (TDM bus is defined using the TDMBusType parameter).1 = A-law; 3 = µ-law	Depends on the PSTN ProtocolType configuration.	1 or 3
PREMIUMSERVICECLASSCONTROLDIFFSERV	Sets the DiffServ for the Premium service class content and control traffic.	40	0 to 63
PREMIUMSERVICECLASSMEDIADIFFSERV	This parameter is used to set the DiffServ for Premium service class content and media traffic.	46	0 to 63
SubnetBroadcastAfterENetSOEnabled	Enables subnet broadcast after Ethernet switchover. 0 = Disable; 1 = Enable	0	0 or 1
TDMBITSClockReference	Configures the BITS clock reference when the device source clock is set to BITS and Fallback is set to manual or non-revertive. 1 = REF_1; 2 = REF_2	1	1 or 2
TDMBITSClockSource	Configures which clock is output to the BITS card and on which output signal. Range: 0 = No output (acTDMBusClockSource_Null) 4 = Network_A (acTDMBusClockSource_Network) 16 = Network_B (acTDMBusClockSource_Network_B) 17 = ATM_A (acTDMBusClockSource_ATM_OC3) 18 = ATM_B (acTDMBusClockSource_ATM_OC3_B)	0	0, 4, & 16 to 21

Table 3-79: Infrastructure Parameters – TP

Parameter Name	Description	Default	Range
TDMBusClockSource	Selects the clock source on which the device synchronizes. Range: 1 = Local oscillator 3 = MVIP 4 = PSTN Network 8 = H.110A 9 = H.110B 10 = NetRef1 11 = NetRef2 12 = SC2M 13 = SC4M 14 = SC8M TP-1610 = 3	1	1, 3, 4, & 8 to 22
TDMBusEnableFallback	Defines the auto fallback of the clock. 0 = Manual 1 = Auto Non-Revertive 2 = Auto Revertive	0	0 to 2
TDMBusFallbackClock	Selects the fallback clock source on which device synchronizes in the event of clock failure. 4 = PSTN Network 8 = H.110A 9 = H.110B 10 = NetRef1 11 = NetRef2	4	4, & 8 to 11
TDMBusLocalReference	When the clock source is set to Network, this parameter selects the Trunk ID to be used as the clock synchronization source of the device. When using H.110/H.100 bus, this parameter also selects the trunk used as the clock source for the NetRef clock generation (in this case, the clock source must not be set to Network.	0	0 to (MAX_TRUNK _NUM-1)
TDMBusmasterSlaveSelection	Sets SC/MVIP/H.100/H.110 to either: 0 = Slave mode (another device in the system must supply clock to TDM bus) or Master mode (the device 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	0	0 to 2
TDMBusNetrefOUTPUTMODE	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	0	0 to 3
TDMBusNetrefSpeed	Determines the NetRef frequency (for both generation and synchronization). 0 = 8 kHz 1 = 1.544 MHz	0	0 to 2

Table 3-79: Infrastructure Parameters – TP

Parameter Name	Description	Default	Range
	2 = 2.048 MHz		
TDMBusOutputPort	Defines the SC/MVIP/H.100/H.110 output port to be used for the device's channel #0. All other channels then occupy the next timeslots sequentially.	0	0 to 15 for SC/MVIP 0 to 31 for H.110
TDMBusOutputStartingChannel	Defines the outgoing TDM Timeslot for device's channel #0. The remaining channels are organized sequentially.	0	0 to 127
TDMBusSpeed	Selects the TDM bus speed according to the Bus Type as follows: SC = 0/2/3 H.110/H.100 = 3 MVIP = 0 0 = 2048 kbps 2 = 4096 kbps 3 = 8192 kbps 4 = 16384 kbps	TP-260/UNI = 2; All other blades = 3	0, 2, 3, 4
TDMBusType	Selects the TDM bus interface to be used (only one TDM bus interface can be enabled at one time although more than one can physically exist on the device). Range: 0 = acMVIP_BUS 1 = acSC_BUS 2 = acFRAMERS 4 = acH100_BUS 5 = EXT TDM 6 = Analog 8 = SW Pstn	TP-1610 = 2 TP-260/UNI = 1	0, 1, 2, & 4 to 8
VLANBRONZESERVICECLASSPRIORITY	Sets the priority for the Bronze service class content.	2	0 to 7
VLANCONTROLVLANID	Sets the control VLAN identifier.	2	1 to 4094
VLANGOLDSERVICECLASSPRIORITY	Sets the priority for the Gold service class content.	4	0 to 7
vlanHeartbeatPriority	Sets the priority value for the heartbeat VLAN tag. Range: A value of 8 will set the priority to the value defined by VLANPREMIUMSERVICECLASSCONTROLPRIORITY parameter. Any other value within the valid range will be set accordingly.	0	0 to 7, and 8
VLANHEARTBEATVLANID	Sets the heartbeat stream VLAN identifier.	0	1 to 4094
VLANMEDIAVLANID	Sets the media VLAN identifier.	3	1 to 4094
VLANMODE	Sets the VLAN functionality. 0 = Disable; 1 = Enable; 2 = PassThru	0	0 to 2

Table 3-79: Infrastructure Parameters – TP

Parameter Name	Description	Default	Range
VLANNATIVEVLANID	Sets the native VLAN identifier.	1	0,1
VLANNETWORKSERVICECLASSPRIORITY	This parameter is used to set the priority for Network service class content.	7	0 to 7
VLANOAMVLANID	Sets the OAMP (Operation, Administration Management and Provisioning) VLAN identifier.	1	1 to 4094
VLANPREMIUMSERVICECLASSCONTROLPRIORITY	Sets the priority for the Premium service class content and control traffic.	6	0 to 7
VLANPREMIUMSERVICECLASSMEDIAPRIORITY	Sets the priority for the Premium service class content and media traffic.	6	0 to 7

3.5.3 Media Processing Parameters

The table below 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 3-80: Media Processing Parameters

ini File Parameter name	Description	Default	Range
AMDDetectionDirection	Determines the AMD (Answer Machine Detector) detection direction. 0 = Detection from the TDM side 1 = Detection from the Network side	0	0 or 1
AMDDetectionSensitivity	Determines the AMD (Answer Machine Detector) detection sensitivity: 0 = Best detection of an answering machine 7 = Best detection of a live call	3	0 to 7
AMRCoderHeaderFormat	Determines the format of the AMR header. 0 = Non standard multiple frames packing in a single RTP frame. Each frame has a CMR & TOC header. 1 = Reserved. 2 = AMR Header according to RFC 3267 Octet Aligned header format. 3 = AMR is passed using the AMR IF2 format.	0	0 to 3
AMRECRedundancyDepth	Sets the AMR/WB-AMR Redundancy depth according to RFC 3267. 0 = No Redundancy 1 = Redundancy depth of a single	0	0 to 3

Table 3-80: Media Processing Parameters

ini File Parameter name	Description	Default	Range
	packet 2 = Redundancy depth of 2 packets 3 = Redundancy depth of 3 packets		
BasicRTPPacketInterval	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	0	0 to 3
BellModemTransportType	Use this parameter to set the Bell modem transport method. 0 = Transparent 2 = Bypass (enum ByPassEnable) 3 = Transparent with Events (enum EventsOnly)	0	0, 2, 3
BrokenConnectionEventActivationMode	Determines if the broken connection mechanism is activated when the RTP stream is activated or when the first RTP packet is received. (acTActivateBrokenConnection) Default = 0 = Activate when the first RTP packet is received		0 to 1

BrokenConnectionEventTimeout	Determines for how long the RTP connection should be broken before the Broken Connection event is issued. In units of 100 msec.
	Range = 3 to 21474836 in units of 100 msec (300 to 0x80000000 msec)
	Default = 3 (= 300 msec)
		See Descr.	See Descr.

CallerIDTransportType	Defines the CallerID Transport type. 0 = Disable 1 = Reserved 2 = Reserved 3 = Mute (events are being generated also).	3	0 to 3
CallerIDType	Defines the supported Caller ID type. 0 = Bellcore 1 = ETSI 2 = NTT 4 = British 16 = ETSI_ETS 17 = Denmark 18 = Indian 19 = Brazilian	0	See Descr.
CallProgressDetectorEnable	Enables or disables detection of Call Progress Tones.	1	0 or 1

Table 3-80: Media Processing Parameters

ini File Parameter name	Description	Default	Range
	0 = Disable 1 = Enable		
CallProgressTonesFilename	Defines Call Progress Tone filenames (downloaded by TFTP).	Null	
CASTransportType	Controls the ABCD signaling transport type over IP. 0 = No Relay over the network 1 = Enable CAS relay according to RFC 2833	0	0 or 1
CNGDetectorMode	Determines the CNG Detector mode. 0 = Disable 1 = Relay 2 = Event Only	0	0 to 2
ConnectionEstablishmentNotificationMode	Determines the notification mode for the RTP connection establishment event acEV_CONNECTION_ESTABLISHED. 0 = Notify only after a broken connection event 1 = Also notify when the first RTP packet is received	0	0 or 1
CPTDetectorFrequencyDeviation	Defines the deviation allowed for the detection of each CPT signal frequency. Units are in Hertz.	10	1 to 30
CPTDETECTORSNR	Defines the value of which CPT Signals with a Signal To Noise Ratio below this value will not be detected. Units are in dB.	15	10 to 60
DisableNAT	Enables or disables the NAT feature. 0 = Do not disable NAT 1 = Disable NAT	1	0 or 1
DisableRTCPRandomize	Controls whether RTCP report intervals are randomized or whether each report interval accords exactly to the parameter defining RTCP Mean Tx Interval. 0 = Randomize 1 = Don't Randomize	0	0 or 1
DJBufMinDelay	Defines the Dynamic Jitter Buffer Minimum Delay (in msec). Recommended value for a regular voice call is 10.	10	0 to 150
DJBufOptFactor	Defines the Dynamic Jitter Buffer frame error/delay optimization. recommended value for a regular voice call is 10.	10	0 to 12
DSPVersionTemplateNumber	Selects the DSP load number. Each load has a different coder list, a	0	0 to 255

Table 3-80: Media Processing Parameters

ini File Parameter name	Description	Default	Range
	different channel capacity and different features supported.		
DTMFDetectorEnable	Enables or disables detection of DTMF signaling. 0 = Disable 1 = Enable	1	0 or 1
DTMFGenerationTwist	Defines a delta (in dB) between the high and low frequency component in the DTMF signal. dB Positive values cause the higher frequency component to be stronger than the lower one. Negative values cause the opposite effect. For any parameter value, both components change so that their average is constant.	0	-10 to 10
DTMFTransportType	Defines the type of DTMF transport. 0 = Erase DTMFs from voice transport not relayed to remote 2 = DTMFs not erased are not relayed to remote 3 = DTMFs are muted from the voice stream and relayed according to RFC 2833 7 = DTMFs are sent according to RFC 2833 and muted when received	3	0, 2, 3, 7
DTMFVolume	Defines and controls the DTMF generation volume [-dBm].	-11	-31 to 0
EchoCancellerAggressiveNLP	User can enable or disable the Aggressive NLP at first 0.5 second of the call by setting this parameter. 0 = Disable 1 = Enable	0	0 to 1
ECHybridLoss	Sets the worst case ratio between the signal level transmitted to the hybrid and the echo level returning from hybrid. Set this per worst hybrid in the system in terms of echo return loss. Refer to the enumeration acTECHybridLoss. 0 = 6 dBm 2 = 0 dBm 3 = 3 dBm	0	0, 2, 3
EnableContinuityTones	Enables or disables Continuity Test tone detection and generation according to the ITU-T Q.724 recommendation. 0 = Disable 1 = Enable	0	0 or 1

Table 3-80: Media Processing Parameters

ini File Parameter name	Description	Default	Range
EnableEchoCanceller	Enables or disables the Echo Canceler. 0 = Disable 1 = Enable	1	0 or 1
EnableEVRCVAD	Enables or disables the EVRC Voice Activity detector. 0 = Disable 1 = Enable	0	0 or 1
EnableFaxModemInbandNetworkDetection	Enables or disables inband network detection related to fax/modem. 0 = Disable 1 = Enable	0	0 to 1
EnableMediaSecurity	Enables or disables Media Security protocol (SRTP) . Enabling this parameter might reduce the device channel capacity. 0 = Disable 1 = Enable	0	0 or 1
EnableNoiseReductionSupport	Enables or disables Noise Reduction. Enabling this parameter might reduce the device channel capacity. 0 = Disable 1 = Enable	0	0 or 1
EnablePatternDetector	Enables or disables activation of the PD (Pattern Detector). 0 = Disable 1 = Enable	0	0 or 1
EnableRFC2658Interleaving	When enabled, RTP packets include an interleaving byte for VBR coders. 0 = Disable 1 = Enable	0	0 or 1
EnableSilenceCompression	Enables or disables Silence Suppression Mode. 0 = Disable = SILENCE_COMPRESION_DISABLE 1 = Enable = SILENCE_COMPRESION_ENABLE 2 = Enable without adaptation = SILENCE_COMPRESION_ENABLE_NOISE_ADAPTATION_DISABLE	0	0 to 2
EnableStandardSIDPayloadType	When set to 1 (Enable), SID packets are sent with the RTP SID type (RFC 3389). 0 = Disable 1 = Enable Determines whether Silence Indicator (SID) packets that are sent and received are according to RFC 3389.	0	0 or 1
EnableSTUModemDetection	Enables or disables detection of two tones required for an STU modem. 0 = Disable 1 = Enable	0	0 or 1

Table 3-80: Media Processing Parameters

ini File Parameter name	Description	Default	Range
EVRCDTXMax	Defines the maximum gap between two SID frames, when using the EVRC voice activity detector.	32	0 to 20000
EVRCDTXMin	Defines the minimum gap between two SID frames, when using the EVRC voice activity detector.	12	0 to 20000
EVRCRate	Used to configure the EVRC coder bit rate. 0 = Variable Rate 1 = 1 kbps 2 = 4 kbps 3 = 8 kbps	0	0 to 3
FaxBypassOutputGain	Defines the fax bypass output gain control in dB.	0 (No Gain)	-31 to +31 in 1 dB step
FaxBypassPayloadType	Users can use this parameter to modify the Fax Bypass Mode RTP packet's payload type. In the case of congestion (if the selected payload type is already 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.	102	0 to 127
FaxModemBypasDJBufMinDelay	Determines the Jitter Buffer constant delay (in milliseconds) during a Fax & Modem Bypass session. (The minimum Jitter Buffer Size).	40	0 to 150
FaxModemBypassBasicRTPPacketInterval	Sets the basic Fax / Modem Bypass RTP packet rate. 0 = Default (set internally) 1 = 5 msec (PACKET_INTERVAL_5_MSEC) 2 = 10 msec (PACKET_INTERVAL_10_MSEC) 3 = 20 msec (PACKET_INTERVAL_20_MSEC)	0	0 to 3
FaxModemBypassCoderType	Users can use this parameter to set the fax/modem bypass coder (according to acTCoders). 0 = G.711 A-Law	0	0 to 64
FaxModemBypassM	Defines the number of basic frames to generate one RTP fax/modem bypass packet.	1	1 or 2
FaxModemRelayVolume	Determines the fax gain control. The range -18 to -3 relates to -18.5	-12	-18 to -3

Table 3-80: Media Processing Parameters

ini File Parameter name	Description	Default	Range
	dBm to -3.5 dBm in steps of 1 dBm.		
FaxRelayECMEnable	Enables or disables the using of ECM mode during Fax Relay. 0 = Disable 1 = Enable	1	0 or 1
FaxRelayEnhancedRedundancyDepth	Determines the number of repetitions to be applied to control packets when using the T.38 standard. 0 = No redundancy 1 = 1 packet redundancy 2 = 2 packet redundancy 3 = 3 packet redundancy 4 = Maximum redundancy	4	0 to 4
FaxRelayMaxRate	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	5	0 to 5
FaxRelayRedundancyDepth	Determines the depth of redundancy for fax packets. This parameter is applicable only to non-V.21 packets. 0 = No redundancy 1 = 1 packet redundancy 2 = 2 packet redundancy	0	0 to 2
FaxTransportMode	Sets the Fax over IP transport method. 0 = Transparent 1 = Relay 2 = Bypass 3 = Transparent with Events	1	0 to 3
G729EVLocalMBS	Determines the maximal bitrate, which may be used by the G.729EV coder at a specific channel. This parameter is defined per channel and may vary between the parties. The initial generation bit rate is the minimum between the MaxBitRate and the MBS values. Possible values are: 0 = G729EV_RATE_8_KBPS 1 = G729EV_RATE_12_KBPS 2 = G729EV_RATE_14_KBPS 3 = G729EV_RATE_16_KBPS 4 = G729EV_RATE_18_KBPS 5 = G729EV_RATE_20_KBPS 6 = G729EV_RATE_22_KBPS 7 = G729EV_RATE_24_KBPS	0	0 to 11,15

Table 3-80: Media Processing Parameters

ini File Parameter name	Description	Default	Range
	8 = G729EV_RATE_26_KBPS 9 = G729EV_RATE_28_KBPS 10 = G729EV_RATE_30_KBPS 11 = G729EV_RATE_32_KBPS, 15 = G729EV_RATE_UNDEFINED		
G729EVMaxBitRate	Determines the maximum generation bitrate for all participants in a session using G.729EV coder. This parameter is defined per session and is equal for all the parties. The initial generation bit rate is the minimum between the MaxBitRate and the MBS values. Possible values are: 0 = G729EV_RATE_8_KBPS 1 = G729EV_RATE_12_KBPS 2 = G729EV_RATE_14_KBPS 3 = G729EV_RATE_16_KBPS 4 = G729EV_RATE_18_KBPS 5 = G729EV_RATE_20_KBPS 6 = G729EV_RATE_22_KBPS 7 = G729EV_RATE_24_KBPS 8 = G729EV_RATE_26_KBPS 9 = G729EV_RATE_28_KBPS 10 = G729EV_RATE_30_KBPS 11 = G729EV_RATE_32_KBPS 15 = G729EV_RATE_UNDEFINED	0	0 to 11, 15
G729ECReceiveMBS	Determines the value of the MBS field of the G.729EV frames to be sent to the other party. This parameter reflects the maximum bit rate, which the local G.729EV supports as a receiver. Possible values are: 0 = G729EV_RATE_8_KBPS 1 = G729EV_RATE_12_KBPS 2 = G729EV_RATE_14_KBPS 3 = G729EV_RATE_16_KBPS 4 = G729EV_RATE_18_KBPS 5 = G729EV_RATE_20_KBPS 6 = G729EV_RATE_22_KBPS 7 = G729EV_RATE_24_KBPS 8 = G729EV_RATE_26_KBPS 9 = G729EV_RATE_28_KBPS 10 = G729EV_RATE_30_KBPS 11 = G729EV_RATE_32_KBPS	0	0 to 11, 15

Table 3-80: Media Processing Parameters

ini File Parameter name	Description	Default	Range
	15 = G729EV_RATE_UNDEFINED		
IBSDetectionRedirection	Determines the IBS (In-Band Signaling) Detection Direction. 0 = PCM 1 = Network	0	0 or 1
IdleABCDPattern	Defines the ABCD (CAS) pattern to be applied on the signaling bus before it is changed by the user or the PSTN protocol. This is only relevant when using the PSTN interface with CAS protocols. Range = 0x0 to 0xF	-	See Descr.
IdlePCMPattern	Defines the PCM pattern applied to the E1/T1 timeslot (B-channel) when the channel is idle. Default: 0xFF if PCMLawSelect is Mu-Law 0xD5 if PCMLawSelect is A-Law Range = 0x00 to 0xFF	See Descr.	See Descr.

InputGain	Defines the PCM input gain.
	Range = -32 dB to +31 dB in 1 dB steps.
	Default = No Gain
		0	-32 to +31

LowDSPResourcesEventHyst	Determines the space between the low and hi watermarks of the DSP resource notifications. Range = 0 to the maximum number of DSP channels	0	See Descr.
LowDSPResourcesEventThreshold	Determines when a notification indicating a 'low number of DSP resources' is issued. Range = Between 0 and the maximum number of DSP channels	0	See Descr.
MaxDTMFDigitsInCIDString	Determines the maximum number of DTMF digits in a DTMF-based Caller ID string.	26	0 to 26
MaxEchoCancellerLength	Defines the maximum device EC (Echo Canceler) length capability. 0 = EC length determined internally to reach maximum channel capacity. 4 = 32 milliseconds 11 = 64 milliseconds 22 = 128 milliseconds Using 64 or 128 msecs reduces the channel capacity to 200 channels.	0	See Descr.

Table 3-80: Media Processing Parameters

ini File Parameter name	Description	Default	Range
MFSS5DetectorEnable	Enables or disables detection of MF SS5 line signaling. 0 = Disable 1 = Enable	0	0 or 1
MFTransportType	Defines the type of MF transport. 0 = Erase MFs from voice transport not relayed to remote 2 = MFs not erased are not relayed to remote 3 = MFs are muted from the voice stream and relayed according to RFC 2833	3	0,2 & 3
MinDTMFDigitsInCIDString	Determines the minimum number of DTMF digits in a DTMF-based Caller ID string.	0	0 to 26
ModemBypassOutputGain	Defines the modem bypass output gain control in dB.	0 (No Gain)	-31 to +31 in 1 dB step
ModemBypassPayloadType	Users can use this parameter to modify the Modem Bypass Mode RTP packet's payload type. In the case of congestion (if the selected payload type is already 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's responsibility to avoid congestion with other payload types.	103	0 to 127
NoiseReductionActivationDirection	Noise Reduction activation direction 0 = from TDM side 1 = from Network side	0	0 to 1
NoiseReductionIntensity	Noise Reduction Intensity: 0 - Weakest 8 - Normal 15 - Strongest	8	See Descr.
NoOpEnable	Enable / disable the Noop packets sending mode. 0 = Disable 1 = Enable	0	0 to 1
NoOpInterval	Sets the Noop packets sending interval. Parameter value in milliseconds default value - 10 sec (10000 msec) Range = 20 to 600000	10000	20 to 600000

Table 3-80: Media Processing Parameters

ini File Parameter name	Description	Default	Range
	(20 msec - 10 min - 10 min = 600000)		
NSEMode	Enables or disables Cisco's NSE fax / modem automatic pass-through mode. 0 = Disable 1 = Enable	0	0 or 1
NSEPayloadType	Users can use this parameter to modify the NSE packet's payload type.	105	96 to 127
NTTDIDSignallingForm	Configures the signaling format used when generating an NTT DID. 0 = FSK Signal 1 = DTMF Based Signal	0	0 to 1
PDPattern	Defines the patterns that can be detected by the Pattern Detector. Range = 0 to 0xFF	-	0 to 0xFF
PDThreshold	Defines the number of consecutive patterns to trigger the pattern detection event.	5	0 to 31
PrerecordedTonesFileName	Defines the name (and path) of the file containing the Prerecorded Tones. Range = String of ASCII characters	-	See Descr.
QCELP13Rate	Configures the QCELP13 coder bit rate. 0 = Variable Rate 1 = 1 kbps 2 = 3 kbps 3 = 7 kbps 4 = 13 kbps	0	0 to 4
QCELP8Rate	Configures the QCELP8 coder bit rate. 0 = Variable Rate 1 = 1 kbps 2 = 2 kbps 3 = 4 kbps 4 = 8 kbps	0	0 to 4
R1DetectionStandard	Determines which one of the R1 MF protocol flavors will be used for detection. 0 = ITU 1 = R1.5	0	0 to 1
RFC2198PayloadType	This parameter sets the RFC 2198 (RTP Redundancy) packet's parameter 'RTP Payload Type'.	104	96 to 127
RFC2833RxPayloadType	Controls the RFC 2833 Relay RTP Payload type of received packets.	96	96 to 127
RFC2833TxPayloadType	Controls the RFC 2833 Relay RTP	96	96 to 127

Table 3-80: Media Processing Parameters

ini File Parameter name	Description	Default	Range
	Payload type of sent packets.		
RTPNOOPPayloadType	User can modify the Noop packets RTP Payload type by setting this parameter.	120	96 to 127
RTPRedundancyDepth	Enables or disables generation of RFC 2198 redundancy packets. 0 = Disable 1 = Enable	0	0 or 1
RxDtmfHangOverTime	Used to configure the Voice Silence time (in ms units) after playing DTMF or MF digits to the TDM side that arrived as Relay from the Network side.	1000	0 to 2000
SITDetectorEnable	Enables or disables SIT (Special Information Tone) detection according to the ITU-T recommendation E.180/Q.35. 0 = Disable 1 = Enable	0	0 or 1
SerialPortAuditIntervalMin	Defines the interval timeout in minutes, of the Serial Port audit. If set to “0”, the audit does not run.	0	0 to 60
TestMode	Defines the type of testing mode applied: 0 = Coder Loopback performs an encoder/decoder loopback inside the DSP device 1 = PCM Loopback loops back an incoming PCM to the outgoing PCM. 2 = ToneInjection generates a 1000 Hz tone to the outgoing PCM 3 = NoLoopback sets the channel to work in normal mode	3	0 to 3
TTYTransportType	Defines the transferring method of TTY signals during a call. 0 = Disable 2 = Relay	0	0 or 2
TxDtmfHangOverTime	Voice Silence time (in ms units) after detecting the end of DTMF or MF digits at the TDM side when the DTMF Transport Type is either Relay or Mute. This feature allows the user to configure the silence time.	100	0 to 2000
UDTDetectorFrequencyDeviation	Defines the deviation allowed for the detection of each signal frequency. Units are in Hertz.	50 Hz	1 to 50
UserDefinedToneDetectorEnable	Enables or disables detection of User Defined Tones signaling. 0 = Disable	0	0 or 1

Table 3-80: Media Processing Parameters

ini File Parameter name	Description	Default	Range
	1 = Enable		
V22ModemTransportType	Sets the V.22 modem transport method. 0 = Transparent 2 = Bypass 3 = Transparent with Events	2	0 to 3
V23ModemTransportType	Sets the V.23 modem transport method. 0 = Transparent 2 = Bypass 3 = Transparent with Events	2	0 to 3
V32ModemTransportType	Sets the V.32 modem transport method. 0 = Transparent 2 = Bypass 3 = Transparent with Events	2	0 to 3
V34ModemTransportType	Sets the V.34 modem transport method. 0 = Transparent 2 = Bypass 3 = Transparent with Events	2	0 to 3
VBRCoderHeaderFormat	0 - payload only (no header, no toc, no m-factor) 1- support 2658 format, 1 byte for interleaving header (always 0) and toc, no m-factor). Similar to RFC 3558 Header Free format. 2 – payload including toc only, allow m-factor 3- RFC 3358 Interleave/Bundled format	0	0 to 3
VoicePayloadFormat	This parameter describes the bit ordering of the G.726/G.727 payload. 0 = Little Endian 1 = Big Endian	0	0 or 1
VoicePromptsFileName	Defines the name (and path) of the file containing the Voice Prompts. Range = String of ASCII characters	-	See Descr.
VoiceVolume	Defines the voice output gain control.	0	-32 to +31
VQMONEnable	Enable voice quality monitoring and RTCP xr reports. 0 = Disable 1 = Enable	0	0 or 1

3.5.3.1 Template Mix Feature

Support for 2 DSP templates being used on a single device was added. This feature is supported via MEGACO and VoPLIb. Currently, only Template 1 (AMR) and Template 2 (EVRC) are supported with this feature. The channel count using this setup is:

Table 3-81: Template Mix Feature – Channel Count

DSP Template No.	IPM-260/UNI	IPM-1610	IPmedia 2000	IPmedia 3000	IPM-6310
1 (AMR)	96	48	48	960	960
2 (EVRC)	96	60	60	924	924

3.5.4 PSTN Parameters

The table below lists and describes the PSTN parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 3-82: PSTN Parameters - ALL

Parameter Name	Description	Default	Range
PSTNTransmissionType	Sets the PSTN Transmission type for the device. Relevant only when TDMBusType=acFRAMERS (2). Transmission type values are: 0 = None, not defined 1 = Optical SONET or SDH 2 = Copper DS3 (T3) 3 = Copper E1 or DS1 (T1)	TP-6310 = 0 Other devices = 3	0 to 3

Table 3-83: PSTN Parameters - TP

Parameter Name	Description	Default	Range
AutoClockTrunkPriority	Defines the trunk priority for auto-clock fallback Priority range is 0 to 100 (0 to 99 are settings, in which 0 = highest Priority; 100 = Do not choose this trunk)	0	0 to 100
BriLayer2Mode	Indicates point to point or point to Multipoint mode for layer2. Applicable in BRI trunks only. Point-to-point = 0; Point-to-Multipoint = 1	0	0 or 1
CASAddressingDelimiters	The PSTN call originator can add delimiters (e.g., '*', '#', 's0', 'sT', 'KP', etc.) at the start or the end of the Address or the ANI. 1 = Enabled = the CAS engine shows the added delimiters in the string 0 = Disabled = the CAS engine does not allow the delimiters in the string to be included and sends only the digits	0	0 or 1
CASFileName	This is a pointer to the CAS filename index (0-7). The index is CASFileName_X. CASFileName_0 through to CASFileName_7 are the path and names of	NULL	0 to 7

Table 3-83: PSTN Parameters - TP

Parameter Name	Description	Default	Range
	the CAS protocol configuration files.		
CASProtocolEnable	Enable or disables the possible CAS protocol configuration. When this parameter is enabled the conference will be disabled. 0 = Disable; 1 = Enable	TP-6310 = 0 Other devices = 1	0 or 1
CasStateMachineCollectANI	Controls the state machine to collect or discard ANI, in cases when the state machine handles the ANI collection (not related for MFCR2). This is a reconfiguration of CAS state machine global parameter. 0 = Don't Collect ANI; 1 = Collect ANI	-1	0, 1
CasStateMachineDigitSignalingSystem	Defines which Signaling System to use MF or DTMF for detection & generation. 0 = DTMF; 1 = MF	-1	0,1
CasStateMachineDTMFMaxOnDetectionTime	Overrides the CAS state machine global parameter 'Detect digit maximum on time' (according to DSP detection information event). Values in msec.	-1	Not applicable
CasStateMachineDTMFMinOnDetectionTime	Overrides the CAS state machine global parameter 'Detect digit minimum on time' (according to DSP detection information event); value is in msec. Digit time length must be longer than this value to receive a detection. Any number may be used, less than CasStateMachineDTMFMaxOnDetectionTime.	-1	See Descr.
CasStateMachineGenerateDigitOnTime	Overrides the CAS state machine global parameter 'Generate digit on-time' msec	-1	Not applicable
CasStateMachineGenerateInterDigitTime	Overrides the CAS state machine global parameter 'Generate digit off-time' msec	-1	Not applicable
CasStateMachineMaxNumOfIncomingAddressDigits	Defines the limitation for the maximum number of address digits that need to be collected. Address collection stops when this number is reached.	-1	Up to 40 digits
CasStateMachineMaxNumOfIncomingANIDigits	Defines the limitation for the maximum number of ANI digits that need to be collected. When number of digits has been reached, collection of ANI stops.	-1	Up to 40 digits
CASTableIndex	This parameter determines which CAS protocol file to use on a specific trunk. The index value corresponds to the number configured for the parameter CASFileName_X. Range = not greater than the parameter defining the PSTN CAS Table Num.	0	X = 0 to 7
CASTablesNum	This parameter defines the quantity of CAS	0	0 to 8

Table 3-83: PSTN Parameters - TP

Parameter Name	Description	Default	Range
	tables that are loaded to the device during a reset. The quantity of CAS tables defined should match the value configured for parameter CASFILENAME_X. 0 = there is no CAS table to be loaded		
CasTrunkDialPlanName	Sets the Dial Plan name that will be used on the specific trunk.	""	String 11 characters
ClockMaster	Used to select the trunk clock source. 0 = acCLOCK_MASTER_OFF (clock recovered from the line) 1 = acCLOCK_MASTER_ON (the trunk clock source is provided by internal / TDM bus clock source depending on the parameter TDM Bus Clock Source)	0	0 or 1
DCHConfig	Defines D-channel configuration. This setting is only applicable to ISDN PRI protocols that support NFAS and/or D-channel backup procedures. 0 = Primary; 1 = Backup; 2 = NFAS	0	0 to 2
DIGITALPORTINFO	Digital Port information identifier (a user-defined string).	0	All
DisableTrunkAfterReset	Disables a Trunk - The trunk behaves as if it is not physically connected, i.e., it enters mode of: no transmit on that Trunk. Used to change the transmission state of the PSTN physical device. Enable, Disable (Tri state) or Send Blue Alarm. 0 = Trunk Enabled; 1 = Trunk Disabled	0	0 or 1
DPNSSBehavior	The DPNSSBehavior parameter represents a Bit field parameter. Each bit represents a specific type of DPNSS behavior. **Currently only first 2 bits are in use.** DPNSS_BEHAV_STOP_SABMR_AFTER_NL_AND_NT1 bit: (bit #0, bit mask 0x0001) **When set to 1**: DPNSS stops repeating SABMR after NL and NT1 limits are exceeded. **When set to 0**: DPNSS continues repeating SABMR after NL and NT1 limits are exceeded = Default = 0 (continue repeating SABMR)	0	0 or 1

Table 3-83: PSTN Parameters - TP

Parameter Name	Description	Default	Range
	DPNSS_BEHAV_FULL_STARTUP_SUCCESS bit: (bit #1, bit mask 0x0002) **When set to 1**: the Startup Procedure is considered as a SUCCESS only when ALL DLCs succeeded to Reset; **When set to 0**: the Startup Procedure is considered as a SUCCESS as soon as 1 DLC succeeded to Reset; Default is 0: (only partial reset is considered as a success).		
DPNSSNumRealChannels	This parameter is relevant only to protocol ISDN DPNSS. Defines the number of real channels.	30	1 to 30
DPNSSNumVirtualChannels	This parameter is relevant only to protocol ISDN DPNSS. Defines the number of virtual channels.	30	0 to 30
DS1PMEnable	Use this parameter to enable or disable the DS1 performance monitoring. 0 = DISABLE_PERFORMANCE_MONITORING 1 = ENABLE_PERFORMANCE_MONITORING	1	0 or 1

FramingMethod	Selects the physical framing method 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 (as D)]
	1 = T1 SuperFrame Format (as B).
	a = E1 DoubleFrame Format
	b = E1 CRC4 MultiFrame Format
	c = E1 CRC4 MultiFrame Format extended G.706B
	A = T1 4-Frame multiframe.
	B = T1 12-Frame multiframe (D4).
	C = T1 Extended SuperFrame without CRC6
	D = T1 Extended SuperFrame with CRC6
	E = T1 72-Frame multiframe (SLC96)
	F = J1 Extended SuperFrame with CRC6 (Japan)
		See Descr.	0, 1, a, b, c, A, B, C, D, E, F

ISDNDuplicateQ931BuffMode	Activates / de-activates delivery of raw Q.931 messages. Refer to the VoPLib documentation ('ISDN Flexible Behavior').	0	0 to 255
ISDNGeneralCCBehavior	This is the bit-field used to determine several general ISDN behavior options. Refer to the VoPLib documentation (ISDN Flexible Behavior).	0	Not applicable

Table 3-83: PSTN Parameters - TP

Parameter Name	Description	Default	Range
ISDNIBehavior	Bit-field used to determine several behavior options, which influence how the Q.931 protocol behaves. Refer to the VoPLib documentation (ISDN Flexible Behavior).	0	Not applicable
ISDNInCallsBehavior	This is the bit-field used to determine several behavior options that influence how the ISDN Stack INCOMING calls behave. Refer to the VoPLib documentation (ISDN Flexible Behavior).	0	Not applicable
ISDNNFASInterfaceID	Defines the Interface ID. Works with NS_EXPLICIT_INTERFACE_ID. Refer to the VoPLib documentation (ISDN Flexible Behavior).	(unsigned char)-1	0 to 255
ISDNOutCallsBehavior	This is the bit-field used to determine several behavior options that influence how the ISDN Stack OUTGOING calls behave. Refer to the VoPLib documentation (ISDN Flexible Behavior).	0	Not applicable
IUAInterfaceID	Defines the IUA trunk interface ID value - unsigned integer - in RFC 3057 - SIGTRAN. Default = 0xFFFFFFFF.	0xFFFFFFFF	0 to 0xFFFFFFFF
LineBuildOut.LOSS	Used to select 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	0	0 to 3
LineBuildOut.OVERWRITE	Used to overwrite the Framer's XPM registers values (these registers control the line pulse shape). 0 = No overwrite; 1 = Overwrite	0	0 or 1
LineBuildOut.XPM0	Used to control the Framer's XPM0 register value (line pulse shape control). Applicable only when TrunkConfig.LineBuildOut.Overwrite=1. Should be used only by expert users.	0	0 to 255
LineBuildOut.XPM1	Used to control the Framer's XPM1 register value (line pulse shape control). Applicable only when TrunkConfig.LineBuildOut.Overwrite=1. Should be used only by expert users.	0	0 to 255
LineBuildOut.XPM2	Used to control the Framer's XPM2 register value (line pulse shape control). Applicable only when TrunkConfig.LineBuildOut.Overwrite=1. Should be used only by expert users.	0	0 to 255
LineCode	Use to select line code. B8ZS or AMI for T1 spans and HDB3 or AMI for E1 spans. 0 = Use B8ZS line code (for T1 trunks only = default) 1 = Use AMI line code (for T1 or E1 trunks)	0	0 to 2

Table 3-83: PSTN Parameters - TP

Parameter Name	Description	Default	Range
	2 = Use HDB3 line code (for E1 trunks only)		
NFASGroupNumber	Relevant only to ISDN NFAS trunks, this parameter indicates the group number of the NFAS group. Valid NFAS group numbers are only 1 to 9. 0 indicates that this trunk is not NFAS (in this case the parameters ISDN NFAS Interface ID and Dch Config are ignored).	0	0 to 9

ProtocolType	Used to set the PSTN protocol to be used for this trunk. Relevant only when TDMBusType=acFRAMERS (2).
	Either:
	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 = 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
	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
	E1_FRENCH_VN6_ISDN = 30
	E1_FRENCH_VN3_ISDN = 31
	T1_EURO_ISDN = 34
	T1_DMS100_MERIDIAN_ISDN = 35
	T1_NI1_ISDN = 36
	E1_DUA = 37
	E1_Q931_PACKETS = 38
	T1_Q931_PACKETS = 39
	E1_NI2_ISDN = 40
	acPROTOCOL_TYPE_BRI_EURO_ISDN = 50
	acPROTOCOL_TYPE_BRI_NI2_ISDN = 51
	acPROTOCOL_TYPE_BRI_DMS100_ISDN
		0	1-23, 26, 28-31, & 34-40, 50-57

Table 3-83: PSTN Parameters - TP

Parameter Name	Description	Default	Range

	= 52
	acPROTOCOL_TYPE_BRI_5ESS_10_ISDN = 53
	acPROTOCOL_TYPE_BRI_QSIG = 54
	acPROTOCOL_TYPE_BRI_FRENCH_VN6_ISDN = 55
	acPROTOCOL_TYPE_BRI_NTT_ISDN = 56
	acPROTOCOL_TYPE_BRI_IUA = 57		

PSTNTransmissionType	Sets the PSTN Transmission type for the device. Relevant only when TDMBusType=acFRAMERS (2). Transmission type values are: 0 = None, not defined 1 = Optical SONET or SDH 2 = Copper DS3 (T3) 3 = Copper E1 or DS1 (T1)	TP-6310 = 0 Other devices = 3	0 to 3
Q931RelayMode	Activates / de-activates the ISDN level 3 Q.931 Relay Mode. Choose 0 or ActivateLAPDmessaging or Q931_RELAY_TO_HOST or Layer3_IS_IUA.	0	0 to 3
TDMBusPSTNAutoClockEnable	Use parameter to enable or disable the PSTN trunk auto-fallback clock feature. 0 = PSTN_Auto_Clock_Disable 1 = PSTN_Auto_Clock_Enable	0	0 or 1
TDMBusPSTNAutoClockRevertingEnable	Use this parameter to enable / disable the PSTN trunk auto-fallback clock reverting feature. If the TDMBusPSTNAutoClockEnable parameter is enabled and a trunk returning to service has an AutoClockTrunkPriority parameter which is set higher than the priority of the local reference trunk (in the TDMBusLocalReference parameter). The local reference reverts to the trunk with the higher priority that has returned to service. The “TDMBusPSTNAutoClockRevertingEnable” parameter specifies whether to change the device's TDMBusLocalReference and derive the clock from it. 0 = PSTN_Auto_Clock_Reverting_Disable 1 = PSTN_Auto_Clock_Reverting_Enable	0	0 or 1

Table 3-83: PSTN Parameters - TP

Parameter Name	Description	Default	Range
TDMHairPinning	Define static TDM hairpinning (cross-connection) to be performed at initialization. Connection is between trunks with the option to exclude a single B-channel in each trunk. Format e.g.: T0-T1/B3,T2-T3,T4-T5/B2.	NULL	See Descr.
TerminationSide	Used to set the ISDN Termination to either User or Network. Termination = For ISDN only. User side = 0; Network side = 1	0	0 or 1

TraceLevel	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
		0	0 to 8, 10 to 12, 15

TrunkAdministrativeState	Defines the administrative state of a trunk. 0 = Lock the trunk - stop trunk traffic to configure the trunk protocol type 2 = Unlock the trunk - enable trunk traffic	2	0 or 2
TrunkLifeLineType	This parameter is used to define the type of trunk lifeline activation. Trunk lifeline = Short trunks 1-2, 3-4. 0 = Activate lifeline on power down 1 = Activate lifeline on power down or on detection of LAN disconnect 2 = Activate lifeline on power down or on detection of LAN disconnect or loss of ping	0	0 to 2

3.5.4.1 PSTN SDH/SONET Parameters

Table 3-84: PSTN Parameters

Parameter Name	Description	Default	Range
SDHFbrGrp_SDHSONETMode	Selects SDH / SONET mode for the PSTN interface. Generally per Fiber Group. Single Fiber	0	0 to 2

Table 3-84: PSTN Parameters

Parameter Name	Description	Default	Range
	Group supported in the PSTN interface of TP6310. Applicable only to the TP-6310, Mediant 3000 /6310, IPM-6310, IPmedia 3000/6310 Relevant only when TDMBusType=acFRAMERS (2) and PSTNTransmissionType=Optical SONET or SDH Transmission type(1). 0 = Unknown 1 = STM1 2 = OC3 Should be in coordination with other parameters as follows: - PSTNTransmissionType - SDHFbrGrp_LP_Mapping_Type - ProtocolType When STM1 mode selected, use SDHFbrGrp_LP_Mapping_Type = 1 (VC12) and ProtocolType for E1. When OC3 mode selected, use SDHFbrGrp_LP_Mapping_Type = 0 (VT1.5) and ProtocolType for DS1.		
SDHFbrGrp_LP_Mapping_ Type	Selects the Low Path mapping type (TU/VT signal label + payload mapping type) for the PSTN interface. Generally per Fiber Group. Single Fiber Group supported in the PSTN interface of TP-6310. Applicable only to the TP-6310, Mediant 3000 /6310, IPM-6310, IPmedia 3000/6310 Relevant only when TDMBusType=acFRAMERS (2). 0 = Asynchronous VT1.5 and DS1 1 = Asynchronous TU12 and E1 15 = Undefined Should be in coordination with other parameters as follows: * SDHFbrGrp_LP_Mapping_Type * ProtocolType For more details please see SDHFbrGrpSdhSonetMode parameter description.	15	0, 1, 15
SDHFbrGrp_Protected	Set to true (1) to activate APS (Automatic Protection Switching) mechanism on PSTN interface. Generally per Fiber Group. Single Fiber Group supported in the PSTN interface of TP6310. Applicable only to the TP-6310, Mediant 3000 /6310, IPM-6310, IPmedia 3000/6310. 0 = APS not activated 1 = APS activated	1	0, 1

Table 3-84: PSTN Parameters

Parameter Name	Description	Default	Range
SDHFbrGrp_APS_DirMode	Sets the Automatic Protection Switch Uni-directional/Bi-directional mode for the Fiber Group: 0 = Uni-directional 1 = Bi-directional Applicable only to the TP-6310, Mediant 3000 /6310, IPM-6310, IPmedia 3000/6310	0	0 to 1
SDHFbrGrp_APS_RevertMode	Sets the Automatic Protection Switch Revertive mode for the Fiber Group. 0 = Non-revertive (Default) 1 = Revertive Applicable only to the TP-6310, Mediant 3000 /6310, IPM-6310, IPmedia 3000/6310	0 (Non revertive)	0 to 1
SDHFbrGrp_APS_WTR	Sets the APS Wait-to-restore time for the Fiber Group. Applicable only to the TP-6310, Mediant 3000 /6310, IPM-6310, IPmedia 3000/6310	5 min	5 to 12 min
SdhPmEnable	Enables or disables Performance Monitoring for the fiber group. 0 = Not Active 1 = Active (Default)	1	0 to 1
SonetSdhMediumCircuit Identifier	SDH / SONET circuit name.	“”	NA
SDHFbrGrp_KLM_Numbering_ Scheme	The scheme for VC/VT numbering on STM-1/OC3. Once a scheme is selected, there is mapping of 3 variables (K, L, M) that identify VC/VT inside the STM-1/OC3 to a trunk number (0 to 62 or 0 to 83) K is the TUG3 number (SDH) or OC1 number (SONET) L is the TUG2 number (SDH) or VT group number (SONET) M is the TU number (SDH) or VT number (SONET) Value 0 sets the numbering scheme (ETSI) where M is run first, then L, and then last K. Value 1 sets the numbering scheme (GR-253) where L is run first, then M, and then last K. Value 2 sets the numbering scheme (Hardware timeslots) where K is run first, then L, and then last M. Applicable only to the TP-6310, Mediant 3000 /6310, IPM-6310, IPmedia 3000/6310 Relevant only when TDMBusType=acFRAMERS (2) and	0	0 to 2

Table 3-84: PSTN Parameters

Parameter Name	Description	Default	Range
	PSTNTransmissionType=Optical SONET or SDH Transmission type(1).		

For a more detailed explanation of the above parameters please refer to the paragraphs below.

3.5.4.2 SDH/SONET Configuration

Note: This section is only relevant for PSTN STM-1 / OC-3 ports in TP-6310.

- The blade supports both STM-1 and OC-3 transmission modes. The mode is determined by ***ini*** file parameter SDHFBRGRP_SDHSONETMODE (1: STM-1, 2: OC-3)
- The Tributary Types supported are VC12 in STM-1 and VT1.5 in OC-3. These types are selectable by ***ini*** file parameter SDHFBRGRP_LP_MAPPING_TYPE (0: VT1.5, 1: VC12)
- The blade supports the following two working modes (selected by the two parameters described above):
  - Mapping of E1s into VC12s and than multiplexing of 63 VC12 into an STM-1 frame
  - Mapping of T1s into VT1.5s and multiplexing of 84 VT1.5 into an OC-3 frame
- The blade supports Trunk Numbering in TPNCP and H.248: When working in STM-1, trunk numbers are in the range of 0 to 62 (E1s) and in OC-3, from 0 to 83 (T1s).

3.5.4.3 Trunk Numbering (KLM Numbering)

Trunks are numbered sequentially while corresponding SDH/SONET instances (timeslots/columns) are have 3 referenced numbers built hierarchically. This complex triple numbering is called here KLM-numbering.

Selection of KLM numbering scheme is important when interconnecting different equipment, e.g., TP-6310 and a multiplexer. If different KLM-numbering schemes are selected on either equipment, there will be mismatch in the trunk sequential numbering.

There are 3 popular KLM numbering schemes:

- ETSI - according to ETSI EN 300 417-1-1, Annex D
- GR-253 - according to GR-253-CORE Issue 3 September 2000
- Timeslots - according to ITU-T G.707 clause 7.3.9. (Hardware timeslots)

All three are supported by the TP-6310 blade.

The SDHKlmNumberingScheme *ini* file parameter is used to select the scheme. Values are:

- acSDH_KLM_NUMBERING_SCHEME_MLK /* ETSI */. M is first running number, L is the second and K is the third
- acSDH_KLM_NUMBERING_SCHEME_LMK /* GR-253 */. L is first running number, M is the second and K is the third

- acSDH_KLM_NUMBERING_SCHEME_KLM /* TIMESLOTS */. K is first running number, L is the second and M is the third

Once the user has selected the scheme, the corresponding table will be selected automatically for 63 or 84 tributaries (trunks).

3.5.4.4 E1 Trunk Enumeration (”SDH” Mappings)

The following table is used for converting internal STM-1 (KLM) numbering to sequential trunk numbering for API references. The 3 numbers - TUG3 (K), TUG2 (L) and TU (M) - set the position of the E1 (V-12) trunk inside the STM-1 frame.

Table 3-85: STM-1 Numbering Conversion Table

Trunk	ETSI			GR-253			Timeslots		
	TUG-3	TUG-2	TU-12	TUG-3	TUG-2	TU-12	TUG-3	TUG-2	TU-12
1	1	1	1	1	1	1	1	1	1
2	1	1	2	1	2	1	2	1	1
3	1	1	3	1	3	1	3	1	1
4	1	2	1	1	4	1	1	2	1
5	1	2	2	1	5	1	2	2	1
6	1	2	3	1	6	1	3	2	1
7	1	3	1	1	7	1	1	3	1
8	1	3	2	1	1	2	2	3	1
9	1	3	3	1	2	2	3	3	1
10	1	4	1	1	3	2	1	4	1
11	1	4	2	1	4	2	2	4	1
12	1	4	3	1	5	2	3	4	1
13	1	5	1	1	6	2	1	5	1
14	1	5	2	1	7	2	2	5	1
15	1	5	3	1	1	3	3	5	1
16	1	6	1	1	2	3	1	6	1
17	1	6	2	1	3	3	2	6	1
18	1	6	3	1	4	3	3	6	1
19	1	7	1	1	5	3	1	7	1
20	1	7	2	1	6	3	2	7	1
21	1	7	3	1	7	3	3	7	1
22	2	1	1	2	1	1	1	1	2
23	2	1	2	2	2	1	2	1	2
24	2	1	3	2	3	1	3	1	2
25	2	2	1	2	4	1	1	2	2

Table 3-85: STM-1 Numbering Conversion Table

Trunk	ETSI			GR-253			Timeslots		
26	2	2	2	2	5	1	2	2	2
27	2	2	3	2	6	1	3	2	2
28	2	3	1	2	7	1	1	3	2
29	2	3	2	2	1	2	2	3	2
30	2	3	3	2	2	2	3	3	2
31	2	4	1	2	3	2	1	4	2
32	2	4	2	2	4	2	2	4	2
33	2	4	3	2	5	2	3	4	2
34	2	5	1	2	6	2	1	5	2
35	2	5	2	2	7	2	2	5	2
36	2	5	3	2	1	3	3	5	2
37	2	6	1	2	2	3	1	6	2
38	2	6	2	2	3	3	2	6	2
39	2	6	3	2	4	3	3	6	2
40	2	7	1	2	5	3	1	7	2
41	2	7	2	2	6	3	2	7	2
42	2	7	3	2	7	3	3	7	2
43	3	1	1	3	1	1	1	1	3
44	3	1	2	3	2	1	2	1	3
45	3	1	3	3	3	1	3	1	3
46	3	2	1	3	4	1	1	2	3
47	3	2	2	3	5	1	2	2	3
48	3	2	3	3	6	1	3	2	3
49	3	3	1	3	7	1	1	3	3
50	3	3	2	3	1	2	2	3	3
51	3	3	3	3	2	2	3	3	3
52	3	4	1	3	3	2	1	4	3
53	3	4	2	3	4	2	2	4	3
54	3	4	3	3	5	2	3	4	3
55	3	5	1	3	6	2	1	5	3
56	2	5	2	3	7	2	2	5	3
57	3	5	3	3	1	3	3	5	3
58	3	6	1	3	2	3	1	6	3
59	3	6	2	3	3	3	2	6	3

Table 3-85: STM-1 Numbering Conversion Table

Trunk	ETSI			GR-253			Timeslots		
60	3	6	3	3	4	3	3	6	3
61	3	7	1	3	5	3	1	7	3
62	3	7	2	3	6	3	2	7	3
63	3	7	3	3	7	3	3	7	3

3.5.4.5 T1 Trunk Enumeration (“Sonet” Mappings)

The following table is used for converting internal OC3 numbering to sequential trunk numbering for API references. The 3 numbers - STS-1 (K), TUG2 (L) and TU (M) - set the position of the T1 (V-1.5) trunk inside the OC3 frame.

Table 3-86: OC3 Numbering Conversion Table

Trunk	ETSI			GR-253			Timeslots		
	STS-1	VTG	VT1.5	STS-1	VTG	VT1.5	STS-1	VTG	VT1.5
1	1	1	1	1	1	1	1	1	1
2	1	1	2	1	2	1	2	1	1
3	1	1	3	1	3	1	3	1	1
4	1	1	4	1	4	1	1	2	1
5	1	2	1	1	5	1	2	2	1
6	1	2	2	1	6	1	3	2	1
7	1	2	3	1	7	1	1	3	1
8	1	2	4	1	1	2	2	3	1
9	1	3	1	1	2	2	3	3	1
10	1	3	2	1	3	2	1	4	1
11	1	3	3	1	4	2	2	4	1
12	1	3	4	1	5	2	3	4	1
13	1	4	1	1	6	2	1	5	1
14	1	4	2	1	7	2	2	5	1
15	1	4	3	1	1	3	3	5	1
16	1	4	4	1	2	3	1	6	1
17	1	5	1	1	3	3	2	6	1
18	1	5	2	1	4	3	3	6	1
19	1	5	3	1	5	3	1	7	1
20	1	5	4	1	6	3	2	7	1

Table 3-86: OC3 Numbering Conversion Table

Trunk	ETSI			GR-253			Timeslots		
21	1	6	1	1	7	3	3	7	1
22	1	6	2	1	1	4	1	1	2
23	1	6	3	1	2	4	2	1	2
24	1	6	4	1	3	4	3	1	2
25	1	7	1	1	4	4	1	2	2
26	1	7	2	1	5	4	2	2	2
27	1	7	3	1	6	4	3	2	2
28	1	7	4	1	7	4	1	3	2
29	2	1	1	2	1	1	2	3	2
30	2	1	2	2	2	1	3	3	2
31	2	1	3	2	3	1	1	4	2
32	2	1	4	2	4	1	2	4	2
33	2	2	1	2	5	1	3	4	2
34	2	2	2	2	6	1	1	5	2
35	2	2	3	2	7	1	2	5	2
36	2	2	4	2	1	2	3	5	2
37	2	3	1	2	2	2	1	6	2
38	2	3	2	2	3	2	2	6	2
39	2	3	3	2	4	2	3	6	2
40	2	3	4	2	5	2	1	7	2
41	2	4	1	2	6	2	2	7	2
42	2	4	2	2	7	2	3	7	2
43	2	4	3	2	1	3	1	1	3
44	2	4	4	2	2	3	2	1	3
45	2	5	1	2	3	3	3	1	3
46	2	5	2	2	4	3	1	2	3
47	2	5	3	2	5	3	2	2	3
48	2	5	4	2	6	3	3	2	3
49	2	6	1	2	7	3	1	3	3
50	2	6	2	2	1	4	2	3	3
51	2	6	3	2	2	4	3	3	3
52	2	6	4	2	3	4	1	4	3
53	2	7	1	2	4	4	2	4	3
54	2	7	2	2	5	4	3	4	3

Table 3-86: OC3 Numbering Conversion Table

Trunk	ETSI			GR-253			Timeslots		
55	2	7	3	2	6	4	1	5	3
56	2	7	4	2	7	4	2	5	3
57	3	1	1	3	1	1	3	5	3
58	3	1	2	3	2	1	1	6	3
59	3	1	3	3	3	1	2	6	3
60	3	1	4	3	4	1	3	6	3
61	3	2	1	3	5	1	1	7	3
62	3	2	2	3	6	1	2	7	3
63	3	2	3	3	7	1	3	7	3
64	3	2	4	3	1	2	1	1	4
65	3	3	1	3	2	2	2	1	4
66	3	3	2	3	3	2	3	1	4
67	3	3	3	3	4	2	1	2	4
68	3	3	4	3	5	2	2	2	4
69	3	4	1	3	6	2	3	2	4
70	3	4	2	3	7	2	1	3	4
71	3	4	3	3	1	3	2	3	4
72	3	4	4	3	2	3	3	3	4
73	3	5	1	3	3	3	1	4	4
74	3	5	2	3	4	3	2	4	4
75	3	5	3	3	5	3	3	4	4
76	3	5	4	3	6	3	1	5	4
77	3	6	1	3	7	3	2	5	4
78	3	6	2	3	1	4	3	5	4
79	3	6	3	3	2	4	1	6	4
80	3	6	4	3	3	4	2	6	4
81	3	7	1	3	4	4	3	6	4
82	3	7	2	3	5	4	1	7	4
83	3	7	3	3	6	4	2	7	4
84	3	7	4	3	7	4	3	7	4

3.5.4.6 SDH/SONET APS Parameters

Note: This section is only relevant for PSTN STM-1 / OC-3 ports in TP-6310.

The blade supports a number of Automatic Protection Switch (APS) modes. Please use the ini file parameters below to correctly configure the APS:

- SDHFbrGrp_Protected parameter

1 = "protected", default

0 = "unprotected"

If "protected" is selected, then APS is activated. All APS parameters below are relevant only if "protected" is selected.

- SDHFbrGrp_APS_DirMode parameter allows the selection between Unidirectional and Bidirectional APS modes

0 = Unidirectional APS mode, default

1 = Bidirectional APS mode

- SDHFbrGrp_APS RevertMode parameter allows the setting Revertive mode of APS

0 = Non-revertive switching, default

1 = Revertive switching

- SDHFbrGrp_APS_WTR parameter allows to set the Wait-To-Restore time (in minutes) in the case the Revertive mode was selected by the SDHFbrGrp_APS RevertMode parameter (5 to 12 minutes with a default of 5 minutes).

3.5.5 Analog Parameters (MediaPack and Mediant 1000 Analog only)

The table below lists and describes the analog parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 3-87: Analog Parameters

Parameter Name	Description	Default	Range
AnalogCallerIDTimingMode	Defines the Analog CallerID Timing Mode. 0 = CallerID transferred between first and second rings 1 = CallerID transferred on valid Off ring	0	0 or 1
ANALOGPORTINFO	Analog Port Information	0	Up to 50 chars.
BellcoreCallerIDTypeOneSubStandard	Selects the sub-standard of the Bellcore Caller ID type. 0 = Between_Rings 1 = Not_Ring_Related 2 = Before_Ring_RP_AS	0	0 to 2
BellcoreVMWITypeOneStandard	Use this parameter to select the Bellcore	0	0 to 1

Table 3-87: Analog Parameters

Parameter Name	Description	Default	Range
	VMWI standard. 0 = Between_Rings 1 = Not_Ring_Related		
CallerIDGeneration	Defines the type of Caller ID. 0 = Bell 202; 1 = V23; 2 = DTMF	0	0 to 2
CallProgressTonesFilename	Defines Call Progress Tone filenames (downloaded by TFTP).	Null	0 to 48 chars.
CountryCoefficients	Allows user to modify line characteristic (AC and DC) according to country.	70	0 to 71
CurrentDisconnectDefaultThreshold	Sets voltage threshold for current disconnect detection by reading line voltage. After setting voltage threshold, compare its value to CurrentDisconnectDefaultThreshold value. If measured threshold is smaller than *.ini file parameter's value, update threshold to the same value configured for *.ini file parameter.	4	0 to 20
CurrentDisconnectDuration	Defines current-disconnect duration (msec). Value is used in generation and detection.	900	200 to 1500
DisableAnalogAutoCalibration	Determines whether to enable the analog Autocalibration in the Direct Access Arrangement (DAA).	0	0 to 1
DisconnectToneType	Defines which CPT types are detected as far-end disconnect. The CPT type is based on the acTCallProgressToneType enum. This is valid when FarEndDisconnectType allows CPT detection.	0	An array of up to 4 tone types
DistinctiveRingFreq0	Defines the Distinctive Ringing Frequency, in units of 10 msec.	50	All
EnableAnalogDCRemover	Determines whether to enable the analog DC remover in the DAA. Possible values: 0 = DC remover is disabled 1 = DC remover is enabled	0	0 to 1
ETSICallerIDTypeOneSubStandard	Selects the number denoting the ETSI CallerID Type 1 sub-standard. 0 = ETSI_Between_Rings 1 = ETSI_Before_Ring_DT_AS 2 = ETSI_Before_Ring_RP_AS 3 = ETSI_Before_Ring_LR_DT_AS 4 = ETSI_Not_Ring_Related_DT_AS 5 = ETSI_Not_Ring_Related_RP_AS 6 = ETSI_Not_Ring_Related_LR_DT_AS	0	0 to 6
ETSIVMWITypeOneStandard	Selects the number denoting the ETSI VMWI Type 1 Standard.	0	0 to 6

Table 3-87: Analog Parameters

Parameter Name	Description	Default	Range
	0 = ETSI_VMWI_Between_Rings 1 = ETSI_VMWI_Before_Ring_DT_AS 2 = ETSI_VMWI_Before_Ring_RP_AS 3 = ETSI_VMWI_Before_Ring_LR_DT_AS 4 = ETSI_VMWI_Not_Ring_Related_DT_AS 5 = ETSI_VMWI_Not_Ring_Related_RP_AS 6 = ETSI_VMWI_Not_Ring_Related_LR_DT_AS		
FarEndDisconnectSilenceMethod	Defines the FarDisconnect silence detection method. 0 = None 1 = Packets count 2 = Voice/Energy Detectors 255 = All	2	0 to 2, 255
FarEndDisconnectSilencePeriod	Defines the Silence period to be detected.	120	10 to 28800
FarEndDisconnectSilenceThreshold	Defines the threshold (in percentages) of the packets to be considered as Silence. This is only applicable if Silence is detected according to the packet count (where FarEndDisconnectSilenceMethod = 1).	8	1 to 100
FarEndDisconnectType	This parameter sets the source for the acEV_FAR_END_DISCONNECTED event (or for the relevant control protocol event). It is a bit field parameter, hence (for example) if both CPT and current disconnect are required, the parameter should be set to 5. FarEndDisconnect contributor: 1 = CPT 2 = PolarityReversal 3 = CPT & PolarityReversal 4 = CurrentDisconnect 5 = CPT & CurrentDisconnect 7 = CPT & PolarityReversal & CurrentDisconnect 8 = Silence	15	0 to 15
FlashHookPeriod	Defines the flashhook period (in msec) for both analog and IP sides. For the analog side it defines: ▪ The maximal hook-flash detection period (for FXS gateways). A longer signal is considered offhook / onhook event. ▪ The hook-flash generation period (for	700 msec	25 to 1500

Table 3-87: Analog Parameters

Parameter Name	Description	Default	Range
	FXO gateways). For the IP side it defines the flash-hook period that is reported to IP. Note: For FXO gateways, add constant of 90 msec to required hook-flash period; e.g., to generate 450 msec hook-flash, set 'FlashHookPeriod' to 540.		
FXOCoefFilename	Defines FXO loop coefficient filename.	Not applicab le	Max 48 characters
FXSCoefFilename	Defines FXS loop coefficient filename.	Not applicab le	Max 48 characters
GroundKeyDetection	Enables/disables the analog ground key detection. 0 = Disable; 1= Enable	0	0 or 1
LifeLineType	Defines the Lifeline phone type. The Lifeline phone is available (for FXS only) on port 4 in MP-104 and MP-108, on port 2 in MP-102, on ports 1-4 in the MP-118, and on port 2 of each analog module in the Mediant 1000. 0 = activate Lifeline phone on power down 1 = activate Lifeline phone on power down or on detection of LAN disconnect 2 = activate Lifeline phone on power down, or on detection of LAN disconnect, or on loss of ping	0	0, 1 or 2
MeasPersistence	Defines the time (in msec) that passes from the time of detection until the interrupt signal.	0	All
MeteringOnTime	Setting the metering signal duration to be detected	200	50-1500
MeteringType	Sets the metering method for charging pulses. 0 = 12 kHz sinusoidal bursts 1 = 16 kHz sinusoidal bursts 2 = Polarity Reversal pulses	0	0 to 2
MinFlashHookTime	Sets the minimal time (in msec) for detection of a flash-hook event (for FXS only). Detection is guaranteed for flash hook periods of at least 60 msec (when setting the minimal time to 25). Flash-hook signals that last a shorter period of time are ignored. Note: It is recommended to reduce the detection time by 50 msec from the required value (e.g. if you set the value as	300 msec	25 to 300 msec

Table 3-87: Analog Parameters

Parameter Name	Description	Default	Range
	200 msec, then enter 150 msec, i.e. 200 minus 50).		
MWIndicationType	Defines the type of Message Waiting Indicator (MWI). Relevant for FXS only. 0 = the MWI is generated according to Bellcore (FSK) and ETSI standards 1 = a voltage of 100 VDC is applied to the line, lighting a lamp on the TE equipment	0	0 or 1
PolarityReversalType	Sets the type of the polarity reversal signal used for the network far-end answer and disconnect indications. Smooth reversal prevents negative effects as non-required ringing. 0 = Soft reverse polarity 1 = Hard reverse polarity	0	0 or 1
RingDeglitch	Defines the time (in msec) to prevent detection of glitch/noise as a ring.	0	All
RingPersistence	Defines the time (in msec) from the ring detection to signaling the ring interrupt.	0	All
THRMeasPersistence	Defines the time (in msec) that passes from when the THR INT is detected until the interrupt signal.	0	All
TimeToSampleAnalogLine Voltage	Determines the time to sample the analog line voltage after offhook, for the current disconnect threshold.	1000	100 to 2500
Winktime	Defines the time elapsed between two consecutive polarity reversals.	200	All

3.5.6 SS7 Parameters

The table below lists and describes the SS7 parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 3-88: SS7 Parameters - ALL

Parameter Name	Description	Default	Range
SS7M3UATrafficBehavior	Defines the M3UA behavior when the SS7 links are up, but there is no association to the soft switch. 0 = NONE (no special behavior) 1 = DEACTIVATE (busy links) 2 = SIPO (LPO links)	0	0 to 2
SS7MTP3RdcyMode	This parameter defines the SS7 MTP3-User Adaptation Layer redundancy mode. Determines the redundancy flavor. 0 =	0	0 or 1

Table 3-88: SS7 Parameters - ALL

Parameter Name	Description	Default	Range
	Disabled; 1 = Enabled		
SS7MTP3RdcyBoardNum	Used to define the device number for the Signaling System 7 (SS7) MTP3-User Adaptation Layer redundancy mode. Each device is allocated a unique number. All devices share a single redundancy table.	0	0 to 2
SS7MTP3RdcyKeepAliveInterval	Defines redundancy X-link keep-alive interval in seconds. (x-link between devices in SS7) MTP3-User Adaptation Layer redundancy mode). 0 = no keep-alive mechanism is activated.	1	0 to 30000
SS7MTP3RdcyKeepAlive Window	Defines redundancy X-link keep-alive tolerance window. (x-link between devices in SS7) MTP3-User Adaptation Layer redundancy mode).	2	0 to 15
SS7MTP3RdcyTblSyncInterval	Defines the interval between SS7 tables automatic synchronizations, in minutes. 0 = no automatic synchronization is activated.	0	0 to 30000
SS7MTP3RdcyTransferType	This is an MTP3-User Adaptation Layer parameter of the SS7, used to define the cross-device connection media type for the redundancy feature. 0 = M3BRDCY_CONN_TYPE_NONE 2 = M3BRDCY_CONN_TYPE_TCP	0	0 and 2

3.5.7 Control Protocol Parameters

The table below 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 3-89: Control Protocol Parameters - ALL

Parameter Name	Description	Default	Range
AdminState	Determines the gateway's administrative state 0 = locked 1 = shutting-down (read only) 2 = unlocked.	2	0 to 2
AdminStateLockControl	Defines the time remaining (in seconds) for the shutdown to complete. 0 = immediate shutdown -1 = waits until all calls drop (infinite) >0 = the number of seconds to wait	-1	-1, ≥ 0

Table 3-89: Control Protocol Parameters - ALL

Parameter Name	Description	Default	Range
CallAgentDomainName	Defines a domain name to be used to connect with the Call Agent. The parameter takes precedence over the Call Agent IP and the provisioned Call Agent parameters.	NULL	String[63]
CallWaitingToneDuration	Changes the call waiting tones family duration, in msec.	12,000 msec	300 to 300,000 msec
CnfNoiseSuppressionEnable	Controls VAD feature on Conference 0 = VAD feature on the Conference user is always enabled. 2 = Causes the DSP to disable VAD feature on Conference user when the number of actual users is not more than ConferenceMaxSimultaneousSpeakers parameter.	NULL	0 or 2
CODERTBLFILENAME	This parameter defines the name of an external coders table. In this table, the user can decide which coders will be used in the system. The original file is a text file, and it is converted by DCONVERT to a binary file.	""	String[63]
CPCipherSuiteType	Defines the default cipher type for the control protocol: 0 = none; 1 = TGCP; 2 = SRTP	0	0 or 1
cpPlayCoder	The coder type to be used when playing a file of type .raw. For the legal coder names, refer to the product's User Manual.	G.711	See Descr.
cpRecordCoder	Determines the coder used for recording to all supported file types. One of the following values: PCMU (G.711 µ-law) PCMA (G.71A-law)	PCMU for 6300 device group PCMA for all other devices	See Descr.
CPSDPPROFILE	Controls MGCP/MEGACO functioning for SDP negotiation. The parameter is bitwise. Bit 0 (Value 1) = Perform SDP Negotiation according to RFC 3407 spec. Bit 1 (Value 2) = Support V.152 (VBD) Bit 2 (Value 4) = Perform SDP Negotiation according to RFC 3264 spec. Bit 3 (Value 8) = Perform SDP Negotiation only according to received	unsigned Integer > 0	See Descr.

Table 3-89: Control Protocol Parameters - ALL

Parameter Name	Description	Default	Range
	SDP. Ignore default configure Bit 4 (Value 16) = Replied SDP will contain also T, S and O lines Bit 5 (Value 32) = Hard Coded Configure PTime Value for Transparent Coder. The Value is 10 Every new RFC support should be turned on or off with this parameter.		
CPSDPSESSIONOWNER	Defines the owner/creator of the session	-	String[31].
CPSERVICECHANGEPROFILE	Specifies the Profile (if any) of the supported protocol. At this stage, the profile will not determine the features supported.	"TGW"	String[63]
CPTransportType	Defines the transport type for the control protocol: 0 = UDP; 1 = TCP	0	0 or 1
DefaultPacketizationPeriod	Defines the default packetization period (Frame Size).	20 msec (for G.723 30)	5 to 80
DialedStringPrefix	Defines a prefix to add to the dialed string.	NULL	String[8]
DialToneDuration	Defines the timeout (in seconds) for the dial tone signal.	16	1 to 65535
DigitMapTimeoutTimer	Defines the timeout value (T symbol) in a digit map, in increments of 10. For MEGACO, it's the start timer. For the others, it's the end timer.	16	1 to 65535
DisableDLCXByGW	MGCP: Enables or disables the self-generation of DLCX commands by the media gateway. 0 = DLCX generated by gateway. 1 = DLCX not generated by gateway; the call-agent must issue DLCX commands for active calls.	0	0 or 1
DisconnectBehavior	Determines PBX behavior upon losing connectivity with H.248 Call agent or TPNCP. 1 = No Action = keep routing traffic 2 = Disable Trunks = stop routing traffic BUT RTP remains active 3 = Reset device = Stop all	1	1 to 3
DTMFDigitLength	Defines the time to play DTMF, in msec.	100	0 to 65535
DTMFInterDigitInterval	Defines the time between DTMFs played, in msec.	100	0 to 65535
EnableCallerIDTypeTwo	Enables or disables Caller ID Type 2. If Off (0), Caller ID Type Two is not played	1	0 or 1

Table 3-89: Control Protocol Parameters - ALL

Parameter Name	Description	Default	Range
	(if playing is requested from Call Agent). 0 = Off; 1 = On		
EndpointName	MGCP: Gateway's endpoint 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	MGCP: 'Acgw' MEGACO: 'line' for analog device and '/c' for trunking devices	String[19]
EndpointPREFIX	(MGCP) This parameter generates (together with parameter Trunk Name) a local endpoint name on trunk-enabled media gateways.	NULL	String[19]
GatewayName	Defines the media gateway's identification name. MGCP: Gateway's identification name towards the MGCP Call Agent. If undefined, the gateway name holds the IP address of the device. MEGACO: Prefix of the gateway part of the termination name.	MGCP: AudioCodes.com MEGACO: NULL for analog devices and 'tgw' for trunking devices	String[63]
KeepAliveEnabled	Parameter can be used to enable a KeepAlive message (NOP ServiceChange). 0 = disable; >0 = enable	0	0 or >0
KeepAliveInterval	This parameter is used to define the interval in seconds of a KeepAlive message.	12	1 to 300
MGControlProtocolType	Defines the control protocol type. Choose either: 0 = None 1 = MGCP 2 = MEGACO 4 = H.323 8 = SIP	1	0 to 2, 4, 8
MGCPCommunicationLayer Timeout	Assumed delay of the communication layer used in retransmission. This parameter defines the maximal time to wait for a response before declaring a	30	>0

Table 3-89: Control Protocol Parameters - ALL

Parameter Name	Description	Default	Range
	disconnection (in seconds).		
MGCPCompatibilityProfile	Controls MGCP/MEGACO functioning for vendor-specific compatibility. Refer to the product's documentation or the enumerator mgTMGCPProfile for possible values. **Note: Value "4096" is no longer valid.**	1	Integer > 0
MGCPDefaultCoder	This parameter can be used to set a default coder for channel opening. For the legal coder names, refer to the product's User Manual. Default = cpDPT_G711Mulaw_Coder	G.711	See Descr.
MGCPDefaultPacketizationPeriod	Defines the default packetization period (Frame Size).	20	5 to 120
MGCPDTMFDetectionPoint	Defines when the detection of DTMF events is notified. 0 = at start of DTMF 1 = at the end of DTMF	1	0 or 1
MGCPRetransmissionTimeout	Controls protocols retransmission timeout. Sets the initial time (in msec) for the first retransmission. The retransmission intervals thereafter increase exponentially.	200 msec	0 to 10000 msec
MGCPRetransmitionTimeout	Sets the initial time for the first retransmission. The Retransmission intervals thereafter increase exponentially.	200 msec	0 to 65535 msec
PMCongestionHysteresis	Controls the protocols Congestion Performance Monitoring Hysteresis Value.	2	1 to 20
ProvisionedCallAgents	Use this parameter to define a list of up to 10 (MGCP) or 5 (MEGACO) legal IP addresses separated by a comma ',' or a semi-colon';' for the ServiceChange command. The gateway starts connecting with the first and in case of failure, attempts the others.	NULL	Legal IP Address
ProvisionedCallAgentsPorts	Use this parameter to define a list of up to 10 (MGCP) or 5 (MEGACO) Call Agent UDP ports separated by a comma ',' or a semi-colon ';' for each Call Agent defined by parameter used to specify Allowed Call Agent Address.	2944	0 to 65535
RandomizeTransactionID	Defines if the transactions produced by the device start with a fixed or random number. 1 = Randomize On Refer also to the parameters defining Transaction ID Range and Transaction ID	1	0 or 1

Table 3-89: Control Protocol Parameters - ALL

Parameter Name	Description	Default	Range
	Base.		
RedundantCallAgentDomainName	Defines the redundant MGCP Call Agent domain name.	' ' (empty string)	String[63]
RestartMaximumWaitingDelay	Defines the Maximum Waiting Delay (in msec) before restart service change when Media Gateway is powered on.	2500	> 0
RTCPInterval	Defines the time interval between the adjacent RTCP reports, in msec.	5000	0 to 65535
SingleSIDPacketWithSCEG729	When using a G.729 coder connection and SCE (Silence Suppression Enable) is On, a single SID packet is sent. If set to 1 and the channel was opened or modified to operate with the G.729 coder with Silence Suppression when Silence is detected, only a single SID packet is sent. If set to 0, SID packets are sent frequently, according to energy changes that require a SID packet for each change.	0	0 or 1
TargetMG_ResponseTime	The response time is defined as the time from the arrival of a call set-up request until the response, in msec	200	100 to 1000 in steps of 50
TransactionIDBase	Defines the minimum number for the transaction ID.	2000	> 0
TransactionIDRange	Defines the range for the transaction ID	999997999	> 0
TransparentCoderPayloadType	Alternative payload type used when using transparent coder.	116	0 to 127
USETransparentCoderWithHBR	If this parameter is set to 1 and the connection uses HBR (High Bit Rate) coders, the DTMF transport type is set to Transparent. Coders list: 0 = Do not use; 1 = Use G711Mulaw G726_32 G727_24_16 G727_32_24 G727_40_24 G726_16 G726_40 G727_24 G727_32 G727_40_32	0	0 or 1

Table 3-90: Control Protocol Parameters - IPM

Parameter Name	Description	Default	Range
ConferenceMaxSimultaneous Speakers	Defines the maximum number of users that can speak simultaneously in a conference.	3	1 to 3
ConferenceMaxUsers	Defines the maximal number of users to reserve for a new conference. The actual conference size can be more than this, but never less.	3	3 to 64
ConferenceSignalGeneration Enable	Generates a beep when a participant enters or exits the conference. 0 = Do not generate; 1 = Generate	1	0 or 1
CPConnectionStatistics	Used to enable/disable print of statistical information regarding a connection for digital devices. 0 = Disable; 1 = Enable	1	0 or 1
cpEndOfRecordCutTime	The max amount of audio, in msec, to cut from the end of a recording. This is used to remove the DTMF signals generated by the end user in terminating the record.	0	0 to 65535

Table 3-91: Control Protocol Parameters - MediaPack & Mediant 1000

Parameter Name	Description	Default	Range
CPPlayDigitalVMWI	Selects the method used for VMWI. 0 = Analog (high line voltage) 1 = Digital (play FSK signal as in caller ID)	0	0 or 1

Table 3-92: Control Protocol Parameters - TP

Parameter Name	Description	Default	Range
CPTrunkIdOffset	Sets the trunk numbering offset. CPTRUNKIDOFFSET_2 causes the first trunk number to be 2.	0	0, >0

TrunkName	MGCP This parameter generates (together with the parameter defining the Endpoint Prefix) a local endpoint name on trunk-enabled media gateways.
	MEGACO Prefix of the trunk part of the termination name.
		MGCP = ' ' (empty string)
		MEGACO = ' ' for analog devices
			String[19]

Table 3-92: Control Protocol Parameters - TP

Parameter Name	Description	Default	Range
		and 's' for trunking devices	

3.5.8 IPsec Parameters

Table 3-93: IP Security Parameters - ALL

Parameter Name	Description	Default	Range
IPsecDPDMode	IPsec Dead Peer Detection (RFC 3706) - Mode of Operation. One of the following values: '0' = Disabled (Default); '1' = Periodic; '2' = On demand	0	0, 1, 2
IPsecMode	Secure Internet Protocol (IPsec) Policy Mode of Operation (Transport or Tunneling): 0= Transport; 1= Tunneling	0	0 or 1
IPsecPolicyRemoteSubnetMask	Secure Internet Protocol (IPsec) Policy - Subnet Mask of the Remote IPSec Address.	255.255.255.255	Legal Subnet
IpsecPolicyRemoteTunnelIPAddress	Secure Internet Protocol (IPsec) Policy - IP Address of the Remote IPSec Tunnel Endpoint.	0.0.0.0	Legal IP address

3.5.9 NFS Parameters

The table below lists and describes the NFS parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values. Note that there is additional ***ini*** configuration required for each remote NFS file system that needs to be accessed. Refer to Table Parameters on page 218 for details.

Table 3-94: NFS Parameters

Parameter Name	Description	Default	Range
NFSBasePort	Start of the range of numbers used for local UDP ports used by the device for NFS sessions and for local TCP ports used by the device for HTTP sessions.	28000	0 to 65535

Table 3-94: NFS Parameters

Parameter Name	Description	Default	Range
	A value of 0 indicates that random ports (in the range 32768 to 65535) should be used. NOTE: When configuring this parameter: 1) the RTP port range should be avoided. 2) the random port range (32768 to 65535) should be avoided if using a non-zero value for this parameter.		

3.5.10 MGCP-Specific Parameters

The table below 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 3-95: MGCP Parameters - ALL

Parameter Name	Description	Default	Range
CallAgentIP	The Call Agent IP address, in dotted notation, to be used for the initial Restart in Progress (RSIP) message. Set to 0.0.0.0 to avoid sending RSIP. Parameter overrides the BootP server's Call Agent IP address, if provided.	NULL	Legal IP address
CallAgentPort	Defines the Call Agent port number. Defaults to the MGCP default port number of 2427.	2427	0 to 65534
ClearRequestBuffer	0 = only an empty R: clears the event list and only an empty S: clears and stops the current signals list. Signals and events will be cleared only when new signals/events are requested or an empty signals/events request is mentioned in the command. 1 = if an encapsulated identifier (X:) is present in the command, all TO signals and all events are cleared.	1	0 or 1
ConnectionIDBase	Defines the lowest number for the Connection ID values assigned by the media gateway.	20	> 0
ConnectionIDRange	Defines the range for the Connection ID values assigned by the gateway.	999999999	> 0

Table 3-95: MGCP Parameters - ALL

Parameter Name	Description	Default	Range
DepopulatedchannelsNumber	Enables the Call Agent to access only a subset of the on-blade channel bank. Set to (-1) to use all available channels.	-1	0 to the maximum number of channels in the device.
EnablePiggyBacking	This parameter configures the option to send piggy-backed commands while RSIPs are sent. For example, if the event is triggered by the device and an RSIP was not yet sent, the RSIP will be sent and piggy-back the event along with it. The call manager will get a combined message containing the RSIP and the event. 0 = commands sent by the gateway will not be piggy-backed. 1 = commands sent by the gateway will be piggy-backed.	1	0 or 1
GatewayMGCPPort	Users can use this parameter to force the media gateway to listen to another UDP port instead of to the original 2427, as defined in RFC 2705.	2427	0 to 65535
LongDurationEventTime	Defines the default time to trigger the long duration event (in seconds).	3600 sec	≥0
MGCPBufferingTimeout	Sets the timer for buffering digit after a digit map match was found and until new RQNT arrives. The timer is in sec.	0	0 or 1
MGCPDIGESTPASSWORD	Defines the MGCP Digest security password.	See Descr.	Up to 39 character s
MGCPDIGESTUSERNAME	Defines the MGCP Digest security user name.	See Descr.	Up to 39 character s
MGCPEndpointNamingPattern	Defines endpoint naming pattern for a gateway. The **'*'** signs are replaced with an actual endpoint number or with a wild-card sign. Default value is “ACgw*”.	"ACgw*"	Up to 63 character s
MGCPEndPointNumberingOffset	Enables users to add an offset to endpoints. Parameter functions only with Endpoint Naming configuration. Using this parameter with Trunk Naming configuration is disallowed. For Trunk Naming configuration, use the 'TrunkIdOffset' parameter.	0	> 0

Table 3-95: MGCP Parameters - ALL

Parameter Name	Description	Default	Range
MgcpFxoDiscPortsAlarm	Determines whether to send alarm on FXO ports that are disconnected from the PBX. If the alarm is sent, the port status will be 'forced'.	0	0 or 1
MGCPSendDigitmapMismatchNotification	The MGCP standard defines that if a number does not match the digitmap definition, a notification is not sent. Values: 1 = Send mismatch notification; a digital mismatch notification is sent 0 = Do NOT send mismatch notification; a digital mismatch notification is not sent. Similarly, MGCP can be enabled to send notifications upon matching digitmap.	0	0 or 1
MGCPSendMACWithRSIP	When this parameter exists in the **.ini* file, the generated RSIPs include the media gateway's / device's MAC address in addition to the regular parameters. This parameter is sent as an MGCP extension parameter. 1 = Include the MAC address of the media gateway / device; 0 = Don't include the MAC address of the media gateway / device	0	0 or 1
MGCPTrunkNamingPattern	Defines the trunk and B-channel naming pattern used by the gateway. The '*' signs will be replaced with a trunk or B-channel number or with a wild-card sign.	'ds*/tr*'	String [63]
MGCPUseAudioPortForT38	Defines that T.38 packets will be received on the RTP port.	0	0 or 1
MGCPVersion	Defines the MGCP protocol version.	MGCP 1.0	String[39]
MGCPXUAMAKE	Defines the make part of x-ua response according to RFC 3149. The maximum length of this parameter is 32 bytes. 0 = Disable; 1 = Enable	0	0 or 1
MGCPXUAMODEL	The model part of x-ua response according to RFC 3149. The maximum length of this parameter is 32 bytes. 0 = Disable; 1 = Enable	0	0 or 1

Table 3-95: MGCP Parameters - ALL

Parameter Name	Description	Default	Range
MGEOL	Sets the characters that constitute the EOL in the commands and responses generated by the device. String sets the characters that constitute the EOL in MGCP messages generated by device	See Descr.	See Descr.
MGHistoryBufferTimeLim	Defines the time (in seconds) that a transaction is kept in the history buffer.	30	≥ 0
QuarantineModeState	Sets the default quarantine handling state. When set, the quarantine handling state is set to Lockstep. If not set, it is set to Loop and Discard. 0 = Loop & Discard; 1 = Lockstep When enabled, the Quarantine events are handled according to RFC 2705. In non-quarantine modes, a Notification is sent immediately on event detection.	0	0 or 1
RedundantAgentIP	Defines the 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. Range = IP address in dotted notation	NULL	xxx.xxx.xxx.xxx
RedundantAgentPort	Defines the redundant Call Agent port number. Defaults to the MGCP default port number of 2427.	2427	0 to 65534
RSIPOnNetworkDisconnection	Specifies whether or not to send an RSIP when the LAN is re-connected. 0 = Don't send RSIP; 1 = Send RSIP	1	0 or 1
T38FALLBACKTRANSPORTMODE	Sets the Channel fax transport mode when its default fax transport mode is set to Relay(T.38) and the remote side has not reported T.38 capability in the SDP 'm' line 0 = Transparent 1 = Relay (T.38) 2 = ByPass 3 = Transparent with Events	0	0 to 3
UseBRacketsWithGatewayName	When the Gateway Name is defined as an empty string and this parameter is set to 1, the gateway name takes the device IP address with added brackets; e.g., [10.2.211.11]. 0 = Off; 1 = On	1	0 or 1
UseNewFormatCoderNegotiation	Disables the response of all coders (and descriptions) that are returned	1	0 or 1

Table 3-95: MGCP Parameters - ALL

Parameter Name	Description	Default	Range
	on execution of the CRCX (Create Connection command) or MDCX (Modify Connection command) without a coder and SDP (Session Description Protocol) included in the command. For detailed information, refer to Coder Negotiation in RFC 3136. 1 = Use the new format; 0 = Do not use.		
UseRangeEndpointsWithRSIP	While parameter is set to 1 (default). RSIPs will be sent in range format e.g. "RSIP 1234 ACGw[ep1-ep2]@AUDC.com". If parameter is set to 0, RSIP is sent to each endpoint. On trunking gateways RSIPs are not sent.	1	0 or 1
UseWildCardWithRSIP	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	1	0 or 1

Table 3-96: MGCP Parameters - MediaPack & Mediant 1000

Parameter Name	Description	Default	Range
MGCPActiveEndPoints	Defines a list of active endpoints, separated by commas. Use a hyphen to define the range of endpoints. For example: '1 3 5-7' means that endpoints 1, 3, 5, 6 and 7 are active. Functions only with Endpoint Naming configuration. With Trunk Naming configuration, the results are unexpected.	All endpoints are active	String[19]

Table 3-97: MGCP Parameters - TP

Parameter Name	Description	Default	Range
ActivateallChannelsOnBoardInit	Activates (1) or deactivates (0) all DSPs when the blade is initialized. Used in order to perform	0	0 or 1

Table 3-97: MGCP Parameters - TP

Parameter Name	Description	Default	Range
	signals/events operations prior to CRCX. 0 = Deactivate; 1 = Activate		

3.5.11 MEGACO-Specific Parameters

The table below 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 3-98: MEGACO Parameters - ALL

Parameter Name	Description	Default	Range
DigitMapName	Name of the provisioned digit map.	NULL	String[10]
DIGITMAPPING	The digit map patterns separated by a vertical bar (|), as defined in H.248.1 Section 7.1.14.	NULL	String[151]
EP_BIT_Field_Size	(For binary MEGACO) Defines the bit field size for each name level (level 0 is the left one, i.e., the Trunk number). The total binary name is 32 bits long.	0	0 to 30
EP_Num	Defines the starting number for each name level (level 0 is the left one when looking at the parameter defining Phys Term Name Pattern). Thus, to start trunk numbering from 1, set EP_NUM_0 to 1.	0	Any positive number
LogicalRTPTermPattern	Defines the name pattern of an RTP termination. For example: 'gw/rtp/*'. The '*****' sign stands for the actual number of the RTP termination.	NULL	String [30]
MEGACO_MID	Defines the media gateway's MID towards the MGCP/MEGACO Call Agent. If empty or illegal, the MID holds the IP address of the device.	NULL	String[64]
MEGACOASN1Profile	Used to profile the binary ASN.1 encoding. Refer to the product's User's Manual for possible values.	1	Integer >0
MEGACOCheckLegalityOfMGC	This parameter is specified if MEGACO rejects commands from a MGC not in the provisioned list. 1 = Reject, don't check; 0 = Check	0	0 or 1
MEGACOContextIDOffset	Offset for the context ID generated by the gateway. For example: offset = 100 causes the first context to be 101.	0	0 to 42949672 95 (0 to FFFFFFF

Table 3-98: MEGACO Parameters - ALL

Parameter Name	Description	Default	Range
			F)
MEGACOCOTTESTTYPE	The continuity test type (Set per trunk). One of the following values: 0 = THRH, 1 = THRL, 2 = TLRH	NULL	0, 1, 2
MEGACOEncoding	Sets the MEGACO coding method. 1 = Support MEGACO protocol's binary ASN.1 format 0 = Text mode	0	0 or 1
MEGACOHangTermTimeout	Default timeout (in seconds) for sending Hanging Termination event, when a request for Hanging Termination is sent without parameters.	0	0 -65535
MEGACOTerminationIDOffset	Offset for the ephemeral termination 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'.	0	0 to 4294967295 (0 to FFFFFFFF)
MegacoVersion	Determines the maximum MEGACO Version number that is supported by the device.	NULL	String[10]
MGCExecutionTime	Defines the estimated execution time of the MGC (in msec).	100	0 to 2000
MGCProvisionalResponseTime	Defines the provisional response timer for the MGC (in msec).	100	0 to 20000
MGExecutionTime	Defines the estimated execution time of the media gateway (in msec).	100	0 to 2000
MGProvisionalResponseTime	Defines the provisional response timer for the media gateway (in msec).	100	0 to 20000
PhysTermNamePattern	Defines the name pattern of a physical termination. For Example: 'tgw/t*/c*'. The '*' sign stands for the actual numbers of the trunk and B-channel.	NULL	String [30]
RTP_BIT_Field_Size	(For binary MEGACO) Defines bit field size for each RTP termination name.	0	0 to 30
RTP_Num	Defines the starting number for each name's RTP termination level (level 0 is the left one, i.e., the Trunk number).	0	Positive number

Table 3-99: MEGACO Parameters - IPM

Parameter Name	Description	Default	Range
AASPackagesProfile	Selects the profile for the AAS package specification standard. 0 = TD-51 standard 1 = H.248.9 standard 2 = MGCP Packet Cable 3 = SIP MSCML	0	0 to 3
AudioTermPattern	Defines the name pattern of an audio termination. **Applicable to IPM devices only.**	NULL	String[32]
BCTTermPattern	Defines the name pattern of a BCT termination.	NULL	String[32]
ConferenceTermPattern	Defines the name pattern of a conference termination. **Applicable to IPM devices only.**	NULL	String[32]
MEGACOProvisionedAudioSize	Defines the provisioned audio size indicated by parameter X-PtEngr. **Applicable to IPM devices only.**	60	1 to 65535
MEGACOProvisionedBCTSize	Provisioned BCT size indicated by parameter X-PtEngr. **Applicable to IPM devices only.**	60	1 to 65535
MEGACOProvisionedConfSize	Provisioned conference size indicated by parameter X-PtEngr. The value is dynamically limited according to the number of DSP channels and the used feature key. **Applicable to IPM devices only.**	60	1 to 65535
MEGACOProvisionedTrunkTestingSize	Defines the provisioned TT (trunk testing) size indicated by the parameter X-PtEngr. **Applicable to IPM devices only.**	60	1 to 65535
TrunkTestTermPattern	Defines the name pattern of a trunk test termination. **Applicable to IPM devices only.**	NULL	String[32]

Table 3-100: MEGACO Parameters - TP

Parameter Name	Description TP	Default	Range
ATM_Num	Defines the starting number for each ATM termination level name. **Applicable to TP-6310 only.**	0	Any positive number.
ATM_BIT_FIELD_SIZE	(For binary MEGACO) Defines the bit field size for each ATM termination name level.	0	0 to 30

Table 3-100: MEGACO Parameters - TP

Parameter Name	Description TP	Default	Range
	Applicable to TP-6310 only.		
LogicalATMTermPattern	Defines the name pattern of an ATM termination.	NULL	String [30]
MEGACOTrunkIDOffset	Sets the offset to the trunk numbering. e.g., Offset = 2 causes the first trunk number to be 2. Parameter was replaced by the parameter 'EP_NUM'.	0	0 to 4294967295

3.5.12 MRCP Parameters

Table 3-101: MRCP Parameters - IPM

Parameter Name	Description	Default	Range
MRCPDEFAULTMIMETYPE	Indicates the default format for speech recognition requests for inline grammars. 0 = GRXML should be used 1 = GSL (Nuance format) should be used. GRXML is a standard used by multiple vendors. But GSL is much more concise, and allows larger grammars due to the lack of verbiage required by GRXML.	0	0 or 1
MRCPENABLED	Enables/disables the MRCP functionality. 0 = Disable; 1 = Enable	0	0 or 1
MRCPMAXPORTS	The number of ports that are allocated to running MRCP related activities, such as speech recognition and text-to-speech. A port is considered duplex, so that speech recognition and text-to-speech can run on the same port. The value should not exceed the number of channels for the system.	10	0 to <Maximum_available_channels/2>
MRCPSERVERIP	Defines the IP address for the MRCP speech server. The IP address must be a valid IP address of the machine hosting the ASR/TTS speech server.	0.0.0.0	Valid IP address
MRCPSERVERNAME	Defines the hostname of the MRCP speech server. It is used to build a URI for the server. The name must be a valid alpha-numeric character string reflecting the hostname of the	NULL	alpha-numeric character string

Table 3-101: MRCP Parameters - IPM

Parameter Name	Description	Default	Range
	machine hosting the ASR/TTS speech server.		
MRCPServerPort	The control port on the MRCP speech server.	554	0 - 65535

3.5.13 Web Interface Parameters

The table below 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 3-102: Web Parameters - ALL

Parameter Name	Description	Default	Range
BKGImageFIleName	Changes an AudioCodes Web background image to the user background image, by loading a GIF/JPEG file. **Notes**: 1. Background height should be 85 pixels. 2. Background image is duplicated alongside to fit the screen width.	NULL	String[47]
DenyAuthenticationTimer	Defines the time the next authentication attempt from the last authentication failed IP should be denied.	0	Elapsed time in sec
EnableRADIUS	Enable/disable the RADIUS (Remote Authentication Dial-In User Server/Service). 0 = RADIUS application is disabled 1 = RADIUS application is enabled	0	0 to 1
HTTPPort	Determines the local HTTP port of the device.	80	1 to 65535
HTTPSCertFileName	Defines the name of the HTTPS server certificate file to be downloaded via TFTP. The file must be in base64-encoded PEM format.	NULL	String[47]
HTTPSCipherString	Defines the Cipher string for HTTPS (in OpenSSL cipher list format). Refer to URL http://www.openssl.org/docs/apps/ciphers.html	EXP:RC4	See Descr.
HTTPSOnly	Use this parameter to allow only HTTPS connections (force security). When set to 1, unencrypted HTTP (normally, port 80) is blocked.	0	0-1

Table 3-102: Web Parameters - ALL

Parameter Name	Description	Default	Range
HTTPSPORT	Determine the local Secure HTTPS port of the device. Restrictions may apply in the range	Port 443	1 to 65535
HTTPSRequireClientCertificate	Requires client certificates for HTTPS connection. The client certificate must be preloaded on the gateway, and its matching private key must be installed on the managing computer. Time and date must be correctly set on the gateway, for the client certificate to be verified. Enable = 0, Disable = 1	0	0 or 1
HTTPSRootFileName	Defines the name of the HTTPS trusted root certificate file to be downloaded via TFTP. The file must be in base64-encoded PEM format.	NULL	String[47]
LogoFileName	GIF/JPEG image file name to replace the AudioCodes Web logo image appearing in the upper left hand corner of the Device Web interface pages. (Note: Image height should be 30 pixels.)	NULL	String[47]
LogoWidth	Defines the logo's image width in pixels as it exists in the Web home page. Maximum allowed width value = 200. If a larger value was entered (or any other illegal value, e.g., a negative value), the width will be set to its default.	141 pixels	0 to 200 pixels
RADIUSAuthPort	RADIUS authentication port. Predefined UDP port using for authentication with the RADIUS server.	1645	Any integer
RADIUSAuthServerIP	Use this parameter to define the RADIUS (Remote Authentication Dial-In User Server/Service) authentication server IP address.	0	W, X, Y, Z
RADIUSDoubleDecodeURL	When enabled, the Web server will perform an additional decoding operation to authentication credentials sent by the user via URL to the RADIUS server (in addition to Percentage-Encoding - RFC 3986 specifications). Enable = 1, Disable = 0	0	0 or 1
RADIUSAuthPort	RADIUS authentication port. Predefined UDP port used for authentication with the RADIUS Server.	1645	Any integer
RADIUSRetransmission	RADIUS packets retransmission. Number of retransmissions for the	3	1 to 10

Table 3-102: Web Parameters - ALL

Parameter Name	Description	Default	Range
	same request.		
RADIUSTo	RADIUS Response Time Out. Time to wait for the response before a retransmission is needed.	10 seconds	1 to 30 seconds.
RadiusVSAAccessAttribute	Defines the 'Security Access Level' attribute code in the VSA section of the Radius packet that the device should relate to.	35	0 to 0xFF in a bitwise format
RadiusVSAVendorID	Defines the vendor ID that the device should accept when parsing a Radius response packet.	5003 (AudioCodes)	0 to 0xFFFFFFFF
ScenarioFileName	The name of the scenario file (including preconfigured Web screens) which can be loaded using TFTP or created using the Web interface.	NULL	0 to 47 characters.
SharedSecret	Shared 'secret' between client/server used for the PAP authentication protocol. 'Secret' is used to authenticate the gateway to the RADIUS server. It should be a cryptographically strong password.	FutureRADIUS	0 to 47 characters.
UseProductName	Activates the userProductName parameter. 1 = On = Enables the userProductName string to override any AudioCodes defaults. 0 = Off = userProductName string will have no effect on the product name.	0	0 or 1
UseRProductName	A string of characters to replace the default AudioCodes product name appearing in the upper right hand corner of the device Web interface pages.	NULL	String[29]
UseWeblogo	Enables the webLogoText string to override any loaded logo image file. 1 = Enables the webLogoText string to override any loaded logo image file (and AudioCodes default logo image). 0 = The webLogoText string will have no effect on the logo image.	0	0 or 1
WEBACCESSLIST	Allows IP addresses to connect to the Web interface. Set to zeroes to allow all IP addresses.	0.0.0.0	Valid IP address
WebAuthMode	Selects HTTP basic (clear text) or digest (MD5) authentication for the Web interface. 0, basic authentication (clear text) 1, digest authentication (MD5)	0	0 to 2

Table 3-102: Web Parameters - ALL

Parameter Name	Description	Default	Range
	2, digest authentication (MD5) used for HTTP, while basic authentication used for HTTPS. Note that turning on RADIUS login forces basic authentication.		
WebDebugLevel	Sets the output level of Web debug messages sent by the Gateway. 0 = Deny; 1 = Show	0	0, 1
WebLogoText	Replaces the default AudioCodes logo image, appearing in the upper left hand corner of the device Web interface pages, with a text string. Note: This string also replaces the AudioCodes name in the title bar.	NULL	String[15]
WEBRADIUSLOGIN	Uses the RADIUS (Remote Authentication Dial-In User Server/Service) for Web interface authentication. Make sure that ENABLERADIUS is on. Use of this parameter without HTTPSONLY = 1 is not recommended. Disable = 0, Enable = 1	0	0, 1

3.5.14 SNMP Parameters

The table below lists and describes the SNMP parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 3-103: SNMP Parameters - ALL

Parameter Name	Description	Default	Range
acUserInputAlarmDescription	Defines the Description of the input alarm.	User's Input-Alarm raised.	0 to 100 characters.
acUserInputAlarmSeverity	Defines the severity of the input alarm.	MIB_no_alarm_E	Warning, minor, major, critical, none
AlarmHistoryTableMaxSize	Determines the maximum number of rows in the Alarm History table. The parameter is controllable via the Config Global Entry Limit MIB (located in the Notification Log MIB).	ALARM_HISTORY_DEFAULT_SIZE	MP-1xx: 50 to 100 for all other Devices 50 to 1000
ChassisPhysicalAlias	This object is an 'alias' name for the physical entity as specified by	NULL	String[255]

Table 3-103: SNMP Parameters - ALL

Parameter Name	Description	Default	Range
	a network manager, and provides a non-volatile 'handle' for the physical entity.		
DisableSNMP	Enables or disables SNMP. 0 = Enable; 1 = Disable	0	0 or 1
EnableSNMPTraps2TPNCPEvents	Enables the module that converts traps into TPNCP events. Possible values: 0 = Disable 1 = Enable	0	0 or 1
ifAlias	The textual name of the interface. The value is equal to ifAlias SNMP MIB object.	NULL	String[64]
KeepAliveTrapPort	The port to which the keep-alive traps are sent.	162	0 to 65534
SendKeepAliveTrap	When Enabled, this parameter invokes the keep-alive trap and sends it out every 9/10 of the time defined in the parameter defining NAT Binding Default Timeout. 0 = Disable; 1 = Enable	0	0 or 1
SetCommunityString	User-determined community string with access limited to **.ini* file entered values only. Parameter is singular version of the readWriteCommunityStrings table, and corresponds to readWriteCommunityStrings_0.	NULL	String[19]
SNMPManagerIsUsed	Enables a row in the SNMP Managers table.0 = Disable; 1 = Enable	0	0, 1
SNMPManagerTableIP	Define the SNMP manager server IP address. This is the tabular version of parameter defining SNMP Manager IP.	0	String[15]
SNMPManagerTrapPort	Sets the trap ports to be used by the different managers. Tabular version of parameter defining SNMP Trap Port.	162	100 to 65534
SNMPManagerTrapSendingEnable	Enables the SNMP Manager's IP address for traps to be sent to it. 0 = Disable; 1 = Enable	1	0 or 1
SNMPManagerTrapUser	This parameter can be set to the name of any configured SNMPV3 user to associate with this trap destination. This determines the	An empty string	0 to 33 characters.

Table 3-103: SNMP Parameters - ALL

Parameter Name	Description	Default	Range
	trap format, authentication level and encryption level. By default, the trap is associated with the SNMP trap community string.		
SNMPPort	This parameter specifies the port number for SNMP requests and responses. Generally, it isn't specified and the default is used.	161	100 to 65534
SNMPREADONLYCOMMUNITY STRING	Used to define a read-only community string.	“public”	String[19]
SNMPREADWRITECOMMUNITYS TRING	Used to define a read-write community string.	“private”	String[19]
SNMPSysOid	Used to define the base product system OID via the **.ini* file.	AudioCodes' root OID 1.3.6.1.4.1.5003.8.1.1	String[100]
SNMPTRAPCOMMUNITY STRING	Defines the community string used in traps.	“trapuser”	String[19]
SNMPTrapEnterpriseOid	Used to define a Trap Enterprise OID . Inner shift of trap in AcTrap sub tree is added to end of the OID from **.ini* file.	AudioCodes' Trap root OID: 1.3.6.1.4.1 .5003.9.10.1.21"	String[100]
SNMPTrapManagerHostName	Defines a FQDN of a remote host that is used as an SNMP Manager. The resolved IP address replaces the last entry in the trap manager table (defined by the parameter 'SNMPManagerTableIP_x') and the last trap manager entry of snmpTargetAddrTable in the snmpTargetMIB; e.g.: 'mngr.corp.mycompany.com'.	NULL	String [99]
SNMPTRUSTEDMGR	Defines the IP address of a trusted SNMP manager.	0.0.0.0	String[15]

Table 3-104: SNMP Parameters - MediaPack & Mediant 1000

Parameter Name	Description	Default	Range
ChassisPhysical AssetID	This object is a user-assigned asset tracking identifier for the Mediant 1000	NULL	String[255] Mediant 1000 only.

Table 3-104: SNMP Parameters - MediaPack & Mediant 1000

Parameter Name	Description	Default	Range
	chassis as specified by an EMS, and provides non-volatile storage of this information. For Mediant 1000 only.		

3.5.15 Voice Streaming Parameters (IPmedia 3000 only)

Note: The following **Voice Streaming Parameters** are only applicable to IPmedia 3000.

The table below lists and describes the Voice Streaming parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 3-105: Voice Streaming Parameters

Parameter Name	Description	Default	Range
EnableVoiceStreaming	Enables/disables HTTP and NFS voice streaming. When enabled, the module requires some system resources, such as tasks and memory allocation. 0 = Disable; 1 = Enable	0	0 or 1
NFSClientMaxRetransmission	Since NFS is carried over UDP, transmission is performed for messages with no response. This parameter enables the user to control the maximum number of retransmissions performed for such a command. By default, the parameter is not used and the number of retransmissions is derived from ServerRespondTimeout.	0 (derived from ServerRespondTimeout).	1 to 100
ServerRespondTimeout	Defines the maximum time in milliseconds, that the blade should wait for a response when working with a remote server. This relates to both to HTTP and NFS commands.	5000	1000 to 20000
StreamingCacheSize	Determines the number of megabytes (out of the 32-MB local memory) used for the cache mechanism. Caching uses a portion of the 32 MB of the resident local memory area to cache the remotely played files. The remaining memory	0	See Descr.

Table 3-105: Voice Streaming Parameters

Parameter Name	Description	Default	Range
	is used for saving and playing local IVR (e.g., voice prompts). When this parameter is greater than 0, the mechanism is automatically enabled and active. **Note: This parameter is applicable only to IPmedia 3000/IPM-8410/IPM-6310.**		
StreamingCacheRefreshTime	Determines the dynamic caching refresh rate (in minutes). This refresh timeout avoids the scenario in which files played from the cache have been updated (changed) on the server. At every refresh, the files that saved on the cache memory are 're-fetched' from their remote servers. **Note: This parameter is applicable only to IPmedia 3000/IPM-8410/IPM-6310.**	-1	-1 to 0xFFFF
StreamingCacheNumOf Descriptors	Determines the number of files that the cache can handle.	5000	0 to 10000
StreamingCacheDecisionInterval	Determines the cache mechanism's decision interval (in minutes).	4	-1 to 65535
StreamingPlayingUnderRun Timeout	Defines the maximum time in milliseconds, that the blade is willing to wait for the streaming server to acknowledge data sent to it. An under run condition is defined as one where the blade is not supplying the DSPs with enough data, "starving" the DSPs. Under runs relate to playing data from a server to our blade where due to environment conditions (usually network problems), that data is not passed quickly enough. This condition will result with damaged data being passed to the user. The streaming level will abort the session containing consecutive under runs as derived from this timer. A user may set the timer to longer periods then the default value thus enabling the blade to be more tolerant to under run conditions.	5000	100 to 10000
StreamingRecordingOverRun	Defines the maximum time in	5000	100 to

Table 3-105: Voice Streaming Parameters

Parameter Name	Description	Default	Range
Timeout	milliseconds, that the streaming server is willing to wait to acknowledge a data request sent from the blade. An overrun condition is defined as one where the blade is sending data to the server but is not receiving a response from the server, acknowledging it received the data. Overruns relate to recording data to a remote server and will result with “holes” in the recording. The streaming level will abort the session containing the consecutive overruns as derived from this timer. A user may set the timer to longer periods than the default value, thereby enabling the blade to be more tolerant to overrun conditions.		10000
VoiceStreamUploadMethod	Defines the HTTP request type for uploading the voice stream to the file server. 0 = POST; 1 = PUT	0	0 or 1
VoiceStreamUploadPostUri	Defines the URI used on the POST request, to upload voice data from the media server to a Web server.	-	-

3.5.16 SCTP Parameters

Note: The following parameters are **not** applicable to MediaPack.

Table 3-106: SCTP Parameters - All Digital Devices

Parameter Name	Description	Default	Range
SCTPAssociationsNum	Defines the maximum number of Stream Control Transmission Protocol (SCTP) associations that can be opened.	3	1 to 8
SCTPChecksumMethod	Stream Control Transmission Protocol (SCTP) uses a checksum mechanism in order to authenticate packets on both sides (the receiving side and the transmitting side). Currently, two checksum mechanisms	0	0 or 1

Table 3-106: SCTP Parameters - All Digital Devices

Parameter Name	Description	Default	Range
	are available: 0 = adler32 checksum mechanism 1 = crc32c checksum mechanism (improved mechanism)		
SCTPDNetNum	Defines the maximum number of association transport addresses that can be active.	3	1 to 3
SCTPHBInterval	Defines the SCTP heartbeat interval.	30	1 to 3600
SCTPHOSTNAME	When this parameter is set to any value other than an empty string, SCTP (Stream Control Transmission Protocol) uses the value as the value of the FQDN (Fully Qualified Domain Name) parameter attached to the INIT chunk. In this case, the FQDN parameter replaces any IP address parameters in the INIT chunk. This feature enables overcoming NAT problems where the original IP addresses belonging to the endpoint supports are converted into pseudo addresses. When this parameter is not set (default), the INIT chunk is sent without any FQDN parameter.	NULL	String[42]
SCTPISTRMNum	Defines the maximum number of incoming streams.	10	1 to 200
SCTPMaxAssocInitAttempts	Defines the maximum number of SCTP association initialization attempts.	5000	5 to 10000
SCTPMaxAssocRet	Defines the maximum number of SCTP association retransmission attempts.	10	5 to 20
SCTPMaxDataChunkSize	Defines the maximum length of SCTP data chunks.	500	50 to 1504
SCTPOSTRMNum	Defines the maximum number of outgoing streams.	10	1 to 200
SCTPOutChunksNum	Defines the maximum number of outgoing chunks.	630	50 to 630
SCTPPortsNum	Defines the maximum number of SCTP endpoints that can be opened.	5	1 to 5
SCTPT4SAckTimer	Defines the SCTP T4 SACK timer interval.	3	1 to 5

3.5.17 Advanced Audio Server Parameters

Note: The following parameters are **not** applicable to MediaPack.

The table below lists and describes the Advanced Audio Server parameters contained in the ***ini*** file. Use this table as a reference when modifying ***ini*** file parameter values.

Table 3-107: Advanced Audio Server Parameters

Parameter Name	Description	Default	Range
AMSAllowURLAsAlias	Indicates if play requests for remote URLs should first be checked for local segments with the same alias as the URL. "0" = system will not check local audio first for play requests using URLs. "1" = system will try to find the URL locally first, and if the audio is not found locally, the system will try to play the audio from the remote URL.	1	0 or 1
AMSEnableBundleBurning	Indicates whether VP and XML files (audio bundle) are allowed to be burnt to the device. 0 = Enable 1 = Disable **Parameter only applies to 8410 blade.**	0	See Descr.
AMSForceRepositoryUpdateEnabled	Indicates that a new repository (consisting of VP and XML files) should always be uploaded to the board regardless of whether signals are still being played on the old repository. 0 - Disable 1 - Enable	0	0 or 1
AMSPrimaryLanguage	Defines Primary Language for AMS	NULL	String[3] - language ISO string
AMSProfile	Defines the AMS Profile. The value is a bit mask with these fields: 1 = APSAudioEnable 2 = TrunksInterfaceDisabled 4 = ExtendedDefaultPrtBufferSize Refer to the enumerator acAMSProfileBitMask for values.	0	Integer ≥ 0
AMSSecondaryLanguage	Defines the Secondary Language for the AMS.	NULL	String[3] - language ISO string
APSEnabled	Indicates if an APS will be providing audio for this system, or if the system should access audio in the vp.dat file. 1 = APS will provide audio. 0 = system	1	0 or 1

Table 3-107: Advanced Audio Server Parameters

Parameter Name	Description	Default	Range
	should use vp.dat.		
AudioStagingAutoSwitchoverEnabled	Indicates if audio should automatically be activated after downloading. 0 = audio not automatically activated. 1 = audio is automatically activated. In a Trunking Gateway environment, this parameter should be set to 0 or 1.	1	0 or 1
RTSPConnectionRetryInterval	Time, in seconds, the system waits before trying to create a socket to the RTSP-based speech server if the socket was never created or was created and then brought down.	10	0 to 65535
RTSPENABLED	Enable/disable the RTSP functionality. 0 = Disable; 1 = Enable	0	0 or 1
RTSPMAXPORTS	Indicates how many channels can be simultaneously active in RTSP sessions. Should not exceed number of available channels for device.	10	0 to <maximum_available_channels/2>
RtspServiceDebugMode	Determines whether to enable RTSP debug messages. These messages contain debug information in reference to the RTSP from the point of creation. 0 - Disable 1 - Enable	0	0 or 1

3.5.18 Video Parameters

Note: The following parameters are only applicable to **IPM-8410** and **IPmedia 3000**.

Table 3-108: Video Parameters - Conference

Parameter Name	Description	Default	Range
VideoConferenceParticipantLayout	Participant's view layout. 1 user 2 x 2 users 3 x 3 users	1	0 to 2
VideoConferenceParticipantTriggerMode	Participant trigger mode defines the trigger for modifying the participant's view. 0 - Voice Activated Video Switch - A participant's view is changed according	0	0, 1 or 3

Table 3-108: Video Parameters - Conference

Parameter Name	Description	Default	Range
	to the active speakers' detection. 1 - Timer - Participant's view is rotated every configured time among different video participants in conference. 3 - Fixed - A participant's view is fixed.		
VideoConferenceParticipantType	Indicates a participant's capabilities in the conference. Regular (2) = Regular participant which is able to both send and receive video.	2	2
VideoConferenceSwitchingInterval	Applicable when using timer trigger switching mode. Determines the video view rotation interval in seconds.	20	0 to 65535
VideoEnableActiveSpeakerHighlight	Highlight the speaker for participants viewing a symmetric layout (2x2 or 3x3) 0 = Don't highlight active speaker. 1 = Highlight active speaker.	0	0 or 1
VideoEnableConferenceParticipantSelfView	Self view indication. 0 = Participant cannot view her/him self. 1 = Participant can view her/him self.	0	0 or 1

3.6 The ini File Table Parameters

3.6.1 SS7 ini File Table Parameters

SS7 Table Parameters

- SS7 Signaling Node Timers Table Parameters on page 222
- SS7 Signaling LinkSet Timers Table Parameters on page 220
- SS7 MTP2 Table Parameters on page 221
- SS7 Signaling Link Table Parameters on page 224
- SS7 Signaling LinkSets Table Parameters on page 227
- SS7 Signaling LinkSet-Links Table Parameters on page 227
- SS7 RouteSets Table Parameters on page 228
- SS7 RouteSet-Routes Table Parameters on page 229
- Static Routing Context Table
- SigTran Interface Groups Table Parameters on page 230
- SigTran Interface IDs Table Parameters on page 232
- SS7 MTP3 Redundancy SN Table Parameter

NFS Server Table Parameters

- NFS Servers Table Parameters on page 235

T3 Configuration Table Parameters

- DS3 Configuration Table Parameters on page 233

The following ***ini*** file Table Parameters are provided:

3.6.1.1 SS7 Signaling Node Timers Table Parameters

Table 3-109: SS7 Signaling Node Timers Table Parameters

ini **File Field Name**	**Description**	**Default**	**Range**
SS7_SNTIMERS_INDEX	Index Field for line	0	0 to (MTP3_SN_TIMER_SETS-1)
SS7_SNTIMERS_NAME	String name for SN timer-set	"SN_Timers”	
SS7_SNTIMERS_T6	Delay to avoid message mis-sequencing on controlled rerouting	1200	500 to 4294967295
SS7_SNTIMERS_T8	Transfer prohibited inhibition timer (transient solution)	1200	500 to 4294967295
SS7_SNTIMERS_T10	Waiting to repeat signaling route set test message	60000	500 to 4294967295
SS7_SNTIMERS_T11	Transfer restricted timer	90000	500 to 4294967295
SS7_SNTIMERS_T15	Waiting to start signaling route set congestion test	3000	500 to 4294967295
SS7_SNTIMERS_T16	Waiting for route set congestion status update	2000	500 to 4294967295
SS7_SNTIMERS_T18_ITU	Timer within a signaling point whose MTP restarts for supervising link and link set activation as well as the receipt of routing information	20000	500 to 4294967295
SS7_SNTIMERS_T19_ITU	Supervision timer during MTP restart to avoid possible ping-pong of TFP, TFR and TRA messages	67000	500 to 4294967295
SS7_SNTIMERS_T20_ITU	Overall MTP restart timer at the signaling point whose MTP restarts	60000	500 to 4294967295
SS7_SNTIMERS_T21_ITU	Overall MTP restart timer at a signaling point adjacent to one whose MTP restarts	65000	500 to 4294967295
SS7_SNTIMERS_T24_ITU	Stabilizing timer after removal of local processor outage, used in LPO latching to RPO (national option)	500	500 to 4294967295

Table 3-109: SS7 Signaling Node Timers Table Parameters

ini File Field Name	Description	Default	Range
SS7_SNTIMERS_T22_ANSI	Timer at restarting SP waiting for signaling links to become available	180000	500 to 4294967295
SS7_SNTIMERS_T23_ANSI	Timer at restarting SP, started after T22, waiting to receive all traffic restart allowed messages	180000	500 to 4294967295
SS7_SNTIMERS_T24_ANSI	Timer at restarting SP with transfer function, started after T23, waiting to broadcast all traffic restart allowed messages	5000	500 to 4294967295
SS7_SNTIMERS_T25_ANSI	Timer at SP adjacent to restarting SP waiting for traffic restart allowed message	30000	500 to 4294967295
SS7_SNTIMERS_T26_ANSI	Timer at restarting SP waiting to repeat traffic restart waiting message	12000	500 to 4294967295
SS7_SNTIMERS_T28_ANSI	Timer at SP adjacent to restarting SP waiting for traffic restart waiting message	3000	500 to 4294967295
SS7_SNTIMERS_T29_ANSI	Timer started when TRA sent in response to unexpected TRA or TRW	60000	500 to 4294967295
SS7_SNTIMERS_T30_ANSI	Timer to limit sending of TFPs and TFRs in response to unexpected TRA or TRW	30000	500 to 4294967295

3.6.1.2 SS7 Signaling LinkSet Timers Table Parameters

Table 3-110: SS7 Signaling LinkSet Timers Table Parameters

Parameter Name	Description	Default	Range
SS7_LKSETTIMERS_INDEX	Index Field for line	0	0 to (MTP3_LKSET_TIMER_SETS-1)
SS7_LKSETTIMERS_NAME	String name for SN timer-set	"SN_Timers”	
SS7_LKSETTIMERS_T2SLT	Interval timer for sending signaling link test messages	30000	500 to 4294967295
SS7_LKSETTIMERS_T1	Delay to avoid message mis-sequencing on changeover	1000	500 to 4294967295

Table 3-110: SS7 Signaling LinkSet Timers Table Parameters

Parameter Name	Description	Default	Range
SS7_LKSETTIMERS_T2	Waiting for changeover acknowledgement	2000	500 to 4294967295
SS7_LKSETTIMERS_T3	Time controlled diversion-delay to avoid mis-sequencing on changeback	1200	500 to 4294967295
SS7_LKSETTIMERS_T4	Waiting for changeback acknowledgement (first attempt)	1200	500 to 4294967295
SS7_LKSETTIMERS_T5	Waiting for changeback acknowledgement (second attempt)	1200	500 to 4294967295
SS7_LKSETTIMERS_T7	Waiting for signaling data link connection acknowledgement	2000	500 to 4294967295
SS7_LKSETTIMERS_T12	Waiting for uninhibit acknowledgement	1200	500 to 4294967295
SS7_LKSETTIMERS_T13	Waiting for force uninhibit	1300	500 to 4294967295
SS7_LKSETTIMERS_T14	Waiting for inhibition acknowledgement	3000	500 to 4294967295
SS7_LKSETTIMERS_T17	Delay to avoid oscillation of initial alignment failure and link restart	1500	500 to 4294967295
SS7_LKSETTIMERS_T22_ITU	Local inhibit ITU test timer	180000	500 to 4294967295
SS7_LKSETTIMERS_T23_ITU	Remote inhibit ITU test timer	180000	500 to 4294967295
SS7_LKSETTIMERS_T20_ANSI	Local inhibit ANSI test timer	90000	500 to 4294967295
SS7_LKSETTIMERS_T21_ANSI	Remote inhibit ANSI test timer	90000	500 to 4294967295

3.6.1.3 SS7 MTP2 Table Parameters

Table 3-111: MTP2 Table Parameters

Parameter Name	Description	Default	Range
SS7Mtp2Parms_LinkRate	SS7 SLI link rate: 'A' for 64K, 'D' for 56K.(0 is equivalent to ‘A’).	'A'	'0', 'A','D'
SS7Mtp2Parms_ErrorCorrectionMethod	SLI error correction method: 'B' for basic, 'P' for PCR. 0 is equivalent to 'B'.	'B'	'0', 'B' ,'P'

Table 3-111: MTP2 Table Parameters

Parameter Name	Description	Default	Range
SS7Mtp2Parms_IacCp	SS7 SLI - number of aborted proving attempts.	5	0-10
SS7Mtp2Parms_SuermT	Signal Unit error rate monitor: T threshold (0 to 256).	64	0-256
SS7Mtp2Parms_AermTin	Alignment error rate monitor.	4	0-20
SS7Mtp2Parms_AermTie	Alignment error rate monitor.	1	0-10
SS7Mtp2Parms_SuermSuD	SS7 Signal Unit error rate monitor: D threshold (0 to 256).	256	0-256
SS7Mtp2Parms_OctetCounting	Octet counting N octets (number of Octets received in octet counting mode).	16	0-256
SS7Mtp2Parms_LssuLength	LSSU length in bytes.	1	1-2
SS7Mtp2Parms_PcrN2	Number of message signal unit octets available for re-transmission.	200	0-512
SS7Mtp2Parms_T1	Timer T1: Timer 'alignment ready'.	50000	0-100000
SS7Mtp2Parms_T2	Timer T2: Timer 'not aligned'.	150000	0-200000
SS7Mtp2Parms_T3	Timer T3: Timer 'aligned'	2000	0-20000
SS7Mtp2Parms_T4n	Timer T4: Proving period timer normal period.	8200	0-15000
SS7Mtp2Parms_T4e	Timer T4: Proving period timer emergency period.	500	0-5000
SS7Mtp2Parms_T5	Timer T5: Timer 'sending SIB'.	120	0-2400
SS7Mtp2Parms_T6	Timer T6: Timer 'remote congestion'	6000	0-10000
SS7Mtp2Parms_T7	Timer T7: Timer 'excessive delay of acknowledgment'	2000	0-5000

3.6.1.4 SS7 Signaling Nodes Table Parameters

Table 3-112: SS7 Signaling Nodes Table Parameters

Parameter Name	Description	Default	Range
SS7_SN_INDEX	Index Field for line	0	0 to (MAX_SN_PER_CARD-1)

Table 3-112: SS7 Signaling Nodes Table Parameters

Parameter Name	Description	Default	Range
SS7_SN_NAME	String name for SN	"SN”	
SS7_SN_TRACE_LEVEL	Trace level of signaling node (level 3)	0	0 or 1
SS7_SN_ADMINISTRATIVE_ STATE	Administrative state of signaling node 0 = L3_OFFLINE 2 = L3_INSERVICE	L3_OFFLIN E	0 or 2
SS7_SN_VARIANT	Variant of signaling node 1 = NET_VARIANT_ITU 2 = NET_VARIANT_ANSI 3 = NET_VARIANT_CHINA	NET_VARI ANT_ITU	0 to 3
SS7_SN_NI	Network Indicator of signaling node 0 = INTERNATIONAL 1 = INTERNATIONAL_SPARE 2 = NATIONAL 3 = NATIONAL_SPARE	NET_INDIC ATOR_INT ERNATION AL	0 to 3
SS7_SN_SP_STP	Routing function of signaling node 0 = SP 1 = STP	SN_FUNC TION_IS_S P	0 to 1
SS7_SN_TFC	Currently not supported	0	0 or 1
SS7_SN_OPC	Origination (local) point-code of signaling node	0	0 to 4294967295
SS7_SN_ROUTESET_CONGES TION_ WINSIZE	RouteSet Congestion Size (messages) of signaling node	8	0 to 255
SS7_SN_TIMERS_INDEX	Index of SNTimers tables used for this signaling node	0	0 to (MTP3_SN_ TIMER_SET S-1)
SS7_SN_ISUP_APP	Level 4 application that handles ISUP traffic for this signaling node 0 = NIL 4 = UAL	MTP3_NIL _APP	0 or 4
SS7_SN_SCCP_APP	Level 4 application that handles SCCP traffic for this signaling node 0 = NIL 4 = UAL	MTP3_NIL _APP	0 or 4

Table 3-112: SS7 Signaling Nodes Table Parameters

Parameter Name	Description	Default	Range
SS7_SN_BISUP_APP	Level 4 application that handles BISUP traffic for this signaling node 0 = NIL 4 = UAL	MTP3_NIL_APP	0 or 4
SS7_SN_ALCAP_APP	Level 4 application that handles ALCAP traffic for this signaling node 0 = NIL	MTP3_NIL_APP	0,4 or 5
SS7_SN_BICC_APP	Level 4 application handling BICC traffic for this signaling node. 0 = MTP3_NIL_APP; 1 = ISUP 2 = SCCP; 4 = UAL	MTP3_NIL_APP(0)	0, 1, 2, 4

Table 3-113: SS7 Signaling Link Table Parameters

Parameter Name	Description	Default	Range
SS7MonSUTypeFilter	Defines the SU messages to be filtered at the device or passed to the host. 0 = No SU is passed to the Host. 1 = Filters LSSU and FISU messages (passes only MSUs to the host). 2 = Filters FISU messages (passes Message Signal Unit (MSUs) and LSSUs to the host) 3 = None (default) When no filtering is configured (i.e., 3), all messages are passed to the host (MSU, LSSU, and FISU). **Note**: For SS7 filtering, you need to enable SS7 monitoring for both Layer 2 and 3 link types (i.e., SS7_SubLink_L3_Type and SS7_SubLink_L2_Type must be set to 5 and 4 respectively).	3	0 to 3
SS7_LINK_INDEX	Index Field for line	0	0 to (MAX_SIGNALING_LINKS_PER_CARD-1)

Table 3-113: SS7 Signaling Link Table Parameters

Parameter Name	Description	Default	Range
SS7_LINK_NAME	String name for Link Params	"LINK"	
SS7_LINK_RDCY_BOARD	Device number in which the link is physically connected	0	0 to 2
SS7_LINK_ADMINISTRATIVE_STATE	Administrative state of signaling link 0 = L3_OFFLINE 2 = L3_INSERVICE	L3_OFFLINE	0 or 2
SS7_LINK_TRACE_LEVEL	Trace level of signaling link (level 2)	0	0 or 1
SS7_LINK_L2_TYPE	Link layer type - defines level 2 media of signaling link 0 = SS7_SUBLINK_L2_TYPE_NONE (default) 1 = SS7_SUBLINK_L2_TYPE_MTP2 2 = SS7_SUBLINK_L2_TYPE_M2UA_MGC 3 = SS7_SUBLINK_L2_TYPE_SAAL 4 = SS7_SUBLINK_L2_TYPE_MONITORING	SS7_SUBLINK_L2_TYPE_NONE	0 to 4
SS7_LINK_L3_TYPE	Link high layer type - defines level 3 or L2 high layer of signaling link 0 = SS7_SUBLINK_L3_TYPE_NONE (default) 1 = SS7_SUBLINK_L3_TYPE_M2UA_SG 2 = SS7_SUBLINK_L3_TYPE_MTP3 3 = SS7_SUBLINK_L3_TYPE_MTP2_TUNNELING 4 = SS7_SUBLINK_L3_TYPE_MTP2Oip 5 = SS7_SUBLINK_L3_TYPE_MONITORING	SS7_SUBLINK_L3_TYPE_NONE	0 to 5
SS7_LINK_TRUNK_NUMBER	Trunk number of signaling link (TDM)	0	0 to MAX_TRUNK_CAPACITY - 1
SS7_LINK_TIMESLOT_NUMBER	Time-Slot number of signaling link (TDM)	16	0 to 31
SS7_LINK_LAYER2_VARIANT	Variant (layer 2) of signaling link (TDM) 1 = NET_VARIANT_ITU 2 = NET_VARIANT_ANSI 3 = NET_VARIANT_CHINA	NET_VARIANT_ITU	1 to 3
SS7_LINK_MTP2_ATTRIBUTES	MTP2 attributes of signaling link (TDM)	3	0 to MAX_C7_MTP2_PARAMS_INDE

Table 3-113: SS7 Signaling Link Table Parameters

Parameter Name	Description	Default	Range
			X
SS7_CONGESTION_LOW_MARK	Link congestion low mark of signaling link (TDM)	5	0 to 255
SS7_CONGESTION_HIGH_MARK	Link congestion high mark of signaling link (TDM)	20	0 to 255
SS7_LINK_M2UA_IF_ID	Interface ID of signaling link	0	0 to 4294967295
SS7_LINK_GROUP_ID	Group ID of signaling link	0	0 to 0xFFFF
SS7_LINK_TNL_MGC_LINK_NUMBER	MTP2 Tunneling: MGC link number (MTP2 \other side\ of signaling link	0	0 toMAX_SIGNALING_LINKS_PER_CARD-1
SS7_LINK_TNL_ALIGNMENT_MODE	MTP2 Tunneling: Alignment mode of signaling links in tunnel 0 = M3B_ALIGNMENT_NORMAL 1 = M3B_ALIGNMENT_EMERGENCY	M3B_ALIGNMENT_EMERGENCY	0 to 255
SS7_LINK_TNL_CONGESTION_MODE	MTP2 Tunneling: Congestion mode of signaling links in tunnel 0 = M3B_CONGESTION_ACCEPT 1 = M3B_CONGESTION_DISCARD	M3B_CONGESTION_ACCEPT	0 to 255
SS7_LINK_TNL_WAIT_START_COMPLETE_TIMER	MTP2 Tunneling Timer: wait start complete	30000	500 to 4294967295
SS7_LINK_TNL_OOS_START_DELAY_TIMER	MTP2 Tunneling Timer: OOS start delay	5000	500 to 4294967295
SS7_LINK_TNL_WAIT_OTHER_SIDE_INSV_TIMER	MTP2 Tunneling Timer: wait other side inservice	30000	500 to 4294967295

3.6.1.5 SS7 Signaling LinkSets Table Parameters

Table 3-114: SS7 Signaling LinkSets Table Parameters

Parameter Name	Description	Default	Range
SS7_LINKSET_SN_INDEX	First Index Field for line	0	0 to (MAX_SN_PER_CARD-1)
SS7_LINKSET_LINKSET_INDEX	Second Index Field for line	0	0 to (MAX_LINKSETS_PER_SN-1)
SS7_LINKSET_NAME	String name for LinkSet Params	"LINKSET"	
SS7_LINKSET_ADMINISTRATIVE_STATE	Administrative state of signaling LinkSet 0 = L3_OFFLINE 2 = L3_INSERVICE	L3_OFFLINE	0 or 2
SS7_LINKSET_DPC	Destination Point-Code of signaling LinkSet	0	Unsigned Integer
SS7_LINKSET_MASK	Mask for links within signaling LinkSet	15	0 to 255
SS7_LINKSET_ ALTERNATE_MASK	Alternate mask for links within signaling LinkSet	240	0 to 255
SS7_LINKSET_TIMERS_ INDEX	Timers Index of signaling LinkSet	0	0 to (MTP3_LINKSET_TIMER_SETS-1)

Table 3-115: SS7 Signaling LinkSet-Links Table Parameters

Parameter Name	Description	Default	Range
SS7_LINKSETLINK_SN_INDEX	First Index Field for line: Signaling Node Number	0	0 to (MAX_SN_PER_CARD-1)
SS7_LINKSETLINK_LINKSET_INDEX	Second Index Field for line: Signaling LinkSet Number	0	0 to (MAX_LINKSETS_PER_SN-1)

Table 3-115: SS7 Signaling LinkSet-Links Table Parameters

Parameter Name	Description	Default	Range
SS7_LINKSETLINK_INNER_LINK_INDEX	Third Index Field for line: Inner Link Index in Signaling LinkSet	0	0 to (MAX_LINKS_PER_LINKSET-1)
SS7_LINKSETLINK_LINK_NUMBER	Physical number of signaling link which is part of the LinkSet	MTP3_LINK_NIL	0 to MAX_SIGNALING_LINKS_PER_CARD-1
SS7_LINKSETLINK_LINK_SLC	"Signaling Link Code" of signaling link which is part of the LinkSet	0	0 to MTP3_MAX_SLC

Table 3-116: SS7 RouteSets Table Parameters

ini File Field Name	Description	Default Value	Valid Range
SS7_ROUTESET_SN_INDEX	First Index Field for line: Signaling Node Number	0	0 to (MAX_SN_PER_CARD-1)
SS7_ROUTESET_INDEX	Second Index Field for line: Signaling RouteSet Number	0	0 to (MAX_ROUTESETS_PER_SN-1)
SS7_ROUTESET_NAME	String name for RouteSet Params	"ROUTESET"	
SS7_ROUTESET_ADMINISTRATIVE_STATE	Administrative state of signaling RouteSet 0 = L3_OFFLINE 2 = L3_INSERVICE	L3_OFFLINE	0 or 23
SS7_ROUTESET_DPC	Destination Point-Code of signaling RouteSet	0	
SS7_ROUTESET_MASK	Mask for routes within signaling RouteSet	15	0 to 255

3.6.1.6 SS7 RouteSet-Routes Table Parameters

Table 3-117: SS7 RouteSet-Routes Table Parameters

ini File Field Name	Description	Default Value	Valid Range
SS7_ROUTESETROUTE_SN_INDEX	First Index Field for line: Signaling Node Number	0	0 to (MAX_SN_PER_CARD-1)
SS7_ROUTESETROUTE_ROUTESET_INDEX	Second Index Field for line: Signaling RouteSet Number	0	0 to (MAX_ROUTESETS_PER_SN-1)
SS7_ROUTESETROUTE_INNER_ROUTE_INDEX	Third Index Field for line: Inner Route Index in Signaling RouteSet	0	0 to (MAX_LINKSETS_PER_ROUTESET-1)
SS7_ROUTESETROUTE_LINKSET_NUMBER	Number of signaling LinkSet which is part of the RouteSet	MTP3_LINKSET_NIL	0 to MAX_LINKSETS_PER_SN-1
SS7_ROUTESETROUTE_PRIORITY	Priority of route within RouteSet	0	0 to 254

Table 3-118: Routing Context Table Parameters

ini File Field Name	Description	Default Value	Valid Range
SS7_RC_INDEX	First Index Field for line: Routing Context Index	0	0 to 15
SS7_RC_INNER_INDEX	Second Index Field for line: Routing Context Inner Index	0	0 to 3
SS7_RC_SN_INDEX	This parameter is used to specify the M3UA Routing Context DPC SN-Index.	0	0 to 1
SS7_RC_OPC1	This parameter is used to specify the first element in M3UA Routing Context OPC List.	-1	-1, 0 to 0xFFFFFF
SS7_RC_OPC2	This parameter is used to specify the second element in M3UA Routing Context OPC List.	-1	-1, 0 to 0xFFFFFF
SS7_RC_OPC3	This parameter is used to specify the third element in M3UA Routing Context OPC List.	-1	-1, 0 to 0xFFFFFF

Table 3-118: Routing Context Table Parameters

ini File Field Name	Description	Default Value	Valid Range
SS7_RC_OPC4	This parameter is used to specify the fourth element in M3UA Routing Context OPC List.	-1	-1, 0 to 0xFFFFFF
SS7_RC_SI1	This parameter is used to specify the first element in M3UA Routing Context SI List	-1	-1, 0 to 15
SS7_RC_SI2	This parameter is used to specify the second element in M3UA Routing Context SI List	-1	-1, 0 to 15
SS7_RC_SI3	This parameter is used to specify the third element in M3UA Routing Context SI List	-1	-1, 0 to 15
SS7_RC_SI4	This parameter is used to specify the fourth element in M3UA Routing Context SI List.	-1	-1, 0 to 15

Table 3-119: SigTran Interface Groups Table Parameters

ini File Field Name	Description	Default Value	Valid Range
SS7_SIG_IF_GR_INDEX	Index Field for line	0	0 to 7
SS7_IF_GR_ID	SigTran group id	65534	0 to 65535
SS7_SIG_SG_MGC	UAL group function	83	77(MGC), 83(SG), 1 (NAT)

SS7_SIG_LAYER	SigTran group layer:
	no_layer = 0,
	iua = 1,
	m2ua = 2,
	m3ua = 3,
	m2tunnel = 4,
	dua = 6
		0	0, 1, 2, 3, 4, 6

SS7_SIG_TRAF_MODE	SigTran group traffic mode.	1	1 to 3
SS7_SIG_T_REC	SigTran group T recovery	2000	0 to 10000000
SS7_SIG_T_ACK	SigTran group T Acknowledge	2000	0 to 10000000
SS7_SIG_T_HB	SigTran group T Heartbeat	2000	0 to 10000000

Table 3-119: SigTran Interface Groups Table Parameters

ini File Field Name	Description	Default Value	Valid Range
SS7_SIG_MIN_ASP	SigTran group minimal ASP number	1	1 to 10
SS7_SIG_BEHAVIOUR	SigTran group Behavior bit field	0	0 to 4294967294
SS7_SCTP_INSTANCE	SigTran group SCTP instance	65534	0 to 65534
SS7_LOCAL_SCTP_PORT	SigTran group local SCTP port	65534	0 to 65534
SS7_SIG_NETWORK	SigTran group Network (ITU,ANSI,CHINA)	1	1 to 3
SS7_DEST_SCTP_PORT	SigTran group destination SCTP port	65534	0 to 65534
SS7_DEST_IP	SigTran group destination IP Address (Valid only for MGC or NAT application)	0	0 to 4294967294
SS7_MGC_MX_IN_STREAM	SigTran max number of SCTP inbound streams	2	2 to 65534
SS7_MGC_NUM_OUT_STREAM	SigTran max number of SCTP outbound streams	2	2 to 65534
RdcyBoardNum	Device number in which the group is physically connected	0	0 to 2
SS7_SIG_ RC_INDEX	This parameter indicates the entry in Routing Context table, that contains the routing context rules for this group. This is a mandatory parameter.	-1	0 to 15
SS7_SIG_RC_VALUE	This parameter indicates the Routing Context value for this Application Server. -1: No value for Routing Context	-1	-1, 0 to 2147483647
SS7_SIG_NETWORK_APPEARANCE	This parameter indicates the Network Appearance value for this Application Server. -1: No value for Network Appearance	-1	-1, 0 to 2147483647

Table 3-120: SigTran Interface IDs Table Parameters

ini File Field Name	Description	Default Value	Valid Range
SS7_SIG_IF_ID_INDEX	Index Field for line	0	0 to 15
SS7_SIG_IF_ID_VALUE	SigTran interface Id value field	0	0 to 4294967294
SS7_SIG_IF_ID_NAME	SigTran interface Id string name	"INT_ID"	--
SS7_SIG_IF_ID_OWNER_GROUP	SigTran interface Id owner group field	0	0 to 65534

SS7_SIG_IF_ID_LAYER	SigTran interface Id layer :
	no_layer = 0,
	iua = 1,
	m2ua = 2,
	m2tunnel = 4,
	dua = 6
		0	0, 1, 2, 4, 6

SS7_SIG_IF_ID_NAI	SigTran interface Id NAI field: NAI is physical link number.	65534	0 to 65534

Table 3-121: SS7 MTP3 Redundancy SN Table Parameters

ini File Field Name	Description	Default Value	Valid Range
SS7_RDCYSN_BOARD_INDEX	First Index Field for line: Blade Number	0	0 to (MAX_MTP3_RDCY_BOARDS-1)
SS7_RDCYSN_SN_INDEX	Second Index Field for line: SN number in the blade	0	0 to (MAX_SN_PER_CARD-1)
SS7_RDCYSN_BOARD_IP	Defines the IP address of the blade	0	0 to 0xFFFFFFFF
SS7_RDCYSN_OPC	Define the local point-code of a shared signaling node	0	
SS7_RDCYSN_ALCAP_OPC	Define the local point-code of ALCAP instance on a given blade	0	

3.6.2 DS3 Configuration Table Parameters

Note: 'T3' and 'DS3' are terms used interchangeably. This section is only applicable to the **TP**-6310 and Mediant 3000 devices.

Table 3-122: DS3 Configuration Table Parameters

Parameter Name	Description	Default	Range
DS3CONFIG_FramingMethod	Used to select the physical DS3 framing method for the interface. **Applicable only to TP-6310/T3, IPM-6310/T3, Mediant 3000/T3 and IPmedia 3000/T3.** 0 = M23 framing 1 = C Bit Parity	0	0 or 1
DS3CONFIG_Clock Source	Selects the DS3 Clock mode blade for the interface. **Applicable only to the TP-6310/DS3, IPM-6310/T3, Mediant 3000/T3 and IPmedia 3000/T3.** 0 = DS3Clock is recovered from the line 1 = DS3 trunk clock source is provided by the device's internal clock	0	0 to 1
DS3CONFIG_LineBuildOut	Used to select the DS3 line build out. **Applicable only to the TP-6310/DS3, IPM-6310/T3, Mediant 3000/T3 and IPmedia 3000/T3.** 2 = Level 1 3 = Level 2 4 = Level 3 5 = Level 4 6 = Level 5 7 = Level 6	4	2 to 7
DS3CONFIG_CircuitIdentifier	Interface SNMP-name represented by a string of up to 15 characters	Empty string	string
DS3CONFIG_TrapEnable	Enables or disables SNMP traps for DS3 1 = Enable 0 = Disable	1	0 or 1

Table 3-122: DS3 Configuration Table Parameters

Parameter Name	Description	Default	Range
DS3CONFIG_PmOnOff	Enables or disables DS3 performance monitoring 1 = Enable 0 = Disable	1	0 or 1
DS3CONFIG_TappingEnable	Enables or disables special DS3 Tapping mode. In this mode the interface is capable of receiving a signal with additional 14dB attenuation to the standard maximum length. 1 = Enable 0 = Disable	0	0 or 1
DS3CONFIG_AdminState	Selects the DS3 interface Administrative Status. When the Administrative Status is set to "down", all 28 underlying DS1 interfaces are unavailable. 1 = Administrative status Up 0 = Administrative status Down (currently not supported)	1	0 or 1

3.6.3 Example of DS3 INI file Selection :

```
[ DS3CONFIG ]
;FramingMethod = DS3 M23(0=default) DS3 CBIT PARITY(1)
;PSTNDS3ClockSource = EXTERNAL(0) LOCAL BOARD(1=default)
;LineBuildOut = LEVEL 1(2) LEVEL 2(3)  LEVEL 3(4=default)
; LEVEL 4(5) LEVEL 5(6) LEVEL 6(7)
FORMAT DS3CONFIG Index = DS3CONFIG FramingMethod,
DS3CONFIG ClockSource, DS3CONFIG LineBuildOut ;

DS3CONFIG 0 = 0, 0, 4 ;
DS3CONFIG 1 = 0, 0, 4 ;
DS3CONFIG 2 = 0, 0, 4 ;
[ \DS3CONFIG ]
```

In this example, the line with the “FORMAT” expression defines a sequence of parameters for each T3 interface, which is not to be changed.

For each T3 interface a line “DS3CONFIG…” defines parameter values.

For example, for the first interface the configuration is set by the following expression:

```
DS3CONFIG 0 = 0, 0, 1 ;
```

This means that the interface should be configured with framing method M23, using External clock source and line built-out below 225 feet.

3.6.4 DSP Template Mix Table

The DSP template mix enables working with a combination of two DSP templates (i.e. Template-Mix) in a single device. The DSP templates' values & capabilities are specified in the device's Release Notes document.

The maximum number of templates allowed at once is 2.

The "DSPVersionTemplateNumber" ***ini*** file parameter is ignored when using the parameters specified in the following table.

Table 3-123: DSP Template Table

Parameter Name	Description	Default	Range
DspTemplates_DspTemplateNumber	Selects the DSP load number. Each load has a different coder list, a different channel capacity and different features supported.	0	0 to 255
DspTemplates_DspResourcesPercentage	Sets the distribution ratio of the selected template on the device's DSPs.	0	0 to 100

Example:

```
[DspTemplates]
FORMAT DspTemplates Index = DspTemplates DspTemplateNumber,
DspTemplates DspResourcesPercentage;
DspTemplates 0 = 1, 50;
DspTemplates 1 = 2, 50;
[\DspTemplates]
```

In this example, DSP template 1 will be loaded to 50% of the DSPs, and DSP template 2 will be loaded to the remaining 50%.

3.6.5 NFS Servers Table Parameters

This table defines the attributes to use when accessing remote NFS file systems. Note that one NFS file server can share multiple file systems. There should be a separate line in this table for each file system.

Table 3-124: NFS Servers Table Parameters

Parameter Name	Description	Default	Range
NFSServers_Index	Table row index.	N/A	0 to 4

Table 3-124: NFS Servers Table Parameters

Parameter Name	Description	Default	Range
NFSServers_HostOrIP	The domain name or IP address of the NFS server. If a domain name is provided, then a DNS server must be configured.	None	See description
NFSServers_RootPath	The path to the root of the exported file system.	None	string
NFSServers_Nfs Version	The NFS version to use in accessing this remote NFS file system.	3	2 or 3
NFSServers_Auth Type	Identifies the authentication method to use in accessing this remote NFS file system: 0 = AUTH_NULL 1 = AUTH_UNIX	1	0 to 1
NFSServers_UID	The numerical User ID (UID) to be used for authentication if AUTH_UNIX (1) is selected.	0	0 to 65537
NFSServers_GID	The numerical Group ID (GID) to be used in authentication if AUTH_UNIX is selected.	1	0 to 65537
NFSServers_VLAN Type	The VLAN identifier to use when accessing this remote NFS file system. This parameters applies only if multiple IP addresses are configured on this device. 0 = OAMP 1 = Media	1	0 to 1

4 Network Configuration

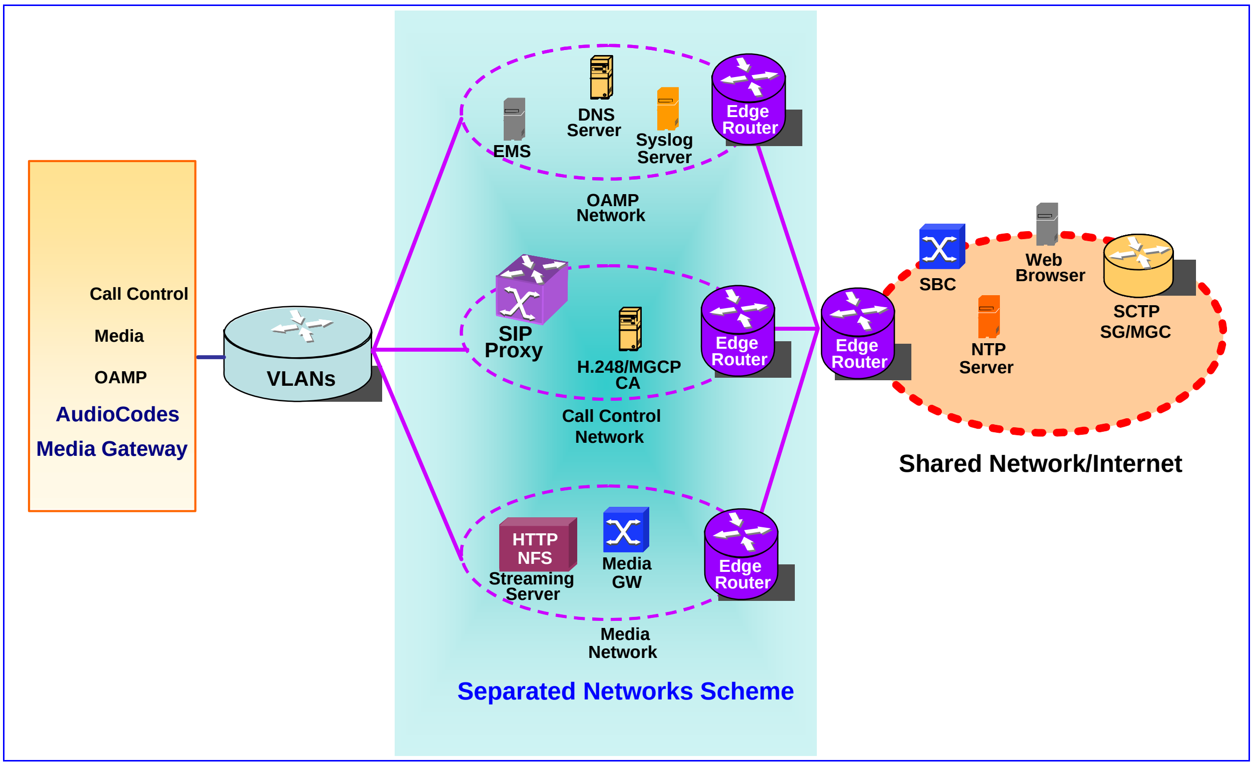
The device allows you to configure up to 16 different IP addresses with associated VLANs, via an Interface Table. Complementing the Interface Table is a configurable Routing Table, allowing the user to define routing rules for non-local hosts/subnets.

This chapter describes the various network configuration options.

4.1 Multiple Network Interfaces and Virtual LANs

A need often arises to have logically separated network segments for various applications (for administrative & security reasons). This can be achieved by employing Layer 2 VLANs & Layer 3 subnets.

Figure 4-1: Multiple Network Interfaces



This figure depicts a typical configuration featuring an AudioCodes Gateway.

The gateway is configured with three network interfaces for:

- Operations, Administration, Maintenance, and Provisioning (OAMP) applications
- Call Control applications
- Media

It is connected to a VLAN aware switch, which is used for directing traffic from (and to) the AudioCodes gateway, to three separated Layer 3 broadcast domains according to VLAN tags (middle pane).

The Multiple Interfaces scheme allows the configuration of up to 16 different IP Addresses, each associated with a unique VLAN ID.

The configuration is performed using the Interface Table, which is configurable via ini file, Web & SNMP interfaces.

4.1.1 Interface Table Overview

The Multiple Interfaces scheme allows the user to define up to 16 different IP Addresses and VLANs in a table format, as shown below.

Table 4-1: Multiple Interface Table

Index	Allowed Application Types	IPv6 Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	0	0	10.31.174.50	16	0.0.0.0	4	ManagementIF
1	2	0	10.32.174.50	16	0.0.0.0	5	ControlIF
2	1	0	10.33.174.50	16	10.33.0.1	6	Media1IF
3	1	0	10.34.174.50	16	0.0.0.0	7	Media2IF
4	1	0	10.35.174.50	16	0.0.0.0	8	Media3IF
5	1	0	10.36.174.50	16	0.0.0.0	9	Media4IF
6	1	0	10.37.174.50	16	0.0.0.0	10	Media5IF
7	1	0	10.38.174.50	16	0.0.0.0	11	Media6IF
8	1	0	10.39.174.50	16	0.0.0.0	12	Media7IF
9	1	0	10.40.174.50	16	0.0.0.0	13	Media8IF
10	1	0	10.41.174.50	16	0.0.0.0	14	Media9IF
11	1	0	10.42.174.50	16	0.0.0.0	15	Media10IF
12	1	0	10.43.174.50	16	0.0.0.0	16	Media11IF
13	1	0	10.44.174.50	16	0.0.0.0	17	Media12IF
14	1	0	10.45.174.50	16	0.0.0.0	18	Media13IF
15	1	0	10.46.174.50	16	0.0.0.0	19	Media14IF

Complementing the network configuration are some VLAN related parameters, determining if VLANs are enabled and the ‘Native’ VLAN ID (refer to the Enabling VLANs & Native VLAN ID sub-sections below) as well as VLAN priorities and DiffServ values for the supported Classes Of Service (refer to Quality of Service Section below).

4.1.1.1 The Interface Table Columns

Each row of the table defines a logical IP Interface, with its own IP Address, Subnet Mask (represented by Prefix Length), VLAN ID (if VLANs are enabled), Name, and application types that are allowed on this interface. One of the interfaces may have a ‘default gateway’ definition. Traffic destined to a subnet which does not meet any of the routing rules (either Local or Static routes) will be forwarded to this gateway (as long this application type is allowed on this interface). Refer to The Gateway Column sub-section below for more details.

4.1.1.2 The Index Column

This column holds the index of each interface. Possible values are 0 to 15. Each interface index must be unique.

4.1.1.3 The Allowed Application Types Column

This column defines the types of applications that are allowed on this interface. The applications are:

- OAMP – Operations, Administration, Maintenance and Provisioning. Examples of OAMP applications include: Web, Telnet, SSH, SNMP.
- CONTROL – Call Control Protocols. Examples of Control applications include: SIP, MGCP, MEGACO.
- MEDIA – RTP streams of Voice/Video.
- Any combination of the above mentioned types.

The following table shows the possible values of this column and their descriptions:

Table 4-2: Allowed Application Types Descriptions

Column Value	Description
0	OAMP: only OAMP applications will be allowed on this interface.
1	MEDIA: only Media (RTP) will be allowed on this interface.
2	CONTROL: only Call Control applications will be allowed on this interface.
3	OAMP & MEDIA: Only OAMP and Media (RTP) applications will be allowed on this interface.
4	OAMP & CONTROL: Only OAMP and Call Control applications will be allowed on this interface.
5	MEDIA & CONTROL: Only Media (RTP) and Call Control applications will be allowed on this interface.
6	ALL: All of the applications will be allowed on this interface

For valid configuration guidelines, refer to the Interface Table Configuration Guidelines sub-section for more information.

4.1.1.4 The IPv6 Interface Mode Column

The IPv6 Interface Mode column will be used for future IPv6 interfaces support. For now, all interfaces must have the value 0, assigned in this column.

4.1.1.5 The IP Address and Prefix Length Columns

These columns allow the user to configure an IPv4 IP Address and its related subnet mask.

The “Prefix Length” column holds the Classless Inter-Domain Routing (CIDR)-style presentation of a dotted decimal subnet notation. The CIDR-style presentation is the latest refinement to the way IP Address are interpreted. Specifically, instead of using eight-bit address blocks, it uses the variable-length subnet masking technique to allow allocation on arbitrary-length prefixes (Refer to http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing for more information).

This column lists the number of ‘1’ bits in the subnet mask. So, a subnet mask of 255.0.0.0 will be represented by a prefix length of 8 (11111111 00000000 00000000 00000000), and a subnet mask of 255.255.255.252 will be represented by a prefix length of 30 (11111111 11111111 11111111 11111100).

Each interface MUST have its own address space. Two interfaces may not share the same address space, or even part of it. The IP address should be configured as a dotted decimal notation. Prefix length values range from 0 to 31.

OAMP Interface Address when Booting using BOOTP/DHCP

When booting using BOOTP/DHCP protocols, an IP address is obtained from the server. This address is used as the OAMP address for this session, overriding the address configured via the Interface Table. The address specified in the Interface Table will be available when booting from Flash again.

This enables it to work with a temporary address for initial management and configuration purposes while retaining the address to be used for deployment.

4.1.1.6 The Gateway Column

This column defines a default gateway for the system. For this reason, only one ‘gateway’ field in the “Gateway” column may be configured. The Gateway Address provided in the “Gateway” column MUST be on the same subnet as the interface address.

The gateway can only be configured on one of the interfaces running Media traffic.

This column configures only the default gateway of the module. However, a separate routing table allows configuring routing rules. Refer to the Routing Table sub-section for more details.

Note: This configured default gateway in the example below (200.200.85.1) will only be available for applications configured on that interface (Media & Control). Outgoing management traffic on interface 0 will never have this gateway as its next hop.

Table 4-3: Configured Default Gateway Example

Index	Allowed Application Types	IPv6 Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	0	0	192.168.85.14	16	0.0.0.0	1	Mgmt
1	5	0	200.200.85.14	24	200.200.85.1	200	CntrlMedia

However, a separate routing table allows configuring routing rules. Configuring the following routing rule, will enable traffic running on interface 0 to access peers on subnet 17.17.0.0 via gateway 192.168.0.1.

Table 4-4: Separate Routing Table Example

Destination	Prefix Length	Subnet Mask	Gateway	Interface	Hop Count
17.17.0.0	16		192.168.0.1	0	1

Refer to the Routing Table sub-section for more details.

4.1.1.7 The VLAN ID Column

This column defines the VLAN ID for each interface. When using VLANs this column MUST hold a unique value for each interface.

4.1.1.8 The Interface Name Column

This column allows the configuration of a short string (up to 16 characters) to name this interface. This name will show in management interfaces (Web, CLI and SNMP) for better readability and has no functional use. This column must have a unique value for each interface (no two interfaces can have the same name) and must not be left blank.

4.1.2 Other Related Parameters

The Interface Table allows the user to configure interfaces and their related parameters, such as their VLAN ID or the interface name. The following sections list more parameters complementing the Interface Table functionality.

4.1.2.1 Enabling VLANs

The Interface Table column “VLAN ID” assigns a VLAN ID to each of the interfaces. Incoming traffic tagged with this VLAN ID will be channeled to the related interface, and outgoing traffic from that interface will be tagged with this VLAN ID.

When VLANs are required, the parameter should be set to 1. Refer to Setting Up Your System on page 249. The default value for this parameter is 0 (disabled).

4.1.2.2 ’Native’ VLAN ID

A ‘Native’ VLAN ID is the VLAN ID that untagged incoming traffic will be assigned to. Outgoing packets sent to this VLAN will be sent only with a priority tag (VLAN ID = 0).

When the ‘Native’ VLAN ID is equal to one of the VLAN IDs in the Interface Table (and VLANs are enabled), untagged incoming traffic will be considered as an incoming traffic for that interface. Outgoing traffic sent from this interface will be sent with the priority tag (tagged with VLAN ID = 0).

When the ‘Native’ VLAN ID is different from any value in the “VLAN ID” column in the Interface Table, untagged incoming traffic will be discarded, and all the outgoing traffic will be fully tagged.

The ‘Native’ VLAN ID is configurable by configuring the VlanNativeVlanId parameter (refer to the Setting up your System sub-section below).

The default value of the ‘Native’ VLAN ID is 1.

Note: If the 'VlanNativeVlanId' is not set (default value = 1), but one of the interfaces has a VLAN ID configured to 1, this interface will still be related to the 'Native' VLAN.

If you do not wish to have a 'Native' VLAN ID, and want to use VLAN ID 1, set the 'VlanNativeVlanId' parameter to a value other than any VLAN ID in the table.

4.1.2.3 Quality of Service Parameters

The system allows you to specify values for Layer-2 and Layer-3 priorities by assigning values to the following service classes:

- Network Service class – network control traffic (ICMP, ARP,)
- Premium Media service class – used for RTP Media traffic
- Premium Control Service class – used for Call Control traffic
- Gold Service class – used for streaming applications
- Bronze Service class – used for OAMP applications

The Layer-2 Quality Of Service parameters enables setting the values for the 3 priority bits in the VLAN tag of frames related to a specific service class (meeting IEEE 802.1p standard).

The Layer-3 Quality Of Service parameters enables setting the values of the DiffServ field in the IP Header of the frames related to a specific service class. The following Quality Of Service parameters can be set:

Table 4-5: Quality of Service Parameters

Parameter Name	Default Value	Description
Layer 2 Class Of Service Parameter (VLAN tag priority field):		
VlanNetworkServiceClassPriority	7	Sets the priority for the Network service class content
VLANPremiumServiceClassMediaPriority	6	Sets the priority for the Premium service class content and media traffic
VLANPremiumServiceClassControlPriority	6	Sets the priority for the Premium service class content and control traffic
VLANGoldServiceClassPriority	4	Sets the priority for the Gold service class content
VLANBronzeServiceClassPriority	2	Sets the priority for the Bronze service class content
Layer 3 Class Of Service Parameters (TOS/DiffServ):		
NetworkServiceClassDiffServ	48	Sets the DiffServ for the Network service class content
PremiumServiceClassMediaDiffServ	46	Sets the DiffServ for the Premium service class content and media

Table 4-5: Quality of Service Parameters

Parameter Name	Default Value	Description
		traffic
PremiumServiceClassControlDiffServ	40	Sets the DiffServ for the Premium service class content and control traffic
GoldServiceClassDiffServ	26	Sets the DiffServ for the Gold service class content
BronzeServiceClassDiffServ	10	Sets the DiffServ for the Bronze service class content

4.1.2.4 Selecting the Application Type

Some applications can be associated with different application types in different setups. These application types are configurable. The applications listed below can be configured to one of two application types:

- DNS
- SCTP traffic
- NTP
- TPNCP

Table 4-6: Application Type Parameters

Parameter Name	Description	Default	Values
EnableDNSasOAM	When this parameter is set to 1, the DNS application will be considered as an OAMP application. If it is set to 0, the DNS application will be considered as a CONTROL application. The DNS application will only operate on interfaces with the matching “Allowed Application Types” column.	1	1 = OAMP 0 = CONTROL
EnableSCTPasControl	When this parameter is set to 1, SCTP traffic will be considered as CONTROL traffic. If it is set to 0, SCTP traffic will be considered as an OAMP traffic. The SCTP transport protocol will only be available on interfaces with the matching “Allowed Application Types” column.	1	1 = CONTROL 0 = OAMP

EnableNTPasOAM	When this parameter is set to 1, the NTP application will be considered as an OAMP application. If it is set to 0, the NTP application will be considered as a CONTROL application. The NTP application will only operate on interfaces with the matching “Allowed Application Types” column.	1	1 = OAMP 0 = CONTROL
EnableTPNCPasOAM	When this parameter is set to 1, TPNCP traffic will be considered as an OAMP application. If it is set to 0, TPNCP based applications will be considered as CONTROL applications. TPNCP based applications will only operate on interfaces with the matching “Allowed Application Types” column.	1	1 = OAMP 0 = CONTROL

4.1.3 Interface Table Configuration Summary & Guidelines

Interface Table configurations must adhere to the following rules:

- Up to 16 different interfaces may be defined.
- The indices used must be in the range between 0 to 15.
- Each interface must have its own subnet. Defining two interfaces with addresses in the same subnet (i.e. two interfaces with 192.168.0.1/16 and 192.168.100.1/16) is illegal.
- Subnets in different interfaces must not be overlapping in any way (i.e. defining two interfaces with 10.0.0.1/8 and 10.50.10.1/24 is illegal). Each Interface MUST have its own address space.
- The Prefix Length replaces the dotted decimal Subnet Mask presentation. This column must have the value of 0-31 for IPv4 interfaces.
- Only one IPv4 interface with OAMP “Allowed Application Types” must be configured. Only one IPv4 interface with CONTROL “Allowed Application Types” must be configured. At least one interface with MEDIA “Allowed Application Types” must be configured. These application types may be mixed (i.e. OAMP and CONTROL). Here are some examples for interface configuration:
  - One interface with “Allowed Application Types” – ALL.
  - One interface with “Allowed Application Types” – OAMP, one other interface with “Allowed Application Types” – CONTROL, and one or more interfaces with “Allowed Application Types” – MEDIA.
  - One interface with “Allowed Application Types” – OAMP & MEDIA, one other interface with “Allowed Application Types” – MEDIA & CONTROL.
  - Other configurations are also possible while keeping to the above-mentioned rule.

- Only one interface may have a Gateway definition. This Gateway address MUST be in the same subnet as this interface; Other Routing Rules may be specified in the Routing Table (Refer to section 1.2 Routing Table for more details).
- Apart from the interface having the default gateway defined, the Gateway column for all other interfaces must be set to “0.0.0.0”.
- The Interface Name column may have up to 16 characters. This column allows the user to name each interface with an easier name to associate the interface with. Although used for better readability of the Interface Table, this column must have a unique value to each interface and must not be left blank.
- For all interfaces, the “IPv6 Interface Mode” column must be set to 0.
- When defining more than one interface, VLANs should be enabled (the VlanMode should be set to 1).
- VLANs will only be available when booting the module from Flash. When booting using BootP/DHCP protocols, VLANs will be disabled to allow easier maintenance access.
- The ‘Native’ VLAN ID may be defined using the ‘VlanNativeVlanId’ parameter. This will relate untagged incoming traffic as if reached with a specified VLAN ID. Outgoing traffic from the interface which VLAN ID equals to the ‘Native’ VLAN ID will be tagged with VLAN ID 0 (priority tag).
- Quality of Service parameters specify the priority field for the VLAN tag (IEEE 802.1p) and the DiffServ field for the IP headers. These specifications relate to service classes.
- When booting using BootP/DHCP protocols, the address received from the BootP/DHCP server will act as a temporary OAMP address, regardless of the address specified in the Interface Table. This address will be available when booting from Flash.
- Network Configuration changes are offline. The new configuration should be saved and will be available at the next startup.

Upon system start up, the Interface Table is parsed and passes comprehensive validation tests. If any errors occur during this validation phase, the system will send an error message to the Syslog server, and will fallback to a “safe mode”, using a single interface and no VLANs. Please be sure to follow the Syslog messages that the device sends in system startup to see if any errors occurred.

4.1.4 Troubleshooting

If any of the above guidelines are violated, the system will fall back to a “safe mode” configuration, consisting of a single interface & no VLANs. For more information on validation failures, consult the Syslog messages.

Validation failures may be caused by one of the following:

- One of the Application Types (OAMP, CONTROL, MEDIA) is missing.
- Too many interfaces with “Allowed Application Types” of OAMP or CONTROL.
- Only one interface defined but the “Allowed Application Types” column isn’t set to “ALL” (numeric value 6).
- Gateway column is filled in more than one row.
- Gateway is defined in an interface not having MEDIA as one of its “Allowed Application Types”.
- Two interfaces have the exact VLAN ID value, while VLANs are enabled.
- Two interfaces have the same name.
- Two interfaces share the same address space or subnet.

Apart from these validation errors, connectivity problems may be caused by one of the following:

- Trying to access the module with VLAN tags while booting from BootP/DHCP.
- Trying to access the module with untagged traffic when VLANs are on, and Native VLAN is not configured properly.
- Routing Table is not configured properly.

4.2 Routing Table

The routing table allows you to configure routing rules. You may define up to 25 different routing rules, via *ini* file, Web Interface & SNMP.

4.2.1 Interface Table Overview

The Routing Table consists of 6 columns:

Table 4-7: Routing Table Layout

Destination	Prefix Length	Subnet Mask	Gateway	Interface	Hop Count
201.201.0.0	16	255.255.0.0	192.168.0.1	0	1
202.202.0.0	16	255.255.0.0	192.168.0.2	0	1
203.203.0.0	16	255.255.0.0	192.168.0.3	0	1
225.225.0.0	16	255.255.0.0	192.168.0.25	0	1

4.2.2 The Routing Table Columns

Each row of the Routing Table defines a routing rule. Traffic destined to the subnet specified in the routing rule is redirected to a specified gateway, reachable via a specified interface.

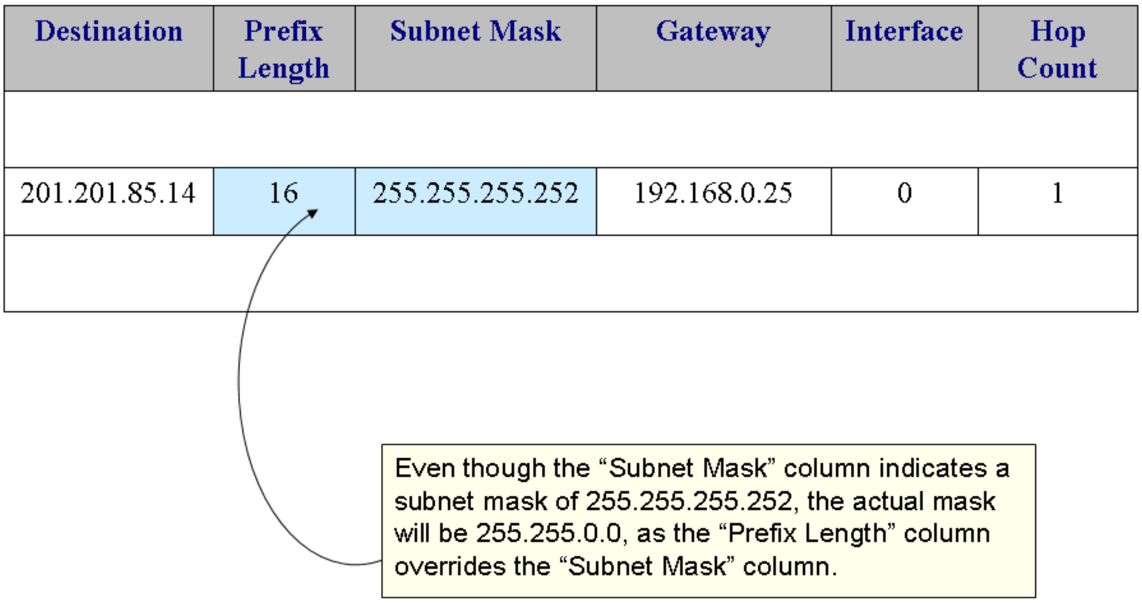
4.2.2.1 The Destination Column

This column holds the destination of the route rule. The destination can be a single host or a whole subnet, depending on the Prefix Length/Subnet Mask specified for this routing rule.

4.2.2.2 The Prefix Length and Subnet Mask Columns

These two columns offer two notations for the mask. The user can either enable the Subnet Mask in dotted decimal notation, or the CIDR-style presentation. Please note that only one of these is needed. If both are specified, the “Prefix Length” column overrides the “Subnet Mask” column:

Figure 4-2: Prefix Length and Subnet Masks Columns



Destination	Prefix Length	Subnet Mask	Gateway	Interface	Hop Count
201.201.85.14	16	255.255.255.252	192.168.0.25	0	1

Even though the “Subnet Mask” column indicates a subnet mask of 255.255.255.252, the actual mask will be 255.255.0.0, as the “Prefix Length” column overrides the “Subnet Mask” column.

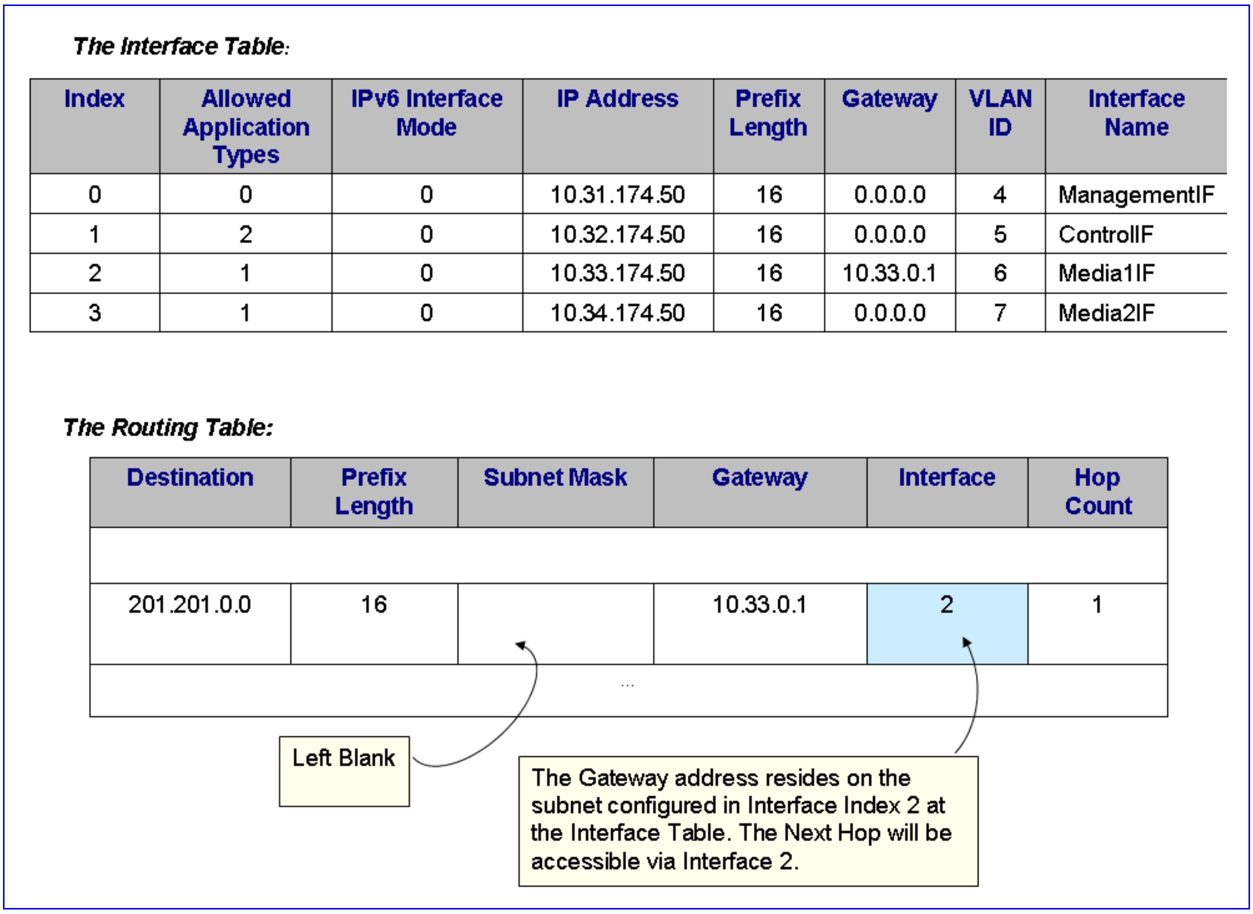
4.2.2.3 The Gateway Column

The gateway column holds the IP Address of the next hop used for traffic, destined to the subnet, as specified by the destination/mask columns. This gateway address MUST be on one of the subnets on which the address is configured in the Interface Table.

4.2.2.4 The Interface Column

This column holds the Interface index (in the Interface Table) from which the gateway address is reached.

Figure 4-3: Interface Column



The Interface Table:

Index	Allowed Application Types	IPv6 Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	0	0	10.31.174.50	16	0.0.0.0	4	ManagementIF
1	2	0	10.32.174.50	16	0.0.0.0	5	ControlIF
2	1	0	10.33.174.50	16	10.33.0.1	6	Media1IF
3	1	0	10.34.174.50	16	0.0.0.0	7	Media2IF

The Routing Table:

Destination	Prefix Length	Subnet Mask	Gateway	Interface	Hop Count
201.201.0.0	16		10.33.0.1	2	1
		...			

Left Blank

The Gateway address resides on the subnet configured in Interface Index 2 at the Interface Table. The Next Hop will be accessible via Interface 2.

4.2.2.5 The Hop Count Column

The Hop Count column MUST be set to 1 for each routing rule.

4.2.3 Routing Table Configuration Summary & Guidelines

Note: Routing Table configurations must adhere to the following rules:

- Up to 25 different routing rules may be defined.
- The user may choose whether to specify “Prefix Length” or “Subnet Mask”. There is no need to specify both.
- If both “Prefix Length“ and “Subnet Mask” are defined, the “Prefix Length” overrides the “Subnet Mask”.
- The “Gateway” IP Address must be available on one of the local subnets.

- The “Interface” column must be set to the Interface that the “Gateway” is configured on.
- The “Hop Count” column must be set to 1.
- The Routing Table configuration, unlike the Interface Table configuration, is online. Therefore, the changes made to the routing rules are applied immediately.

4.2.4 Troubleshooting

When adding or modifying any of the routing rules, the added or modified rule passes a validation test. If errors are found, the route will be disqualified and will not be added to the Routing Table.

Failed routing validations may result in limited connectivity (or no connectivity) to the destinations specified in the bad routing rule.

For any error found in the routing table or failure to configure a routing rule, the system will send a notification message to the Syslog server, reporting the problem.

Common errors configuring routing rules may be:

- The IP Address specified in the “Gateway” column is unreachable from the interface specified in the “Interface” column.
- The same destination was defined in two different routing rules.
- “Subnet Mask” and “Prefix Length” columns were both entered with inconsistent values, and the “Prefix Length” overrides the “Subnet Mask” column.
- More than 25 routing rules were specified.

Note: If a routing rule is required to access OAMP applications (for remote management, for instance) and this route is not configured correctly, the route will not be added, and the device will not be accessible remotely. In order to restore connectivity, the device will have to be accessed locally from the OAMP subnet and configure the required routes.

4.3 Setting Up Your System

4.3.1 Setting Up Your System via Web Interface

The Web interface is a convenient user interface for configuring the module's network configuration. Refer to Configuration for more information.

4.3.2 Setting Up Your System via *ini* File

When configuring the network configuration via *ini* File, use a textual presentation of the Interface and Routing Tables, as well as some other parameters.

The following shows an example of a full network configuration, consisting of all the parameters described in this section.

```
; Interface Table Configuration:
[InterfaceTable]

FORMAT InterfaceTable Index = InterfaceTable ApplicationTypes,
InterfaceTable IPv6InterfaceMode, InterfaceTable IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable VlanID, InterfaceTable InterfaceName;
InterfaceTable 0  = 6, 0, 192.168.85.14,  16, 192.168.0.1, 1,
myAll;
[\InterfaceTable]


; VLAN related parameters:
VlanMode = 0
VlanNativeVlanId = 1


; Routing Table Configuration:
RoutingTableDestinationsColumn = 201.201.0.0, 202.202.0.0
RoutingTableDestinationPrefixLensColumn = 16, 16
RoutingTableGatewaysColumn = 192.168.0.2, 192.168.0.3
RoutingTableInterfacesColumn = 0, 0
RoutingTableHopsCountColumn = 1, 1


; Class Of Service parameters:
VlanNetworkServiceClassPriority = 7
VlanPremiumServiceClassMediaPriority = 6
VlanPremiumServiceClassControlPriority = 6
VlanGoldServiceClassPriority = 4
VlanBronzeServiceClassPriority = 2
NetworkServiceClassDiffServ = 48
PremiumServiceClassMediaDiffServ = 46
PremiumServiceClassControlDiffServ = 40
GoldServiceClassDiffServ = 26
BronzeServiceClassDiffServ = 10


; Application Type for applications:
EnableDNSasOAM = 1
EnableSCTPasControl = 1
EnableNTPasOAM = 1
EnableTPNCPasOAM = 1
```

This *ini* file shows the following:

- An Interface Table with a single interface (192.168.85.14/16, all applications are allowed).
- A Routing Table is configured with two routing rules, directing all traffic for subnet 201.201.0.0/16 to 192.168.0.2, and all traffic for subnet 202.202.0.0/16 to 192.168.0.3.
- VLANs are disabled, 'Native' VLAN ID is set to 1.
- Values for the Class Of Service parameters were assigned.
- The DNS application is configured to act as an OAMP application; SCTP traffic is configured to act as a CONTROL traffic; the NTP application is configured to act as an OAMP application; and TPNCP based applications are configured to act as OAMP applications.

Note: Lines that begin with a semicolon are considered a remark and are ignored.

Note: The Interface Table configuration via *ini* file, MUST have the bolded prefix and suffix, to allow the AudioCodes INI File parser to correctly recognize the Interface Table.

The following sections show some examples of selected network configurations, and their matching *ini* file configuration.

➢ Example 1 – A Simple Single Interface Configuration

The Interface Table, with a single interface for all application types:

Index	Allowed Application Types	IPv6 Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	6	0	192.168.85.14	16	192.168.0.1	1	myAll

VLANS are not required, and the 'Native' VLAN ID is irrelevant.

Class of Service parameters may have the default values.

The required routing table features two routes:

Destination	Prefix Length	Subnet Mask	Gateway	Interface	Hop Count
201.201.0.0	16		192.168.0.2	0	1
202.202.0.0	16		192.168.0.3	0	1

The DNS/SCTP/NTP/TPNCP applications may have their default application types. This example's matching ***ini*** file is shown above. However, since many parameter values equal their default values, they can be omitted. The ***ini*** file can be also written like this:

```
; Interface Table Configuration:
[InterfaceTable]

FORMAT InterfaceTable Index = InterfaceTable ApplicationTypes,
InterfaceTable IPv6InterfaceMode, InterfaceTable IPAddress,
InterfaceTable PrefixLength, InterfaceTable Gateway,
InterfaceTable VlanID, InterfaceTable InterfaceName;

InterfaceTable 0  = 6, 0, 192.168.85.14,  16, 192.168.0.1, 1,
myAll;
[\InterfaceTable]


; Routing Table Configuration:
RoutingTableDestinationsColumn = 201.201.0.0, 202.202.0.0
RoutingTableDestinationPrefixLensColumn = 16, 16
RoutingTableGatewaysColumn = 192.168.0.2, 192.168.0.3
RoutingTableInterfacesColumn = 0, 0
RoutingTableHopsCountColumn = 1, 1
```

➢ Example 2 – Three Interfaces - one for each application exclusively

The Interface Table will be configured with three interfaces, one exclusively for each application type: one interface for OAMP applications, one for Call Control applications, and one for RTP Media applications:

Index	Allowed Application Types	IPv6 Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	0	0	192.168.85.14	16	0.0.0.0	1	ManagementIF
1	2	0	200.200.85.14	24	0.0.0.0	200	myControlIF
2	1	0	211.211.85.14	24	211.211.85.1	211	myMediaIF

VLANs are required. The 'Native' VLAN ID is the same VLAN ID as the AudioCodes Management interface (index 0).

One routing rule is required, to allow remote management:

Destination	Prefix Length	Subnet Mask	Gateway	Interface	Hop Count
176.85.49.0	24		192.168.0.1	0	1

All other parameters will be set to their respective default values.

The *ini* file matching this configuration can be written like this:

```
; Interface Table Configuration:
[InterfaceTable]

FORMAT InterfaceTable Index = InterfaceTable ApplicationTypes,
InterfaceTable_IPv6InterfaceMode, InterfaceTable_IPAddress,
InterfaceTable PrefixLength, InterfaceTable Gateway,
InterfaceTable VlanID, InterfaceTable InterfaceName;

InterfaceTable 0  = 0, 0, 192.168.85.14,  16, 0.0.0.0, 1,
ManagementIF;
InterfaceTable 1  = 2, 0, 200.200.85.14,  24, 0.0.0.0, 200,
myControlIF;
InterfaceTable 2  = 1, 0, 211.211.85.14,  24, 211.211.85.1, 211,
myMediaIF;
[\InterfaceTable]


; VLAN related parameters:
VlanMode = 1
VlanNativeVlanId = 1


; Routing Table Configuration:
RoutingTableDestinationsColumn = 176.85.49.0
RoutingTableDestinationPrefixLensColumn = 24
RoutingTableGatewaysColumn = 192.168.0.1
RoutingTableInterfacesColumn = 0
RoutingTableHopsCountColumn = 1
```

➢ Example 3 - One interface exclusively for management (OAMP applications) and another for Call Control and RTP (CONTROL and MEDIA applications):

The Interface Table will be configured with two interfaces. One is exclusively for Management purposes, and the other for Call Control and RTP Media applications:

Index	Allowed Application Types	IPv6 Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	0	0	192.168.85.14	16	0.0.0.0	1	Mgmt
1	5	0	200.200.85.14	24	200.200.85.1	200	CntrlMedia

VLANs are required. The 'Native' VLAN ID is the same VLAN ID as the AudioCodes Management interface (index 0).

One routing rule is required, to allow remote management:

Destination	Prefix Length	Subnet Mask	Gateway	Interface	Hop Count
176.85.49.0	24		192.168.0.1	0	1

All other parameters will be set to their respective default values.

The *ini* file matching this configuration can be written like this:

```
; Interface Table Configuration:
[InterfaceTable]

FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes,
InterfaceTable IPv6InterfaceMode, InterfaceTable IPAddress,
InterfaceTable PrefixLength, InterfaceTable Gateway,
InterfaceTable_VlanID, InterfaceTable_InterfaceName;

InterfaceTable 0  = 0, 0, 192.168.85.14,  16, 0.0.0.0, 1, Mgmt;
InterfaceTable 1  = 5, 0, 200.200.85.14,  24, 200.200.85.1, 200,
CntrlMedia;
 [\InterfaceTable]


; VLAN related parameters:
VlanMode = 1
VlanNativeVlanId = 1


; Routing Table Configuration:
RoutingTableDestinationsColumn = 176.85.49.0
RoutingTableDestinationPrefixLensColumn = 24
RoutingTableGatewaysColumn = 192.168.0.1
RoutingTableInterfacesColumn = 0
RoutingTableHopsCountColumn = 1
```

4.3.3 VLANs and Multiple Interfaces – A Basic Walkthrough

By default the device operates without VLANs and multiple IPs, using a single IP address, subnet mask and default gateway IP address. This section provides an example of the configuration required to integrate the device into a VLAN and multiple IPs network using the Web interface and ***ini*** file. The table below shows an example configuration that is implemented in the following sections.

Table 4-8: Example of VLAN and Multiple IPs Configuration

Network Type	IP Address	Subnet Mask	Default Gateway IP Address	VLAN ID	External Routing Rule
OAM	10.31.174.50	255.255.0.0	0.0.0.0	4	83.4.87.X
Control	10.32.174.50	255.255.0.0	0.0.0.0	5	130.33.4.6
Media	10.33.174.50	255.255.0.0	10.33.0.1	6	--

Note that since a default gateway is available only for the Media network, for the device to be able to communicate with an external device / network on its OAM and Control networks, IP routing rules must be used.

Note: The values provided are sample parameters only and are to be replaced with actual values appropriate to your system.

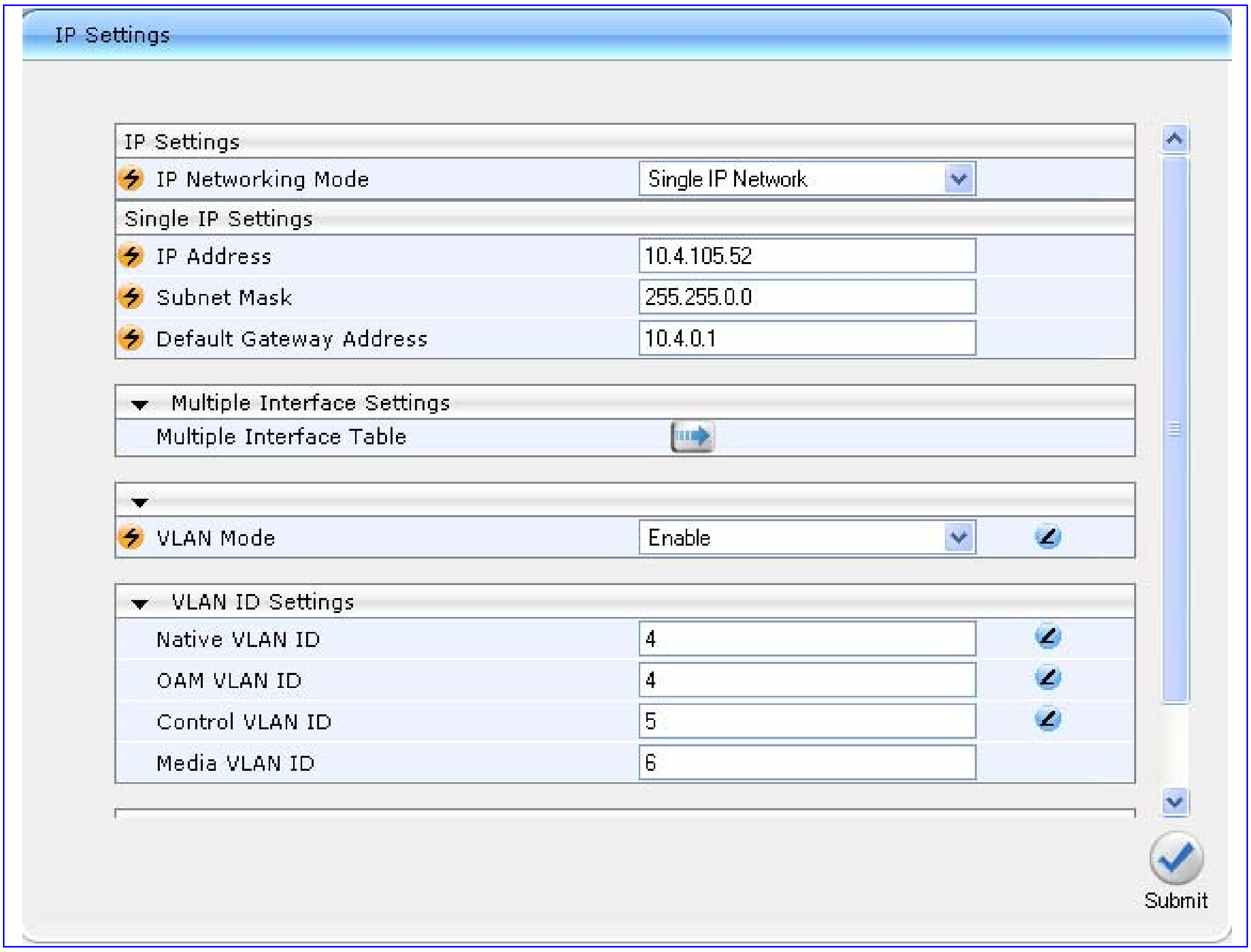
4.3.3.1 VLAN Configuration Using the Web Interface

➢ **To integrate the device into a VLAN and Multiple IPs network using the Web interface, take these 7 steps:**

1. Access the Web Interface (refer to the Web Interface section of the products's User's Manual).
2. Use the Software Upgrade Wizard (Refer to **Software Upgrade Wizard** in the product's User's Manual) to load and burn the firmware version to the device (VLANs and multiple IPs support is available only when the firmware is burned to flash).
3. Configure the VLAN parameters by completing the following steps:
   - Open the ‘VLAN Settings’ screen (Advanced Configuration menu > Network Settings > IP Settings option); the ‘IP Settings’ screen is displayed.

- Modify the VLAN parameters to correspond to the values shown in the figure below.

Figure 4-4: VLAN Settings Screen Example



IP Settings

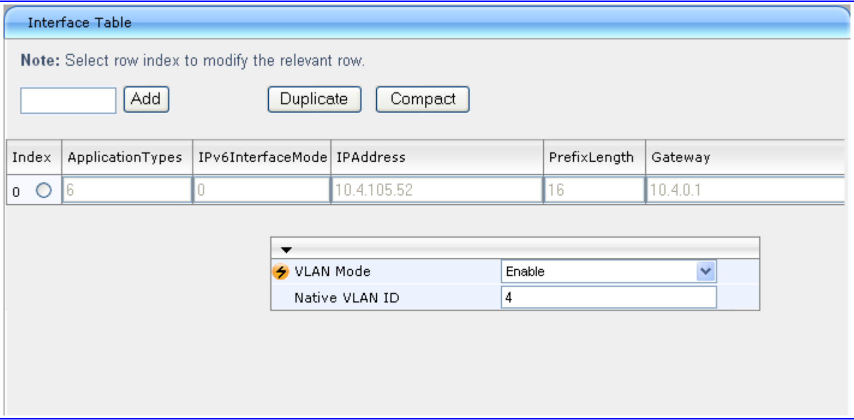
IP Settings	
IP Networking Mode	Single IP Network
Single IP Settings	
IP Address	10.4.105.52
Subnet Mask	255.255.0.0
Default Gateway Address	10.4.0.1
Multiple Interface Settings	
Multiple Interface Table	
VLAN Mode	Enable
VLAN ID Settings	
Native VLAN ID	4
OAM VLAN ID	4
Control VLAN ID	5
Media VLAN ID	6

- Click the Submit button to save your changes.

4. Configure the **Multiple Interface Settings** parameters by completing the following steps:
   - Open the ‘IP Settings’ screen (Configuration menu > Network Settings > IP Settings option); the ‘Interface Table’ screen is displayed.
   - Modify the IP parameters to correspond to the values shown in the figure below. Note that the OAMP, Control and Media Network Settings parameters appear only after you select the option ‘Multiple IP Networks’ in the field ‘IP Networking Mode’.

Note: Configure the OAMP parameters only if the OAMP networking parameters are different from the networking parameters used in the Single IP Network mode.

Figure 4-5: Interface Table



Interface Table

Note: Select row index to modify the relevant row.

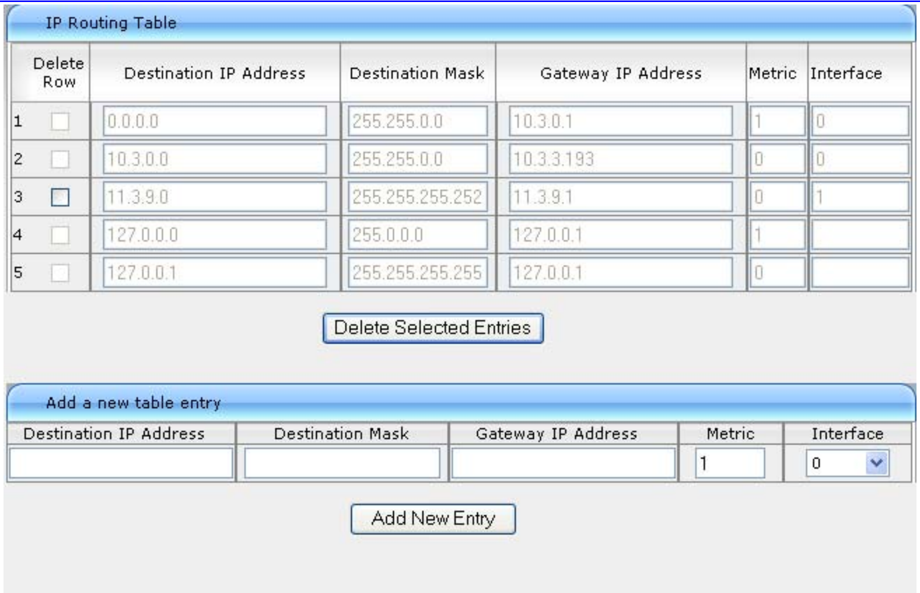
Index	ApplicationTypes	IPv6InterfaceMode	IPAddress	PrefixLength	Gateway
0 ○	6	0	10.4.105.52	16	10.4.0.1

VLAN Mode	Enable
Native VLAN ID	4

- Click the Submit button to save your changes.

5. Configure the IP Routing table by completing the following steps (the IP Routing table is required to define static routing rules for the OAMP and Control networks since a default gateway isn't supported for these networks):
   - Open the 'IP Routing Table' screen (Configuration menu > Network Settings > IP Routing Table option); the 'IP Routing Table' screen is displayed.

Figure 4-6: IP Routing Table



IP Routing Table

	Delete Row	Destination IP Address	Destination Mask	Gateway IP Address	Metric	Interface
1	□	0.0.0.0	255.255.0.0	10.3.0.1	1	0
2	□	10.3.0.0	255.255.0.0	10.3.3.193	0	0
3	□	11.3.9.0	255.255.255.252	11.3.9.1	0	1
4	□	127.0.0.0	255.0.0.0	127.0.0.1	1	
5	□	127.0.0.1	255.255.255.255	127.0.0.1	0	

Add a new table entry

Destination IP Address	Destination Mask	Gateway IP Address	Metric	Interface
			1	0

   - Use the 'Add a new table entry' pane to add the routing rules shown in the table below.

Table 4-9: Example of IP Routing Table Configuration

Destination IP Address	Destination Mask	Gateway IP Address	Hop Count	Network Type
130.33.4.6	255.255.255.255	10.32.0.1	20	Control
83.4.87.6	255.255.255.0	10.31.0.1	20	OAMP

- Click the Submit button to save your changes.

6. Save your changes to flash so they are available after a power fail.
7. Reset the gateway. Click the Reset button on the main menu bar; the Reset screen is displayed. Click the button Reset.

4.3.3.2 Integrating Using the *ini* File

➢ To integrate the device into a VLAN and multiple IPs network using the *ini* file, take these 3 steps:

1. Prepare an ***ini*** file with parameters shown in the figure below (refer to the following notes):
   - If the BootP/TFTP utility and the OAMP interface are located in the same network, the Native VLAN ID (VLANNativeVLANId) must be equal to the OAMP VLAN ID (VLANOamVLANId), which in turn must be equal to the PVID of the switch port the gateway is connected to. Therefore, set the PVID of the switch port to 4 (in this example).
   - Configure the OAMP parameters (LocalOAMIPAddress, LocalOAMSubnetMask and LocalOAMDefaultGW) only if the OAMP networking parameters are different from the networking parameters used in the Single IP Network mode.
   - The IP Routing table is required to define static routing rules for the OAMP and Control networks since a default gateway isn't supported for these networks.

Example of VLAN and Multiple IPs *ini* File Parameters

```
; VLAN Configuration
VlanMode=1
VlanOamVlanId=4
VlanNativeVlanId=4
VlanControlVlanId=5
VlanmediaVLANid=6

; Multiple IPs Configuration
EnableMultipleIPs=1
LocalMediaIPAddress=10.33.174.50
LocalMediaSubnetMask=255.255.0.0
LocalMediaDefaultGW=10.33.0.1
LocalControlIPAddress=10.32.174.50
LocalControlSubnetMask=255.255.0.0
LocalControlDefaultGW=0.0.0.0
LocalOAMIPAddress=10.31.174.50
LocalOAMSubnetMask=255.255.0.0
LocalOAMDefaultGW=0.0.0.0

; IP Routing table parameters
RoutingTableDestinationsColumn = 130.33.4.6, 83.4.87.6
```

```
RoutingTableDestinationMasksColumn = 255.255.255.255 ,
255.255.255.0
RoutingTableGatewaysColumn = 10.32.0.1 , 10.31.0.1
RoutingTableInterfacesColumn = 2 , 0
RoutingTableHopsCountColumn = 20,2
```

2. Use the BootP/TFTP utility to load and burn (-fb option) the firmware version and the ***ini*** file you prepared in the previous step to the device (VLANs and multiple IPs support is available only when the firmware is burned to flash).
3. Reset the device after disabling it on the BootP/TFTP utility.

By default the device works without VLANs and IP separation. To enable these features, the enable parameter should be entered via the ***ini*** file, together with the set of parameters that are needed for correct device operation (i.e. set of IP addresses, set of VLAN IDs, etc.) After the updated ***ini*** file is saved and loaded to the non-volatile flash memory on the device, the device must be reset. The updated configuration, which includes the updated settings to enable VLANS and multiple IP address is implemented.

4.3.4 Setup Example

The configuration directions in this section utilize the sample parameters detailed in this section.

Using default values, the device currently works in single IP mode with parameters acquiring from its BootP server. The following are sample parameters acquired from the BootP server:

- IP = 10.31.174.50
- Subnet = 255.255.0.0
- Default gateway = 10.31.10.1

4.3.5 Preparing the Device for VLANs and Multiple IPs (MI)

To illustrate how to prepare for VLANS and Multiple IPs (MI), without using the Interface Table scheme described in this chapter, two examples are detailed in this section.

Example 1 - The device is to communicate with an external device whose IP address is **130.33.4.6** via the internal Call-Control network

Example 2 - The device is to communicate with an external network whose IP address is **83.4.87.X** via the internal OAM network.

In both of these examples the hop count routing parameter is **20**. Since these internal networks cannot be configured with default gateways, static routes are needed.

➢ To prepare the device for Multiple IPs and VLANs, take these 10 steps:

1. Ascertain values for the following parameters:

Note: The values provided are sample parameters only and are to be replaced with actual values appropriate to your system.

- OAMP network:

    - IP/Subnet/GW = 10.31.174.50 / 255.255.0.0 / 0.0.0.0
    - VLAN ID = 4
- Call Control network:
    - IP/Subnet/GW = 10.32.174.50 / 255.255.0.0 / 0.0.0.0
    - VLAN ID = 5
- Media network:
    - IP/Subnet/GW = 10.33.174.50 / 255.255.0.0 / 10.33.0.1
    - VLAN id = 6

> **Note:** The device is configured only with one default gateway on the media network. Additional routes can be configured statically by Web/SNMP/TPNCP and even via the ***ini*** file.

1. In the ***ini*** file, set the VLAN parameters.
   a. If the BootP/TFTP to be on the same network as the operational OAMP network, the Native VLAN parameter is equal to the OAMP VLAN (both equal to the PVID of the switch port of the device to which it is connected). According to the sample parameters in this example (see above), the OAMP Network VLAN ID = 4. The following is an example of the VLAN parameters in the ***ini*** file with values according to the sample parameters above:

```
VLANOamVlanId=4
VLANNativeVlanId=4
```

   b. In the switch port, configure the PVID = 4.
   c. Set the Call Control network VLAN ID and Media network VLAN ID. According to the sample parameters in this example (see above), the Call Control network VLAN ID = 5 and the Media network VLAN ID = 6 . The following is an example of these parameters in the ***ini*** file with values according to the sample parameters above:

```
VLANControlVlanId=5
VLANMediaVlanId=6
```

   d. Set the VLANMode parameter to 1. The following is an example of this parameter in the ***ini*** file:

```
VLANMode=1
```

2. In the ***ini*** file, set the Multiple IP parameters.
3. In the ***ini*** file, configure the Local Media and Call-Control parameters. The following is an example of these parameters in the ***ini*** file with values according to the sample parameters above:

```
LocalMediaIPAddress=10.33.174.50
LocalMediaSubnetMask=255.255.0.0
LocalMediaDefaultGW=10.33.0.1

LocalControlIPAddress=10.32.174.50
```

```
LocalControlSubnetMask=255.255.0.0
LocalControlDefaultGW=0.0.0.0
```

Note: More than one interface cannot be configured on the same network.

4. Set the required static routes. The table below displays the Routing Table Rules according to the sample parameters in the example 1 and 2 as mentioned above.

Table 4-10: Routing Table Rules

Destination	Subnet Mask	Gateway	Hops Count	Network
130.33.4.6	255.255.255.255	10.32.0.1	20	Call-Control
83.4.87.6	255.255.255.0	10.31.0.1	20	OAMP

The interface networks are designated as follows:

0 - OAMP

1 - Media

2 - Call Control

According to the sample parameters shown in the table above, the Routing table in the ***ini*** file appears like this:

```
The following is an example of the Routing Table parameters
(according to the sample parameters shown in the table above) in
the ini file:

RoutingTableDestinationsColumn = 130.33.4.6, 83.4.87.6
RoutingTableDestinationMasksColumn = 255.255.255.255 ,
255.255.255.0
RoutingTableGatewaysColumn = 10.32.0.1 , 10.31.0.1
RoutingTableInterfacesColumn = 1 , 0
RoutingTableHopsCountColumn = 20,20
```

Each row of these parameters in the ***ini*** file is a column in the table of Routing Table Rules.

5. Set the EnableMultipleIPs parameter to 1. The following is an example of this parameter in the ***ini*** file:

```
EnableMultipleIPs=1
```

6. Save the changes to the ***ini*** file.
7. If your device is using a software version earlier than 4.6, use BootP to burn the updated *cmp* version to the non -volatile flash memory (**-fb** option in the BootP application).
8. Reset the device. The updated ***ini*** file is implemented. The device has all the required information to enable the advanced VLAN and IP separation features.
9. To re-establish its network infrastructure, reset the device again.

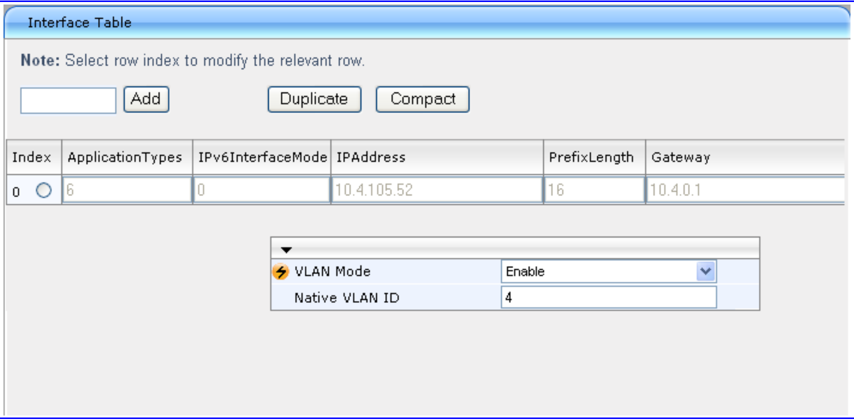
4.3.6 Verifying the VLANS and Multiple IP Settings Using the Web Interface

For details on using the Web Interface, refer to the Web Interface section of the product’s User's Manual.

➢ **To verify the VLANS and Multiple IP settings using the Web interface, take the following 2 steps:**

1. In the Web interface, access the IP Setting screen (Advanced Configuration->Network Settings->IP Settings).

Figure 4-7: Interface Table



Interface Table

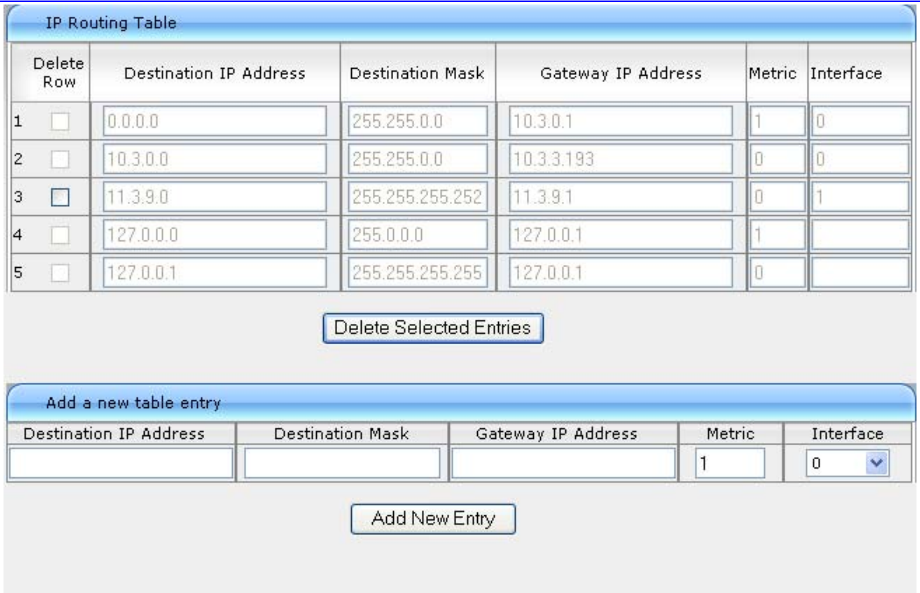
Note: Select row index to modify the relevant row.

Index	ApplicationTypes	IPv6InterfaceMode	IPAddress	PrefixLength	Gateway
0 ○	6	0	10.4.105.52	16	10.4.0.1

VLAN Mode	Enable
Native VLAN ID	4

2. Access the Routing Table screen (Configuration->Network Settings->IP Routing Table):

Figure 4-8: IP Routing Table



IP Routing Table

	Delete Row	Destination IP Address	Destination Mask	Gateway IP Address	Metric	Interface
1	□	0.0.0.0	255.255.0.0	10.3.0.1	1	0
2	□	10.3.0.0	255.255.0.0	10.3.3.193	0	0
3	□	11.3.9.0	255.255.255.252	11.3.9.1	0	1
4	□	127.0.0.0	255.0.0.0	127.0.0.1	1	
5	□	127.0.0.1	255.255.255.255	127.0.0.1	0	

Add a new table entry

Destination IP Address	Destination Mask	Gateway IP Address	Metric	Interface
			1	0

4.3.7 OAMP Parameters

If the your network architecture requires the OAMP network settings be different than the parameter values acquired in the BootP process, you must set the local OAMP configuration line in the ***ini*** file to the required parameters. The example below shows the form of these parameter settings using the following sample parameter values:

- OAMP network:
  - IP/Subnet/GW = 10.34.174.50 / 255.255.0.0 / 0.0.0.0
  - VLAN ID = 7

➢ To set the local OAMP configuration, take these 4 steps:

1. In the ***ini*** file, set the local OAMP configuration line similar to the following:

```
LocalOAMIPAddress=10.34.174.50
LocalOAMSubnetMask=255.255.0.0
LocalOAMDefaultGW=0.0.0.0
```

2. In the ***ini*** file, change the OAMP VLAN tag line to:

```
VLANOamVlanId=7
```

3. Save the changes to the ***ini*** file and load it to the device.
4. Reset the device. The updated ***ini*** file is implemented. The device has all the required information to enable the OAMP parameters.

4.3.8 MI and VLAN Parameters

The following table lists the Multiple IP parameters.

Table 4-11: Multiple IP Parameters

Parameter Name	Default Value	Comments
LocalMediaIPAddress	0.0.0.0	The source address of the device in the Media network
LocalMediaSubnetMask	0.0.0.0	The subnet applied to the device in the Media network
LocalMediaDefaultGW	0.0.0.0	The default gateway in the Media network
LocalControlIPAddress	0.0.0.0	The source address of the device in the Control network
LocalControlSubnetMask	0.0.0.0	The subnet applied to the device in the Control network
LocalControlDefaultGW	0.0.0.0	The default gateway in the Control network (currently not in use)

Table 4-11: Multiple IP Parameters

Parameter Name	Default Value	Comments
LocalOAMIPAddress	0.0.0.0	The source address of the device in the OAMP network
LocalOAMSubnetMask	0.0.0.0	The subnet applied to the device in the OAMP network
LocalOAMDefaultGW	0.0.0.0	The default gateway in the OAMP network (currently not in use)

The following table lists the VLAN parameters.

Table 4-12: VLAN Parameters

Parameter Name	Default Value	Comments
VLANMode	0	When burned in the flash, the device will try to initialize VLANs feature in subsequent boots
VLANNativeVLANId	1	The PVID of the switch port which the device is connected to
VLANOamVLANId	1	The VLAN Identifier of the OAMP
VLANControlVLANId	2	The VLAN Identifier of the Control
VLANMediaVLANId	3	The VLAN Identifier of the Media
VLANSendNonTaggedOnNative	0	Specifies whether to send non-tagged packets on the native VLAN
VLANNetworkServiceClassPriority	7	Sets the priority for the Network service class content
VLANPremiumServiceClassMediaPriority	6	Sets the priority for the Premium service class content and media traffic
VLANPremiumServiceClassControl Priority	6	Sets the priority for the Premium service class content and control traffic
VLANGoldServiceClassPriority	4	Sets the priority for the Gold service class content
VLANBronzeServiceClassPriority	2	Sets the priority for the Bronze service class content
NetworkServiceClassDiffServ	48	Sets the DiffServ for the Network service class content
PremiumServiceClassMediaDiffServ	46	Sets the DiffServ for the Premium service class content and media traffic

Table 4-12: VLAN Parameters

Parameter Name	Default Value	Comments
PremiumServiceClassControlDiffServ	40	Sets the DiffServ for the Premium service class content and control traffic
GoldServiceClassDiffServ	26	Sets the DiffServ for the Gold service class content
BronzeServiceClassDiffServ	10	Sets the DiffServ for the Bronze service class content

The following parameters are used for both the VLAN and MI features.

Table 4-13: Shared VLAN and MI Parameters

Parameter Name	Default Value	Range	Comments

EnableDNSasOAM	1	Enable = 1
		Disable = 0
			For MI: If enabled, the DNS services are on the OAMP network. If disabled, they are on the Control network
			For VLANS: If enabled, the DNS services are on the OAMP VLAN. If disabled, they are on the Control VLAN

EnableNTPasOAM	1	Enable = 1
		Disable = 0
			For MI: If enabled, NTP services are on the OAMP network. If disabled, they are on the Control network
			For VLANS: If enabled NTP services are on the OAMP VLAN. If disabled, they are on the Control VLAN

EnableSCTPasControl	1	Enable = 1
		Disable = 0
			For MI: If enabled, SCTP services are on the Control network. If disabled, they are on the OAMP network
			For VLANS: If enabled, SCTP services are on the Control VLAN. If disabled, they are on the OAMP VLAN.

EnableTPNCPasOAM	1	Enable = 1
		Disable = 0
			For MI: If enabled, TPNCP services are on the OAMP network. If disabled, they are the Control network
			For VLAN: If enabled, TPNCP services are on the OAMP VLAN. If disabled, they are on the Control VLAN.

4.3.9 Getting Started with the Mediant 3000 System in High Availability Mode

Note: This sub-section is only applicable to **Mediant 3000** and **Mediant 3000 with TP-8410 blades**.

In High Availability (HA) mode, the Mediant 3000 system features two blades which are redundant to each other. Both blades share the same configuration, while only one of them is active. (The default state is the Active module in Slot 1 and the Redundant module in Slot 3).

Each of the blades in the Mediant 3000 system will boot as a standalone. The blade will also be assigned with its own private address (which may have been acquired via BootP/DHCP or configured manually) which will be used throughout the boot process (prior to entering HA mode).

The active blade will switch to the address configured by the M3KGlobalIpAddr parameter once the boot process has completed, replacing its existing OAMP address. The redundant blade will disconnect from network once the system has gone into HA mode.

4.3.9.1 The Mediant 3000 Internal Link

In Mediant 3000 systems, the two modules keep an active communication channel between them. Via this channel, the redundant blade is updated constantly by the active blade and monitors the state of the active blade, ready to take over in the case of a failure.

The internal link is used solely for internal communication and will not be reachable from an external network. The system uses a pre-configured IP Addresses based on the slot that each module was inserted in.

The blade in Slot #1 (which is the active blade at system startup) allocates the IP address 11.3.9.1 with the prefix length of 30 bits (equal to a subnet mask of 255.255.255.252). The blade in Slot #3 (which is the redundant blade at system startup) allocates the IP address 11.3.9.2 with the prefix length of 30 bits (equals to a subnet mask of 255.255.255.252).

Note: The internal link addresses should not be configured as one of the system addresses. Communication with any external peer carrying this address is also forbidden.

4.3.9.2 Planning Multiple Interfaces Scheme with Mediant 3000 Systems

When working with Mediant 3000 systems, it is important to know, that the interface supporting OAMP applications, will change its address to the Mediant 3000 Global IP Address.

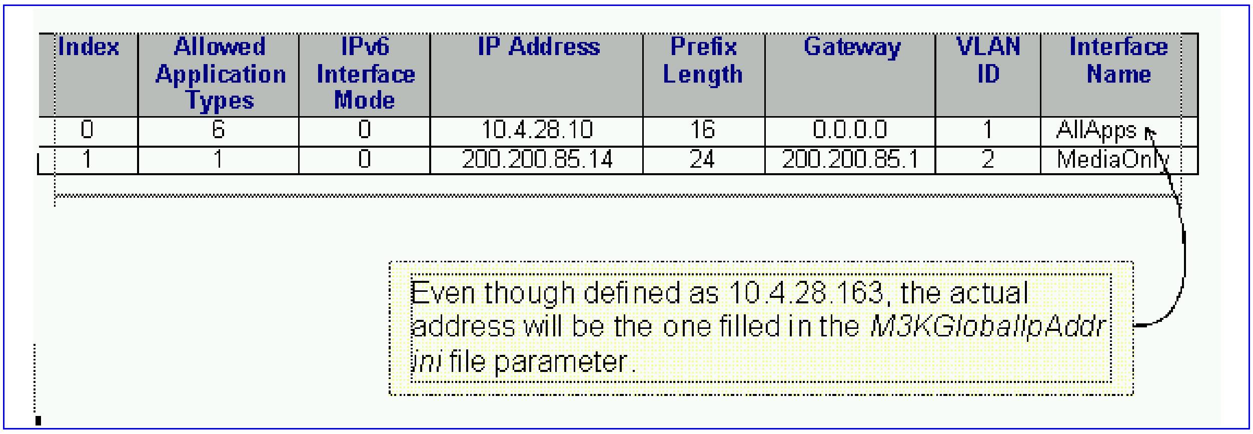
This affects the system configuration when working with Multiple Interfaces. If, for instance, the following Interface Table is used:

Index	Allowed Application Types	IPv6 Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	6	0	10.4.28.163	16	0.0.0.0	1	AllApps
1	1	0	200.200.85.14	24	200.200.85.1	2	MediaOnly

* The M3KGlobalIPAddress parameter is configured to 10.4.28.10

We should be aware of the fact, that the first interface (supporting Media & Call Controls in addition to OAMP applications) will change its IP Address to the address assigned to the M3KGlobalIpAddr parameter.

Therefore the actual configuration will be:

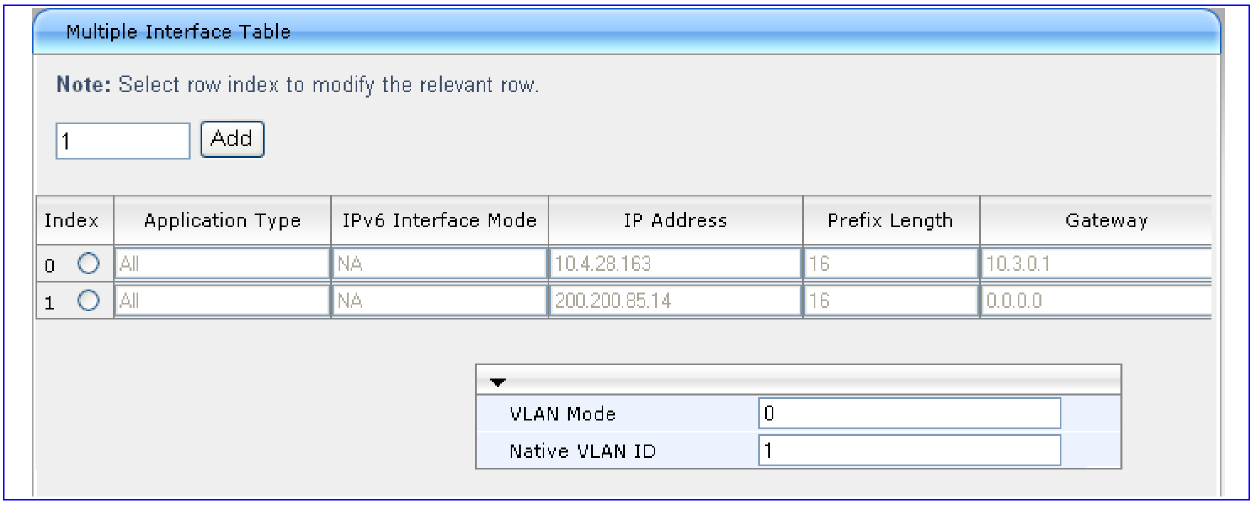


Index	Allowed Application Types	IPv6 Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	6	0	10.4.28.10	16	0.0.0.0	1	AllApps
1	1	0	200.200.85.14	24	200.200.85.1	2	MediaOnly

Even though defined as 10.4.28.163, the actual address will be the one filled in the *M3KGlobalIpAddr ini* file parameter.

This may seem confusing to some, as the Interface Table will show exactly as configured:

Figure 4-9: Interface Table



Multiple Interface Table

Note: Select row index to modify the relevant row.

Index	Application Type	IPv6 Interface Mode	IP Address	Prefix Length	Gateway
0 ○	All	NA	10.4.28.163	16	10.3.0.1
1 ○	All	NA	200.200.85.14	16	0.0.0.0

VLAN Mode	0
Native VLAN ID	1

The Interface Table shows that the IP address of interface #0 is 10.4.28.163, but actually it was changed to 10.4.28.10. (Refer to the address field of the browser.)

4.3.9.3 Configuring the Mediant 3000 for Multiple Interfaces via *ini* File

When configuring Mediant 3000 systems for Multiple Interfaces take into consideration that one of the interfaces (the one that supports OAMP applications) address will be changed to the Mediant 3000 Global IP Address.

When configuring the system for High Availability, the “M3KGlobalIpAddr” parameter must be added, which will act as the Management IP Address (the OAMP interface address) of the whole system.

The following example shows a sample configuration of three interfaces.

First, let’s allocate three IP Addresses to the system:

IP Address	Details
192.168.85.14	The M3K Global IP Address. This address will be used as the management IP Address of the system.
192.168.85.15	A private IP Address for the blade in slot #1
192.168.85.16	A private IP Address for the blade in slot #2

The Interface Table will be configured with three interfaces, one exclusively for each application type:

- OAMP applications
- Call Control applications
- RTP Media applications

The interface of the OAMP applications will be configured to the private IP Address of the blade at slot #1.

Index	Allowed Application Types	IPv6 Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	0	0	192.168.85.15	16	0.0.0.0	1	Management IF
1	2	0	200.200.85.14	24	0.0.0.0	200	myControlIF
2	1	0	211.211.85.14	24	211.211.85.1	211	myMediaIF

VLANs are required. The ‘Native’ VLAN ID is the same VLAN ID as the AudioCodes management interface (index 0).

One routing rule is required, to allow remote management:

Destination	Prefix Length	Subnet Mask	Next Hop	Interface	Metric
176.85.49.0	24		192.168.0.1	0	1

All other parameters will be set to their respective default values.

The *ini* file matching this configuration can be written like this:

```
; M3K system Global IP Address:
M3KGlobalIpAddr = 200.200.85.14

; Interface Table Configuration:
[InterfaceTable]

FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes,
InterfaceTable_IPv6InterfaceMode, InterfaceTable_IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable_VlanID, InterfaceTable_InterfaceName;

InterfaceTable 0  = 0, 0, 192.168.85.15,  16, 0.0.0.0, 1,
ManagementIF;
InterfaceTable 1  = 2, 0, 200.200.85.14,  24, 0.0.0.0, 200,
myControlIF;
InterfaceTable 2  = 1, 0, 211.211.85.14,  24, 211.211.85.1, 211,
myMediaIF;
[\InterfaceTable]


; VLAN related parameters:
VlanMode = 1
VlanNativeVlanId = 1


; Routing Table Configuration:
RoutingTableDestinationsColumn = 176.85.49.0
RoutingTableDestinationPrefixLensColumn = 24
RoutingTableGatewaysColumn = 192.168.0.1
RoutingTableInterfacesColumn = 0
RoutingTableHopsCountColumn = 1
```

The same *ini* file should be loaded to both blades.

4.3.9.4 Using Separate Physical Network Interfaces with your Mediant 3000

Note: This sub-section is only applicable to **Mediant 3000 with TP-8410 blades**.

When the Mediant 3000 operates in the Multiple Interfaces scheme, it uses a single physical Ethernet port, located on its RTM. Using VLAN tags and an external VLAN aware switch, the traffic can be directed to separate physical networks. In the physical interfaces separation mode, each application type (OAMP, Call Control and RTP Media) has its own dedicated Ethernet port.

A Mediant 3000 with a TP-8410 blade offers the user to split the traffic of different application types into different physical ports. This eliminates the need of a VLAN aware switch, redirecting the traffic by its VLAN tag.

In this mode, the system actually implements the same scheme internally. This means that VLAN tags are used internally and traffic will not be sent or received with VLAN tags.

Note: Physical Network Separation is currently supported only with the three interfaces scheme. All three interfaces must be configured.

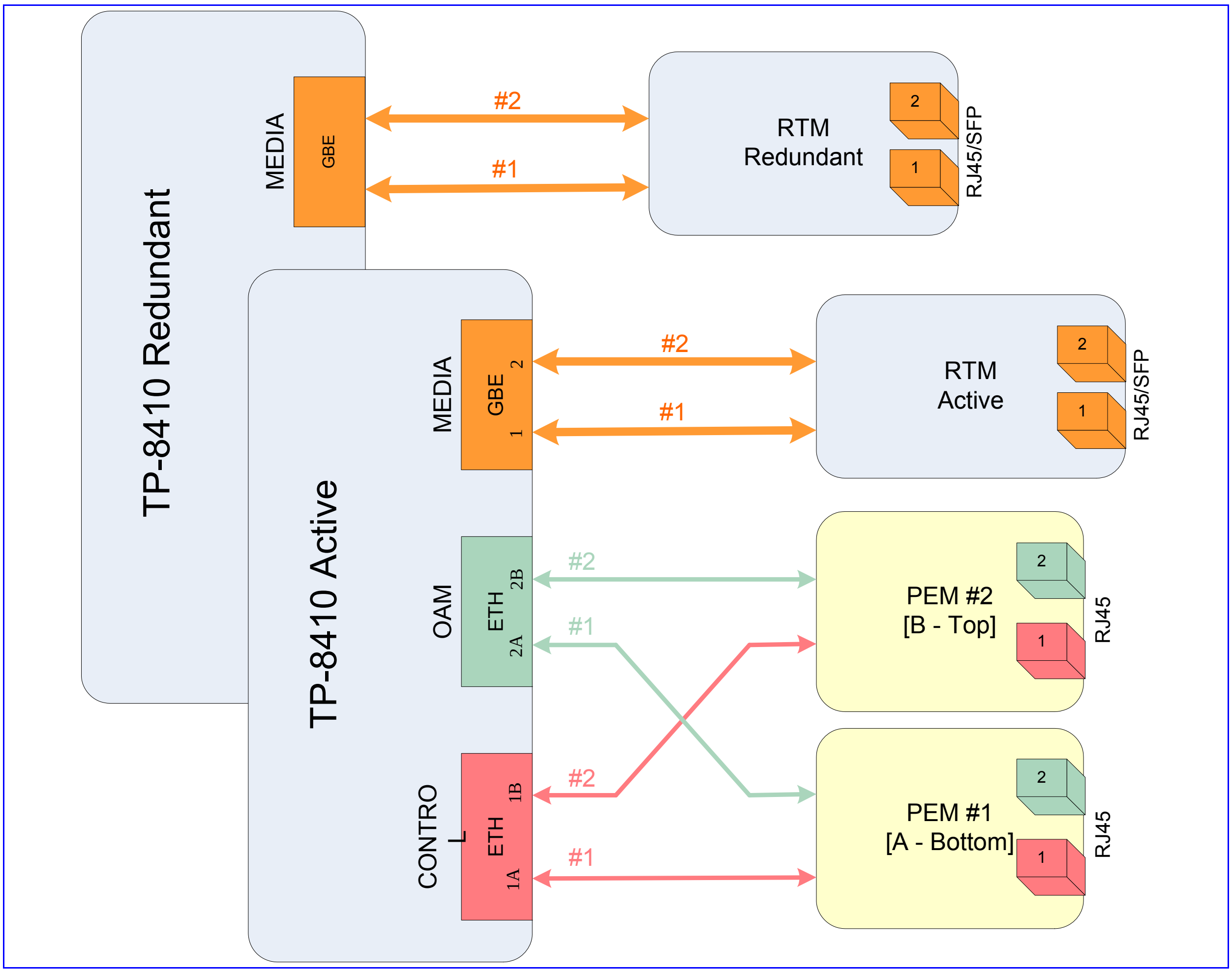
> **Note:** When working in this mode, OAMP traffic is sent and received from/to the designated physical port (on the PEM); this port is not the same port used when working with a single physical interface.

Changing the operation mode (enabling or disabling network physical interfaces separation) is done by setting the following *ini* file parameter:

```
EnableNetworkPhysicalSeparation = 1
```

The following figure illustrates the connectivity of the system when working with network physical interfaces separation:

Figure 4-10: Network Separate Physical Interfaces on Mediant 3000 + TP-8410 Block Diagram



Each of the TP-8410 blades (Active and Redundant) has its own dedicated and redundant physical GBE port for Media traffic. This interface is directly available through the RTM module. The other four Ethernet ports are available through the PEM module, but are shared by both blades. The active blade is the one connected to these ports.

4.3.9.4.1 Configuring your System to the Separate Physical Interfaces Scheme

➢ **To prepare the Mediant 3000 to work with Multiple Interfaces and**

network physical interfaces separation, take these 8 steps:

Prepare an ini file with parameters as shown in Setting Up Your System via ini File on page 249.

1. Insert only a single blade into the system. Each blade should be configured separately. There is no importance to the order.
2. Make sure that your Ethernet cable is connected to the RTM of the inserted blade.

Use BootP/TFTP to load the INI file you prepared in the first step to the blade (Multiple Interfaces and physical interfaces separation are available when booting from flash), or use the Web interface to set the configuration.

3. Verify that the following message is sent to the Syslog: *"Updating Flash to work in Network Separation Mode in the next Boot".*
4. Repeat steps 3 to 5 for the second blade.
5. Insert both blades into the system and connect a separate Ethernet cables for each network. Remember that your OAMP applications are now available at the PEM module. Power up the system.
6. Verify that the following message is sent to the SysLog from each blade: *"Board Is Working in Network Separation Mode"*.

➢ To prepare the Mediant 3000 to work with Multiple Interfaces and without network physical interfaces separation, take these 7 steps:

Prepare an ini file with parameters as shown in Setting Up Your System via ini File on page 249.
Make sure that the *ini* file parameter *"EnableNetworkPhysicalSeparation"* is added and set to 0.

1. Insert a single blade into the system. Each blade should be configured separately, while the other blade is not inserted into the M3K. There is no importance to the order.
2. Make sure your Ethernet cable is connected to the PEM.
3. Use BootP/TFTP to load the INI file you prepared in the first step to the blade (Multiple Interfaces and VLANs are available when booting from flash), or use the Web interface to set the configuration.
4. Verify that the following message is sent to the Syslog: *"Updating Flash to work in Non Network Separation Mode in the next Boot".*
5. Repeat steps 4 to 5 with the second blade.
6. Insert both blades into the system and connect two separate Ethernet cables, one for each RTM. Remember that your OAMP applications are now available at the RTM module, as well as your Media and Call Control applications. Power up the system.

5 Standard Control Protocols

Note: Some IETF URL links referred to in this User Manual may not be active and in such a case may require an Internet search for the text required.

The device 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 VoPLib API (over PCI or over TPNCP).

For information on TPNCP, refer to the section on TPNCP in VoPLib Application Developer's Manual, Document #: LTRT-844xx).

Note: **(Applicable to** TP-260 and IPM-260 only**)**

MEGACO is currently not supported on the TP-260 and IPM-260 product family. Contact your local sales representative for further details.

5.1 MGCP Control Protocol

5.1.1 MGCP Overview

MGCP (Media Gateway Control Protocol) is a standards-based network control protocol (based on the IETF RFC 3435 and RFC 3660). 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 or require the user to use any specific software library, in order to construct a Call Agent. The user may choose any one of many such stacks available in the market.

Note: MGCP and MEGACO protocols cannot co-exist on the same device.

5.1.2 MGCP Operation

5.1.2.1 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 the 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 or more 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.

5.1.2.2 MGCP Call Agent Configuration

The Call Agent can be configured using three different methodologies:

5.1.2.2.1 Resolving the Host Name Fully Qualified Domain Name (FQDN) Address via the DNS Server

In the first option, the Call Agent is defined as an FQDN address, to be resolved by a DNS server. The DNS server can return a single IP address or a list of up to 10 IP addresses.

The restart procedure is complete only if the DNS successfully returns the DNS query. In this case, the host name is resolved and the device can work with the IP address (or list of addresses).

If resolving the host name fails, the device keeps trying to resolve it until the DNS returns the resolution successfully and a valid IP address is issued.

If first IP address in the DNS list stops responding, the re-transmission mechanism continues trying to send its commands to the next IP address in the DNS list.

The DNS look-up methodology *ini* file configuration is shown below:

```
CallAgentDomainName = 'domain name'
DNSPRISERVERIP = IP address
DNSSECSERVERIP = IP address
CallAgentPort = Port number
```

Note: In this setup, the CallAgentIP and RedundantAgentIP parameters are ignored.

5.1.2.2.2 Configuring Primary and Secondary IP Addresses

In this option, up to two IP addresses are configured. One for the primary call agent and the other (optional) for the secondary IP address. If the primary IP address stops responding, the re-transmission mechanism tries sending its commands to the secondary IP (if the secondary IP is configured).

The IP addresses methodology *ini* file configuration is show below:

```
CallAgentDomainName = '' (two single commas indicating this is an
empty string)
CallAgentIP = IP address A
RedundantAgentIP = IP address B
CallAgentPort = Port A
RedundantAgentPort = Port B
```

5.1.3 Using a Configuration Table to Assign Endpoints

The third option offers a new matrix called CPCallManagerGroups, which is used to define the relationship between a group of trunks (endpoints in Analog gateways), and primary and secondary MGCs. The new matrix uses the following parameters:

- ProvisionedCallAgents

- ProvisionedCallAgentsPorts
- CallAgentDomainName (Note that this is the same parameter used in option one, but here it has an enhanced meaning).

Note: This matrix is an offline parameter, and changes are applied only by resetting the device.

Each row in the matrix includes the following:

- List of relevant trunks/endpoints
- Call agent type (DNS or IP)
- Index of the primary call agent in the ProvisionedCallAgents
- Index of the secondary call agent in the ProvisionedCallAgents

If the Call Agent type is DNS, the index fields are not relevant, as this refers to the CallAgentDomainName parameter. This implies that when DNS is used, there is no primary and secondary call agent. However, DNS can be used for one line in the matrix, while IP can be used for another. Note also that only **one** DNS can be configured on the device by using the CallAgentDomainName parameter. Therefore, there is no possibility to configure a different DNS to different trunk groups.

Up to ten lines can be added to the matrix, as only ten different MGCs (or one DNS) are allowed.

Matrix Example:

Assuming the following parameters are defined in the *ini* file:

ProvisionedCallAgents = 10.0.0.1,10.0.0.2,10.0.0.3,10.0.0.4,10.0.0.5
ProvisionedCallAgentsPorts = 2427, 2427, 2427, 2427, 2427
CallAgentDomainName = 'user.corp.com'

and the following matrix appears as follows:

Table 5-1: CPCallManagerGroups Example

Group Id	Group Members	MGC Type	Primary MGC	Secondary MGC
1	"1,5-8"	0	1	2
2	“0,2,4,10-13”	0	3	0
3	“3,14,15”	1	0	0
4	“DEFAULT LINE”	0	4	5

In the above matrix, the primary MGC of Group 1 is 10.0.0.1 and the secondary MGC is 10.0.0.2.

The primary MGC of group 2 is 10.0.0.3 and there is no secondary MGC.

Group 3 uses ‘user.corp.com’ DNS entry to locate the MGC.

Group 4 is the default group for all the trunks not defined in previous groups. The primary MGC is 10.0.0.4 and the secondary MGC is 10.0.0.5.

5.1.3.1.1 Field Descriptions

1. **Group Members** - This field contains (in string format) the list of all the members of this group. The list is separated by commas, and can include ranges. The numbers refers to trunks in digital gateways and to endpoints in analog gateways. A special line can be used as the default group. This line MUST contain the string “DEFAULT LINE”. The meaning of this line is that every trunk/endpoint which is not defined in one of the other groups will belong to this line. Note that if two default lines are entered, the first one only will be used.
2. **MGC Type -** The field values are:
   a. 0 – IP address (use the ProvisionedCallAgents parameter)
   b. 1 – DNS (use the CallAgentDomainName parameter)
3. **Primary MGC** - This field contains the index of the primary MGC in the ProvisionedCallAgents parameter. Note that the first index is 1, and the value 0 means no value.
4. **Secondary MGC** - This field contains the index of the secondary MGC in the ProvisionedCallAgents parameter. Note that the first index is 1, and the value 0 means no value.

5.1.3.1.2 Configuring the Matrix in the ini File:

The following is an example of an ***ini*** file table, to be used as a basis for validation tests. Note that the FORMAT line is a single line (word-wrapped for readability).

```
[ CPCallManagerGroups]
FORMAT CPCallManagerGroups Index =
CPCallManagerGroups GroupMembersList, CPCallManagerGroups MGCType,
CPCallManagerGroups PrimaryMGCIndx,
CPCallManagerGroups SecondaryMGCIndx ;
CPCallManagerGroups 0 = "1,5-8",        0, 1, 2 ;
CPCallManagerGroups 1 = "0,2,4,10-13",  0, 3, 0 ;
CPCallManagerGroups 2 = "3,14,15",      1, 0, 0 ;
CPCallManagerGroups 3 = "DEFAULT LINE", 0, 4, 5 ;
[ /CPCallManagerGroups]
```

5.1.3.2 Configuration and Update of the Endpoint's Notified Entity

All endpoints used by the same gateway can hold up to 20 different FQDN addresses. All commands containing the twenty-first or higher FQDN are rejected using the error code 502. If an IP address is used to identify notified entities, the number of IP addresses is limited to the number of endpoints, e.g., each endpoint may hold a different IP address of its notified entity.

The notified entity configuration is done using the N: line in accordance with RFC 3435.

5.1.4 MGCP Endpoints Names

MGCPTrunkNamingPattern and MGCPEndPointNamingPattern .ini file parameters are used to configure the MGCP endpoint naming:

- MGCPTrunkNamingPattern - A string for parameter with a maximum length of 64 characters, used only for digital devices. The parameter enables a user to configure the endpoint name using a string pattern such as "ds/tr*/*". The asterisks will be replaced by the trunk and B-channel accordingly. The pattern must be of the form "STR1*STR2*", where STR1 and STR2 are free text strings that should contain one slash sign ("/") each.

- MGCPEndPointNamingPattern - A string for parameter with a maximum length of 64 characters, used only for analog devices. The parameter enables a user to configure the endpoint name using a string pattern such as "ACgw*". The asterisk will be replaced by the relevant line number. The pattern must be of the form "STR1*", where STR1 is free text string that should contain one slash sign ("/").

5.1.5 MGCP KeepAlive Mechanism

The KeepAlive mechanism maintains a constant connection with the Call Agent. In case of a Call Agent failure, the device will enter into a disconnected state and will switch over to its redundant Call Agent. Moreover, since constant transportation is running between the Call Agent and the device, using the KeepAlive mechanism 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 loses 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.

KeepAlive *ini* file parameters:

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

KeepAliveInterval = 12 by default - This parameter is used to define the interval in seconds of a KeepAlive message

KeepAlive examples:

While working in endpoint naming conventions:

RSIP 2200 *@audiocodes.com MGCP 1.0

RM: X-KeepAlive

While working in trunk naming conventions:

RSIP 2420 ds/tr/*/*@audiocodes.com MGCP 1.0

RM: X-KeepAlive

5.1.6 MGCP Piggy-Back Feature

The RFC 3435 and PacketCable specifications define a piggy-backing mechanism that group commands according to their destination and send them as a single UDP command.

This feature is set via the EnablePiggyBacking ***ini*** file parameter (default = ON, e.g., EnablePiggyBacking = 1).

If the piggy-back feature is active, all outgoing commands are kept in a buffer according to its destination. Every 30 msec, all occupied buffers are cleared and all commands held in the buffers are piggy-backed and sent according to the FIFO methodology.

5.1.7 Device Distinctive Ringing Mechanism

Note: The following sub-section on Distinctive Ringing Mechanism is applicable to **MediaPack** only.

The device supports an advanced Distinctive-Ringing mechanism This feature configures the ringing frequency and multiple ringing cadences.

The ringing types are configured inside the Call Progress Tone file. For configuration and call progress tone creation refer to, 'Modifying the Call Progress Tones File & Distinctive Ringing File' on page 584.

The MGC instructs the gateway to play a specific ringing signal using the following commands: l/r1 or h/r1 (where r1 can be replaced with r0-r7).

Note that since Ringing Pattern #0 (in the CPT file) is reserved for the l/rg or h/rg signal commands, the signals that are used in the l/r1 or h/r1 commands are offset by one (i.e., r0 is represented by Ringing Pattern #1, r1 is represented by Ringing Pattern #2 and so forth).

5.1.8 SDP Support in MGCP

MGCP supports basic SDP (Session Description Protocol), as defined in RFC 2327. It also supports the Silence Suppression attribute defined in SDP-ATM. The 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 codec 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 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)

Note: Additional extensions to the SDP are also supported. The RTPMAP attribute must appear before FMTP.

- Other specific functionalities are defined in the following sections.

5.1.8.1 RFC 3407 Support - Capability Declaration

RFC 3407 defines a capability declaration feature in SDP by defining a set of new SDP attributes. Together, these attributes define a capability set, which consists of a capability set sequence number ('sqn') followed by one or more capability descriptions ('cdsc'). Each capability description in the set contains information about supported media formats.

In order for the gateway to support this feature the ini file parameter 'CPSdpProfile' must be set to a value that turns on the first bit (e.g., CPSdpProfile = 1 or CPSdpProfile = 3).

A returned SDP containing a capabilities description may look like the following (capability parts are bold):

```
v=0
o=- 298209245 1 IN IP4 10.4.3.96
s=-
c=IN IP4 10.4.3.96
a=sqn: 1
t=0 0
m=audio 4020 RTP/AVP 4
a=rtpmap:4 G723/8000/1
a=fmtp:4 bitrate=6.3;annexa=yes
a=cdsc: 1 image udptl t38
a=cpar: a=T38FaxUdpEC:t38UDPRedundancy
a=cpar: a=T38FaxMaxBuffer:1024
a=cpar: a=T38FaxMaxDatagram:238
a=cdsc: 2 audio RTP/AVP 0 8 97 98 2 99 105 106 107 108 109 110 111 112 113 4 80 18 3 116 96 104 13 120
a=cpar: a=rtpmap:97 G726-16/8000/1
a=cpar: a=rtpmap:98 G726-24/8000/1
a=cpar: a=rtpmap:99 G726-40/8000/1
a=cpar: a=rtpmap:105 X-G727-16/8000/1
a=cpar: a=rtpmap:106 X-G727-24-16/8000/1
a=cpar: a=rtpmap:107 X-G727-24/8000/1
a=cpar: a=rtpmap:108 X-G727-32-16/8000/1
a=cpar: a=rtpmap:109 X-G727-32-24/8000/1
a=cpar: a=rtpmap:110 X-G727-32/8000/1
a=cpar: a=rtpmap:111 X-G727-40-16/8000/1
a=cpar: a=rtpmap:112 X-G727-40-24/8000/1
a=cpar: a=rtpmap:113 X-G727-40-32/8000/1
a=cpar: a=rtpmap:4 G723/8000/1
a=cpar: a=fmtp:4 bitrate=*;annexa=yes
a=cpar: a=rtpmap:80 G723/8000/1
a=cpar: a=fmtp:80 bitrate=*;annexa=yes
a=cpar: a=fmtp:18 annexb=yes
a=cpar: a=rtpmap:116 X-CCD/8000/1
a=cpar: a=rtpmap:96 telephone-event/8000
a=cpar: a=fmtp:96 0-15
a=cpar: a=rtpmap:104 RED/8000
a=cpar: a=fmtp:104
a=cpar: a=rtpmap:120 no-op/8000
```

5.1.9 MGCP Fax

5.1.9.1 MGCP Fax Configuration

MGCP offers the following Fax configurations.

- MGCP Fax package

- Proprietary change-fax-transport type in the local connection options (refer to 'Fax Transport Type Setting with Local Connection Options' on page 284) – enables changing the fax transport type without using the T.38 fax package.
- MGCP Fax profile “Display Fax Port on Second SDP M Line” (refer to 'MGCP Profiling' on page 288). enables negotiating the T.38 fax port without using the T.38 fax package.

Note: The following table is NOT applicable to MediaPack.

Table 5-2: MGCP Fax Package Gateway Mode

Gateway CH 0	Call Agent	Gateway CH 1
200 17501 OK	RQNT 17501 ACgw0@[10.4.4.129] MGCP 1.0 X: 12 S: L/dl R: D/X(D) D: 2xxx	
NTFY 2075 ACgw0@[10.4.4.129] MGCP 1.0 X: 12 O: 2580	200 2075 OK	

200 17502 OK
I: 34
 	
v=0
o=- 767771419 0 IN IP4 10.4.4.129
s=-
c=IN IP4 10.4.4.129
t=0 0
m=audio 4000 RTP/AVP 8
	CRCX 17502 ACgw0@[10.4.4.129] MGCP 1.0
	C: 1
	L: a:PCMA , fxr/fx:gw
	M: recvonly
	X: 12
	R: fxr/gwfax	

	CRCX 17503 ACgw1@[10.4.4.129] MGCP 1.0
	C: 1
	L: a:PCMA , fxr/fx:gw
	M: sendrecv
	X: 12
	R: fxr/gwfax
	
	v=0
		200 17503 OK
		I: 35
		
		v=0
		o=- 1973242229 0 IN IP4 10.4.4.129
		s=-
		c=IN IP4 10.4.4.129
		t=0 0
		m=audio 4010 RTP/AVP 8

Table 5-2: MGCP Fax Package Gateway Mode

Gateway CH 0	Call Agent	Gateway CH 1

	c=IN IP4 10.4.4.129
	m=audio 4000 RTP/AVP 8
	a=sqn: 0
	a=cdsc: 1 audio RTP/AVP 8
	a=cdsc: 22 image udptl t38	

200 17504 OK
 	
v=0
o=- 767771419 1 IN IP4 10.4.4.129
s=-
c=IN IP4 10.4.4.129
t=0 0
m=audio 4000 RTP/AVP 8
	MDCX 17504 ACgw0@[10.4.4.129] MGCP 1.0
	C: 1
	I: 34
	X: 12
	R: fxr/gwfax
	L: a:PCMA
	M: sendrecv
	
	v=0
	c=IN IP4 10.4.4.129
	m=audio 4010 RTP/AVP 8
	a=sqn: 0
	a=cdsc: 1 audio RTP/AVP 8
	a=cdsc: 22 image udptl t38	

200 17505 OK	RQNT 17505 ACgw0@[10.4.4.129] MGCP 1.0 X: 12 R: fxr/gwfax	
	200 2076 OK	NTFY 2076 ACgw1@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/gwfax(start)
	RQNT 17506 ACgw1@[10.4.4.129] MGCP 1.0 X: 12 R: fxr/gwfax	200 17506 OK
NTFY 2077 ACgw0@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/gwfax(start)	200 2077 OK	

Table 5-2: MGCP Fax Package Gateway Mode

Gateway CH 0	Call Agent	Gateway CH 1
NTFY 2078 ACgw0@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/gwfax(stop)	200 2078 OK	
	200 2079 OK	NTFY 2079 ACgw0@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/gwfax(stop)
250 17507 OK	DLCX 17507 ACgw0@[10.4.4.129] MGCP 1.0	
	DLCX 17508 ACgw1@[10.4.4.129] MGCP 1.0	250 17508 OK

Note: The following table is applicable to MediaPack only.

Table 5-3: MGCP Fax Package Loose Mode

Gateway CH 0	Call Agent	Gateway CH 1
NTFY 2095 ACgw0@[10.4.4.129] MGCP 1.0 X: 12 O: hd	200 2095 OK	
200 16823 OK	RQNT 16823 ACgw0@[10.4.4.129] MGCP 1.0 X: 12 S: L/dl R: D/X(D) D: 2xxx	
NTFY 2096 ACgw0@[10.4.4.129] MGCP 1.0 X: 12	200 2096 OK	

Table 5-3: MGCP Fax Package Loose Mode

Gateway CH 0	Call Agent	Gateway CH 1
O: 2580		

200 16824 OK
I: 39
 	
v=0
o=- 1932071854 0 IN IP4 10.4.4.129
s=-
c=IN IP4 10.4.4.129
t=0 0
m=audio 4000 RTP/AVP 18
a=fmtp:18 annexb=yes
a=sqn: 21
a=cdsc: 1 audio RTP/AVP 18
a=cdsc: 2 image udptl t38
a=T38FaxUdpEC:t38UDPRedundancy
a=fmtp:18 annexb=no
	CRCX 16824 ACgw0@[10.4.4.129] MGCP 1.0
	C: 1
	X: 12
	L: a:G729 , fxr/fx:t38
	M: recvonly
	R: fxr/t38	

	CRCX 16825 ACgw1@[10.4.4.129] MGCP 1.0
	C: 1
	X: 12
	L: a:G729 , fxr/fx:t38
	M: sendrecv
	R: fxr/t38
	S: L/rg
	
	v=0
	c=IN IP4 10.4.4.129
	m=audio 4000 RTP/AVP 18
	a=sqn: 0
	a=cdsc: 1 audio RTP/AVP 18
	a=cdsc: 2 image udptl t38
		200 16825 OK
		I: 40
		
		v=0
		o=- 1895854000 0 IN IP4 10.4.4.129
		s=-
		c=IN IP4 10.4.4.129
		t=0 0
		m=audio 4010 RTP/AVP 18
		a=fmtp:18 annexb=yes
		a=sqn: 22
		a=cdsc: 1 audio RTP/AVP 18
		a=cdsc: 2 image udptl t38
		a=T38FaxUdpEC:t38UDPRedundancy
		a=fmtp:18 annexb=no

	200 2097 OK	NTFY 2097 ACgw1@[10.4.4.129] MGCP 1.0 X: 12 O: hd

200 16826 OK
 	
v=0
o=- 1932071854 1 IN IP4
	MDCX 16826 ACgw0@[10.4.4.129] MGCP 1.0
	C: 1	

Table 5-3: MGCP Fax Package Loose Mode

Gateway CH 0	Call Agent	Gateway CH 1

10.4.4.129
s=-
c=IN IP4 10.4.4.129
t=0 0
m=audio 4000 RTP/AVP 18
a=fmtp:18 annexb=yes
a=sqn: 23
a=cdsc: 1 audio RTP/AVP 18
a=cdsc: 2 image udptl t38
a=T38FaxUdpEC:t38UDPRedundancy
a=fmtp:18 annexb=no
	I: 39
	X: 12
	R: fxr/t38
	L: a:G729
	M: sendrecv
	
	v=0
	c=IN IP4 10.4.4.129
	m=audio 4010 RTP/AVP 18
	a=sqn: 0
	a=cdsc: 1 audio RTP/AVP 18
	a=cdsc: 2 image udptl t38	

	200 2098 OK	NTFY 2098 ACgw1@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/t38(start)
NTFY 2099 ACgw0@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/t38(start)	200 2099 OK	

200 16827 OK
 	
v=0
o=- 1932071854 2 IN IP4 10.4.4.129
s=-
c=IN IP4 10.4.4.129
t=0 0
m=image 4002 udptl t38
a=sqn: 24
a=cdsc: 1 audio RTP/AVP 0
a=cdsc: 2 image udptl t38
a=T38FaxUdpEC:t38UDPRedundancy
	MDCX 16827 ACgw0@[10.4.4.129] MGCP 1.0
	C: 1
	I: 39
	X: 12
	R: fxr/t38
	L: a:G729
	M: sendrecv
	
	v=0
	c=IN IP4 10.4.4.129
	m=image 4012 udptl t38
	a=sqn: 0
	a=cdsc: 1 audio RTP/AVP 18
	a=cdsc: 2 image udptl t38	

NTFY 2100 ACgw0@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/t38(start)	200 2100 OK	
	MDCX 16828	200 16828 OK

Table 5-3: MGCP Fax Package Loose Mode

Gateway CH 0	Call Agent	Gateway CH 1

	ACgw1@[10.4.4.129] MGCP 1.0
	C: 1
	I: 40
	X: 12
	R: fxr/t38
	L: a:G729
	M: sendrecv
	
	v=0
	c=IN IP4 10.4.4.129
	m=image 4002 udptl t38
	a=sqn: 0
	a=cdsc: 1 audio RTP/AVP 18
	a=cdsc: 2 image udptl t38
		v=0
		o=- 1895854000 1 IN IP4 10.4.4.129
		s=-
		c=IN IP4 10.4.4.129
		t=0 0
		m=image 4012 udptl t38
		a=sqn: 25
		a=cdsc: 1 audio RTP/AVP 0
		a=cdsc: 2 image udptl t38
		a=T38FaxUdpEC:t38UDPRedundancy

NTFY 2101 ACgw0@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/t38(stop)	200 2101 OK	
	200 2102 OK	NTFY 2102 ACgw1@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/t38(start)
RQNT 16829 ACgw0@[10.4.4.129] MGCP 1.0 X: 12 R: fxr/t38	200 16829 OK	
	200 16830 OK	RQNT 16830 ACgw1@[10.4.4.129] MGCP 1.0 X: 12 R: fxr/t38
NTFY 2103 ACgw0@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/t38(start)	200 2103 OK	
	200 2104 OK	NTFY 2104 ACgw1@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/t38(stop)
NTFY 2105	200 2105 OK	

Table 5-3: MGCP Fax Package Loose Mode

Gateway CH 0	Call Agent	Gateway CH 1
ACgw0@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/t38(stop)		
NTFY 2106 ACgw0@[10.4.4.129] MGCP 1.0 X: 12 O: hu	200 2106 OK	
	200 2107 OK	NTFY 2107 ACgw1@[10.4.4.129] MGCP 1.0 X: 12 O: hu
DLCX 16831 ACgw0@[10.4.4.129] MGCP 1.0	250 16831 OK	
	250 16832 OK	DLCX 16832 ACgw1@[10.4.4.129] MGCP 1.0

5.1.10 Fax Transport Type Setting with Local Connection Options

In addition to the T.38 Fax package described in Fax Package Definition - FXR on page 315, the parameter, “x-faxtranstype” can set the Fax Transport Type of each connection to either Transparent, Relay or Transparent with Events. If this parameter is not placed in the Local Connection Options, (LCO) command, then the default value configured by the FaxTransportMode ***ini*** file parameter is set.

Table 5-4: Fax Transport Type

Fax Mode	Description
x-faxtranstype:transparent	Fax events are ignored
x-faxtranstype:relay	Faxes are transmitted on T.38
x-faxtranstype:transparentwithevents	Fax is transmitted in-band and fax events are detected

5.1.10.1 Display Fax Port on Second M Line

This feature enables users to negotiate the T.38 fax port without using the T.38 fax package. To set this feature, the FaxTransportType parameter in the ***ini*** file should be configured to relay T.38. Avoid setting the fax transport type through the MGCP local connects options field (such as in 'Fax Transport Type Setting with Local Connection Options' above).

When this feature is enabled, an SDP response includes an additional media line such as:

```
m=image 4342 udptl t38
```

This example indicates the T.38 fax port 4342 is used.

5.1.11 Voice Band Data (VBD) for MGCP

VBD is a way for a number of endpoints participating in the same connection, to decide on appropriate default coders for transporting fax and modem data. This is useful in cases where not all the participants support the T.38 ITU-T recommendation.

The feature enables the gateway to negotiate over a VBD coder used for bypass mode. That is, if we are in bypass mode, then in case a fax/modem event occurred, we will switch to the VBD coder and when the event ended, we will return to our default voice coder.

For further information regarding Voice Band Data, see ITU-T V.152 "Procedures for supporting Voice-Band Data over IP Networks".

The relevant ***ini*** file parameter is CPSDPPROFILE. Turn on the second bit to enable the VBD functionality.

There are 4 coders supporting bypass and therefore can be used for VBD:

1. PCMU
2. PCMA
3. G726-32
4. G726-40

The one who initiates the VBD coder negotiation must include in its offer, either PCMA or PCMU or both in the list of VBD coders it offers. The one that answers the offer must indicate support for at least one VBD coder which does not need to be PCMU or PCMA.

The payload type marked for VBD treatment should be a dynamic payload type.

If a coder with a static payload type (voice coder) is requested for the VBD mode, then the coder must be duplicated where the second occurrence of the coder will have a dynamic payload and will be used for VBD.

VBD coders can be used through an SDP or through the Local connection options.

5.1.11.1 SDP Usage

VBD coders are used through the SDP by the 'gpmd' (general-purpose media descriptor) attribute for associating payload types in a media information ('m') line with the VBD coder. The syntax should be:

```
"a=gpmd:<payload> vbd<yes | no>
```

For example:

```
m=audio 4000 RTP/AVP 0 96
a=rtpmap:96 PCMU/8000/1
a=gpmd:96 vbd=yes
```

A dynamic payload was given in the 'm' line. The *rtpmap* line indicated that the payload is associated with the PCMU coder and the *gpmd* line indicates that this coder is a VBD coder.

5.1.11.2 LCO Usage:

The usage of the VBD coders through the LCO is done by a new attribute added to MD package "gpmd". The command may be used in a number of ways:

1. A *gpmd* field for each of the coders:

```
L: a:codec1;codec2, md/gpmd:"codec1 vbd=yes",
                 md/gpmd:"codec2 vbd=yes"
```

2. One *gpmd* field for all of the coders:

```
L: a:codec1;codec2, md/gpmd:"codec1 vbd=yes" ;
   "codec2 vbd=yes"
3. Reference to a specific coder if same coder appears a number of
time:
L: a:codec1;codec1, md/gpmd : "codec1:2 vbd=yes".
```

(The vbd coder is the second codec1).

Note that is it possible to use the "gpmd" attribute without the package "md" prefix , that is replacing all occurrences of "md/gpmd" with "gpmd" alone.

Table 5-5: VBD Examples

Command	Expected Results

CRCX 29630 ACgw0@[10.4.4.123] MGCP 1.0
C: 1
M: recvonly
 	
v=0
o=- 776407889 0 IN IP4 10.4.4.123
s=-
c=IN IP4 10.4.4.123
t=0 0
m=audio 4020 RTP/AVP 0
a=gpmd:0 vbd=yes
m=image 4022 udptl t38
	534 29711 FAIL
	
	(The command failed since only one voice coder was given and the remote asked for this voice coder to support VBD.)

CRCX 29630 ACgw0@[10.4.4.123] MGCP 1.0
C: 1
M: recvonly
L: a:PCMU;G729;PCMA;G726-32;G726-40;PCMA,md/gpmd:"G729 vbd=yes";"G726-32 vbd=yes";"G726-40 vbd=yes",gpmd:"PCMA vbd=yes"
	200 29718 OK
	I: 21
	
	v=0
	o=- 379071889 0 IN IP4 10.4.4.123
	s=-
	c=IN IP4 10.4.4.123
	t=0 0
	m=audio 4000 RTP/AVP 0 96 97 99 8
	a=rtpmap:96 PCMA/8000/1
	a=gpmd:96 vbd=yes
	a=rtpmap:97 G726-32/8000/1
	a=gpmd:97 vbd=yes
	a=rtpmap:99 G726-40/8000/1
	a=gpmd:99 vbd=yes

Table 5-5: VBD Examples

Command	Expected Results
	(All the list of given VBD codecs are used for VBD)

CRCX 29630 ACgw1@[10.4.4.123] MGCP 1.0
C: 1
M: recvonly
 	
v=0
o=- 379071889 0 IN IP4 10.4.4.123
s=-
c=IN IP4 10.4.4.123
t=0 0
m=audio 4000 RTP/AVP 0 96 97 99 8
a=rtpmap:96 PCMA/8000/1
a=gpmd:96 vbd=yes
a=rtpmap:97 G726-32/8000/1
a=gpmd:97 vbd=yes
a=rtpmap:99 G726-40/8000/1
a=gpmd:99 vbd=yes
	200 29720 OK
	I: 23
	
	v=0
	o=- 1966564099 0 IN IP4 10.4.4.123
	s=-
	c=IN IP4 10.4.4.123
	t=0 0
	m=audio 4010 RTP/AVP 0 96
	a=rtpmap:96 PCMA/8000/1
	a=gpmd:96 vbd=yes
	
	(Only the first VBD coder is used for supporting VBD.)

CRCX 29739 ACgw0@[10.4.4.123] MGCP 1.0
C: 1
M: recvonly
L: a:PCMU;PCMA;G726-32;G726-40;G729;G729,md/gpmd:"G729:2 vbd=yes"
 	
(G729 doesn't support VBD and will be discarded)
	200 29742 OK
	I: 21
	
	v=0
	o=- 1754678337 0 IN IP4 10.4.4.123
	s=-
	c=IN IP4 10.4.4.123
	t=0 0
	m=audio 4000 RTP/AVP 0 8 2 99 18 96 97 98 99
	a=rtpmap:99 G726-40/8000/1
	a=fmtp:18 annexb=yes
	a=rtpmap:96 PCMU/8000/1
	a=gpmd:96 vbd=yes
	a=rtpmap:97 PCMA/8000/1
	a=gpmd:97 vbd=yes
	a=rtpmap:98 G726-32/8000/1
	a=gpmd:98 vbd=yes
	a=rtpmap:99 G726-40/8000/1
	a=gpmd:99 vbd=yes

Table 5-5: VBD Examples

Command	Expected Results
	(Note that the first VBD coder is PCMU and not G726-40 as expected and the list contains 4 VBD supporting coders)
CRCX 29747 ACgw1@[10.4.4.123] MGCP 1.0 C: 1 M: recvonly L: a:G726-40,md/gpmd:"G726-40:1 vbd=yes"	534 29794 FAIL (There must also be one voice coder which doesn't support VBD.)

5.1.12 MGCP Profiling

MGCP uses profiles for saving backward compatibility and certain modes of MGCP behavior. A MGCP profile can be set through the ***ini*** file “MGCPCompatibilityProfile” parameter. Different profiles are presented below. For further profiling information please contact AudioCodes support personnel.

5.1.13 TGCP Compatibility

To use Trunking Gateway Control Protocol (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.
- Support of specific TGCP packages.

5.1.14 TDM Hairpin

It is possible to "hairpin" two endpoints directly through the TDM hardware, without passing through the DSP and IP layers. This functionality ensures that no delay is induced, and correct bit-sensitive data is transmitted between the endpoints. Note that while the connection exists, no events can be detected and no signals can be sent.

To use this feature, the MGC should open a connection with a second endpoint, using the transparent coder.

1. To create the connection, issue CRCX WITH Z2 (second endpoint) + coder X-CCD (transparent). The coder name can also be written as "clearmode".

Example:

```
CRCX 19278 ds/tr0/1@[10.31.4.93] MGCP 1.0
l:a:x-ccd
M: sendrecv
z2:ds/tr0/2@[10.31.4.93]
```

2. To remove the hairpin connection, issue DLCX to both connections on both endpoints.

5.1.15 AMR Policy Management

Note: The sub-section on AMR Policy Management is applicable to **6310/8410/3000 devices**.

The AMR coder contains 8 rate options. However, the AMR specification, RFC 3267, does not define the rules for selecting among those rates. The device provides a proprietary definition policy for selecting a rate according to packet loss measurement.

This policy management enables both ends - the device gateway and the remote end (e.g., a handset) - to negotiate the current AMR rate and current AMR redundancy depth according to the voice quality of the line.

The voice quality is defined by measuring the packet loss. When one side of the call detects that the packet loss exceeded a pre-defined value, it sends (in a special field in the AMR packet called CMR) a command to change the current rate. The values of packet loss and Hysteresis are defined in the 3G specification 44.318.

Table 5-6: MultiRate Configuration Information Element

Octet 3 - n				
Threshold Value	**Frame Loss Ratio**	**Hysteresis Value**		**Frame Loss Ratio**
0	= 0 %	0	=	0 %
1	= 0.25 %	1	=	0.25 %
…		2	=	0.5 %
19	= 4.75 %	3	=	0.75 %
20	= 5 %	4	=	1 %
21	= 5.5 %	5	=	1.5 %
…		6	=	2 %
39	= 14.5 %	7	=	2.5 %
40	= 15 %	8	=	3 %
41	= 16 %	9	=	4 %
…		10	=	5 %
50	= 25 %	11	=	6 %
51	= 26 %	12	=	8 %
52	= 28 %	13	=	10 %
…		14	=	13 %
62	= 48 %	15	=	17 %

Table 5-6: MultiRate Configuration Information Element

Octet 3 - n			
Threshold Value	Frame Loss Ratio	Hysteresis Value	Frame Loss Ratio
63	= 50 %		

Each side of the call must have the same table that defines each rate that the AMR redundancy requires, as well as the change definitions accordingly.

Setting the policy table is done via the LCO part of the MGCP command. The following is a command example:

```
crcx 3 ds/01/05@TPM0Slot6.M5K  MGCP 1.0
C: 00000100AC100005496FDC41851D0505
L: p:20, a:AMR, e:off, s:off, fmtp:”amr mode-set=1,3,5,7;start-
mode=7; r=2,1,0,0;policy1= l:25,
h:10;policy2=l:15,h:10;policy3=l:5,h:4;AlertPolicy=l:40,lh:10”
M: inactive
```

This command defines the following:

1. Four rates to be used in the call. (mode-set=1,3,5,7).
2. Which of the rates is to be used at the start of the call. (A new parameter “start-mode=7”).
3. Four values for the AMR redundancy depth for each of the defined rates. (“r=2,1,0,0”).
4. Three policy management fields, (“policy1=l:25, h:10;policy2=l:15,h:10;policy3=l:5,h:4”) The first defines the rule for moving from the first rate (1 in the example) to the second rate (3 in the example) and back. The second defines the rule for moving from the second rate (3 in the example) to the third rate (5 in the example) and back, and so on. Up to 7 policy fields can be defined, for 8 different rates. Each policy field contains the following sub-fields:
   a. “l” is the packet loss level which causes moving to that rate from a higher rate.
   b. 'h” is the hysteresis level for the above packet loss level which causes moving back to the higher rate.
   c. The values of both sub-fields are enumerations according to the table “MultiRate Configuration Information Element” (shown below) as defined in the 3G specification 44.318.
5. Alert Policy Field ("AlertPolicy=l:40,lh:10") - This field defines a value of packet loss for an immediate alert. The changing of rates always is done to the neighboring rate, but the alert and alert cease events are sent immediately. The alert event is defined in the RTP package (RFC 3660) - r/qa. Unfortunately, when the normal condition return occurs, it is also required to be reported. But there is no such event in the package. Therefore, an extension to define a "quality alert cease" event - r/x-qac is used.

Table 5-7: MGCP AMS Alert Policy

Gateway	Call Agent
200 4 OK	mdcx 4 ds/01/05@TPM0Slot6.M5K MGCP 1.0

Table 5-7: MGCP AMS Alert Policy

Gateway	Call Agent
	C: 00000100AC100005496FDC41851D0505 I: 30000 L: p:20, a:AMR, e:off, s:off M: sendrecv X: 1 R:r/qa
When packet loss exceeds the permitted value, notification is sent: ntfy 20 ds/01/05@TPM0Slot6.M5K MGCP 1.0 C: 00000100AC100005496FDC41851D0505 X: 1 O: r/qa	The MGC replies and sets a request for quality alert cease notification: 200 20 OK X:2 R:r/x-qac
When conditions are back to normal, the gateway notifies: ntfy 30 ds/01/05@TPM0Slot6.M5K MGCP 1.0 C: 00000100AC100005496FDC41851D0505 X: 2 R:r/x-qac	200 2096 OK

5.1.16 Creating Conference Calls

Note: The sub-section on Creating Conference Calls is only applicable to **IPmedia** family devices.

MGCP does not support virtual endpoints even though the functionality is mentioned in RFC 3435 (though not appropriately defined). The AudioCodes conference package supplies an efficient alternative. It has been built to provide users with conference with up to 64 users per call while still allowing the device to handle its resources instead of the Call Agent. The maximum number of users in all conference calls is specific to the relevant device, and can be up to 2016 users for the 6310/8410/3000 devices and 120 users for the 1610/2000 devices.

5.1.16.1 Creating a Conference Call

When the Call Agent initiates a new conference call, it is highly recommended to include the "any endpoint" symbol ('$') as the endpoint number when creating the connection. This allows the device to look for the available resource (for a simplicity free endpoint). The returned endpoint is the conference handler, e.g., from now until the last conference user is removed, all new users are added to the call by creating the connection through this endpoint. An endpoint that becomes the conference handle rejects all non-conference calls.

The Call Agent or device selects an "endpoint" and uses it as a conference handler (point of contact for a specific conference call).

When the first connection is made on an endpoint with “conference” as the connection mode, the endpoint is marked as a conference handler. Users can be added to the conference with create connection commands to the same endpoint.

In the case of an RTP user who would like to join the conference call, a DSP is allocated to process the RTP stream. For this purpose, the gateway runs a “free endpoint” algorithm (refer to "Searching for the “Free Endpoint” Algorithm" below), to search for an endpoint and uses its DSP resource. When this endpoint is found, it is blocked until it is removed from the conference call. Since the Call Agent is not aware of the selected “free endpoint”, it may try to create another conference call on one of the endpoints that have no DSP resources. To avoid call failures, it is strongly recommended that the device be allowed to select the conference handler by creating the connection with a “use any” flag (refer to "Adding an RTP User" below).

Additional users can be added to the conference by create connection to the same “conference handler” responding from the device. Users can select the endpoint to be the conference handler as long as the endpoint is “free”.

5.1.16.2 Searching for the “Free Endpoint” Algorithm

The Endpoint is considered “free” if:

- The endpoint signals database signals list is empty.
- The endpoint database requested events list is empty. (Persistent events are ignored.)
- No connections were found in the endpoint database.

5.1.16.3 Adding an RTP Conference User

To add an RTP stream to a conference call, the user creates a connection with remote SDP parameters.

When the user adds an RTP stream to a specific conference call, the gateway must supply the DSP resource to process the RTP stream. When a “free” endpoint is found, the endpoint is blocked from getting any commands until it is removed from the conference call.

If the endpoint is blocked, it cannot be available for regular calls and cannot be added to other conference calls.

5.1.16.4 Adding a TDM Conference User

To add a TDM Conference user, use Z2 in the CRCX command to specify the requested trunk B-channel. (For examples, refer to Creating a Conference on page 293.) This supports endpoint naming and trunk naming formats.) When an endpoint is added to the conference, no other connections can be made over this endpoint until it is removed from the conference. The user can add the conference handler endpoint to the conference by placing it in the second endpoint ID.

5.1.16.5 Conference Restrictions

- Searching for a "free endpoint" algorithm ignores the persistent events ***ini*** file parameter, in which case the gateway may find an endpoint with an empty requested events list for an RTP conference user.
- If the Call Agent requests a new conference call without a "use any"

($@AudioCodes) flag and the endpoint is already used in another active call, a 502 error may be issued to the Call Agent.

- If the TDM user is asked to be added to an active conference call and it was previously used by a "free endpoint" algorithm to add an RTP user, a 502 error may be issued to the Call Agent.

5.1.17 Conference Configuration

5.1.17.1 *ini* File Configuration (Optional)

- ConferenceMaxUsers - Sets the initial number of users in a conference call that have reserved resources. This parameter is set in conference call setup. Range is “3 to 64”. Default value is 3.
- ConferenceMaxSimultaneousSpeakers - Default value is 3.
- ConferenceSignalGenerationEnable - Default value is 1.

Conference Parameter Package

Package Name: cnf

Version: 0

While using this package, the user can set each conference call properties individually.

- ConferenceMaxUsers - local connection options parameter. When a new conference call is created and the number of users is previously known, the device can save conference resources for all users in advance. Using this parameter could prevent failure as a result of insufficient resources while adding new users. If the Call Agent adds users to a conference call and the number of users exceeded the value set for ConferenceMaxUsers, a new user is added if the gateway has free conference resources, If the gateway is out of conference resources, the CRCX command is rejected and an Error 502 is returned, indicating an out of resources error.

Package format

L: cnf: maxconfusers =number, confusertype= confusertypevalue

L: cnf: maxconfusers =number, cnf, confusertype= confusertypevalue

Maxconfusers Number = 3 to 64

Confusertypevalue = regular\listener\master or 1\2\3

regular - the user can talk and listen

listener - the user can listen only.

master - the user has priority to talk within “conferencemaxsimultaneousspeakers” range.

5.1.18 Examples of Creating a Conference

To create a new conference call, use the "any endpoint" wild card to signal the device that a new conference call is to start. The device provides the conference handler. The device responds with a specific endpoint as the conference handler. All other users are added\removed from this conference-handler\endpoint.

5.1.18.1 Creating a Conference Using RTP

Conference users can each specify a coder, packetization, etc. All regular call rules are valid, e.g., a coder can be specified in the local connection options as well as in the remote SDP. The device treats all RTP users as a standard "RTP to B-channel" connection, but the connection's conference mode directs the stream to the conference chip.

```
First user is an RTP side user

CRCX 1931 $@[10.4.10.126] MGCP 0.105
C: 111
L: cnf:maxconfusers = 5;confusertype = regular
M: confrnce

v=0
c=IN IP4 10.4.12.12
m=audio 5000 RTP/AVP 0

200 1931 OK
I: 27
Z: ACgw0@AudioCodes.Com

v=0
o=- 1178418554 0 IN IP4 10.4.10.126
s=phone-call
c=IN IP4 10.4.10.126
t=0 0
m=audio 4000 RTP/AVP 0
```

5.1.18.1.1 Adding an RTP User

The Call Agent asks for the conference handler provided by the device to add another RTP user.

```
CRCX 31376 ACgw0@[10.4.10.126] MGCP 0.105
C: 1
L: cnf:confusertype = regular
M: confrnce

v=0
c=IN IP4 10.4.13.67
m=audio 6540 RTP/AVP 0

200 31376 OK
I: 22

v=0
o=- 1596935742 0 IN IP4 10.4.10.126
s=phone-call
c=IN IP4 10.4.10.126
t=0 0
m=audio 4010 RTP/AVP 0
```

5.1.18.1.2 Creating a Conference Using TDM

First user is a TDM side user - The device allocates the conference handler connection for the first TDM user and returns the conference-handler endpoint to the Call Agent in the specific endpoint field:

```
CRCX 31359 $@[10.4.10.126] MGCP 0.105
C: 1
L: cnf:confusertype = regular
M: confrnce
Z2: ACgw10@[10.4.10.126]
```

```
200 31359 OK
I: 27
Z: ACgw0@AudioCodes.Com
```

5.1.18.1.3 Adding a TDM User

The Call Agent asks the conference handler provided by the gateway to add another TDM user.

```
CRCX 31357 ACgw0@[10.4.10.126] MGCP 0.105
C: 1
L: cnf:confusertype = regular
M: confrnce
Z2: ACgw11@[10.4.10.126]

200 31373 OK
I: 21
```

5.1.19 CALEA (Communications Assistance for Law Enforcement Agencies)

CALEA Electronic Surveillance enables the conduct of lawfully-authorized electronic surveillance.

While Electronic Surveillance is activated, both bi-directional connection RTP streams are duplicated and sent to the LEA server.

The connection sides are not effected while Electronic Surveillance is performed.

The feature is described in the PacketCable specification, Appendix H of

www.packetcable.com/downloads/specs/PKT-SP-TGCP-I10-050812.pdf

Activating this feature requires the BCT/CALEA Feature key.

If the CALEA is activated on a call, and, due to lack of device resources, electronic surveillance is not possible, the call does not fail and appropriate error messages are issued notifying that call was established without CALEA (error codes 211-214). For more information, refer to the specification mentioned above.

Note that the PacketCable specification forbids enabling electronic surveillance on a connection which was initiated without electronic surveillance. ES parameters must be specified in the CRCX command, and may later be modified in MDCX commands.

The following is an example of a CRCX command with electronic surveillance parameters:

```
CRCX 1204 ACgw0@AudioCodes.com MGCP 1.0
C: 1
L: p:20, a:PCMU, es-cci:123456, es-ccd:[1.2.3.4]:9000
M: recvonly
X: 1234
```

The following is an example of a MDCX command with electronic surveillance parameters:

```
MDCX 1206 ds ACgw0@AudioCodes.com MGCP 1.0
C: 1
I: 21
L: p:20, a:PCMU, es-cci:123456, es-ccd:[1.2.3.4]:9000
M: sendrecv
X: 1234
```

5.1.20 RTP Media Encryption - RFC 3711 Secure RTP

The SRTP (RFC 3711) media encryption standard is partially supported.

RFC 3711 defines a media profile “RTP/SAVP” which is used when working in secured streams.

The SRTP defines how to encrypt the media, but does not define how to negotiate the encryption keys on the control level. The method used to negotiate the encryption keys is defined in RFC 4568.

This RFC defines a cryptographic attribute for SDP to be used for media encryption.

There is no official definition for how to use this in MGCP. Therefore, rules were developed for the device and detailed below.

5.1.20.1 Supported Suites

SRTP implementation is limited to AES_CM_128_HMAC_SHA1_32 and AES_CM_128_HMAC_SHA1_80. All other suites are ignored.

The only supported key parameter is *MKI* (Master Key Identifier). The length of the *MKI* is limited to 4 bytes. If the remote side sends a longer *MKI*, this specific key will be ignored. This means that if this is the only key, the call will fail.

The key lifetime field is not supported. However, if it is included in the key it will be silently ignored and the call will not fail.

While an SRTP suite may hold many keys and key parameters, the device supports a single key. Suites that are provided with more than one valid key are ignored and marked as not valid.

5.1.20.2 Supported Session Parameters

The following session parameters are supported:

- UNENCRYPTED_SRTP
- UNENCRYPTED_SRTCP
- UNAUTHENTICATED_SRTP

Session parameters should be the same for both the local and remote sides. When the device initiates the call, the session parameters will be defined according to *ini* file parameters (see below). When the device is the answering side, the parameters are adjusted according to the remote offering.

Unsupported session parameters are ignored, and will not cause a call failure. Note, however, that our implementation has a limitation in supporting un-authentication and un-encryption together on the same side. This combination will cause the specific line to be ignored.

5.1.20.3 Configuration and Activation

The following defines the encryption support level and default values:

1. **DSP template** - Configures the DSP Template that supports SRTP.
2. **Feature Key** – Defines if media encryption is enabled on the device.
3. ***ini* file parameter** – The *EnableMediaSecurity* parameter defines SRTP support when set to Enable, e.g., EnableMediaSecurity = 1.
4. ***ini* file parameter** – The *SRTPTxPacketMKIASize* parameter defines the length of the local MKI, used to identify the local key. The range of this parameter is 0-4.

5. ***ini* file parameter** – The *RTPEncryptionDisableTx* parameter can be set in order to work with non-encrypted RTP.
6. ***ini* file parameter** – The *RTCPEncryptionDisableTx* parameter can be set in order to work with non-encrypted RTCP.
7. ***ini* file parameter** – The *RTPAuthenticationDisableTx* parameter can be set in order to work with non-authenticated RTP.

The local descriptor may contain more parameters regarding the encryption, and these are described in the following paragraphs.

5.1.20.4 SRTP Local Connection Option Format

Use of SRTP LCO parameters is described below, in Secured Connection Negotiation.

Table 5-8: SRTP ABNF Parameter Description

Parameter	Description
LocalOptionValue=	("srtp" ":" EncryptionAlgorithm) Or ("x-srtp" ":" EncryptionAlgorithm)
EncryptionAlgorithm =	algorithmName 0*(";" algorithmName)
algorithmName =	AES_CM_128_HMAC_SHA1_32, AES_CM_128_HMAC_SHA1_80 F8_128_HMAC_SHA1_32 SRTP_SUITE_NULL

5.1.20.5 SDP Definition

The following attribute is defined in RFC 4568.

```
a=crypto:<tag> <crypto-suite> <key-params> [<session-params>]
```

The fields "tag", "crypto-suite", "key-params|, and "session-params" are described below. An example of the crypto attribute for the "RTP/SAVP" transport is provided, i.e., the secure RTP extension to the Audio/Video Profile [srtp].

In the following, new lines are included for formatting reasons only:

```
a=crypto:1_AES_CM_128_HMAC_SHA1_80
inline:PS1uQCVeeCFCanVmcjkpPywjNWhcYD0mXXtxaVBR|2^20|1:32
```

All mandatory and optional fields in the SRTP crypto attribute are parsed, but non-supported fields are ignored.

Valid SRTP attribute line format:

```
a=crypto:1 AES CM 128 HMAC SHA1 80
inline:PS1uQCVeeCFCanVmcjkpPywjNWhcYD0mXXtxaVBR
```

Endpoint Capability

While SRTP is enabled, upon auditing the endpoints, all supported SRTP suites will be returned. Refer to the example below:

```

Audit Endpoint
AUEP 15959 ds/tr0/1@[10.4.4.129] MGCP 1.0
F: A
Audit redpond
200 15959 OK
```

```
A: nt:IN , v:G;D;T;L;R;A;M;MS;DT;MD;MO;BL;FXR;FM;IT,
a:PCMA;PCMU;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;
G723;G723Low;G729A;G728;Transparent;G729E;Telepone-Event;RED;CN;no-
op;image/t38,m:sendonly;recvonly;sendrecv;inactive;netwloop, x-
srtp:AES_CM_128_HMAC_SHA1_32;AES_CM_128_HMAC_SHA1_80;SRTP_SUITE_N
ULL
```

5.1.20.6 Secured Connection Negotiation

The examples below show the creation of a secured connection via CRCX and MDCX commands.

5.1.20.6.1 SRTP Negotiation

1. If the User\ Call Agent does not provide LCO SRTP information or SDP line attributes, the Gateway returns the supported suites.

Simple create connection.

```
CRCX 15936 ds/tr0/1@[10.4.4.129] MGCP 1.0
C: 1
L: a:PCMA
M: recvonly
```

All supported packages are provided in local connection options.

```
200 15936 OK
I: 22

v=0
o=- 1147873153 0 IN IP4 10.4.4.129
s=-
c=IN IP4 10.4.4.129
t=0 0
m=audio 4000 RTP/AVP 8
a=rtcp-xr:
a=crypto:1 AES CM 128 HMAC SHA1 32
inline:MKHEBFC/PMKHEB+CJfvspnkheifcZW
a=crypto:2 AES CM 128 HMAC SHA1 80
inline:9630xvspsqnkhecZEC/8520xurolpm
m=image 4002 udptl t38
```

Terminated side gets originated side information.

```
CRCX 15938 ds/tr0/2@[10.4.4.129] MGCP 1.0
C: 1
L: a:PCMA
M: sendrecv

v=0
o=- 1147873153 0 IN IP4 10.4.4.129
s=-
c=IN IP4 10.4.4.129
t=0 0
m=audio 4000 RTP/AVP 8
a=rtcp-xr:
a=crypto:1 AES CM 128 HMAC SHA1 32
inline:MKHEBFC/PMKHEB+CJfvspnkheifcZW
```

```
a=crypto:2 AES CM 128 HMAC SHA1 80
inline:9630xvspsgnkhecZEC/8520xurolpm
m=image 4002 udptl t38
```

Terminated response with its local information.

```
200 15938 OK
I: 23

v=0
o=- 294810044 0 IN IP4 10.4.4.129
s=-
c=IN IP4 10.4.4.129
t=0 0
m=audio 4010 RTP/AVP 8
a=rtcp-xr:
a=crypto:1 AES CM 128 HMAC SHA1 32
inline:EbYVSPNKNLIFC/966j030xvspmjheh
m=image 4012 udptl t38
```

Originating side gets termination side information.

```
MDCX 15940 ds/tr0/1@[10.4.4.129] MGCP 1.0
C: 1
I: 22
L: a:PCMA
M: sendrecv

v=0
o=- 294810044 0 IN IP4 10.4.4.129
s=-
c=IN IP4 10.4.4.129
t=0 0
m=audio 4010 RTP/AVP 8
a=rtcp-xr:
a=crypto:1 AES CM 128 HMAC SHA1 32
inline:EbYVSPNKNLIFC/966j030xvspmjheh
m=image 4012 udptl t38
```

Update succeeded.

```
200 15940 OK
```

2. LCO Only

Selection of a specific package.

```
CRCX 15963 ds/tr0/1@[10.4.4.129] MGCP 1.0
C: 1
L: a:PCMA,x-SRTP:AES CM 128 HMAC SHA1 32
M: recvonly
200 15963 OK
I: 21

v=0
o=- 377373126 0 IN IP4 10.4.4.129
s=-
c=IN IP4 10.4.4.129
```

```
t=0 0
m=audio 4000 RTP/AVP 8
a=rtcp-xr:
a=crypto:1 AES CM 128 HMAC SHA1 32
inline:NLIFC/QNKHEB/8520tqtrolifcaXnk
m=image 4002 udptl t38
```

Selection of a valid package

```
CRCX 15964 ds/tr0/2@[10.4.4.129] MGCP 1.0
C: 1
L: a:PCMA,x-SRTP:F8 128 HMAC SHA1 32;AES CM 128 HMAC SHA1 32
M: recvonly
200 15964 OK
I: 22

v=0
o=- 1862533257 0 IN IP4 10.4.4.129
s=-
c=IN IP4 10.4.4.129
t=0 0
m=audio 4010 RTP/AVP 8
a=rtcp-xr:
a=crypto:1 AES CM 128 HMAC SHA1 32
inline:pA9631yvspnkhebZKPMJGEB+OLJGDA
m=image 4012 udptl t38
```

5.1.20.6.2 Negotiation Errors

- If LCO was provided with no valid SRTP suites, a 532 error will be returned.
- If SDP was provided with no valid SRTP suites, a 505 error will be returned.
- If both local connection and remote connection SRTP were provided and no match was found, a 506 error will be returned.

5.1.20.6.3 SDP Crypto Grammar

(For more information on ABNF, refer to RFC 4568.

The ABNF grammar for the generic crypto attribute is listed below, followed by the ABNF grammar for the SRTP specific use of the crypto attribute.

```
   The ABNF grammar for the crypto attribute is defined below:

   "a=crypto:" tag 1*WSP crypto-suite 1*WSP key-params
                                           *(1*WSP session-param)

   tag              = 1*9DIGIT
   crypto-suite     = 1*(ALPHA / DIGIT / " ")

   key-params       = key-param *(";" key-param)
   key-param        = key-method ":" key-info
   key-method       = "inline" / key-method-ext
   key-method-ext   = 1*(ALPHA / DIGIT / " ")
   key-info         = %x21-3A / %x3C-7E ; visible (printing)
characters
                                        ; except semi-colon
   session-param    = 1*(VCHAR)         ; visible (printing)
characters

   where WSP, ALPHA, DIGIT, and VCHAR are defined in [RFC2234].
```

```
SRTP "Crypto" Attribute Grammar
```

Below is an Augmented BNF [RFC2234] grammar for the SRTP-specific use of the SDP crypto attribute:

```
     crypto-suite   = srtp-crypto-suite
     key-method= srtp-key-method
     key-info       = srtp-key-info
     session-param  = srtp-session-param

     srtp-crypto-suite   = "AES_CM_128_HMAC_SHA1_32" /
                           "F8_128_HMAC_SHA1_32" /
                           "AES_CM_128_HMAC_SHA1_80" /
                           srtp-crypto-suite-ext

     srtp-key-method= "inline"
     srtp-key-info       = key-salt ["|" lifetime] ["|" mki]

     key-salt       = 1*(base64)   ; binary key and salt values
                                   ; concatenated together, and
then
                                   ; base64 encoded [section 6.8 of
                                   ; RFC2046]

     lifetime      = ["2^"] 1*(DIGIT)   ; see section 5.1 for "2^"
     mki            = mki-value ":" mki-length
     mki-value      = 1*DIGIT
     mki-length     = 1*3DIGIT   ; range 1..128.

     srtp-session-param  = kdr /
                           "UNENCRYPTED_SRTP" /
                           "UNENCRYPTED_SRTCP" /
                           "UNAUTHENTICATED_SRTP" /
                           fec-order /
                           fec-key /
                           wsh /
                           srtp-session-extension

     kdr  = "KDR=" 1*2(DIGIT)  ; range 0..24, power of two

     fec-order = "FEC_ORDER=" fec-type
     fec-type  = "FEC_SRTP" / "SRTP_FEC"
     fec-key   = "FEC_KEY=" key-params

     wsh       = "WSH=" 2*DIGIT    ; minimum value is 64
     base64    =  ALPHA / DIGIT / "+" / "/" / "="

     srtp-crypto-suite-ext  = 1*(ALPHA / DIGIT / "_")
     srtp-session-extension = ["-"] 1*(VCHAR)  ;visible chars
[RFC2234]
                              ; first character must not be dash
("-")
```

5.1.21 MGCP Coder Negotiation

5.1.21.1 General Background

Control protocols such as MGCP and MEGACO use a special protocol to define the stream characteristics. This protocol is called SDP – "Simple Session Description

Protocol" – and it is defined in RFC 2327. The SDP defines (among other things) the IP address and port for the session, the media type (audio for voice, data for fax), and codecs to be used for this session. Every codec is represented with the encode method and payload number.

There are two kinds of RTP payloads:

The first type is the fixed payload that was assigned to a known codec. When this kind of payload is used, there is no need for further data, as the number is worldwide accepted. Refer to 'RTP/RTCP Payload Types' on page 611 for the complete list of fixed coders.

The second type is the dynamic payload, and it is used to define any codec. The range of the dynamic payloads is 96 to 127. When defining a dynamic payload, extra data is needed to map the number to a known codec. This data can be found in the MIME registration of each codec.

5.1.21.2 MGCP Coder Negotiation (RFC 3435)

RFC 3435 defines three coder lists for coder negotiation:

- Internal coders list – this list contains the coders supported by the gateway.
- LCO list – list supplied by the Call Agent.
- RCO list – list supplied by the remote side.

Note: Refer to RFC 3435, Section 2.6, 'Use of Local Connection Options and Connection Descriptors'.

While negotiating coders, the gateway must use the following methodology:

1. If the Call Agent supplies an LCO list, the media gateway takes an intersection of the LCO and the internal coders lists.
   If no match is found, an Error 534 is returned indicating a coder negotiation error.
2. If the Call Agent supplies both an LCO and an RCO, the media gateway takes an intersection of the list from step a (above) and the RCO list.
   If no match is found, an Error 534 is returned indicating a coder negotiation error.
3. If a match is found, e.g., coders are supported by the device and appear in both lists, the media gateway uses the first voice coder. This coder appears first in the SDP response.
4. If the RCO list is supplied, an intersection is made between the RCO list and internal list.
   If no match is found, an Error 505 is returned, indicating an unsupported remote connection descriptor error.
5. If no LCO list and no RCO list were provided, the media gateway responds with all of its supported coder list i.e., Internal coder list.

   The default coder configured in the MGCPDefaultCoder ini file parameter is the first in the list.

MGCP and SDP RFCs distinguish between two types of coders: voice coders (G.711, G.729, GSM, etc.) and non-voice coders (RFC 2833, Comfort noise, VBD coder, etc.). Coder negotiation fails if no voice coder is found during the coder negotiation process.

If several voice coders and non-voice coders are supplied, the SDP response will show voice coder first in list and non-voice coders are next in list. Coder negotiation is performed on both voice coders and non-voice coders.

5.1.21.3 Coder Negotiation Configurations

The default coder can be modified in the ***ini*** file parameter, 'MGCPDefaultCoder'. An example is: MGCPDefaultCoder='G726'.

Users can load the device with a pre-defined coder table (see Coder Table File on page 587). The coder table allows the user to define per each coder its payload type, textual representation in the MGCP messages and required packetization period.

According to coder negotiation scheme above, if no coder is reported in the LCO, the default coder is used and all supported coders are reported in the SDP response. When the parameter 'UseNewFormatCoderNegotiation' is set to 1 (default), the internal coder list is reported. To prevent the gateway from sending this list, set the parameter to 0 in the ***ini*** file.

5.1.21.4 Mapping of Payload Numbers to Coders

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

Table 5-9: MGCP Mapping of Payload Numbers to Coders

Coder	**Encoding Name**	**Default Payload Number**
AMR (10.2)	"AMR",“AMR_10_2”, “AMR1020”	70
AMR (12.2)	"AMR",“AMR_12_2”,”AMR1220”	71
AMR (4.75)	"AMR",“AMR_4_75”,”AMR475”	64
AMR (5.15)	"AMR",“AMR_5_15” , “AMR515”	65
AMR (5.9)	"AMR",“AMR_5_9” ,”AMR590”	66
AMR (6.7)	"AMR",“AMR_6_7“, “AMR670”	67
AMR (7.4)	"AMR",“AMR_7_4”, “AMR740”	68
AMR (7.95)	"AMR",“AMR_7_95”, “AMR795”	69
Comfort Noise	“CN”,” COMFORT-NOISE”	13
EVRC	"EVRC"	60
EVRC (TFO)	"X-EVRC_TFO"	81
EVRC (TTY)	"X-EVRC_TTY"	85
G.711 µ–law	“PCMU”, “G711”, ”G.711”,”G.711U”, “G.711MULAW”, “G711MULAW”	0
G.726_32	"G726_32"	2
G.729E	"G729E",”G.729E”	63
G.711 A law_64	"PCMA", “G.711A”,”G.711ALAW”	8
G.723 (High)	"G723" “G.723” , “G723”, “G723HIGH”	4
G.723 (Low)	"G723LOW"	80

Table 5-9: MGCP Mapping of Payload Numbers to Coders

Coder	Encoding Name	Default Payload Number
G.726_16	"G726_16"	35
G.726_24	"G726_24"	36
G.726_40	"G726_40"	38
G.727_16	"X-G727_16",” G727”	39
G.727_24	"X-G727_24"	41
G.727_24_16	"X-G727_24_16"	40
G.727_32	"X-G727_32"	44
G.727_32_16	"X-G727_32_16"	42
G.727_32_24	"X-G727_32_24"	43
G.727_40_16	"X-G727_40_16"	45
G.727_40_24	"X-G727_40_24"	46
G.727_40_32	"X-G727_40_32"	47
G.728	"G728"	15
G.729	"G729","G.729","G729A"	18
GSM	"GSM"	3
GSM-EFR	“GSM-EFR”	84
QCELP_13	"QCELP"	62
QCELP_13_TFO	"X-QCELP_TFO"	83
QCELP_8	"X-QCELP_8"	61
QCELP_8_TFO	"X-QCELP_8_TFO"	82
Redundancy per RFC 2198	“RED”	104
RFC 2833	“telephone-event”	96
T.38 Fax	“IMAGE/T38”	No Payload
Transparent	"X-CCD", “TRANSPARENT”	56

5.1.21.5 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.

5.1.21.6 Field Descriptions

Notes for all MGCP Package tables:

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.

5.1.21.7 Generic Media Package - G

Table 5-10: 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

5.1.21.8 DTMF Package - D

Table 5-11: 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	

Table 5-11: DTMF Package - D

Symbol	Definition	R	S	Duration
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		

5.1.21.9 Line Package - L

Table 5-12: Line Package - L

Symbol	Definition	R	S	Duration
0-9, #, *, a,b,c,d	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
rg	Ringing		TO	180 sec
cf	Confirmation tone		BR	

Table 5-12: Line Package - L

Symbol	Definition	R	S	Duration
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	
r0-r7	Distinctive Ringing	x		

* Persistence Events

Note: The following paragraphs on VMWI Signal and Network Disconnect are applicable to **MediaPack and Mediant 1000 only.**

VMWI Signal

A VMWI signal can be generated as an analog signal, e.g. when an analog device raises the voltage on the telephone line, or the VMWI can be played as FSK modem signal, e.g. VMWI is transmitted in same way as Caller ID is played. The user can configure the VMWI method using the "CPPlayDigitalVMWI" ***ini*** file parameter, 0 = Analog VMWI turn the line voltage high (default), 1 = play FSK signal like caller ID.

It is highly recommended to play an FSK VMWI signal with a ringing signal, since most handsets that support the digital VMWI feature, detect the FSK signal only after the first ring.

The analog VMWI signal can be turned ON\OFF asynchronously with no relation to other signals.

Network Disconnect (OSI)

Signal Generation - Network Disconnect signal can be played on device FXS devices only. The Hook current is disconnected according to ***ini*** file parameter CurrentDisconnectDuration.

Signal Detection - Network Disconnect signal can be detected on device FXO devices only. Network disconnect can be detected by: polarity reversal, current disconnect and call progress tone.

The FarEndDisconnectType ***ini*** file parameter selects which of the methods is to be used: 1:CPT 2:PolarityReversal or 4:CurrentDisconnect

If cpt is selected, the user must specify the tone type using DisconnectToneType = call progress tone type.

For example, DisconnectToneType = 1 means DialTone triggers the network disconnected event. DisconnectToneType = 3 means BusyTone triggers the network disconnected event.

Flash Hook Event (from the IP side)

Once a Flash hook event is received via RFC 2833 (e.g., flash hook from the IP side) a wink is generated towards the relevant analog interface.

5.1.21.10 Handset Emulation Package - H

Table 5-13: 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
r0-r7	Distinctive Ringing	x		

5.1.21.11 Trunk Package - T

Note: Trunk Package - T is NOT applicable to **MediaPack.**

Table 5-14: Trunk Package - T

Symbol	Definition	R	S	Duration/Comment
co1	Continuity tone		TO	2 sec
co2	Continuity test		TO	2 sec
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 sec
of	Report failure	x		

5.1.21.12 PacketCable (NCS) Line Package - L

Table 5-15: PacketCable (NCS) Line Package - L

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 sec
cf	Confirmation tone		BR	
ci(ti, nu,na)	Caller ID		BR	$_{ti}$ denotes time $_{nu}$ denotes number $_{na}$ denotes name
dl	Dial tone		TO	
ft	Fax tone	x		
hd	Off-hook transition	P,S		
hf	Flash hook	P		

Table 5-15: PacketCable (NCS) Line Package - L

Symbol	Definition	R	S	Duration/Comment
hu	On-hook transition	P,S		
mt	Modem tones	x		
mwi	Message waiting indicator		TO	16 sec
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 sec
ro	Reorder tone		TO	180 sec
rt	Ring back tone		TO	30 sec
sl	Stutter dial tone		C,TO	180 sec
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	

5.1.21.13 Announcement Package - A

Table 5-16: Generic Media Package - A

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

5.1.21.14 RTP Package - R

Table 5-17: 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	

Note: Continuity Tests are not supported in the **MediaPack.**

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 - starts after first incoming RTCP packet. While set to zero the timer starts at once.

The following is an 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 shown in this example:

```
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)
```

Note: This is NOT applicable to MediaPack.

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

Continuity tone generation/detection is configuration dependent. To generate continuity tones and allow for their detection (if required), they must be defined by adding the following to the ***ini*** file :

ForceEchoOff=0

ENABLECONTINUITYTONES = 1

USERDEFINEDTONEDETECTORENABLE = 1

The tones should also be defined as part of the call progress tone file loaded into the device.

5.1.21.15 CAS Packages

Note: The CAS Packages are NOT applicable to **MediaPack**.

MGCP supports two CAS packages:

- MF FGD Operator Services Package (MO)
- MF Terminating Protocol Package (MT)

These packages use the E&M (ear and mouth) signaling system 'Feature Group D': MO for originating, MT for terminating.

MO is defined both in MGCP (RFC 3064) and in TGCP. MT is defined in TGCP only. The MO package is mainly used for 911, as well as general FGD originating usage. MT is used for the FGD terminating.

To use these packages, the appropriate CAS tables should be loaded.

The following events/signals are not supported:

- [MO package]
- Call Answer (ans)
- Reverse Make Busy (rbz)
- Operator Recall (rel)
- Start Wink (swk)
- [MT package]
- Call Answer (ans)

- Call Block (bl)
- Operator Interrupt (oi)
- Permanent Signal Tone (pst)

5.1.21.15.1 MF FGD Operator Services Package - MO

Table 5-18: MF FGD Operator Services Package - MO

Code	Description	Event	Signal	Additional Info
ans	Call Answer	P	-	See Notes(1)
oc	Operation Complete	x	-	
of	Operation Fail	-	-	
orbk	Operator Ringback	x	-	
rbz	Reverse make busy	P,S	-	
rcl	Operator Recall	-	BR	
rel	Release Call	P	BR	
Res	Resume Call	-	BR	
Rlc	Release complete	P,S	BR	
Sup	Call Setup	-	TO	Place call
Sus	Suspend Call	-	BR	
Swk	Start Wink	x	-	

5.1.21.16 ISUP Trunk Package - IT

Note: The ISUP Trunk Package - IT is NOT applicable to **MediaPack.**

Table 5-19: ISUP Trunk Package - IT

Symbol	Definition	R	S	Duration/Comment
co1	Continuity tone 1		TO	Time-out = 2 sec
co2	Continuity tone 2		TO	Time-out = 2 sec
ft	Fax tone		-	
ma	Media start	C	-	
mt	Modem tone		-	
oc	Operation complete		-	

Table 5-19: ISUP Trunk Package - IT

Symbol	Definition	R	S	Duration/Comment
of	Operation failure		-	
ro	Reorder tone	-	TO	Time-out = 30 sec
rt	Ring back tone	-	TO,C	Time-out = 180 sec

5.1.21.17 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)

G.723 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

G.729 Family

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

5.1.21.18 Fax Package Definition - FXR

Table 5-20: 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, which includes the following event parameters:
  - 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 includes 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, which includes the following event parameters:
  - 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

5.1.21.19 Conference Package - CNF

Note: The Conference Package - CNF is only applicable to **IPmedia**.

Package Name: cnf

Version: 0

While using this package, the user can set each conference call's properties independently.

Package format

L: cnf: maxconfusers =number; confusertype= confusertypevalue

L: cnf: maxconfusers =number,cnf: confusertype= confusertypevalue

Maxconfusers Number = 3-64

Confusertypevalue = regular\listener\master or 1\2\3

- Regular - The user can talk and listen.

- Listener - The user can listen only.
- Master - The user has priority to talk within "ConferenceMaxSimultaneousSpeakers" range.

5.1.21.20 Extended Line Package - XL

Table 5-21: Extended Line Package - XL

Symbol	Definition	R	S	Duration/Comment
rev	activates or switches off line reversal on an endpoint	x	TO	Infinite

5.1.21.21 V5 Package Definition X-v5

Note: The following V5 Package Definition table is NOT applicable to **MediaPack.**

Table 5-22: V5 Package Definition

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

Note: The following Base Package Definition table is only applicable to **IPM.**

5.1.21.22 Base Audio Package - BAU

The Base Audio Package (BAU) is defined in the PacketCable Audio Server Protocol Specification, PKT-SP-ASP-I02-010620. This event package provides support for the standard IVR operations of PlayAnnouncement, PlayCollect, and PlayRecord. It supports direct references to simple audio, as well as indirect references to simple and complex audio. It provides audio variables, control of audio interruption, digit buffer control, special key sequences, and support for re-prompting during data collection.

Table 5-23: BAU Package Definition

Symbol	Definition	R	S	Duration/Comment
ma(parms)	Manage audio		BR	
Oc	Operation complete	x		
of(parms)	Operation failed	x		
pa(parms)	Play announcement		TO	variable
pc(parms)	Play collect		TO	variable
pr(parms)	Play record		TO	variable

5.1.21.23 Signal List Package - SL

The Signal List package allows the playing of more than one brief or timeout signal at a time. The package is defined in RFC 3660.

Table 5-24: Signal List Package Definition

Symbol	Definition	R	S	Duration/Comment
oc	Operation complete	x		
of	Operation failed	x		
s(list)	Signal List		TO	variable

5.1.21.24 NCS V5 SCN Line Package - E (Applicable to MediaPack only)

The E package (as defined in ETSI TS 101 909-4 V 1.3.1 (2002-12) maps the V5 protocol messages into signals and events. The package is supported only under the NCS profile, and support is offered as described in the table below.

Table 5-25: Table 0- NCS V5 SCN Line Package Definition

Symbol	Definition	R	S	Duration/Comment
oc	Operation Complete	x		
of	Operation Failed	x		
ss(lt=parm)	Steady Signal		BR	parm = rp (Reversal Polarity) or np (Normal Polarity).
ps(lt=mpb, rep=1)	Pulsed Signal		BR	The only supported parameters are the mpb (Metering Pulse Burst) and

Table 5-25: Table 0- NCS V5 SCN Line Package Definition

Symbol	Definition	R	S	Duration/Comment
				rep=1 (one repetition).
cr(parm)	Cadence Ringing		TO	parm is the Cadence Ringing index, which can be 0-7.

5.1.22 Compression Coders

Note: The following sub-section on Compression Coders is NOT applicable to **MediaPack**.

MGCP supports the compression Coders listed in the Coder Table File on page 587 section.

The following table lists potential coders (actual coder support depends on the specific DSP template version set on the device) and their default textual representation in MGCP (textual representation may be changed via Coder Table file).

Table 5-26: Compression Coders

Coder	MGCP Textual Name
AMR (10.2)	“AMR”, “AMR_10_2”, “AMR-10-2”, “AMR1020”, ”AMR2”
AMR (12.2)	“AMR”, “AMR_12_2”, “AMR-12-2”,”AMR1220”, ”AMR2”
AMR (4.75)	“AMR”, “AMR_4_75”, “AMR-4-75”, ”AMR475”, ”AMR2”
AMR (5.15)	“AMR”, “AMR_5_15” , “AMR-5-15” , “AMR515”, ”AMR2”
AMR (5.9)	“AMR”, “AMR_5_9” , “AMR-5-9” , ”AMR590”, ”AMR2”
AMR (6.7)	“AMR”, “AMR_6_7“,“AMR-6-7“, “AMR670”, ”AMR2”
AMR (7.4)	“AMR”, “AMR_7_4”, “AMR-7-4”, “AMR740”, ”AMR2”
AMR (7.95)	“AMR”, “AMR_7_95”, “AMR-7-95”, “AMR795”, ”AMR2”
AMR-WB	"AMR-WB", "AMR_WB"
Comfort Noise	“CN”,” COMFORT-NOISE”
EVRC	"EVRC"
EVRC (TFO)	"X-EVRC-TFO", "EVRC_TFO", "EVRC-TFO"
EVRC (TTY)	"X-EVRC-TTY", "EVRC_TTY", "EVRC-TTY"
EVRC0	“EVRC0”

Table 5-26: Compression Coders

Coder	MGCP Textual Name
EVRC1	“EVRC1”
EVRCB	“EVRCB”
EVRCB0	“EVRCB0”
EVRCB1	“EVRCB1”
G.711 µ law	"PCMU", "G.711", "G.711U", "G.711MULAW", "G711", "G711MULAW"
G.711 A law_64	"PCMA", “G.711A”,”G.711ALAW”
G.722	"G722", "G.722"
G.723 (High)	"G723", "G.723", "G723HIGH"
G.723 (Low)	"G723", “G723LOW”
G.726_16	"G726-16", "G726_16"
G.726_24	"G726-24", "G726_24"
G.726_32	"G726-32", "G726_32"
G.726_40	"G726-40", "G726_40"
G.727_16	"X-G727-16", "G727_16", "G727-16"
G.727_24	"X-G727-24", "G727_24", "G727-24"
G.727_24_16	"X-G727-24-16", "G727_24_16", "G727-24-16"
G.727_32	"X-G727-32", "G727_32", "G727-32"
G.727_32_16	"X-G727-32-16", "G727_32_16", "G727-32-16"
G.727_32_24	"X-G727-32-24", "G727_32_24", "G727-32-24"
G.727_40_16	"X-G727-40-16", "G727_40_16", "G727-40-16"
G.727_40_24	"X-G727-40-24", "G727_40_24", "G727-40-24"
G.727_40_32	"X-G727-40-32", "G727_40_32", "G727-40-32"
G.728	"G728"
G.729	"G729", "G.729", "G729A", "G.729A"
G.7291	"G7291", "G.729.1", "G729EV", "G.729EV"
G.729E	"G729E",”G.729E”
GSM	"GSM"
GSM-EFR	"GSM-EFR", "GSM_EFR"

Table 5-26: Compression Coders

Coder	MGCP Textual Name
QCELP_13	"QCELP", "QCELP_13", "QCELP-13"
QCELP_13_TFO	"X-QCELP-TFO", "QCELP_13_TFO", "QCELP-13-TFO", "QCELP-TFO"
QCELP_8	"X-QCELP-8", "QCELP_8", "QCELP-8"
QCELP_8_TFO	"X-QCELP-8-TFO","QCELP_8_TFO", "QCELP-8-TFO"
Redundancy per RFC 2198	“RED”
RFC 2833	“telephone-event”
T.38 Fax	“IMAGE/T38”
Transparent	"X-CCD", "TRANSPARENT", "CCD", "clearmode"
iLBC13	"iLBC", "iLBC13", "iLBC_13", "iLBC-13"
iLBC15	"iLBC", "iLBC15", "iLBC_15", "iLBC-15"
BV16	"BV16", "BV_16", "BV-16"
NOOP	"no-op"

The following is an example of creating a connection command with G.711 coders:

```
CRCX 10060 Acgw0@[10.1.37.5]
C: 35
L: a:G.711
```

5.1.23 STUN - Simple Traversal of User Datagram Protocol in MGCP

An AudioCodes Gateway residing in a Local Area Network can discover the presence and types of NATs (Network Address Translator) and firewalls between it and the public internet using the STUN (Simple Traversal of User Datagram) protocol as described in RFC 3489. STUN is a client server protocol where the gateway is the STUN client and requires a STUN server running on the public internet.

STUN enables a gateway residing inside a LAN to provide a public internet address to its signaling and media ports although it resides in a LAN behind a NAT/firewall..

Using STUN, a gateway will display its signaling address (upon RSIP, etc) as the address which appears to the public internet. Media addresses will also be displayed as well as public addresses in the appropriate SDP field.

To use STUN a user should specify the following parameters in the ***ini*** file:

ENABLESTUN = 1

STUNSERVERPRIMARYIP = primary server IP address

STUNSERVERSECONDARYIP = secondary server IP address; may be omitted

NATBINDINGDEFAULTTIMEOUT = binding default timeout, use 20 for default

Because the signaling connection goes through a type of NAT, you must keep this connection alive. One of the options to assure this connection remains alive, is to activate the keep-alive mechanism. Refer to MGCP KeepAlive Mechanism on page 275.

5.1.24 Connection Statistics (CDR)

The ***ini*** file parameter, CPConnectionStatistics, enables the call detail report to print out to the Syslog. The statistical information is printed when the connection is deleted.

Syslog Output Example:

```
11:35:52.636 : 10.4.4.125 : NOTICE  : MGCP CALL STATISTICS:
Endpoint Name: ds/tr0/1,  Connection deleted by Call Agent, Coder:
PCMU, ConnectionID: 22, CallId: 1, Call duration: 120 seconds,
Local RTP port: 4000, Remote RTP address: 10.4.4.125 port 4010,
Tx/Rx bytes 602400/1158560, Tx/Rx packets 3765/7241
```

Note: The Tx/Rx bytes and packets contain an informative value only if, in the DLCX command, the CallID and ConnectionID parameters are defined. If they are not defined, a zero value is printed (Tx/Rx bytes 0/0, Tx/Rx packets 0/0).

5.1.25 Disabling the Delete Connection Functionality from the Gateway Side

The ***ini*** file parameter, DisableDLCXByGW, enables or disables the DLCX (Delete Connection) functionality from the gateway's side (issued when the gateway determines that the connection is no longer valid). Note that when the CA issues a DLCX command to delete multiple connections, AudioCodes recommends using wildcarding in the DLCX commands when possible.

For example, if the DLCX from the gateway side is disabled and the CA receives a forced RSIP for a specific trunk with connections on all the B-channels, the CA should attempt to delete the connections on the trunk with one wildcard DLCX instead of 31 separate DLCX commands for each of the B-channels.

5.1.26 RTCP Extended Reports (RTCP-XR) VoIP Metrics Data

RTCP Extended Reports, defined in RFC 3611, provides additional data beyond the ones provided by RTCP. The VoIP metrics report block, can be toggled and reported using MGCP.

Implementation is according to *draft-auerbach-mgcp-rtcpxr-00.txt*. A new Local Connection Options parameter is introduced, **xrm/mcr**. For example, toggling RTCP-XR data collection, reporting and responding is done by:

L: xrm/mcr:on

Note the difference between this LCO parameter and the SDP parameter defined in RFC 3611. For the full reference, consult the above-mentioned draft. Note that, currently, MGCP reports only the remote RTCP-XR data.

Table 5-27: RTCP XR Example Flow

Gateway CH 0	Call Agent	Gateway CH 1
	← CRCX 4390 Acgw0@[10.11.10.215] MGCP 1.0 TGCP 1.0 C: 1234 L: p:20 , a:PCMU , xrm/mcr:on M: recvonly	

200 4390 OK
I: 25
v=0
o=- 1329622418 0 IN IP4 10.11.10.215
s=-
c=IN IP4 10.11.10.215
t=0 0
m=audio 4000 RTP/AVP 0
a=rtcp-xr:voip-metrics
m=image 4002 udptl t3		

	CRCX 4391 Acgw1@[10.11.10.215] MGCP 1.0 TGCP 1.0
	C: 1234
	L: p:20 , a:PCMU , xrm/mcr:on
	M: sendrecv
	
	v=0
	o=- 1329622418 0 IN IP4 10.11.10.215
	s=-
	c=IN IP4 10.11.10.215
	t=0 0
	m=audio 4000 RTP/AVP 0
	a=rtcp-xr:voip-metrics
	m=image 4002 udptl t38	

		200 4391 OK
		I: 26
		v=0
		o=- 1509771038 0 IN IP4 10.11.10.215

Table 5-27: RTCP XR Example Flow

Gateway CH 0	Call Agent	Gateway CH 1

		s=-
		c=IN IP4 10.11.10.215
		t=0 0
		m=audio 4010 RTP/AVP 0
		a=rtcp-xr:voip-metrics
		m=image 4012 udptl t38

	←
	MDCX 4392 Acgw0@[10.11.10.215] MGCP 1.0 TGCP 1.0
	C: 1234
	I: 25
	M: sendrecv
	
	v=0
	o=- 1509771038 0 IN IP4 10.11.10.215
	s=-
	c=IN IP4 10.11.10.215
	t=0 0
	m=audio 4010 RTP/AVP 0
	a=rtcp-xr:voip-metrics
	m=image 4012 udptl t38	

200 4392 OK
v=0
o=- 1329622418 1 IN IP4 10.11.10.215
s=-
c=IN IP4 10.11.10.215
t=0 0
m=audio 4000 RTP/AVP 0
m=image 4002 udptl t38		

	← AUCX 17374 Acgw0@[10.11.10.215] MGCP 1.0 TGCP 1.0 I: 27 F: XRM/RVM	
200 17374 OK XRM/RVM: NLR=0, JDR=0, BLD=0, GLD=0, BD=0, GD=131, RTD=0, ESD=90,		

Table 5-27: RTCP XR Example Flow

Gateway CH 0	Call Agent	Gateway CH 1
SL=127, NL=127, RERL=127, GMN=16, NSR=91, XSR=127, MLQ=41, MCQ=41, JBN=70, JBM=70, JBS=44		
	DLCX 4393 Acgw0@[10.11.10.215] MGCP 1.0 TGCP 1.0 C: 1234 I: 25	
250 4393 OK P: PS=102, OS=16320, PR=106, OR=16960, PL=0, JI=0, LA=0 XRM/RVM: NLR=0, JDR=0, BLD=0, GLD=0, BD=0, GD=0, RTD=1, ESD=0, SL=127, NL=127, RERL=127, GMN=16, NSR=92, XSR=127, MLQ=41, MCQ=41, JBN=70, JBM=70, JBS=44		
	DLCX 4394 Acgw1@[10.11.10.215] MGCP 1.0 TGCP 1.0 C: 1234 I: 26	
		250 4394 OK P: PS=105, OS=16800, PR=102, OR=16320, PL=0, JI=0, LA=0 XRM/RVM: NLR=0, JDR=173, BLD=255, GLD=0, BD=183, GD=183, RTD=0, ESD=90, SL=127, NL=127, RERL=127, GMN=16, NSR=31, XSR=127, MLQ=17, MCQ=15, JBN=70, JBM=70, JBS=44

5.1.27 Controlling Jitter Buffer Settings with MGCP

Users may control the jitter buffer settings on the gateway, per each connection that is established by MGCP.

The jitter buffer settings that can be configured are:

1. Jitter Buffer Minimum Delay [msec] (0-150) – which sets the minimum period of time for delaying incoming packets. (ini file name: DJBufMinDelay)

2. Jitter Buffer Optimization Factor (0-13) – which sets the rate at which the jitter buffer grows or shrinks according to actual network conditions. (ini file name: DJBufOptFactor)

Configured Jitter Buffer Settings example:

```
CRCX 25070 ds/tr0/1@[10.4.3.96] MGCP 1.0
C: 11
L: a:PCMA, x-jitter-buffer-min-delay:75, x-adaptive-jitter-buffer-
ratio:7
```

Configuration can be done either in a CRCX or MDCX command, at the Local Connection Options line only (L: line).

The syntax used is the AudioCodes proprietary syntax and hence has the prefix of ‘x-‘.

If MGCP does not use these settings, as described above, the connection will be opened with the values set in the ini file.

When Jitter buffer settings are set in a CRCX or MDCX command, they can be monitored through an AUCX command auditing the Local Connection Options:

AUCX 18568 ds/tr0/1@[10.4.3.96] MGCP 1.0 I: 21 F: L	200 18568 OK L: p:20 , a:PCMA , e:off , s:off , t:b8 , nt:NI ,x-jitter-buffer-min-delay:75 , x-adaptive-jitter-buffer-ratio:7

When checking the endpoint’s capabilities through an AUEP command, both setting options will appear:

AUEP 18584 ds/tr0/1@[10.4.3.96] MGCP 1.0 F: A	200 18584 OK A: nt:IN , v:G;D;T;L;R;A;M;MS;DT;MD;MO;BL;FXR;FM;IT, a:PCMU;PCMA;G726-16;G726-24;G726-32;G726-40;X-G727-16;X-G727-24-16;X-G727-24;X-G727-32-16;X-G727-32-24;X-G727-32;X-G727-40-16;X-G727-40-24;X-G727-40-32;G729;G728;X-CCD;G729E;iLBC;iLBC;telephone-event;CN;no-op;image/t38,m:sendonly;recvonly;sendrecv;inactive;netwloop,sc-rtp:64/51;62/51;60/51;60/50,sc-rtcp:81/71;82/71;81/70;82/70;80/70,x-srtp:SRTP_SUITE_NULL, x-jitter-buffer-min-delay, x-adaptive-jitter-buffer-ratio, es

5.1.28 DigitMap Special Handling

5.1.28.1 DigitMap Prefix

Ordinarily, when a digit map collection has completed, the collected string is reported back to the MGC as is. However, sometimes there is a need to add a prefix to the collected digits (e.g., area code 02- Seoul, 03 – Ulsan, etc.) which will be added automatically within the Notification event, to the number dialed.

The prefix is defined using the *DialedStringPrefix ini* file parameter.

- The maximum length of the string parameter is 8 characters. The default value is NULL (i.e. no prefix).
- The prefix is considered to be a part of the total event’s size, i.e. when the

parameter is not defined, the dialed string length can be up to 32 characters. If the prefix is defined as X (< = 8) then the dialed string size is reduced and can have **32 minus X** characters. if the *"Dialed String Exceeds 32 - X Error Message"* is generated.

5.1.28.2 Notification for Digitmap Mismatch

MGCP does not send a Notify Message if a digit map collection completed unsuccessfully (i.e., no match found). However, if there is a need to always get the dialed digits regardless of success or failure, use the *MGCPSendDigitmapMismatchNotification ini* file parameter. This is a Boolean parameter, so it can be set to either "0" or "1". The default value is equal to 0 (i.e. do not send digitmap mismatch notification).

5.1.29 Digest Authentication

Note: The following sub-section on Digest Authentication is only applicable to **MediaPack**.

5.1.29.1 Overview

Digest Authentication provides an access authentication scheme based on the HTTP digest authentication scheme defined in RFC 2617 (HTTP Authentication: Basic and Digest Access Authentication). The formal definition is available from the Multi Service Forum, MSF-IA-MGCP.002-FINAL.

The access scheme is a challenge-response scheme where the call agent is challenging the gateway for authentication. When challenged, the gateway will calculate an appropriate response and notify the call agent. The response appears as an addition to a MGCP command. The call agent examines the response and decided wherever to approve the gateway (continue the session) or not (end the session).

The call agent and the gateway must pre-agree on a password and a username. On the gateway side, the password is configured using the MGCPDigestPassword ***ini*** file parameter and the username is configured using the MGCPDigestUsername ***ini*** file parameter. The former parameter is mandatory: the later may be omitted. If so, the username will be taken from the relevant MGCP command's gateway name. These parameters must be identical to the call agent's one for a successful authentication. Only one call agent data per gateway can be defined.

Following a successful authentication, the gateway will include the authentication data for each command (NTFY and RSIP) it issues upon the call agent.

Two methods are defined for Quality of Protection (qop): 'auth' and 'auth-int' where both are supported. 'auth' means only a username and password should be replied and the computed response will not include the entire message body. 'auth-int' means integrity protection as well. This requires the gateway to maintain (and send in each command) a sequential number of the commands sent in the 'nc' field. The message body is used when calculating the response.

The only algorithm supported for digest authentication is MD5.

5.1.29.2 Digest Authentication Sample

The gateway and the call agent are pre-configured for a password. In this case, the digest username is not defined for the gateway, so the current gateway name will be used. The gateway starts and sends RSIP (with no authentication):

```
RSIP 2018 *@[10.4.4.138] MGCP 1.0
RM: restart
```

The call agent, requiring authentication to proceed, challenges the gateway:

```
401 2018 OK
X+WWW-Authenticate: Digest realm="actestvoiceservice",qop="auth-
int",nonce="/Jb1RTsTDYkKHNuuMiCaUO5AeDb9Ekky9p59",opaque="SdytTxTyR
EBm0GvEMLcyO6Ei9iKRneGL"
```

The gateway, using the parameters supplied by the call agent and the username/password it holds, computes a response (in a new RSIP):

```
RSIP 2019 *@[10.4.4.138] MGCP 1.0
RM: restart
X+Authorization: Digest
username="[10.4.4.138]",realm="actestvoiceservice",nonce="/Jb1RTsTD
YkKHNuuMiCaUO5AeDb9Ekky9p59",uri="MGCP",qop=auth-
int,nc=00000001,cnonce="",response="5171b9dca748868813ce004b147c962
5",opaque="SdytTxTyREBm0GvEMLcyO6Ei9iKRneGL"
```

The call agent validates the gateway.

```
200 2019 OK
```

5.1.29.3 Other Methods of Authentication

Usually the call agent will challenge the gateway on the RSIP message (as it's the first one it receives). Although on authentication the gateway will add authentication data to any message it sends, the call agent may re-challenge the gateway using an AUCX command:

```
AUEP 22142 ACgw0@[10.4.4.138] MGCP 1.0
F:
X+WWW-Authenticate: Digest realm="actestvoiceservice",qop="auth-
int",nonce="/Jb1RTsTDYkKHNuuMiCaUO5AeDb9Ekky9p59",opaque="SdytTxTyR
EBm0GvEMLcyO6Ei9iKRneGL"
```

And the gateway replies with:

```
X+Authorization: Digest
username="[10.4.4.138]",realm="actestvoiceservice",nonce="/Jb1RTsTD
YkKHNuuMiCaUO5AeDb9Ekky9p59",uri="MGCP",qop=auth-
int,nc=00000002,cnonce="",response="adf6df39a6d0ac44fc57b4096542f9d
a",opaque="SdytTxTyREBm0GvEMLcyO6Ei9iKRneGL"
```

5.1.30 RSIP Restart Method Usage

The gateway sends a Restart In Progress (=RSIP) message upon boot up and upon any change to an endpoint (or group of endpoints) state. RSIP messages are aggregated when possible. The following restart method parameters are supported:

- **Restart:** The current entity (dsX) is in service.
- **Forced:** The current entity is shutting down. Active calls are lost.
- **Disconnected:** The current entity had lost its connectivity with the call agent and is trying to re-establish a connection.
- **Graceful:** The current entity is about to start graceful shutdown; no new calls will

be available, where active calls are not affected. Refer to Graceful Shutdown on page 95.

- **Cancel-Graceful:** The graceful shutdown operation for the current entity was cancelled.
- **Keep Alive:** Used for keep-alive messages. Refer to MGCP KeepAlive Mechanism on page 275.

5.2 MGCP Compliance

The MGCP Compliance Matrix Table below summarizes the supported MGCP features respectively. The Reference column in the table refers to IETF RFC 3435 from January 2003 (which replaced RFC 2705).

Table 5-28: MGCP Compliance Matrix

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
1	"*" Wild-carding	Yes		
2	"$" Wild-carding	Yes		
3	Domain name for Call Agent	Yes	IP address is used to identify Call Agent	Pages 23, 96
4	**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
5	Timer indication - T	Yes	Interdigit timer Fixed Timer of 4 sec is used	
6	**Digits and Letters**			
7	#	Yes		
8	X	Yes		
9	X.	Yes	X. - Arbitrary number of X Occurrences	
10	*	Yes		
11	[0-9]	Yes	For digit maps	
12	A,B,C,D	Yes		
13	Event names	Yes		
14	Wildcard notations (X, $, *,all)	Yes		

Table 5-28: MGCP Compliance Matrix

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
15	Optional connection ID (G/rt@A3F58)	Yes		
16	**Signals**			2.1.7
17	On/Off (OO)	Yes		
18	Time out (TO)	Yes		
19	Brief (BR)	Yes		
20	Using "+", "-" to turn on/off the "OO" Signal	Yes		
21	**Connection modes**			3.2.2.6
22	Inactive	Yes		
23	Send only	Yes		
24	Receive only	Yes		
25	Send/receive	Yes		
26	Conference	Yes		
27	Data	No		
28	Loopback	Yes		
29	Continuity test	Yes		
30	Network loop back	Yes		
31	Network continuity (netwtest)	Yes		
32	**Endpoint Configuration command**	Yes		2.3.2
33	**Notification Request command**			2.3.3
34	Endpoint ID	Yes		
35	Notified Entity	Yes		
36	RequestedEvents (with associate actions)	Yes	If not specified, notifications is send to command originator	

Table 5-28: MGCP Compliance Matrix

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
37	RequestIdentifier	Yes		
38	DigitMap	Yes		
39	Defined explicitly or through a previous command	Yes		
40	SignalRequests	Yes		
41	**Quarantine Handling**			
42	Discard	Yes		
43	Process loop	Yes		
44	Process	Yes		
45	Loop	No		
46	**Process step by step**			
47	Requested events	Yes		
48	Digit map	Yes		
49	DetectEvents	Yes		
50	Encapsulated Endpoint Configuration	Yes		
51	**Event associated actions**			
52	Notify event immediately with all accumulated events	Yes		
53	Swap audio	No		
54	Accumulate event in buffer, but do not notify yet	Yes		
55	Accumulate according to digit map	Yes		
56	Keep signal active	Yes		

Table 5-28: MGCP Compliance Matrix

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
57	Process Embedded Notification Request	Yes		
58	Ignore the event	Yes		
59	Supporting two or more actions, hf(S,N)	Yes	Combining up to 2 actions	
60	Persisted events	Yes	Configurable	
61	Number of active connections on an endpoint	1 to 3	1 when using encryption; otherwise up to 3	
62	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	
63	Notification request with empty signal list for stopping tone generation	Yes		
64	Detection of events on Connections	Yes		
65	**Notifications**			2.3.4
66	EndpointID	Yes		
67	NotifiedEntity	Yes		
68	RequestIDentifier	Yes		
69	ObservedEvents	Yes		
70	**Create Connection command**			2.3.5
71	CallID	Yes		
72	Endpoint	Yes		
73	NotifiedEntity	Yes		

Table 5-28: MGCP Compliance Matrix

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
74	Multiple connections per endpoint	Yes	- Up to 3 connections - Only one of them can be in send/send receive mode - 1 connection when using encryption	
75	LocalConnection Options			
76	Encoding method	Yes	One value List of values not supported	
77	Packetization period	Yes		
78	Bandwidth	Yes	Parsing only	
79	Type of Service (TOS)	Yes	2 Hex digits	
80	Echo cancelation	Yes		
81	Silence suppression	Yes		
82	Gain control	Yes	-32 to +31 value	
83	Reservation service	No		
84	RTP security	Yes	Providing Key as per RFC 2327	
85	Type of network (IN, Local)	Yes		
86	Vendor specific extensions	Yes		
87	Mode	Yes		
88	RemoteConnectionDescriptor	Yes		
89	SecondEndpointID	Yes		
90	**Encapsulated Notification Request**			
91	R:	Yes		
92	S:	Yes		
93	Encapsulated Endpoint Configuration	Yes		

Table 5-28: MGCP Compliance Matrix

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
94	**Create Connection return parameters**			
95	ConnectionID	Yes		
96	SpecificEndpointID ("Z")	Yes		
97	LocalConnection Descriptor	Yes		
98	SecondEndpointID	Yes		
99	Secondconnection ID	Yes		
100	Second M line for Fax t38	Yes		
101	**ModifyConnection**			2.3.6
102	CallID	Yes		
103	Endpoint	Yes		
104	Connection ID	Yes		
105	NotifiedEntity	Yes		
106	LocalConnection Options	Yes	See CreateConnectionCmd above	
107	Mode	Yes		
108	RemoteConnectionDescriptor	Yes		
109	**Encapsulated Notification Request**			
110	R:	Yes		
111	S:	Yes		
112	Encapsulated Endpoint Configuration	No		
113	**Modify Connection Return Parameters**			

Table 5-28: MGCP Compliance Matrix

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
114	LocalConnection Descriptor	Yes	Returns if local connection parameters were modified	
115	**Delete Connection (from Call Agent)**			2.3.7
116	CallID	Yes		
117	EndpointID	Yes		
118	ConnectionID	Yes		
119	**Encapsulated Notification Request**			
120	R:	Yes		
121	S:	Yes		
122	Encapsulated Endpoint Configuration	No		
123	**Delete Connection return Parameters**			
124	**Connection Parameters**			
125	Number of packets send	Yes		
126	Number of octets send	Yes		
127	Number of packets received	Yes		
128	Number of octets received	Yes		
129	Number of packets lost	Yes		
130	Inter-packet arrival jitter	Yes		
131	Average transmission delay - latency	Yes		
132	Delete Connection	Yes		2.3.8

Table 5-28: MGCP Compliance Matrix

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
	(from gateway)			
133	CallID	Yes		
134	EndPointID	Yes		
135	ConnectionID	Yes		
136	ReasonCode	Yes		
137	Connection Parameters	Yes		
138	**DeleteConnection (multiple connections)**	Yes		2.3.9
139	CallID	Yes		
140	EndPointID	Yes		
141	**Audit Endpoint**			
142	EndpointID	Yes		
143	RequestedInfo	Yes		
144	Wildcard convention * ("all of")	Yes		
145	**AuditEndpoint Return Parameters**			2.3.10
146	Endpoint ID list, "Z='	Yes		
147	RequestedEvents	Yes		
148	Including actions associated with the events	Yes		
149	DigitMap	Yes		
150	SignalRequests TO signals currently active On/Off signals currently ON Pending Brief signals	Yes		
151	RequestIDentifier	Yes		

Table 5-28: MGCP Compliance Matrix

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
152	NotifiedEntity	Yes		
153	Connection Identifiers	Yes		
154	DetectEvents	Yes		
155	ObservedEvents	Yes		
156	EventStates	Yes		
157	Bearer Information	No		
158	RestartReason	Yes		
159	RestartDelay	Yes		
160	ReasonCode	No		
161	**Capabilities**			
162	List of supported codecs	Yes		
163	Packetization Period	No		
164	Bandwidth	No		
165	Echo Cancelation	No		
166	Silence Suppression	No		
167	Gain Control	No		
168	Type of Service	No		
169	Resource Reservation	No		
170	Encryption Key	No		
171	Encryption Suites	Yes		
172	Type of Network	Yes		
173	Supported Event Packages	Yes		
174	Connection Modes	Yes		
175	Audit Connection	Yes		
176	ConnectionID	Yes		

Table 5-28: MGCP Compliance Matrix

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
177	RequestedInfo	Yes		
178	**Audit Connection Return Parameters**			2.3.11
179	CallID	Yes		
180	Notified Entity	Yes		
181	Local Connection Options	Yes		
182	Mode	Yes		
183	Remote Connection Descriptor	Yes		
184	LocalConnection Descriptor	No		
185	Connection Parameters	Yes		
186	**Restart in Progress (RSIP)**			2.3.12
187	EndpointID			
188	"All of" wildcard (*)	Yes		
189	Restart Method	Yes		
190	Graceful	Yes		
191	Forced	Yes		
192	Restart	Yes		
193	Disconnected	Yes		
194	Cancel-graceful	Yes		
195	Restart Delay	Yes		
196	ReasonCode	No		
197	Restart in progress return parameters (notified entity &	No		

Table 5-28: MGCP Compliance Matrix

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
	return code)			
198	**Return Codes and Error Codes**	Partially		2.4
199	100	Yes	The transaction is currently being executed An actual completion message will follow later	
200	200	Yes	The requested transaction was executed normally	
201	210-214	Yes	CALEA return codes	
202	250	Yes	The connection was deleted	
203	400	Yes	The transaction couldn't be executed due to a transient error	
204	401	Yes	The phone is already off hook	
205	402	Yes	The phone is already on hook	
206	405	Yes	The transaction could not be executed, because the endpoint is "restarting".	
207	500	Yes	The transaction could not be executed because the endpoint is unknown	
208	501	Yes	The transaction could not be executed because the endpoint is not ready	
209	502	Yes	The transaction could not be executed because the endpoint does not have sufficient resources	
210	503	No	"All of" wildcard not fully supported The transaction contained an "all of" wildcard, however NotificationRequests non-empty	

Table 5-28: MGCP Compliance Matrix

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
211	504	Yes	Unknown or unsupported command.	
212	505	Yes	Unsupported RemotedConnectionDescriptor	
213	506	Yes	Unable to satisfy both LocalConnectionOptions and RemoteConnection Descriptor	
214	507	Yes	Unsupported functionality	
215	510	Yes	The transaction could not be executed because a protocol error was detected	
216	511	Yes	The transaction could not be executed because of the command contained an unrecognized extension	
217	512	No	The transaction could not be executed because the gateway is not equipped to detect one of the requested events	
218	513	Yes	The transaction could not be executed because the gateway is not equipped to generate one of the requested signals	
219	514	Yes	The transaction could not be executed because the gateway cannot send the specified announcement	
220	515	Yes	The transaction refers to an incorrect connection ID	
221	516	Yes	The Transaction refers to an unknown call ID	
222	517	Yes	Unsupported or invalid mode	
223	518	Yes	Unsupported or unknown package	
224	519	Yes	Gateway does not have a digit map	

Table 5-28: MGCP Compliance Matrix

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
225	520	Yes	The transaction could not be executed because the GateWay is restarting	
226	521	Yes	Endpoint redirected to another Call Agent endpoint is restarting	
227	522	Yes	No such event or signal	
228	523	Yes	Unknown action or illegal combination of actions	
229	524	Yes	Internal inconsistency in localConnectionOptions	
230	525	Yes		
231	526	No		
232	527	Yes		
233	528	Yes		
234	529	No		
235	530	No		
236	531	Yes		
237	532	Yes	Unsupported value in LocalConnectionOptions	
238	533	Yes	Response too big	
239	534	Yes	Codec negotiation failure	
240	535	Yes	Packetization period not supported	
241	536	No	Unknown or unsupported RestartMethod	
242	537	Yes	Unknown or unsupported digit map extension	
243	538	Yes	Event or Signal error	
244	6xx	Yes	Basic and advanced audio packages	

Table 5-28: MGCP Compliance Matrix

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
245	Reason Codes (900, 901, 902)	No		
246	**MGCP Command Header**			3.2
247	Endpoint identifier	Yes		
248	Notified entity	Yes		
249	In notified entity, If port # is omitted, using default MGCP port (2427)	Yes		
250	Response Acknowledgement	Yes (receive side only)		
251	**Encoding of Session Description - SDP**			3.5
252	SDP parameters: v,c,m,a	Yes		
253	Using RTPMAP attribute to define encoding of dynamic audio formats	Yes		
254	Optional Ptime attribute to define packet duration	Yes		
255	IP address of remote/local gateways	Yes		
256	**Transmission over UDP**			3.5
257	Transaction identifiers	Yes		
258	Receiving Duplicated transaction IDs	Yes		

Table 5-28: MGCP Compliance Matrix

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
259	Retransmission timers	Yes		
260	Piggy backing	Yes		
261	Provisional responses	Yes		
262	MultipleCall Agents and Call Agent Redundancy	Yes		
263	States, failover and race conditions	Yes		
264	**Failover Assumptions and Highlights**			4.1
265	Call Agents DNS	Yes		
266	Notified Entity for endpoint	Yes		
267	Responses send to source address	Yes		
268	Backup Call Agent	Yes		
269	**Retransmission, Detection of Lost Associations**			4.3
270	Commands retransmission	Yes		
271	Disconnecting endpoint/gateway	Yes		
272	**Race Conditions**			4.4
273	Quarantine list	Yes		
274	Explicit detection	Yes		
275	Ordering of commands	Yes		
276	Restart avalanche	Yes		
277	Disconnected endpoints	Yes		

Table 5-28: MGCP Compliance Matrix

#	Feature	Support	AudioCodes Comments	References (to IETF RFC 3435)
288	**Security requirements**			5
289	MGCP IP security (RFC 1825)	No		

5.3 MEGACO (Media Gateway Control) Protocol

Note: The following section on MEGACO is NOT applicable to **MediaPack** and the **TP/IPM-260** product family.

5.3.1 MEGACO Overview

MEGACO (**ME**dia **GA**teway **CO**ntrol) Protocol is a standards-based network control protocol (originally based on IETF RFC 3525 and ITU-T H.248.1 V1. The current version is H.248.1 V3). 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., B-channel), 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).

H.248.1 V2 (April 2003) has made some changes compared to V1 (V1 is also called RFC 3525).

AudioCodes' version 5.0 and earlier supports H.248.1 V1.

AudioCodes' version 5.2 supports H.248.1 V2 and is backward compatible with V1.

AudioCodes' version 5.4 partially supports H.248.1 V3 and is backward compatible with V1 and V2.

Since this is a standards-based control protocol, AudioCodes does not provide or require the user to use any specific software library in order to construct a Call Agent. (Users may choose any of many such stacks available in the market.)

Note: MGCP and MEGACO protocols cannot co-exist 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, but the binary encoding has not been fully tested. 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. 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 continues 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 “Light” Virtual Media Gateway

Currently the application does not support the standard virtual media gateway. In order to achieve some of the functionally, however, the following support is included:

- The above authorization enables the gateway to reply to more than one MGC.
- Notifications are sent to the MGC that requested them by sending a command with the events descriptor to the device.

Note: The serviceChange commands are sent **ONLY** to the controlling MGC. This implies that if one of the non-controlling MGC stopped responding, a disconnection service change is sent to the controlling MGC after retransmission timer expiration.

5.3.2.6 Transport over SCTP

MEGACO protocol messages may be transmitted over the Stream Control Transmission Protocol (SCTP), as defined in the H.248.4 sub-series. This is done by configuring the Transport Type (cpTransportType) to be equal to "2".

When working with the SCTP transport, there can be only one MGC that sends commands, as this is a point-to-point protocol. Therefore, the "Light" Virtual Media Gateway is not applicable.

The H.248.4 sub-series defines the ability of working with multiple streams. Our implementation, however, supports only one stream.

Some SCTP parameters have an effect on the H.248 over SCTP behavior, primarily related to the H.248 message size. The *SCTPMaxDataChunkSize* parameter can be set to the maximum value of 1450 bytes. Therefore, the maximum message length that can be sent and received on the SCTP stream is defined by this parameter. An attempt to send the gateway a longer message will fail. If the gateway needs to reply with a longer message, it will send an Error 533 message.

Another set of parameters relates to the retransmission behavior of the SCTP and H.248. Theoretically, there is no need for retransmission on the H.248 level, as the SCTP is a reliable protocol. However, there can always be the case in which the application level is down while the communication level is still up. In order to avoid such a state, it is recommended to set the H.248 retransmission timer (*MGCPRetransmissionTimeout)* to be larger than *SCTPHBInterval* (at least by two) and the *MGCPCommunicationLayerTimeout* parameter which defines the time frame after which the H.248 will declare disconnection to be twice the value of (SCTPHBInterval * SCTPMaxAssocRet). This will ensure that the SCTP retransmission will take precedence, but still an application failure will be noticed.

5.3.2.7 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 values are set via SNMP, Web or ***ini*** file parameter or MEGACO command using the DS package. (The diffserv package changes only the media path.) Note, that changing the value of the control path requires the Gateway to be reset.

The value of the control path is now controlled by an ini file parameter called 'PremiumServiceClassControlDiffServ', with the default value of 40. If this parameter is not set, and the old parameter 'ControlDiffServ' is, the old parameter is used.

The value for the media path is now set via the ***ini*** file parameter 'PremiumServiceClassMediaDiffServ', with the default value of 46. If this parameter is not set, and the old parameter 'IPDiffServ' is, the old parameter is used '.

5.3.2.8 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 "Playing Signals" below), a new EventDescriptor, or both. The embedded descriptor replaces the current descriptor.

5.3.2.9 Playing Signals

Signals in MEGACO reside in a SignalsDescriptor. The signal combination options in the signal descriptor are:

- One signal request
- One signal list
- Two signal requests - One of the signal requests is from Group 1 and the other signal request is from Group 2 (both groups and combinations are shown below). This is applicable only if these signals are supported in the current configuration.

Table 5-29: General Signal Combination Options

Group 1	Notes	Group 2	Notes
ToneGen/*	Including all the inheriting packages	al/ri (cpSignal)	
an/* (cpSignal)		alert/ri (cpSignal)	
ct/*		alert/rs (cpSignal)	
alert/cw (cpSignal)	Only when analog device	xal/* (cpSignal)	
andisp/* (cpSignal)	Only when analog device	gb/*	(3G)
aasb/* (cpSignal)	Including all the inheriting packages	bt/*	(3G)
nttrk/* (cpSignal)			
ctyp/*			

Table 5-30: Signal Combination Options for CAS Support

Group 1	Notes	Group 2	Notes
ToneGen/*	Including all the inheriting packages	bcas/sza	
an/*		bcas/ans	
ct/*		bcas/idle	
alert/cw		rbs/*	
bcasaddr/addr		icas/cf	
oses/*		icas/cb	
osext/*		icas/rlg	
icas/congestion		casblk/*	
icas/status		Bcas/sz	
icasc/*			
aasb/*	Including all the inheriting packages		
nttrk/*			
ctyp/*			

A signal list can contain 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 (For timeout signals only) can be defined as a parameter in the signal. If omitted, a default value is used (refer to the package's description in the beginning of this section).

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 **CPT ID** column in the **MEGACO Call Progress Tone Signals** table below.

When a CPT file is missing, the device defines default values only for the following signals:

- Dial tone
- Ringing tone
- Busy tone

5.3.2.10 MEGACO Supported Signals

Table 5-31: MEGACO Call Progress Tone Signals

Symbol	Definition	Type	Duration	CPT ID
cg/dt	Dial tone	TO	180 sec	1
cg/rt	Ringing tone	TO	180 sec	2
cg/bt	Busy tone	TO	180 sec	3
cg/ct	Congestion tone	TO	180 sec	4
cg/sit	Special Information tone	BR	2 sec	5
cg/wt	Warning tone	BR	1sec	6
cg/pt	Payphone Recognition tone	TO	180 sec	38
cg/cw	Call Waiting tone	BR	1 sec	9
cg/cr	Caller Waiting tone	TO	180 sec	15

Table 5-31: MEGACO Call Progress Tone Signals

Symbol	Definition	Type	Duration	CPT ID
xcg/cmft	Comfort tone	TO	180 sec	18
xcg/roh	Off-hook warning tone	TO	180 sec	16
xcg/nack	Negative Acknowledgement	TO	180 sec	19
xcg/vac	Vacant Number tone	TO	180 sec	20
xcg/spec	Special Conditions dial tone	TO	180 sec	21
srvtn/rdt	Recall dial tone	TO	180 sec	22
srvtn/conf	Confirmation tone	BR	1 sec	8
srvtn/ht	Held tone	TO	180 sec	23
srvtn/mwt	Message Waiting tone	TO	180 sec	17
xsrvtn/xferdt	Call Transfer Dial Tone	TO	180 sec	24
xsrvtn/cft	Call Forward Tone	BR	1 sec	25
xsrvtn/ccst	Credit Card Service Tone	BR	1 sec	26
xsrvtn/srdt	Special Recall Dial Tone	TO	180 sec	27
bcg/bdt	Dial tone	TO	180 sec	1
bcg/brt	Ringing tone	TO	180 sec	2
bcg/bbt	Busy tone	TO	180 sec	3
bcg/bct	Congestion tone	TO	180 sec	4
bcg/bsit	Special Information tone	BR	2 sec	5
bcg/bwt	Warning tone	BR	1 sec	6
bcg/bpt	Payphone Recognition tone	TO	180 sec	38

Table 5-31: MEGACO Call Progress Tone Signals

Symbol	Definition	Type	Duration	CPT ID
bcg/bcw	Call Waiting tone	BR	1 sec	9
bcg/bcr	Caller Waiting tone	TO	180 sec	15
carr/cdt	Carrier Dial tone	BR		216
carr/ans	Carrier Answer tone	BR		217
carr/chg	Carrier Charging tone	BR		218
carr/ldi	Long Distance Indicator tone	BR		219
prectn/preconf	preset conference notification tone	BR		42
prectn/pcprec	preset conference precedence notification tone	BR		41
prectn/precrt	precedence ringing tone	TO	180 sec	44
prectn/preempt	pre-emption tone	BR		43
conftn/enter	conference entrance tone	BR		33
conftn/exit	conference exit tone	BR		34
conftn/lock	conference lock tone	BR		35
conftn/unlock	conference unlock tone	BR		36
conftn/timelim	a time limit warning tone	BR		37

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}
             }
     }
  }
}
```

5.3.2.11 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. 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. Not applicable to 6310/8410/3000 devices. For these devices two DSP are allocated.
- When both users use non-G.711 and different coders, two DSPs are allocated
- When one side uses RFC 2833 and the other does not, or the payloads of RFC 2833 are different, 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.

5.3.2.12 Create a Conference

Note: The sub-section on Create a Conference is only applicable to **IPM**.

When using MEGACO, all the terminations in an active context can connect with each other. As such, if the user puts more than two terminations in a context and according to the existing topology one of them has at least two inputs, MEGACO assumes that it should be a conference. The terminations types can be a mix of physical and ephemeral terminations, each with its own coder.

The flow direction within the conference users is controlled via the Topology descriptor. The Topology defines the media flow direction for each couple of terminations in a context. The default flow direction is **Bothway**, meaning that this termination sends and receives data. For example, if there are five terminations in a context - Term/1 to Term/5, and the following topology: {*, Term/1, ONEWAY, Term/1, Term/2, BOTHWAY} means that Term/1 hears all the other terminations, but only the Term/2 participant can hear it (Coach mode).

The conference works in a **conference bridge** mode. The voices of all participants are mixed, and each participant received the sum of all others. This is not similar to three-way conferencing, where each user has two input streams that are mixed, and one output stream that is split and sent to the two other users.

Each conference participant needs a DSP resource, which is allocated internally.

```
A conference creation command is shown in the following example:
MEGACO/1 [10.2.207.141]:2944
Transaction = 1237 {
    Context = $ {  Topology{*, *, bothway},CA{TP},
    Add  = $ {Media {LocalControl {Mode = sendreceive },
      Local {
v=0
c=IN IP4 $
m=audio $ RTP/AVP 0
 },
    Remote {
v=0
c=IN IP4 10.2.221.1
m=audio 4000 RTP/AVP 0}}},

Add  = $ {Media { LocalControl {Mode = sendreceive },
     Local {
v=0
c=IN IP4 $
m=audio $ RTP/AVP 0
 },
      Remote {
v=0
c=IN IP4 10.2.221.1
m=audio 4010 RTP/AVP 0}}},

Add  = $ {Media {   LocalControl { Mode = sendreceive},
     Local {
v=0
c=IN IP4 $
m=audio $ RTP/AVP 0
 },
    Remote {
v=0
c=IN IP4 10.2.221.1
m=audio 4020 RTP/AVP 0}}}}}
```

In the example, a new conference is created, and the media flow is send - receive for all participants.

5.3.2.13 STUN - Simple Traversal of User Datagram Protocol in MEGACO

STUN - Simple Traversal of User Datagram Protocol in MEGACO functions similar to STUN in MGCP. (Refer to STUN - Simple Traversal of User Datagram Protocol in MGCP on page 320.)

As in MGCP, because the signaling connection goes through a type of NAT, you must keep this connection alive. One of the options to assure this connection remains alive, is to activate the keep-alive mechanism. Refer to KeepAlive Notifications From the Gateway on page 344.

5.3.2.14 CAS Protocols Support in MEGACO

5.3.2.14.1 MFCR2 Support in MEGACO

The MFCR2 trunk protocols are supported in MEGACO by using the 'bcas' package defined in H.248.25, the 'icas' and 'casblk' packages defined in H.248.28 and 'icasc' package defined in H.248.29

Using these packages, the device converts from the MFCR2 protocol, which is a PSTN protocol, to the MEGACO protocol, thereby bridging the PSTN world with the IP world.

When MEGACO and MFC-R2 protocols share control of a channel, their timings are synchronized so that MEGACO commands do not cause damage to the MFC-R2 protocol's negotiation. For example, MFC-R2 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 MFC-R2 protocol's negotiation is not yet finished, the entire negotiation could be damaged. To avoid this problem, the MEGACO does not change the echo canceler state until the call was accepted by the answering side.

The actual call should start only after the accept signal is finished. (See the call flow of call start).

The application supports a special option called re-answer. In this option, the answering side can put down the phone, and pick it up again. The phone close will result with the 'icas/cb' event, but if the phone is taken up again, the 'bcas/ans' event will be sent. The timing of this action is defined by the MGC. It is the MGC responsibility to decide when the call should be disconnected by sending the 'icas/cf' signal. (refer to the figures below for the call flow of the call disconnect for the use of these signals and events). Note that even though the re-answer timer is controlled by the MGC, the device still keeps its own timer (currently hard-coded to be 256 seconds), so that it does not get stuck in case of command loss.

Blocking the Bchannel is done by using the 'casblk' package. The 'blk' and 'ublk' events are reported only if the action was done by the remote side. The reason for this is that the local side already knows its status. Unfortunately, sometimes the MGC loses the state and needs to synchronize with the current status. The recommended command for this is to send the 'bcas/idle' signal, and ask for the 'bcas/idle' and 'casblk/blk' events. This results in idling the line in case of a partial call, and getting the current state of the line: Idle (After idling completed) or Blocked (If blocked by the other side).

Figure 5-1: MEGACO-R2 Call Start Flow Diagram

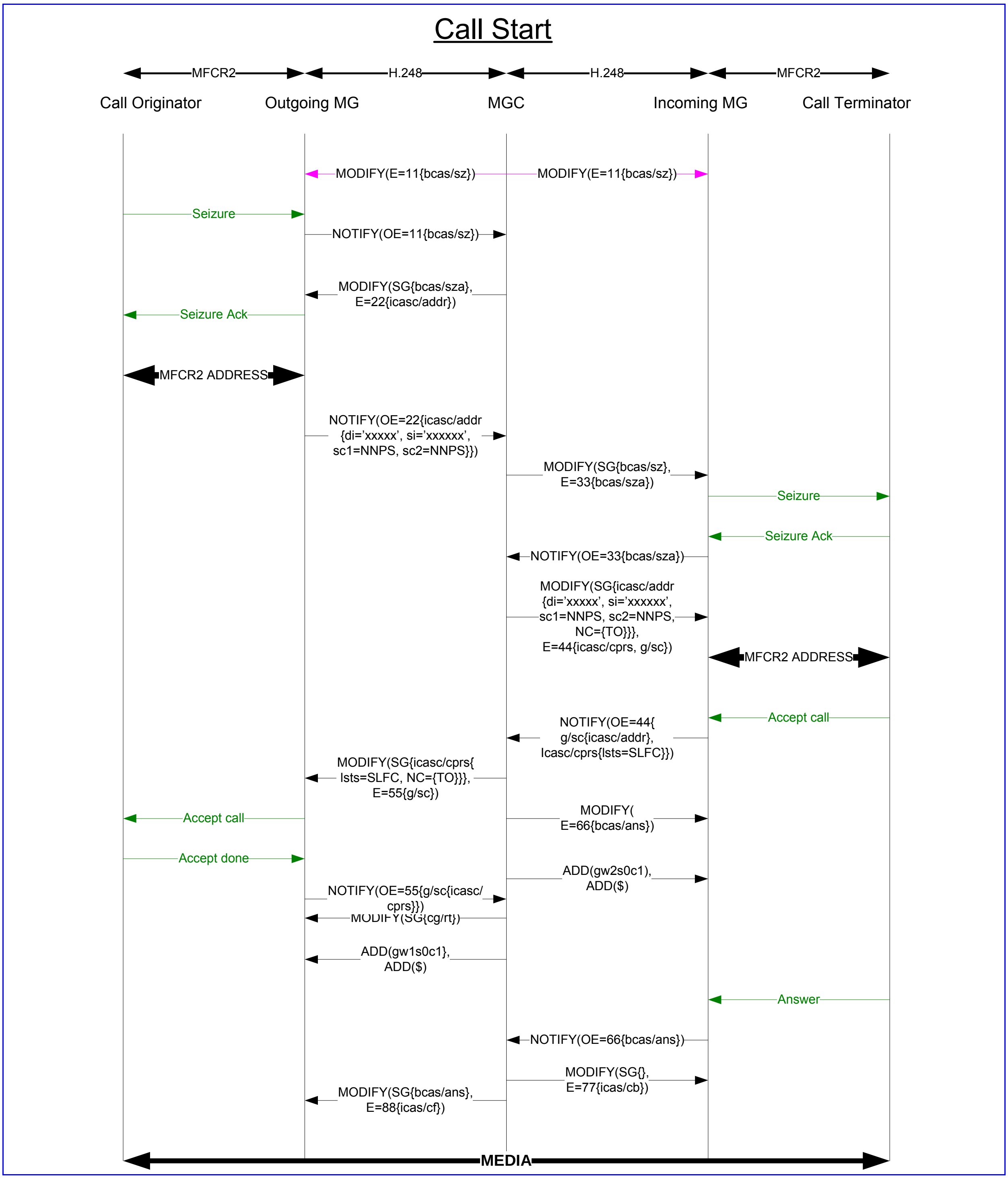
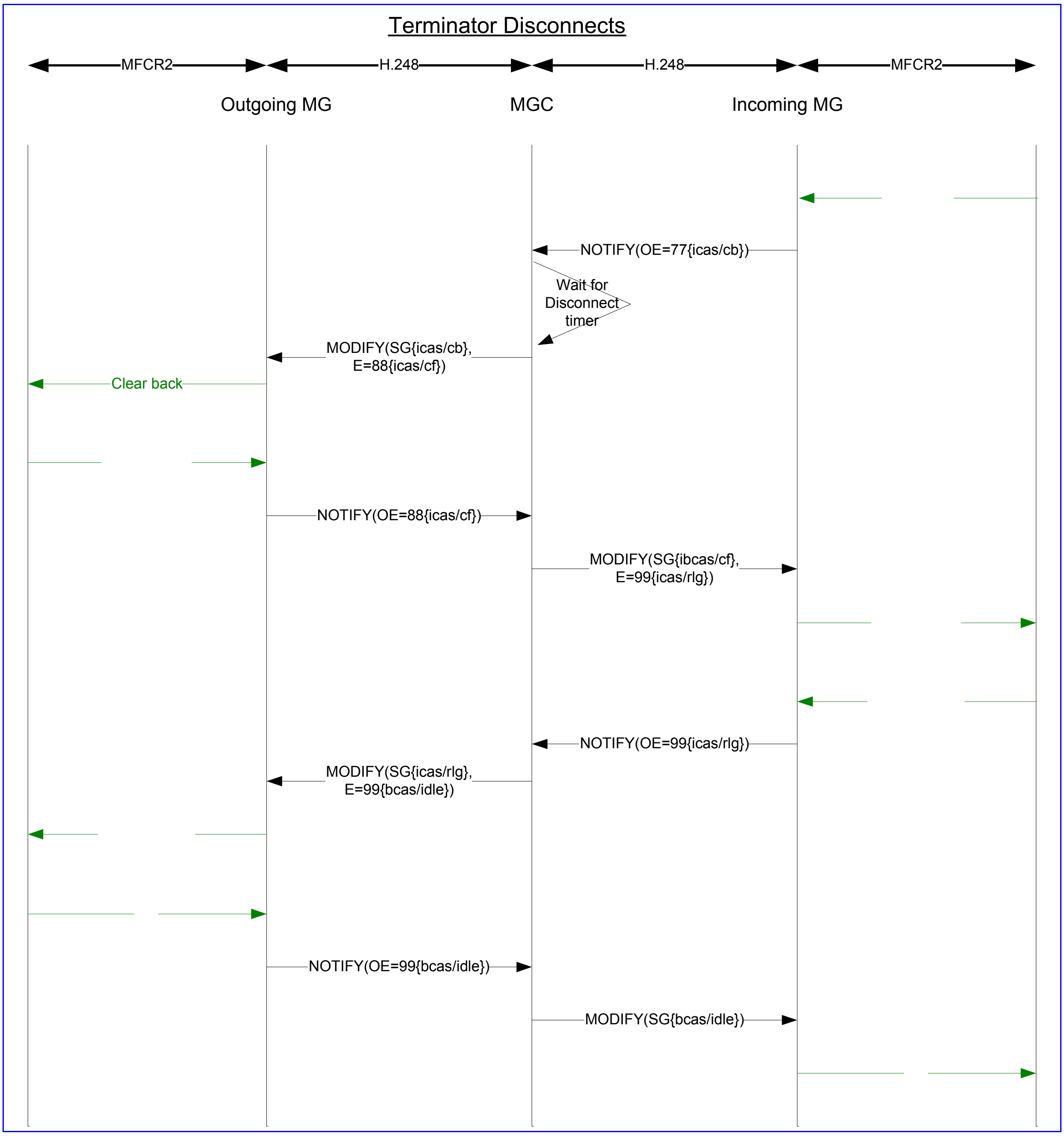

Call Start
MFCR2
H.248
H.248
MFCR2
Call Originator
Outgoing MG
MGC
Incoming MG
Call Terminator
MODIFY(E=11{bcas/sz})
MODIFY(E=11{bcas/sz})
Seizure
NOTIFY(OE=11{bcas/sz})
MODIFY(SG{bcas/sza}, E=22{icasc/addr})
Seizure Ack
MFCR2 ADDRESS
NOTIFY(OE=22{icasc/addr {di='xxxxx', si='xxxxxx', sc1=NNPS, sc2=NNPS}})
MODIFY(SG{bcas/sz}, E=33{bcas/sza})
Seizure
Seizure Ack
NOTIFY(OE=33{bcas/sza})
MODIFY(SG{icasc/addr {di='xxxxx', si='xxxxxx', sc1=NNPS, sc2=NNPS, NC={TO}}}, E=44{icasc/cprs, g/sc})
MFCR2 ADDRESS
Accept call
NOTIFY(OE=44{ g/sc{icasc/addr}, Icasc/cprs{lsts=SLFC}})
MODIFY(SG{icasc/cprs{ lsts=SLFC, NC={TO}}}, E=55{g/sc})
Accept call
MODIFY(E=66{bcas/ans})
Accept done
ADD(gw2s0c1), ADD($)
NOTIFY(OE=55{g/sc{icasc/ cprs}})
MODIFY(SG{cg/rt})
ADD(gw1s0c1}, ADD($)
Answer
NOTIFY(OE=66{bcas/ans})
MODIFY(SG{}, E=77{icas/cb})
MODIFY(SG{bcas/ans}, E=88{icas/cf})
MEDIA

Figure 5-2: MEGACO-R2 Call Disconnect Flow Diagram



Note: The disconnection from the originator side looks the same. It only starts from the 'Clear forward' line signal. Also, even though the 'idle' notification is sent regardless of the 'bcas/idle' signal, this signal is still required for the internal state machine.

5.3.2.15 E911 Support in MEGACO

The following attributes distinguish the E911 trunk:

- There are only outgoing calls. The 911 operator never calls any number.
- The 911 operator may hold the call so that the caller cannot disconnect it. Even if the caller closes the call, the operator may ring back. This feature is not supported by all E911 operators.

All of the required E911 support functionality is defined in H.248.25 - Basic CAS packages:

- ' bcas' - Full support of all line signals.
- ' bcasaddr' - Supports dialing and detecting a string of digits in MF and DTMF .
- ' rbs' - Supports the wink and flash hook.
- ' oses' - Supports operator ring-back generation and detection.
- ' osext' - Support operator extended services. Not relevant for 911, but applicable for the General E & M Case.

The following diagrams show call start in 911 and operator ring back:

Figure 5-3: MEGACO-911 Call Start Flow Diagram

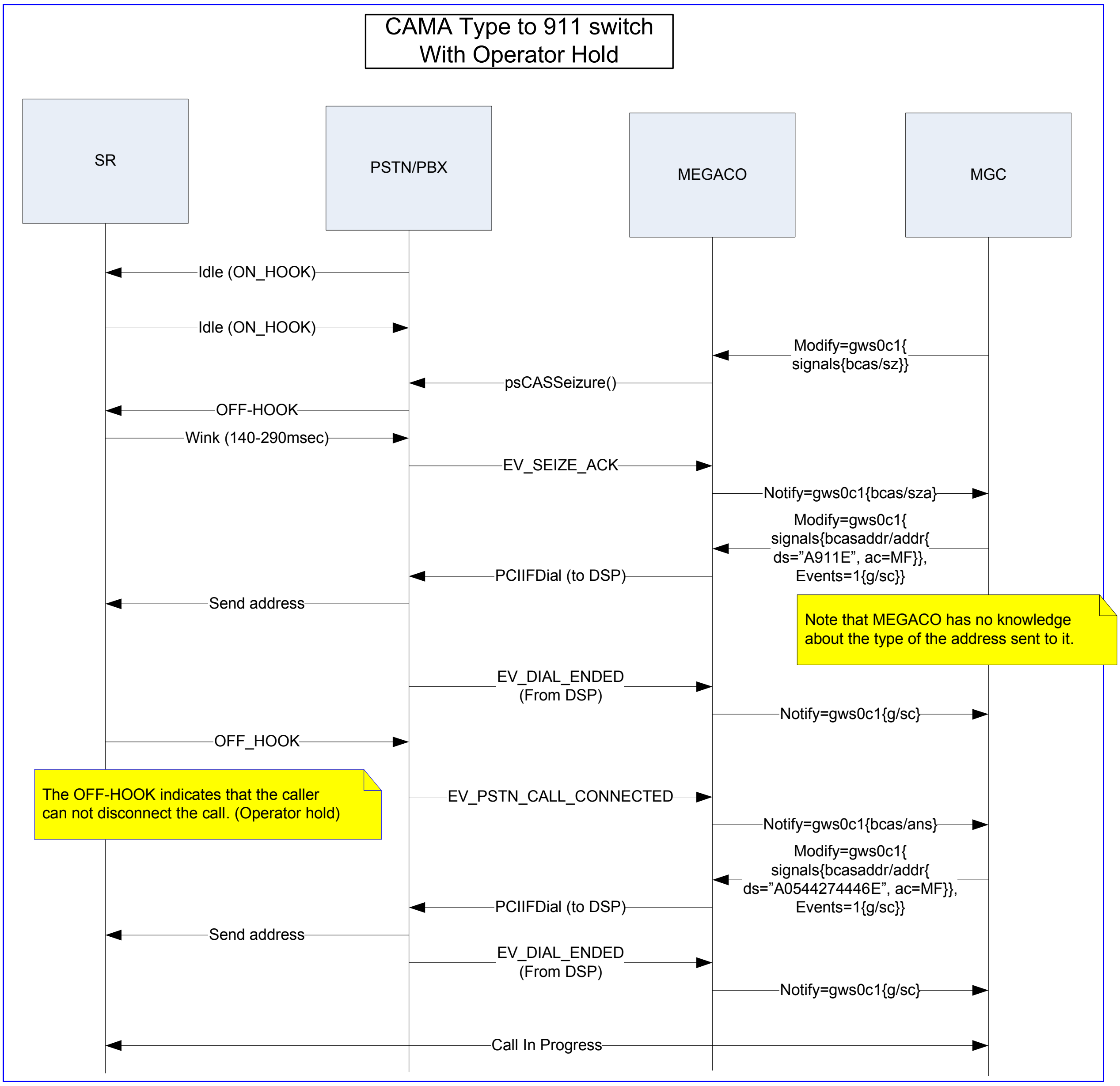
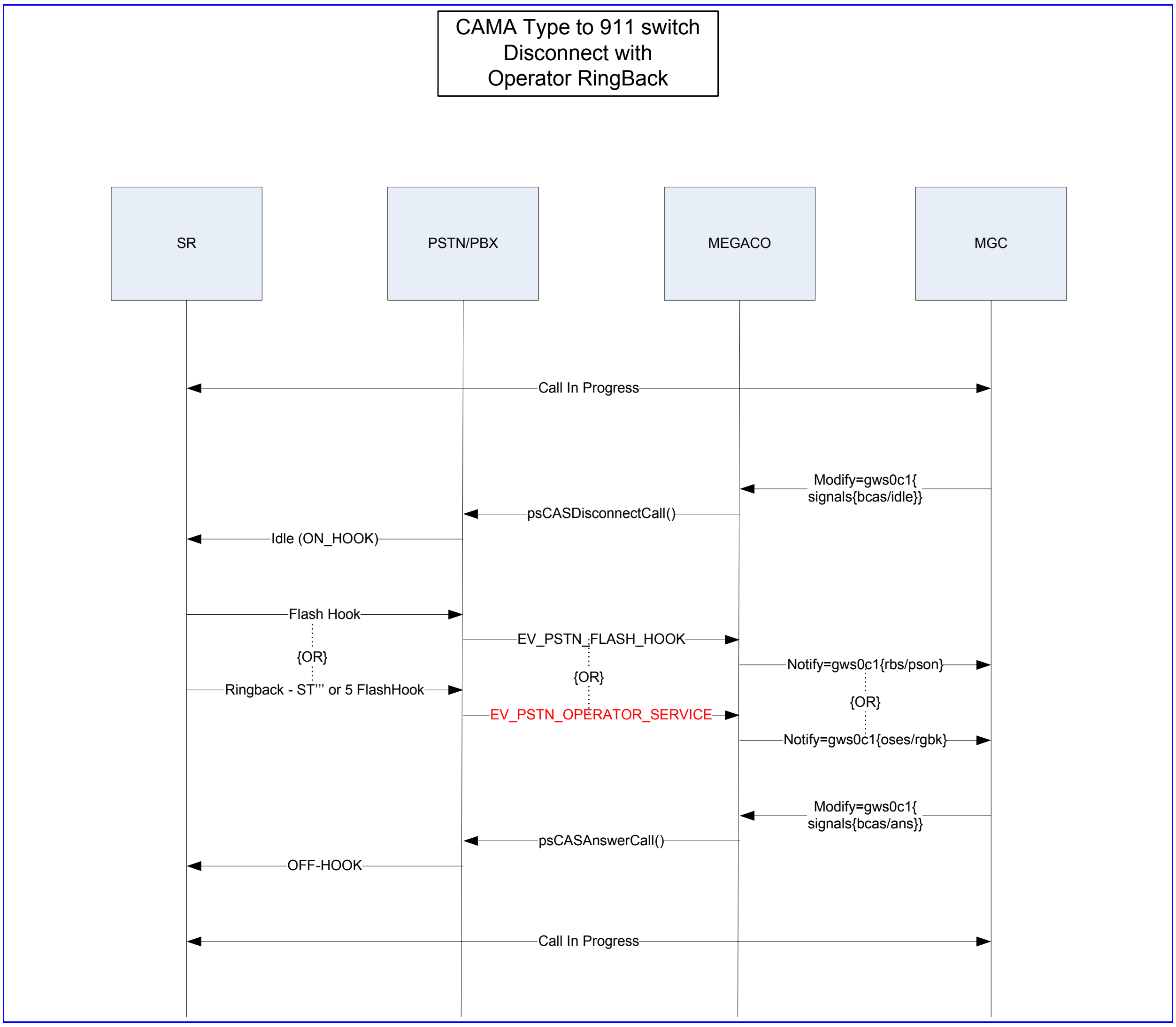


Figure 5-4: MEGACO-911 Operator Ringback Flow Diagram



5.3.2.16 E&M and MF Trunks

MEGACO fully supports any CAS trunk. The 911 trunks described above is a particular case in which the E&M feature group D is implemented. However, the full range of trunks can be supported using the H.248.25 protocol package set.

Note: Not all of the CAS state machine files enjoy MEGACO support. Therefore, before trying to activate a trunk other than a 911 trunk, contact AudioCodes Technical Support to receive the proper CAS table.

5.3.2.17 VLAN Support (Applicable to 3000/6310/8410 only)

The VLAN package, as defined in H.248.56, is supported and allows the Media Gateway Controller to select the interface that is to be used as the source interface for the outgoing 'Media' stream (i.e., from which VLAN the call should be initiated from). The selection is performed using the VLAN Tags package property.

Note: The implementation supports only one VLAN tag. The Ethernet Priority property is not in use.

5.3.2.18 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.

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.

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
```

The '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).

RFC 2833 negotiation behavior is defined by the "Negotiation Type" BIT of "CPSDPProfile" parameter (bit 3, value 8). When this bit is turned on, strict negotiation rules are applied. Otherwise, non-strict rules are applied.

When non-strict negotiation is used, negotiation is performed according to the following rules:

- If both sides specify the 'telephone-event' in the SDP, the device uses the RFC 2833 transport type.
- If one of the sides 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, there is an asymmetric transmit and receive.

Therefore, if you need to activate RFC 2833 only when both sides agree on it, you should configure the default value (e.g., Transparent). to be different than that of RFC 2833.

When strict negotiation is used, negotiation is performed according to the same rules above, except that if one of the sides does not specify the 'telephone-event' in the SDP, the device does not use RFC 2833 as transport type. The device uses the **Transparent** transport type if it is either transparent or mute.

For backward compatibility, the same functionality can be enabled by the value of bit 2 (value 4) for the MEGACO profiling parameter, MGCPCompatibilityProfile.

5.3.2.19 Silence Suppression Support

Silence suppression can be enabled in two ways:

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

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

Silence suppression can be disabled by:

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

2. 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.

3. 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.

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

The table below summarizes the operation of silence suppression:

Table 5-32: 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

5.3.2.20 Digits Collection Support

The following methods for digit collection are supported:

- One by one collection using the single events in the 'dd' package (e.g., dd/d3). Note that if the wildcarded format is used (dd/*), we will report the start digit and end digit events (e.g. dd/std{tl=d1} and dd/etd{tl=d1}) and not the specific digit event (e.g. dd/d1).
- Collection according to digit map. This includes the basic collection 'dd/ce' event defined in the basic package and the 'xdd/xce' and 'edd/mce', both defined it

H.248.16. The maximal pattern length is 150 bytes, and the maximal collected number is 30 digits. For the extended digit collection, the buffering of type ahead digits continues up to the limit of 30 digits. New digits after that are lost.

5.3.2.21 Reporting Fax Events

Some of the Fax events can be reported using the packages from H.248.2: "CTYP" and "IPFAX". The only Fax events reported by the "CTYP" package are the "V21flag" and “cng”, 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 device. "EOF" is reported when the MEGACO application gets "EVENT_END_FAX" from the device.

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.

5.3.2.22 Reporting Media Stream Creation Failure

When attempting to establish a media stream, the operation may fail as a result of a problem in the Address Resolution Protocol (ARP), which is used for mapping IP network addresses to hardware addresses and finding addresses of a computer in a network.

MEGACO will notify media stream creation failures resulting from ARP errors using the MEGACO cause event (eventide:cause) defined in generic package (packageID: g) with reasons FT (“Failure, Temporarily”) and FP (“Failure, Permanent”).

The following are the possible errors:

1. ***An ARP request was sent and no response was supplied (FT cause)*** – Indicating that the internal blade’s configuration is correct and an ARP request was sent, with no response from the remote side. This is treated as a temporary condition. MEGACO will issue a generic event with a cause of “Failure, Temporarily”.
2. ***An ARP request was not sent due to internal configuration problems (FP cause)*** – Indicating that as a result of some error in the internal configuration, the ARP message was not sent to the remote side. MEGACO will issue a generic event with a cause of “Failure, Permanent”.

5.3.2.23 Loss of H.248 Connectivity

Loss of H.248 connectivity (for events such as Ethernet cable disconnections, loss of media stream (H.248) due to peripheral failures, etc.), is passed on to the TDM side by the device, i.e., connectivity loss is signaled to the connected PBXs, which stops routed traffic to a gateway that cannot presently handle any calls.

Upon service re-connection, a Service Change command is generated for all the trunks. The trunks with alarms (i.e., not synchronized) are reported as “FORCED”. The remaining trunks are reported as “RESTART”.

5.3.2.23.1 Enabling the Mechanism

The *DisconnectBehavior* parameter configures the mechanism behavior.

The valid settings are:

1 – No Action (default: do nothing)

2 – Disable Trunks

3 – Reset Device (If there is an existing call, then the software will reset the device; otherwise no action is taken.)

Note: It is the user's responsibility, as a preliminary condition to activate the KeepAlive mechanism.(i.e. implementing Inactivity Timer Package event)

5.3.2.24 RTCP-XR support (H.248.30)

Note: RTCP-XR Support is NOT applicable to **6310/8410/3000** devices.

RTCP extended Reports (XR) are defined in RFC 3611. It expands RTCP with an additional seven blocks of information. One of these blocks of information, the basis for this feature, is the VoIP metrics report block (Block 7). This block provides metrics for monitoring VoIP calls.

The MEGACO ITU standard H.248.30 defines two packages to configure the RTCP-XR and report the statistics: RTCPXR and XRBM.

MEGACO controls the activation of the RTCP-XR statistics calculation and reports the statistics gathered by it. In addition, the activation of RTCP-XR statistics calculation can be carried out by ***ini*** file parameter configuration.

5.3.2.24.1 Activation

If the Feature key allows for it, a call is marked to return RTCP-XR statistics, provided that one or more of the following conditions is fulfilled:

- MEGACO receives an ADD command in which there is an RTCP-XR in the remote SDP with specification of block 7 (a=rtcp-xr: voip-metrics),
- In the ***ini*** file RTCP-XR is set to ON
- The localControl descriptor for this call contained a reference for the RTCP-XR packages. It activates the Voice Engines API for collecting statistics, (if not already active).

5.3.2.24.2 Deactivation

The MEGACO deactivates sending RTCP-XR packets if it gets a remote SDP with an empty RTCP-XR attribute. (a=rtcp-xr:)

5.3.2.24.3 Statistics Report

The statistics for RTCP-XR is reported if the following conditions exist:

- The lower level (Voice Engines) received statistics.
- The call is marked for returning RTCP-XR statistics.

The report is issued when the call manager requests statistics, usually when closing a call (Subtract), but also during the call (Audit).

5.3.3 Graceful Management via MEGACO

Graceful Management (i.e. Locking / Unlocking the device with either Graceful or Forced setting) can be implemented via the H.248 protocol, using the ServiceChange message on the ROOT termination sent from the call agent.

Note: Graceful Management can also be performed using the Web Interface. Refer to the Web Interface section of the product's User's Manual.

Example for ServiceChange:

```
MEGACO/2 [10.4.10.84]:2944
T=5563{
C = - {
SC=ROOT{
SV{
MT=RS, AD = 2944,V=2,PF=TGW/1,RE="905 Termination taken out of
service",DL=578,20000101T00070135}}}}
```

The following fields contribute to the management functionality:

- Method (MT)
- Reason (RE)
- Delay (DL).

5.3.3.1 Graceful Shutdown

Graceful shutdown is performed when MT = Graceful (GR) and DL is greater than 0. Graceful Shutdown is activated with the delay set in the DL. If DL is absent or set to zero, the delay value is considered to be infinite.

5.3.3.2 Canceling a Graceful Shutdown

Graceful Shutdown can be canceled within the time delay set in the DL field. The MT field should be set to Restart (i.e MT = RS). The action following is dependant on the RE field. If RE = 918 (i.e Cancel Graceful), the Graceful Shutdown will be canceled and the gateway will continue to function regularly.

While the device is in the graceful shutdown state when the graceful shutdown is canceled, the device sends a graceful cancel ServiceChange message and the device returns from the graceful shutdown state to the normal running state.

The service change that is sent contains the following:

```
Method=restart
Reason=Cancel Graceful
```

5.3.3.3 Restart

To restart, set the MT field to Restart (MT = RS). If the RE field is not equal to 918 (Cancel Graceful), the gateway will reset.

5.3.3.4 Force Shutdown

To force shutdown, set the MT field to Forced (MT = FO). As a result, the gateway will be blocked immediately and all calls will be dropped.

Note: The way to handle the unlock and release from force lock is to use another administrative tool (e.g., Web interface, SNMP or TPNCP). There is no way to release from force lock using MEGACO.

5.3.3.5 Configurable Profile Names

One of the ServiceChangeDescriptor parameters is the optional ServiceChangeProfile parameter, which specifies the profile (if any) of the protocol supported. The ServiceChangeProfile parameter can be configurable using the *ini* file parameter: "cpServiceChangeProfile" (Type: string, max length: 63 chars, default value: "TGW"). The parameter can also be configured using the Web.

Note: At this stage, the profile will not determine the features supported.

5.3.4 SDP Support in MEGACO

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

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 CPSDPProfile by turning on bit 4 (value 16).

For backward compatibility, the same functionality can be enabled by the value of bit 3 (value 8) for the MEGACO profiling parameter, MGCPCompatibilityProfile.

The ‘o’ line can be configured (if Private Labeling for the gateway is required) via the CPSDPSessionOwner ***ini*** file parameter. The ***ini***i file's parameter’s default value for MEGACO and MGCP remains:'-'. The maximum length for this parameter is 31 characters.

In the ‘a’ line, the general supported attributes in SDP are:

- SILENCESUPP:VAL
  (VAL=on or off) - To turn silence suppression on or off (defined in RFC 3108)

- MAXCONFUSERS:n
  (n - The number of users reserved for this conference). This is a proprietary parameter that allows reservation of conference size. The actual conference length is less than this. The default is 3.

- CONFUSERTYPE:n
  (n - is 0 for not active, 1 for regular user, 2 for listener only and 3 for master). This is a proprietary parameters that defines the user type. The default is 1 - Regular.

- 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 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)

Note: RTPMAP attribute must appear before the FMTP.

- 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
```

Other attributes are supported according to specific feature required (see below).

5.3.4.1 SDP Support Profiling

While adding support for new SDP features, the old behavior must be retained. This is carried out by adding a new ***ini***/Web parameter – cpSDPProfile. This parameter is a bit map, which currently allows for the following:

- Bit 0 (Value 1) enables the support of RFC 3407 (Simple capabilities).
- Bit 1 (Value 2) enables the support of V.152 (Voice Band Data).
- Bit 2 (Value 4) enables the support of RFC 3264 (offerer / answerer).
- Bit 3 (Value 8) controls the type of the SDP Negotiation. If this bit is turned on, strict SDP negotiation is performed, which means that configured default values are ignored in most cases. (Refer to RFC 2833 Support on page 358 and Fax T.38 & Voice Band Data Support on page 368.)
- Bit 4 (Value 16) enables the extra lines in the outgoing SDP ('t' 's' and 'o' lines). (Refer to SDP Support in MEGACO on page 364.)
- Bit 5 (Value 32) - Default packetization period (ptime) for the transparent coder is 10 msec. Using the SDP attribute ptime overrides it.
- Bit 8 (Value 256) - Symmetric payloads. This feature forces the gateway to use the remote dynamic payload for a coder instead of using a default one. Should be used in cases where the remote gateway does not support asymmetric payloads.

5.3.4.2 Selecting a Coder or Ptime Using an Under-Specified Local Descriptor

Before the current version, the supported under-specified fields in the SDP had been:

- IP address
- Port
- Payload

Added from version 4.8, the Profile and Ptime are also supported, as in the following example:

- c=IN IP4 $
- m=audio $ $ $
- a=ptime:$

The reply is a list of all supported coders.

5.3.4.2.1 Selecting a Payload for a Known Coder

There is sometimes a need to set a coder, but let the device define a payload type for it. This is done by setting the payload type to CHOOSE, while specifying the coder name. Only one CHOOSE can appear in a media line. The following example illustrates the usage:

- c=IN IP4 $
- m=audio $ RTP/AVP 18 0 $ 96
- a=rtpmap:$ telephone-event
- a=rtpmap:96 G7291
- a=ptime:$

5.3.4.2.2 Support of Asymmetric Tx/Rx Payloads

In the MEGACO commands, up to two SDP sessions are received. One SDP session for the local side and the another SDP session for the remote side. Each SDP can contain a different definition of the payloads to be used for the same coder. In the reply the result of the negotiation between the local SDP and remote SDP is returned. The reply contains the negotiated coder, and the payload type to be used is included in the SDP session from the local side (not the SDP session from the remote side). As a result, the media stream SENT FROM the device uses the payload received in the remote SDP, but the media stream RECEIVED IN the device is to use the payload type defined in the local SDP.

5.3.4.3 RFC 3407 Support – Simple Capabilities

RCF 3407 defines a minimal and backward-compatible capability declaration feature in SDP by defining a set of new SDP attributes. Together, these attributes define a capability set, which consists of a capability set sequence number followed by one or more capability descriptions. Each capability description in the set contains information about supported media formats, but the endpoint is not required to use any of these. In order to actually use a declared capability, session negotiation must be carried out by the call manager.

Example 1

The following call flows example illustrates the usage of this capability:

```
MEGACO/1 [10.2.1.228]:2944
Transaction = 10264 {
       Context = $ {
          Add  = $ {Media {
                    LocalControl {
                       Mode = Receiveonly
                    },
         Local {
   v=0
   c=IN IP4 $
   m=audio $ RTP/AVP 0
  m=image $ UDPTL t38
},
         Remote {
   v=0
   c=IN IP4 10.4.4.46
   m=audio 4020 RTP/AVP 0
}}}}}
```

The reply to this is:

```
MEGACO/1 [10.4.4.46]:2944
P=10264{
C=2{
A = gwRTP/1{
M{
L{

v=0
c=IN IP4 10.4.4.46
a=sqn: 0
a=cdsc: 1 image udptl t38
a=cpar: a=T38FaxMaxBuffer:1024
a=cpar: a=T38FaxMaxDatagram:238
m=audio 4010 RTP/AVP 0
a=ptime:20
a=silencesupp:off - - - -
}}}}}
```

In this case, the local was requested to use both audio and image, but the remote supports only the audio. The reply will return the image as a capability, in the session level.

Example 2

In this example the capabilities are displayed both in the session and in the media level:

```
MEGACO/1 [10.2.1.228]:2944
Transaction = 10265 {
       Context = $ {
          Add  = $ {Media {
                    LocalControl {
                       Mode = Receiveonly
                    },
         Local {
   v=0
   c=IN IP4 $
   m=audio $ RTP/AVP 0 18 4
  m=image $ UDPTL t38
},
         Remote {
   v=0
   c=IN IP4 10.4.4.46
   m=audio 4020 RTP/AVP 0
}}}}}
```

The reply to this is:

```
MEGACO/1 [10.4.4.46]:2944
P=10265{
C=3{
A = gwRTP/2{
M{
L{

v=0
c=IN IP4 10.4.4.46
a=sqn: 0
a=cdsc: 1 image udptl t38
a=cpar: a=T38FaxMaxBuffer:1024
a=cpar: a=T38FaxMaxDatagram:238
m=audio 4020 RTP/AVP 0
a=ptime:20
a=silencesupp:off - - - -
a=cdsc: 2 audio RTP/AVP 0 18 4
a=cpar: a=fmtp:18 annexb=yes
a=cpar: a=rtpmap:4 G723/8000/1
a=cpar: a=fmtp:4 bitrate=6.3;annexa=yes
a=cpar: a=ptime:20
a=cpar: a=silencesupp:off - - - -
}}}}}
```

In Example 2, the local was requested to use both audio and image, but the remote supports only the audio and parts of the coders. The reply returns the image as a capability in the session level and the fully supported coders in the media level.

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

5.3.4.4.1 Support of Fax and Modem Type by Default Parameters

To support T.38 by default, without MEGACO interference, configure the device as follows:

- Bit 3 (Value 8) of CPSDPProfile parameter is turn off
- FaxTransportType is set 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 the value, 4.

Following these rules, the channel is opened with T.38 and transition to T.38 is performed automatically upon detection. The T.38 fax port is assumed to be the RTP port + 2, both for the local and remote side

Bypass (VBD) mode can also be supported by default, without MEGACO interface, by configuring ***ini*** file parameters:

- Bit 3 (Value 8) of CPSDPProfile parameter is turned off
- FaxTransportType is set to Bypass.
- VxModemTransportType (x stands for 21, 22, 23, 32, 34) is set to Bypass.
- The packetization period is configured by the parameter FaxModemBypassBasicRTPPacketInterval.
- The payload to be used is configured by the parameter FaxBypassPayLoadType and ModemBypassPayloadType.

5.3.4.4.2 Negotiating Fax and Modem type via SDP

Support of the Fax type (T.38, Bypass or Transparent) and modem type (Bypass, 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 3 (value 8) of the SDP profiling parameter CPSDPProfile. If this bit is not set, the device uses a non strict negotiation (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.
- The modems transport type and payload will be set according to the configuration defaults as before.

However, if this bit is set, the negotiation is strict and rules are as follows:

- If the 'm=image' line is not received both in local AND remote descriptors, T.38 is NOT used.
- If the local SDP "m=audio" line contains the G.711 coder in addition to another voice coder, the fax (if not set previously to T.38) and modem mode is Bypass (VBD), and the G.711 payload type is used for the fax and modem. Note that this is a proprietary way to define a VBD coder. This can be avoided by using the V.152 VBD attribute (See next section).
- If the additional G.711 coder is not offered in the local SDP the Fax (if not set to T.38) and modem Transport Type is Transparent.
- If the fax redundancy attribute line does not appear both in local and remote descriptors, redundancy for non-V21 packets is NOT used.

For backward compatibility, the same functionality can be enabled by the value of bit 2 (value 4) for the MEGACO profiling parameter, MGCPCompatibilityProfile.

5.3.4.5 Media Encryption (SRTP) using RFC 3711

SRTP (RFC 3711) details the media encryption standard. The device partially implements it. RFC 3711 defines a new media profile “RTP/SAVP” for use in secured streams. (The non-secured profile is “RTP/AVP”).

SRTP defines how to encrypt the media, but does not define how to negotiate the key. For negotiation with the key, the method used is defined in RFC 4568.

This RFC defines a cryptographic attribute for SDP to use for media encryption.

There is no official definition for how to use this in MEGACO, therefore, the following describes the implementation.

5.3.4.5.1 Supported Suites

The device SRTP implementation is limited to AES_CM_128_HMAC_SHA1_32 and AES_CM_128_HMAC_SHA1_80. All other suites are ignored.

The only supported key parameter is MKI. The length of the MKI is limited to 4 bytes. If the remote side sends a longer MKI, this specific key will be ignored. This means that if this is the only key, the call will fail.

The key lifetime field is not supported. However, if it is included in the key it will be silently ignored and the call will not fail.

The SRTP suite may hold many keys and key parameters. The device supports a single key and no key parameters. Suites that are provided with more than one valid key are ignored, and marked as not valid.

5.3.4.6 Supported Session Parameters

The following session parameters are supported:

- UNENCRYPTED_SRTP
- UNENCRYPTED_SRTCP
- UNAUTHENTICATED_SRTP

Session parameters should be the same for both the local and remote sides. When the device initiates the call, the session parameters will be defined according to *ini* file parameters (see below). When the device is the answering side, the parameters will be adjusted to the remote offering.

Unsupported session parameters are ignored, and will not cause a call failure. Note, however, that our implementation has a limitation in supporting un-authentication and un-encryption together on the same side. This combination will cause the specific line to be ignored.

5.3.4.6.1 Configuration and Activation

The device supports two packages of media encryption, TGCP and SRTP. MEGACO, however, supports only SRTP.

The following defines the encryption support level:

1. DSP template - Templates 0 and 2 support SRTP and template 3 supports TGCP.
2. Feature Key – Enables/Disables media encryption on the device.
3. *ini* file parameter – The *EnableMediaSecurity* parameter, defines SRTP support when set to Enable.
4. *ini* file parameter – The *SRTPTxPacketMKIASize* parameter defines the length of the local MKI, used to identify the local key. The range of this parameter is 0-4.
5. *ini* file parameter – The RTPEncryptionDisableTx parameter can be set in order to work with non-encrypted RTP.
6. *ini* file parameter – The RTCPEncryptionDisableTx parameter can be set in order to work with non-encrypted RTCP.
7. *ini* file parameter – The RTPAuthenticationDisableTx parameter can be set in order to work with non-authenticated RTP.

Even if the device is configured to support encryption, the actual activation must be done on a per command basis. Activation of a secured connection is done by sending to the device a local descriptor in which the transport method is “RTP/SAVP” (defined in RFC 3711). The local descriptor may contain more parameters regarding the encryption as described below.

5.3.4.6.2 SDP Definition

The following attribute is defined in RFC 4568.

```
a=crypto:<tag> <crypto-suite> <key-params> [<session-params>]
```

The fields tag, crypto-suite, key-params, and session-params are described in the sub-sections below, and an example is provided of the crypto attribute for the "RTP/SAVP" transport, i.e., the secure RTP extension to the Audio/Video Profile [srtp].

In the following, new lines are included for formatting purposes only:

```
a=crypto:1 AES CM 128 HMAC SHA1 80
inline:PS1uQCVeeCFCanVmcjkpPywjNWhcYD0mXXtxaVBR|2^20|1:32
```

In MEGACO, the following fields are allowed to be under specified:

- **Tag** – If the tag is under specified, the rest of the line can be omitted. This means that the gateway returns **all** the supported suites. for the device, the following is expected when sending 'a=crypto:$':

  a=crypto:1 AES_CM_128_HMAC_SHA1_32
inline:MKHEBFC/PMKHEB+CJfvspnkheifcZW

  a=crypto:2 AES_CM_128_HMAC_SHA1_80
inline:9630xvspsqnkhecZEC/8520xurolpm

- **crypto-suite** – If the crypto suite is under specified, the gateway may chose one of the supported suites. In this case, however, the key params field should also exist and contain '$'. The answer to 'a=crypto:1 $ $' is, for example:

  a=crypto:1 AES_CM_128_HMAC_SHA1_32
inline:MKHEBFC/PMKHEB+CJfvspnkheifcZW

- **key-params** – When the key param is under specified, it means that the sender wants a specific suite, and wants the gateway to produce the key. an example of the request is:

  _'a=crypto:1 AES_CM_128_HMAC_SHA1_80 $'

  and the reply:

  a=crypto:1 AES_CM_128_HMAC_SHA1_80
inline:9630xvspsqnkhecZEC/8520xurolpm

- session-params – When the session parameters are omitted, a default will be taken according to the blade configuration. (refer to the configuration section above). Setting values to these parameters, however, will override the defaults.

5.3.4.6.3 Connection Negotiation

The examples below show the creation of a secured connection via the ADD command. This can also be done by the Modify command. In this case, the connection starts in a non-secured mode and updated to a secured mode. (The opposite is also possible – to start with secured mode and move to a non-secured mode).

Simple Offerer for Secured Connection

In this example, the call manager sends an under specified SDP, and requests a secured connection. Note that there are no attribute lines for SRTP, and this is considered as if 'a=crypto:$' was received: (Refer to the previous section, item 1).

The MGC sends:

```

MEGACO/1 [10.2.1.228]:2944
   Transaction = 1 {
       Context = $ {
          Add = $ {Media {LocalControl {
                   Mode = Receiveonly},
                   Local {
v=0
c=IN IP4 $
m=audio $ RTP/SAVP 0
a=ptime:20
 }}}}}
```

The Gateway answers:

```
MEGACO/1 [10.4.4.46]:2944
   P = 1 {
       C = $ {
          A = $ {M {O {
                   MO = Receiveonly},
                   L {
v=0
c=IN IP4 10.4.4.46
m=audio 4000 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_32
inline:MKHEBFC/PMKHEB+CJfvspnkheifcZW
a=crypto:2 AES CM 128 HMAC SHA1 80
inline:9630xvspsqnkhecZEC/8520xurolpm
a=ptime:20
 }}}}}
```

Simple Offerer for Both Secured and Non-Secured Connection

In this example, the call manager sends an under specified SDP, but this time requests both secured and non-secured connections. This is the more general scenario, as the MGC must make sure that if the remote side does not support SRTP, the call does not fail (assuming that there is no request for a secured only call).

Note that there are no attribute lines for SRTP, and this is considered as if ‘a=crypto:$’ was received: (Refer to the previous section, item 1).

The MGC sends:

```
MEGACO/1 [10.2.1.228]:2944
   Transaction = 2 {
       Context = $ {
          Add = $ {Media {LocalControl {
                   Mode = Receiveonly},
                   Local {
v=0
c=IN IP4 $
m=audio $ RTP/SAVP 0
m=audio $ RTP/AVP 0
a=ptime:20
 }}}}}
```

The Gateway answers:

```
MEGACO/1 [10.4.4.46]:2944
   P = 2{
       C = 2 {
          A = GWRTP/2 {M {O {
                   MO = Receiveonly},
                   L {
v=0
c=IN IP4 10.4.4.46
m=audio 4010 RTP/SAVP 0
a=crypto:1 AES CM 128 HMAC SHA1 32
inline:MKHEBFC/PMKHEB+CJfvspnkheifcZW
a=crypto:2 AES CM 128 HMAC SHA1 80
inline:9630xvspsqnkhecZEC/8520xurolpm
m=audio 4010 RTP/AVP 0
a=ptime:20
 }}}}}
```

Oferrer – Choosing One Suite

In this example, the MGC wants the Gateway to choose one suite. The Gateway chooses the suite and also the key.

The MGC sends:

```
MEGACO/1 [10.2.1.228]:2944
   Transaction = 3 {
       Context = $ {
          Add = $ {Media {LocalControl {
                   Mode = Receiveonly},
                   Local {
v=0
c=IN IP4 $
m=audio $ RTP/SAVP 0
a=crypto:1 $ $
a=ptime:20
 }}}}}
```

The Gateway answers:

```
MEGACO/1 [10.4.4.46]:2944
   P = 3 {
       C = 3 {
          A = GWRTP/3 {M {O {
                   MO = Receiveonly},
                   L {
v=0
c=IN IP4 10.4.4.46
m=audio 4020 RTP/SAVP 0
a=crypto:1 AES CM 128 HMAC SHA1 32
inline:MKHEBFC/PMKHEB+CJfvspnkheifcZW
a=ptime:20
 }}}}}
```

Oferrer – Suite is Defined

In the example, the MGC wants the Gateway to work with a specific suite and produce the key. The Gateway returns the chosen key:

The MGC sends:

```
MEGACO/1 [10.2.1.228]:2944
   Transaction = 4 {
       Context = $ {
          Add = $ {Media {LocalControl {
                   Mode = Receiveonly},
                   Local {
v=0
c=IN IP4 $
m=audio $ RTP/SAVP 0
a=crypto:1 AES CM 128 HMAC SHA1 32 $
a=ptime:20
 }}}}}
```

The Gateway answers:

```
MEGACO/1 [10.4.4.46]:2944
   P = 4 {
       C = 4 {
          A = GWRTP/4 {M {O {
                   MO = Receiveonly},
                   L {
v=0
c=IN IP4 10.4.4.46
m=audio 4030 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_32
inline:MKHEBFC/PMKHEB+CJfvspnkheifcZW
a=ptime:20
 }}}}}
```

Answerer – Local Parameters Not Defined

In this example, the MGC sends the basic SDP to the local side and the offered data from the remote side. The Gateway negotiates the data and returns the result:

The MGC sends:

```
MEGACO/1 [10.2.1.228]:2944
   Transaction = 4 {
       Context = $ {
          Add = $ {Media {LocalControl {
                   Mode = Receiveonly},
                   Local {
v=0
c=IN IP4 $
m=audio $ RTP/SAVP 0
a=ptime:20
 },
                   Remote {
v=0
c=IN IP4 10.4.4.46
m=audio 4000 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_32
inline:MKHEBFC/PMKHEB+CJfvspnkheifcZW
a=crypto:2 AES CM 128 HMAC SHA1 80
inline:9630xvspsqnkhecZEC/8520xurolpm
a=ptime:20
}}}}}
```

The Gateway answers:

```
MEGACO/1 [10.4.4.46]:2944
   P = 4 {
       C = 5 {
          A = GWRTP/5 {M {O {
```

```
                   MO = Receiveonly},
                   L {
v=0
c=IN IP4 10.4.4.46
m=audio 4040 RTP/SAVP 0
a=crypto:1 AES CM 128 HMAC SHA1 32
inline:8375hrytsqnkhecPOE/8732xurnrtd
a=ptime:20
 }}}}}
```

Error Cases

The negotiation results in an error if there is no supported SDP at the end of it. This can be caused by one of the following:

1. The MGC requests a secured connection ONLY, but the Gateway does not support it.
2. The Gateway supports SRTP, but not the suite requested by the MGC.
3. The remote side sends SDP with a secured connection ONLY and the Gateway does not support it.
4. The suites sent by the remote side are not supported by the Gateway.
5. The suite (sent by MGC or remote side) is supported, but there are session parameters that are not supported or contain more than one key.

5.3.4.7 Support of RFC 3264

The terms, “oferrer” and “answerer” used in device, are originally defined in RFC 3264. This RFC is currently partially supported. The device can receive a media line with port number 0, and treat it as a statement that this is not supported.

5.3.4.8 EVRC Family Coders

5.3.4.8.1 EVRC Coders

The EVRC coders are supported according to the following standards:

1. RFC3558 - two types of coders exist in the EVRC family:
   a. EVRC0 - This is the Header-free format. Each frame can include only one packet and each can be of a different rate.
   b. EVRC - This is the Bundling format. Each frame can include more than one packet, each with a different rate. In order for the receiver to encode the packets, there is a TOC at the beginning. The RFC also defines an interleaving format, but we do not support it. The parameter "maxinterleave" which is defined in the RFC will always be returned as zero.
2. Draft-ietf-avt-compact-bundled-evrc-11.txt - defines one more EVRC coder:
   a. EVRC1 - This is similar to EVRC0 (Header free), but adds the ability to define which rate to use. The default rate behavior (fixed or variable) is defined by a configuration parameter "EVRCRate". This default can be overridden by using the parameter defined by the fmtp parameter "evrcfixedrate", which is defined in the aforementioned draft, and valid only for EVRC1.

In addition to the official coders defined above, two proprietary coders are supported:

1. X-EVRC-TTY - This is an EVRC coder with TTY support.

2. X-EVRC-TFO - This is an EVRC coder with TFO support.

For these two, there is no way to define Header-free or Bundling. Therefore, the parameter "maxinterleave" is used to distinguish between the two formats. If the parameter was used in the command, the Bundling format is used. Otherwise, the device default will be used as defined by the configuration parameter "VBRCoderHeaderFormat".

Note: The following EVRCB coders are applicable to **6310/8410/3000** products.

5.3.4.8.2 EVRCB Coders

The draft, "Draft-ietf-avt-compact-bundled-evrc-11.txt" defines a new coder type - EVRCB. In parallel to the EVRC coders, it defines the following:

- EVRCB - Bundled format (Parallel to EVRC)
- EVRCB0 - The header free format with variable rate (Parallel to EVRC0)
- EVRCB1 - The header free format (Parallel to EVRC1).

5.3.4.9 Silence Suppression Support in EVRC Coders

The 'Draft-ietf-avt-compact-bundled-evrc-11.txt' draft defines 4 new parameters to support silence suppression in EVRC. It also defines that the default for the silence suppression is ON. Here is an example for SDP session which defines these parameters:

```
v=0
   c=IN IP4 $
   m=audio $ RTP/AVP 97
   a=rtpmap:97 EVRC1
   a=fmtp:97 fixedrate=0.5
   a=fmtp:97 silencesupp=1 dtxmax=100 dtxmin=5 hangover=0
   a=ptime:60
```

If both sides returns the silence suppression as ON, it will be turned on in the call. Note that we use the default value for the "hangover" parameter.

5.3.4.10 AMR Coders Rate Change

Note: The sub-section on AMR Coders Rate Change is only applicable to **6310/8410/3000** devices.

A pre-defined table can be configured to give a set of rules for an automatic AMR rate change. The decision for the change is based upon the packet loss rate. For more information about this option, contact AudioCodes Technical Support.

5.3.4.11 V.152 - VBD Attribute Support

The V.152 defines a way to declare support of VBD (Voice Bond Data), and define which coder and payload will be used for it. This is done by using a new SDP attribute 'gpmd'.

(See www.ietf.org/internet-drafts/draft-rajeshkumar-mmusic-gpmd-03.txt for more information about this “General Purpose Media Description” attribute.)

The below is an example from V.152. It shows how we define the VBD support using the new attribute:

```
    m=audio 3456 RTP/AVP 18 0 13 96 98 99
    a=rtpmap:96 telephone-event/8000
    a=fmtp:96 0-15, 34, 35
    a=rtpmap:98 PCMU/8000
    a=gpmd:98 vbd=yes
    a=rtpmap:99 G726-32/8000
    a=gpmd:99 vbd=yes
```

In the example the sender supports voice on G729 and PCMU, and VBD data on both PCMU with payload 98, and G726-32 with payload 99. This new attribute enables the use of the same coder but with two different payload types.

The behavior of the device will depend on a new bit in the SDP profile parameter (See above). If the profile is off, the behavior stays as before, with the following additions:

- If we get only under specified local descriptor from the MGC (Oferrer), and it contains a VBD attribute, our answer will include it.
- If we get both local and remote descriptors (Answerer), and the remote contains a VBD attribute, our answer will include it if the negotiation succeeds.

When the profile bit is turned on, the behavior will be as follows:

- If we get only under specified local descriptor from the call manager (Oferrer), we will include the VBD attribute if the G.711 coder existed in the request (for backward compatibility) or a VBD was specified in the request. In the first case (only G.711), we will return the G.711 as a normal voice coder, but also add a new dynamic payload with the same G.711 coder to indicate that we support VBD for it. (See example 1 below)
- If we get both local and remote descriptors (Answerer), and the local contains G.711, and the remote contains this G.711 and VBD, we will adopt the payload of the remote for our VBD. (See example 2 below)

```
Example 1:
The received SDP:
Local{
    v=0
    c=IN IP4 $
    m=audio $ RTP/AVP 18 0
}

The reply for this will be:
Local{
    v=0
    c=IN IP4 10.4.4.46
    m=audio 4000 RTP/AVP 18 0 104
    a=rtpmap:104 PCMU/8000
    a=gpmd:104 vbd=yes
}

Example 2:
The received SDP:
Local{
    v=0
    c=IN IP4 $
    m=audio $ RTP/AVP 18 0
},
remote{
    v=0
    c=IN IP4 10.4.4.46
    m=audio 4000 RTP/AVP 18 0 104
```

```
     a=rtpmap:104 PCMU/8000
     a=gpmd:104 vbd=yes
}

The reply for this will be:
Local{
     v=0
     c=IN IP4 10.4.4.46
     m=audio 4010 RTP/AVP 18 0 104
     a=rtpmap:104 PCMU/8000
     a=gpmd:104 vbd=yes
}
```

5.3.5 Mapping Payload Numbers to Coders

The table below 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.

These values can be overridden by the external CoderTable.

Table 5-33: MEGACO Mapping Payload Numbers to Coders

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

Table 5-33: MEGACO Mapping Payload Numbers to Coders

Default Payload Number	Encoding Name	Coder
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
56	"X-CCD"	Transparent
60	"EVRC0"	EVRC0
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)
100	“iLBC”	iLBC (13)
101	“iLBC”	iLBC (15)
102	“BV16”	BV16
96	“telephone-event”	RFC 2833
104	“RED”	Redundancy per RFC 2198
13	“CN”	Comfort Noise
121	“EG711A”	EG711 ALAW
122	“EG711U”	EG711 MULAW

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.

5.3.6 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 protocol supports the basic set of packages as defined in Annex E of RFC 3015 (Refer to the document at www.letf.org/rfc/ - refer to the '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.

5.3.6.1 General Packages

Table 5-34: General Packages

Package Name	Pkg Id	Standard	Supporting Devices	First Supported Version	Note
Generic Package	g	H.248.1	All	3.8	
Base Root Package	root	H.248.1	All	3.8	
Tone Generator Package	tonegen	H.248.1	All	3.8	
Tone Detection Package	tonedet	H.248.1	All	3.8	
Basic DTMF Generator Package	dg	H.248.1	All	3.8	
DTMF detection Package	dd	H.248.1	All	3.8	
Call Progress Tones Generator Package	cg	H.248.1	All	3.8	
Call Progress Tones Detection Package	cd	H.248.1	All	3.8	
Network Package	nt	H.248.1	All	3.8	
RTP Package	rtp	H.248.1	All	3.8	
TDM Circuit Package	tdmc	H.248.1	All	3.8	
Call Type Discrimination Package	ctyp	H.248.2	All	4.2	
IP Fax Package	ipfax	H.248.2	All	4.2	

Table 5-34: General Packages

Package Name	Pkg Id	Standard	Supporting Devices	First Supported Version	Note
Generic Announcement Package	an	H.248.7	All	3.8	
Congestion Handling Package	chp	H.248.10	device family only	5.2	
Overload Control Package	ocp	H.248.11	device family only	5.2	
Inactivity Timer Package	it	H.248.14	All	4.6	
Extended DTMF Detection Package	xdd	H.248.16	All	4.2	
Enhanced DTMF Detection Package	edd	H.248.16	All	4.2	
Enhanced alerting package	Alert	H.248.23	All	4.6	Only Call Waiting is supported on all devices
Multi-frequency tone generation package	mfg	H.248.24	All	4.2	
Multi-frequency tone detection package	mfd	H.248.24	All	4.2	
Conferencing Tones Generation Package	conftn	H.248.27	All	4.8	
Carrier Tones Generation Package	carr	H.248.27	All	5.0	
RTCP XR Base Package	rtcpxr	H.248.30	Not device family	5.0	
RTCP XR Burst Metrics Package	xrbm	H.248.30	Not device family	5.0	
Detailed Congestion Reporting Package	dcr	H.248.32	device family only	5.0	
Hanging Termination Detection Package	hangterm	H.248.36	All	5.2	
IP NAT Traversal Package	ipnapt	H.248.37	All	5.2	
Basic Call Progress Tones Generator with Directionality	bcg	Q.1950	All	4.6	
Basic Services Tones Generator Package	srvtn	Q.1950	All	4.6	

Table 5-34: General Packages

Package Name	Pkg Id	Standard	Supporting Devices	First Supported Version	Note
Expanded Services Tones Generation Package	xsrvtn	Q.1950	All	4.6	
Expanded Call Progress Tones Generator Package	xcg	Q.1950	All	4.6	
Differentiated Services Package	ds	ETSI TS102.333 Annex A	All	5.0	

5.3.6.2 Trunking Gateway Packages

Table 5-35: Trunking Gateways Packages

Package Name	Pkg Id	Standard	Supporting Devices	First Supported Version	Note
Basic Continuity Package	ct	H.248.1	All	4.2	Only 4 wire supported
Basic CAS package	bcas	H.248.25	All	4.6	
Basic CAS addressing package	bcasaddr	H.248.25	All	4.8	
Robbed bit signalling package	rbs	H.248.25	All	4.8	Signals generation only from 5.2
Operator services and emergency services package	oses	H.248.25	All	4.8	
Operator services extension package	osext	H.248.25	All	5.2	
International CAS Package	icas	H.248.28	All	4.6	
CAS Blocking Package	casblk	H.248.28	All	4.6	
International CAS Compelled Package	icasc	H.248.29	All	4.6	

Note: The following 3G Packages Table is only applicable to 6310/8410/3000 devices.
For further information contact AudioCodes Technical Support.

5.3.6.3 3G Packages

Table 5-36: 3G Packages

Package Name	Pkg Id	Standard	Supporting Devices	First Supported Version	Note
Bearer Characteristics	bcp	Q.1950	6310 only	4.6	
Bearer Network Connection Cut Through	bnct	Q.1950	6310 only	4.6	
Generic Bearer connection	gb	Q.1950	6310 only	4.6	
Bearer Control Tunneling	bt	Q.1950	6310 only	4.6	
3G User Plane	Threegup	3GPP TS 29.232	6310 only	4.6	
3G TFO Control	Threegtfoc	3GPP TS 29.232	6310 only	4.6	

Note: The following Media Server Packages Table is only applicable to **IPmedia.**

5.3.6.4 Media Server Packages (IPmedia only)

Table 5-37: Media Server Packages

Package Name	Pkg Id	Standard	Supporting Devices	First Supported Version	Note
Basic Announcement Syntax Package	bannsyx	H.248.9			Syntax only
Voice Variable Syntax Package	vvsyx	H.248.9			Syntax only
Announcement Set Syntax Package	setsyx	H.248.9			Syntax only

Table 5-37: Media Server Packages

Package Name	Pkg Id	Standard	Supporting Devices	First Supported Version	Note
General Text Variable Type Package	phrsyx	H.248.9			Syntax only
Advanced Audio Server Base Package	aasb	H.248.9	All IPM	4.2	
AAS Digit Collection Package	aasdc	H.248.9	All IPM	4.2	
AAS Recording Package	aasrec	H.248.9	All IPM	4.2	
AAS Segment Management Package	aassm	H.248.9	All IPM	4.2	
Floor Control Package	fcp	H.248.19	8410 + Video	5.2	
Indication of Being Viewed Package	indview	H.248.19	8410 + Video	5.2	
Voice Activated Video Switch Package	vavsp	H.248.19	8410 + Video	5.2	
Lecture Video Mode Package	lvmp	H.248.19	8410 + Video	5.2	

Note: The following Media Server Packages Table is only applicable to **IPmedia 2000**.

5.3.6.5 Media Server Packages (IPmedia 2000)

Table 5-38: Media Server Packages

Package Name	Pkg Id	Standard	Supporting Devices	First Supported Version	Note
Basic Announcement Syntax Package	bannsyx	H.248.9			syntax only
Voice Variable Syntax Package	vvsyx	H.248.9			syntax only
Announcement Set Syntax Package	setsyx	H.248.9			syntax only
General Text Variable Type Package	phrsyx	H.248.9			syntax only
Advanced Audio Server Base Package	aasb	H.248.9	All IPM	4.2	

Table 5-38: Media Server Packages

Package Name	Pkg Id	Standard	Supportin g Devices	First Supported Version	Note
AAS Digit Collection Package	aasdc	H.248.9	All IPM	4.2	
AAS Recording Package	aasrec	H.248.9	All IPM	4.2	
AAS Segment Management Package	aassm	H.248.9	All IPM	4.2	
Floor Control Package	fcp	H.248.1 9	8410 + Video	5.2	
Indication of Being Viewed Package	indview	H.248.1 9	8410 + Video	5.2	
Voice Activated Video Switch Package	vavsp	H.248.1 9	8410 + Video	5.2	
Lecture Video Mode Package	lvmp	H.248.1 9	8410 + Video	5.2	

5.3.7 MEGACO Profiling

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

- Bit 2 (Value 4) - Controls the type of support for the Fax T.38 negotiation. (Refer to Fax T.38 and Voice Band Data Support (Bypass Mode) on page 368 and controls the type of RFC 2833 negotiation (refer to RFC 2833 Support on page 358). This bit has been deprecated as of Version 5.2 and replaced by Bit 3 (value 8) of CPSDPProfile parameter. (It is to be removed in a later software version.)
- Bit 3 (Value 8) - Enables the extra lines in the outgoing SDP ('t' 's' 'o' lines). (Refer to SDP Support in MEGACO on page 364.) This bit has been deprecated as of Version 5.2 and replaced by Bit 4 (value 16) of CPSDPProfile parameter. (It is to be removed in a later software version.)
- 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. This functionality has been deprecated as of Version 5.2 and replaced by Bit 5 (value 32) of CPSDPProfile parameter. (It is to be removed in a later software version.)
  - 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. This functionality has been deprecated as of Version 5.2 and replaced by Bit 3 (value 8) of CPSDPProfile parameter. (It is to be removed in a later software version.)
  - When sending a notification transaction request, the device does not mark it

as optional.

5.3.8 MEGACO Termination Naming

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

5.3.8.1 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 cannot 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.

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

Note: The following four patterns are only applicable to **IPmedia family devices**.

- AUDIOTERMPATTERN - Pattern for ephemeral terminations used ONLY for voice prompts and call progress tones playing.
- TRUNKTESTTERMPATTERN - Pattern for ephemeral terminations used for trunk testing.
- CONFERENCETERMPATTERN - This pattern does not represent a specific termination type. It is the name of the conference pool. It is used in the proprietary "X-UPDATE" service change to report the status of the pool.
- BCTTERMPATTERN - This pattern does not represent a specific termination type. It is the name of the Bearer Channel Tandeming pool. It is used in the proprietary "X-UPDATE" service change to report the status of the pool.

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
    - EP_NUM_0 - Defines the starting trunk number
    - EP_NUM_1 - Defines the starting channel number
- RTP_NUM - Defines the starting number for the RTP terminations. (The default value is 0.)

Endpoints can also be hierarchical to groups i.e. Endpoints (or Terminations) can be addressed as "ds3*/tr*/*", instead of the serial numbering of trunks as described above. The hierarchy can be up to 5 levels. This can be used for DS3 numbering of trunks.

Endpoints hierarchy is set using the *ini* file parameters:

EP_Num_0 - EP_Num_4 specifies the start index within each group.

EP_Min_0 - EP_Min_4 should be set to 0.

EP_Max_0 - EP_Max_4 specifies the member count for each group, counting from the corresponding EP_Min parameter.

PHYSTermNamePattern describes the pattern of the hierarchy, e.g., a 3 level hierarchy can be set as: DS3*/*/*

Example of Setting *ini* File Parameters for DS3 Numbering of Trunks:

```
PHYSTermNamePattern=DS3*/*/*
EP Num 0 = 0
EP Num 1 = 4 --> example for start index that is not 0, in this
case the trunks within each trunk groups start with index 4
EP Num 2 = 1

EP Min 0 = 0
EP Min 1 = 0
EP Min 2 = 0

EP Max 0 = 1--> 2 trunk groups
EP Max 1 = 3 --> each group holds 4 trunks
EP_Max_2 = 31
```

5.3.8.2 Defining Field Width in the Termination Name

As explained above, a field is defined in a pattern by using '*'. If defining a fixed width for this field is needed, a leading zero should be used in the pattern definition. The possible width options are 1 or 2 characters. Therefore, only one leading zero is supported. For example:

- Pattern is "gws0*chan0*" - In this format, the name for trunk 1 and channel 1 is "gws01chan01", and the name for trunk 1 channel 12 is "gws01chan12".
- Pattern is "gws*chan*" - In this format, the name for trunk 1 and channel 1 is "gws1chan1", and the name for trunk 1 channel 12 is "gws1chan12".

5.3.9 MEGACO Version Negotiation

The device supports version negotiation with the Gateway Controller.

H.248 V1 messages are identified by the MEGACO/1 header.

H.248 V2 messages are identified by the MEGACO/2 header.

H.248 V3 messages are identified by the MEGACO/3 header.

The first ServiceChange sent by the device is encoded as a V1 message but with the "Version=n" parameter, where n is the version number of H.248.1 supported by the device (the "suggested version").

By default, the suggested version is "2", but this value can be over-written by the MegacoVersion configuration parameter.

For example:

```
 MEGACO/1 [10.4.4.175]:2944
 Transaction=4630{
 Context = - {
 ServiceChange=ROOT{ Services{ Method=Restart,ServiceChangeAddress
= 2944,
                        Version=3,  Profile=TGW/1,Reason="901 MG
Cold Boot"}}}}
```

The Media Gateway Controller should reply with the *v=n* parameter where *n* is the highest version supported by the Media Gateway Controller, that is either smaller or equal to the suggested version.

For example, if the Media Gateway Controller supports V3, it should reply with the *"v=3"* parameter.

```
MEGACO/1 [10.4.2.67]:2944
P=4630{
C=-{SC=ROOT{SV{V=3}}}}
```

Note: The version number in the reply header is of no significance.

All following messages should conform to V3.

5.3.10 H.248.1 V2 - Main Changes

H.248.1 V2 (April 2003) supports ABNF messages. ASN.1 messages are supported for V1 only.

These are the main changes of V2:

1. **Ability to audit specific properties, events, signals and statistics.**

For example, if you want to audit the local control mode of terminations, instead of auditing full media details of terminations, do the following:

```
MEGACO/2 [10.2.207.145]:2944
  TRANSACTION = 1845 {
       CONTEXT = *{AUDITVALUE = GWS0c*{
                   AUDIT{media}
       }
    }
}
```

Audit only the value you are interested in:

```
MEGACO/2 [10.2.207.145]:2944
    TRANSACTION = 1845 {
       CONTEXT = *{AUDITVALUE = GWS0c*{
                   AUDIT{media{LocalControl  { Mode }}}
       }
    }
}
```

2. **Allowing topology to be set per stream.**

For example:

```
MEGACO/2 [10.2.1.228]:2944
Transaction = 1237 {
       Context = 1 {
        Topology{gws0c4, gws0c3, bothway}
}}
```

This can be expanded with the stream parameter (in which case this topology is relevant for the specific stream only).

```
MEGACO/2 [10.2.1.228]:2944
Transaction = 1237 {
       Context = 1 {
        Topology{gws0c4, gws0c3, bothway,Stream=3}
}}
```

3. **The GW supports version negotiation with the controller.**

   The suggested version number is "2". Refer to MEGACO Version Negotiation sub-section above for more details.

5.3.11 H.248.1 V3 - Main Changes

The following are the main V3 changes supported by the device:

1. Parsing of H.248.1 V3 messages - Although not all of the V3 features are supported, the parsing of V3 syntax is supported and appropriate error codes are returned for unsupported features.
2. Statistics descriptor in the Media Gateway Controller requests – This feature is partially supported and if the statistics descriptor is received in and Add or Modify command on the termination level, then only the required statistics will be returned when needed. However, the reset of the statistics is not supported.
3. Out of Service Flag - This is a new ServiceChange parameter which indicates that the device is not ready yet to work and is still in registration or restart phase. The first ServiceChange which is encoded as a V1 message, carries the flag in the form of an extension parameter, X-SC="SIC=ON".

For example:

```
MEGACO/1 [10.4.4.176]:2944
   T=20339{
   C = - {
      SC=GWS0C*{
          SV{ MT=RS,RE="900 Service Restored",V=3,X-SC="SIC=ON"
}}}}
```

The ServiceChange message which is encoded as V3 will be in the regular SIC form.

For example:

```
MEGACO/3 [10.4.4.176]:2944
   T=20339{
   C = - {
      SC=GWS0C*{
          SV{ MT=RS,RE="900 Service Restored", SIC }}}}
```

5.3.12 CAS to Analog Mapping Protocol

Note: This section is applicable to **TP-6310 and Mediant 3000**.

This section describes the device implementation of the AudioCodes proprietary CAS to Analog Mapping Protocol.

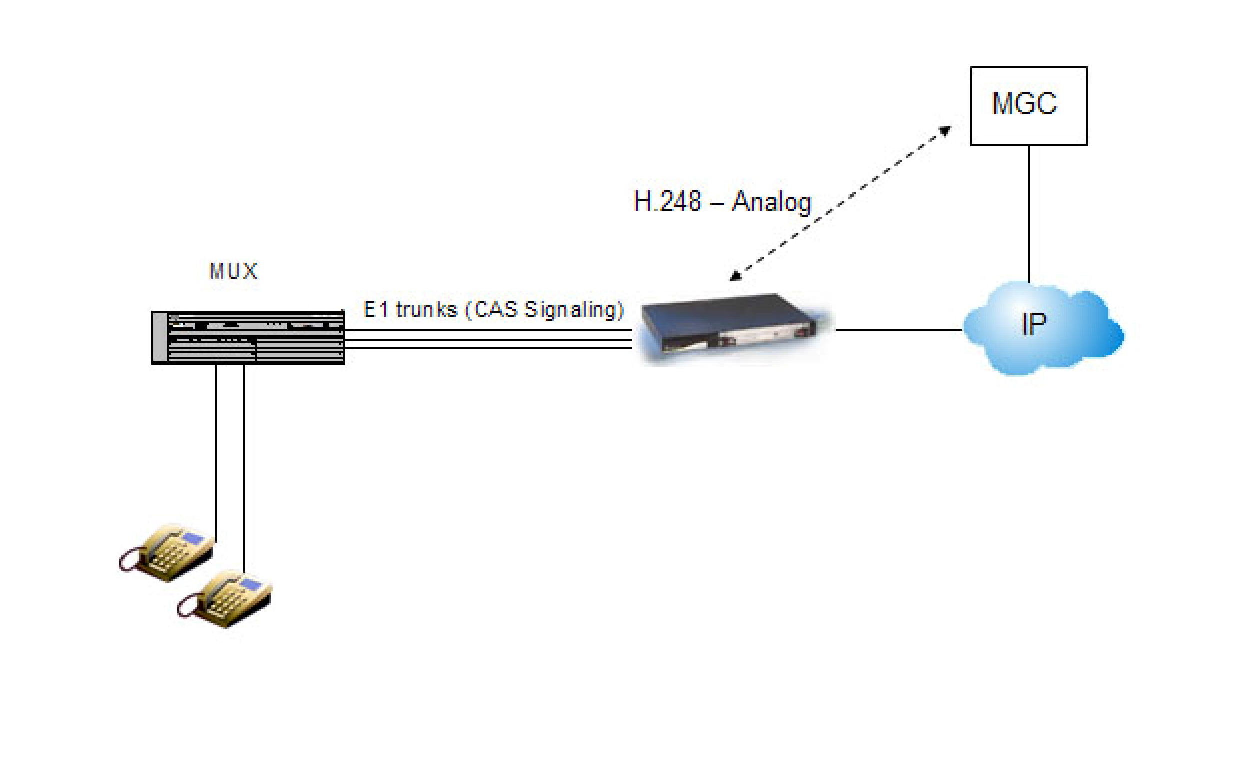
Note 1: There is no standard protocol that defines how to map CAS to Analog.

Note 2: This mapping protocol is only supported in MEGACO.

Note 3: Currently only MelCAS, Ground Start and Loop Start CAS protocols are supported.

When configuring the device to work in this mode (when the physical PSTN layer is CAS) the gateway will appear as a regular Analog Gateway to the GWC (GW controller). The gateway will support Analog packages and will translate them to CAS commands (signals).

Figure 5-5: CAS to Analog Mapping Protocol



In this configuration, the following line to trunk/b-channel mapping is described in the table below.

Table 5-39: Trunk/B-channel Mapping

Trunking Channels		Analog Channels
Trunk 1	b-channel 1	line 1
Trunk 1	b-channel 2	line 2
...	...	...
Trunk 1	b-channel 15	line 15
Trunk 1	b-channel 16	- (see note below)
Trunk 1	b-channel 17	line 16
Trunk 1	b-channel 18	line 17

Table 5-39: Trunk/B-channel Mapping

Trunking Channels		Analog Channels
...	...	...
Trunk 2	b-channel 1	line 31
Trunk 2	b-channel 2	Line 32
…	…	…
Trunk 3	b-channel 1	Line 61
…	…	…

Note: B-Channel 16 in each trunk is skipped as it used for CAS signaling.

Table 5-40: Mapping Table

CAS	Analog	MEGACO
Incoming CAS -->	**Analog relevant detection -->**	**MEGACO relevant event**
Clear Back/Forward	On Hook	al/on event
Release	On Hook	al/on event
Answer/Re-Answer	Off Hook	al/of event
Seize	Off Hook	al/of event
Flash On/Off	Flash Hook	al/fl event
Outgoing CAS	**<- Analog relevant generation**	**<-- MEGACO relevant signals/command**
Block	-	Option 1: Modify physical termination state to InService command Option 2:

Table 5-40: Mapping Table

CAS	Analog	MEGACO
		Service Change on physical termination with Force method command
Unblock	-	Option 1: Modify physical termination state to OutOfService command Option 2: Service Change on physical termination with Restart method command
Answer	Line side answer	xal/las signal
Clear (Back)	Network disconnect	xal/nd signal
Release process	-	subtract physical termination command
Ring On/Off	Ringing	alert/ri signal

5.3.13 *ini* File Configuration

➢ **To enable 'CAS to Analog' mapping, take these 3 steps:**

1. Configure the device as a CAS gateway with the relevant PSTN configuration and CAS table.
2. Configure the MEGACO relevant parameters.
3. Set the TrunkingToAnalogFunctionalityProfile to 1.

5.3.14 Pulse Dial Detection

Pulse Dial detection is the detection of pulse dialed digits via the ABCD CAS bits.

This feature provides pulse dial signaling detection to be able to work with pulse dial telephone endpoints.

Note 1: Pulse dial detection does not prevent the detection of digits that were dialed via any other method (e.g., DTMF).

Note 2: There is NO generation of dial pulses, only detection.

To enable the pulse dial detection, the user must set the TrunkingToAnalogFunctionalityProfile INI file the parameter to 1.

5.3.15 CAS Table Configuration

To configure a CAS table to work with pulse dial detection, refer to the CAS Protocol Table section, in the User Manual .

In order to work with pulse dial detection, some parameters must be exist and initiated in the UserProt_defines_xxx.h file, as described in the CAS Protocol Table.

The parameters are:

ABCD_DURING_BREAK - the ABCD CAS bits pattern during the break interval.

ABCD_DURING_MAKE - the ABCD CAS bits pattern during the make interval.

PULSE_DIAL_BREAK_MINIMUM_TIME – the minimum break interval time.

PULSE_DIAL_BREAK_MAXIMUM_TIME – the maximum break interval time.

PULSE_DIAL_MAKE_MAXIMUM_TIME – the minimum make Interval time.

PULSE_DIAL_INTER_DIGIT_MINIMUM_TIME - The minimum time between 2 dial pulse digits.

PULSE_DIAL_INTER_DIGIT_MAXIMUM_TIME - The maximum time between 2 dial pulse digits in the same dialed number.

When a pulse dial digit is detected, a timer is set for this time. If another digit is detected before the timer expires, the new digit is considered part of the same dialed. When this timer expires, the last digit that was detected will be considered as the last digit of the dialed number and an appropriate event will be sent to the host.

Note: When enabling pulse dial detection (by setting the TrunkingTo AnalogFunctionalityProfile INI file the parameter to 1) these parameters MUST exist and be initiated in the CAS table, otherwise the trunk will not be configured.

These parameters are already configured in the MelCAS CAS table.

5.4 MEGACO Compliance

The MEGACO Compliance Matrix table below summarizes the supported MEGACO features. The Reference column in the table refers to IETF RFC 3015 from September 2002.

Note: MEGACO is not currently supported in this version for **260** devices. Contact your local sales representative for further details.

Table 5-41: MEGACO Compliance Matrix

Reference (in RFC 3015)	Item	Support	Comments
7	**Commands supported:**		
	Add	Yes	
	Modify	Yes	

Table 5-41: MEGACO Compliance Matrix

Reference (in RFC 3015)	Item	Support	Comments
	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	
	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	

Table 5-41: MEGACO Compliance Matrix

Reference (in RFC 3015)	Item	Support	Comments
	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	Yes	This is treated as if the default is 'yes'
	ReserveValue	Yes	This is treated as if the default is ‘yes’
7.1.8	**Local & Remote Descriptors:**		
	Unspecified Local Descriptor	Yes	
	Unspecified Remote Descriptor	Yes	
	Empty Local Descriptor	Yes	
	Empty Remote Descriptor	Yes	
	Multiple groups	Yes	
7.1.9	**Event Descriptor**		
	EventBufferControl		

Table 5-41: MEGACO Compliance Matrix

Reference (in RFC 3015)	Item	Support	Comments
	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	Yes	Up to 2, according to the Defines Table
	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	

Table 5-41: MEGACO Compliance Matrix

Reference (in RFC 3015)	Item	Support	Comments
	ServiceChangeVersion	Yes	
	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**		
	Isolate	Yes	
	Oneway	Yes	
	Bothway	Yes	
	CHOOSE wildcard	Yes	
	ALL wildcard	Yes	
7.2	**Command API**		
7.2.1	**Add**		

Table 5-41: MEGACO Compliance Matrix

Reference (in RFC 3015)	Item	Support	Comments
	Termination ID	Yes	
	MediaDescriptor	Yes	
	ModemDescriptor	No	
	MuxDescriptor	No	
	EventsDescriptor	Yes	
	SignalsDescriptor	Yes	Up to 2 signals 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	Up to 2 signals 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	

Table 5-41: MEGACO Compliance Matrix

Reference (in RFC 3015)	Item	Support	Comments
	SignalsDescriptor	Yes	Up to 2 signals 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	
	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	

Table 5-41: MEGACO Compliance Matrix

Reference (in RFC 3015)	Item	Support	Comments
	Statistics	Yes	
	Packages	Yes	
7.2.7	**Notify**		
		Yes	
7.2.8	**Service Change**		
	Termination ID	Yes	
	Wildcard	Yes	
	“Root” Termination	Yes	
	ServiceChangeMethod		
	Graceful	Yes	This method can be used only for the whole gateway (ROOT termination), and only when sent from the gateway to the MGC
	Forced	Yes	
	Restart	Yes	
	Disconnected	Yes	
	Handoff	Yes	
	Failover	Yes	
	Extension	No	
	ServiceChangeReason		
	900 Service Restored	Yes	
	901 Cold Boot	Yes	
	902 Warm Boot	No	
	903 MGC Direct Change	Yes	
	904 Termination Malfunctioning	No	
	905 Term Taken out of Service	Yes	
	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	

Table 5-41: MEGACO Compliance Matrix

Reference (in RFC 3015)	Item	Support	Comments
	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	Yes	
	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	Only for H.248v1
7.3	**Command Error**		
	400 - Bad Request	Yes	
	401 - Protocol Error	Yes	
	402 - Unauthorized	Yes	
	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	

Table 5-41: MEGACO Compliance Matrix

Reference (in RFC 3015)	Item	Support	Comments
	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	
	501 - Not Implemented	Yes	

Table 5-41: MEGACO Compliance Matrix

Reference (in RFC 3015)	Item	Support	Comments
	502 - Not ready	Yes	
	503 - Service Unavailable	Yes	
	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	Yes	
	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**		

Table 5-41: MEGACO Compliance Matrix

Reference (in RFC 3015)	Item	Support	Comments
	ContextID	Yes	
	CHOOSE Wildcard	Yes	
	All Wildcard	Yes	
8.2	**Transaction API**		
8.2.1	**Transaction Request**		
	Multiple actions per request	Yes	
8.2.2	**Transaction Reply**		
	Multiple actions per reply	Yes	
8.2.3	**Transaction Pending**		
	Transaction Pending Support	Yes	
	normalMGEcecutionTime	Yes	
	normalMGCEcecutionTime	Yes	
8.3	**Messages**		
	Receive Messages	Yes	
	Send Messages	Yes	
9	**Transport**		
	Transport over UDP	Yes	
	Transport over TCP	Yes	
	Transport over SCTP	Yes	
9.1	**Ordering of Commands**		
		Yes	
9.2	**Protection Against the Restart Avalanche**		
	Use of default MWD per platform	Yes	
	Random restart delay	Yes	
	Random seed selection	Yes	
	Detection of local activity	No	
10	**Security Considerations**		
		No	
11	**MG-MGC Control Interface**		
11.1	**Multiple Virtual Gateways**		

Table 5-41: MEGACO Compliance Matrix

Reference (in RFC 3015)	Item	Support	Comments
		No	Not supported, however, receiving commands from more than one MGC and sending notification to the MGC which requested it is supported.
11.2	**Cold Start**		
	Primary Call Agent support	Yes	
	Secondary Call Agents support	Yes	
	Cyclic check for Call Agent	Yes	
11.3	**Negotiation of Protocol Version**		
		Yes	
11.4	**Failure of an MG**		
		No	
11.5	**Failure of an MGC**		
		Yes	

Reader's Notes

6 SS7 Functionality & Configuration

Note: This SS7 Configuration Guide is not applicable to **MediaPack and Mediant 1000**.

Several SS7 network elements are available. This section provides a brief description of each network element, and corresponding configuration description.

Part of the various network elements described below includes the use of SigTran (M2UA, M3UA), as implemented in devices such as device.

The device can implement Signaling gateways for IUA, DUA, M2UA and M3UA layers. This Signaling gateway node can be connected to two MGC nodes using the Override mode. For further information please refer to
RFC 3057, RFC 3331 or RFC 3332.

Note: please refer to the description of SS7 configuration parameters in Table of Parameter Value Constructs on page 32 and in Table Parameters on page 218.

6.1 SS7 Network Elements

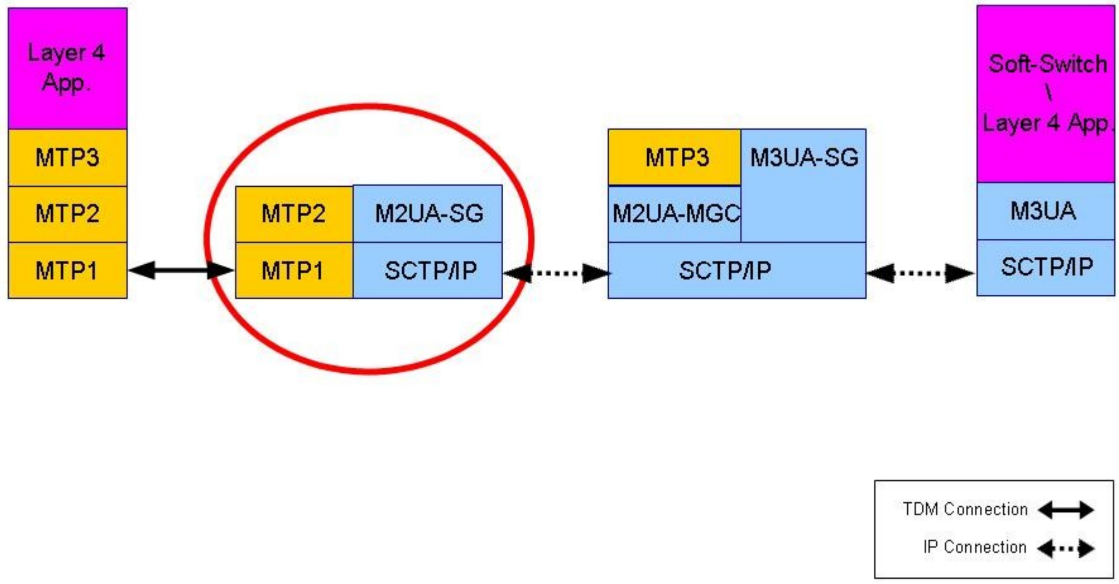
The SS7 network elements include these basic configurations:

- SS7 M2UA - SG Side
- SS7 M2UA – Media Gateway Controller Side
- SS7 MTP3 Node
- SS7 MTP2 Tunneling
- SS7 SN Redundancy - MTP3 Shared Point Code

6.1.1 SS7 M2UA - SG Side

For the SS7 M2UA - SG side network element, the MTP2 link from the SS7 network side is sent via SCTP (IP) to Media-Gateway side.

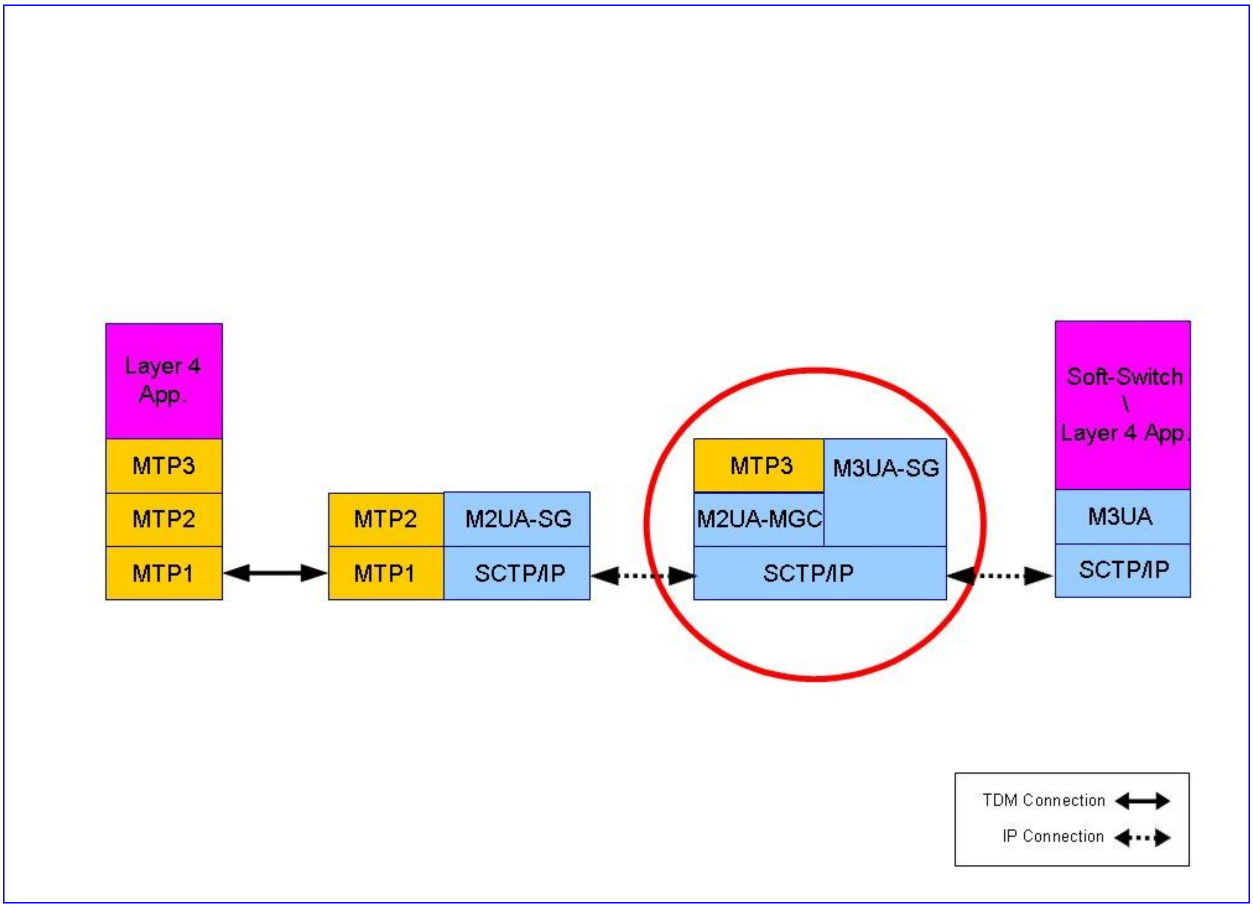
Figure 6-1: SS7 M2UA - SG Side



6.1.2 SS7 M2UA – Media Gateway Controller Side

For the SS7 M2UA – Media Gateway Controller side network element, the M2UA Media Gateway Controller link is from the IP side. MTP3 is supported in the device's software. The MTP3 payload is sent via M3UA to the Soft-Switch. (MTP3 can also route MSUs to other SS7 network elements via other links).

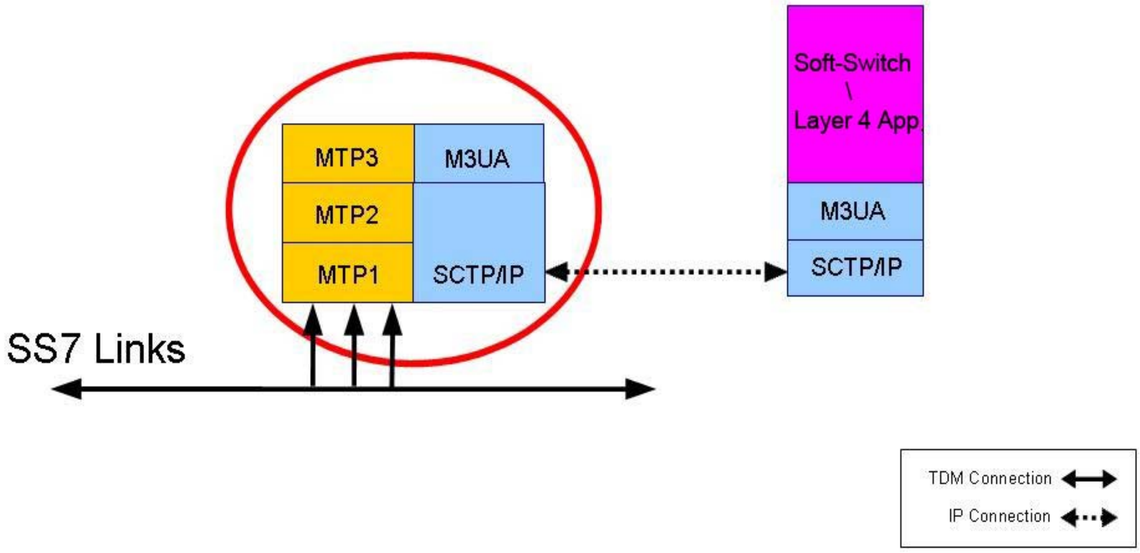
Figure 6-2: SS7 M2UA - MGC Side



6.1.3 SS7 MTP3 Node

The SS7 MTP3 Node is a classic MTP3 over MTP2 configuration. Links are incoming from the SS7 Network. MTP3 payload is sent via M3UA to the Soft-Switch or routed to other SS7 network elements according to the MSU headers.

Figure 6-3: SS7 MTP3 Node

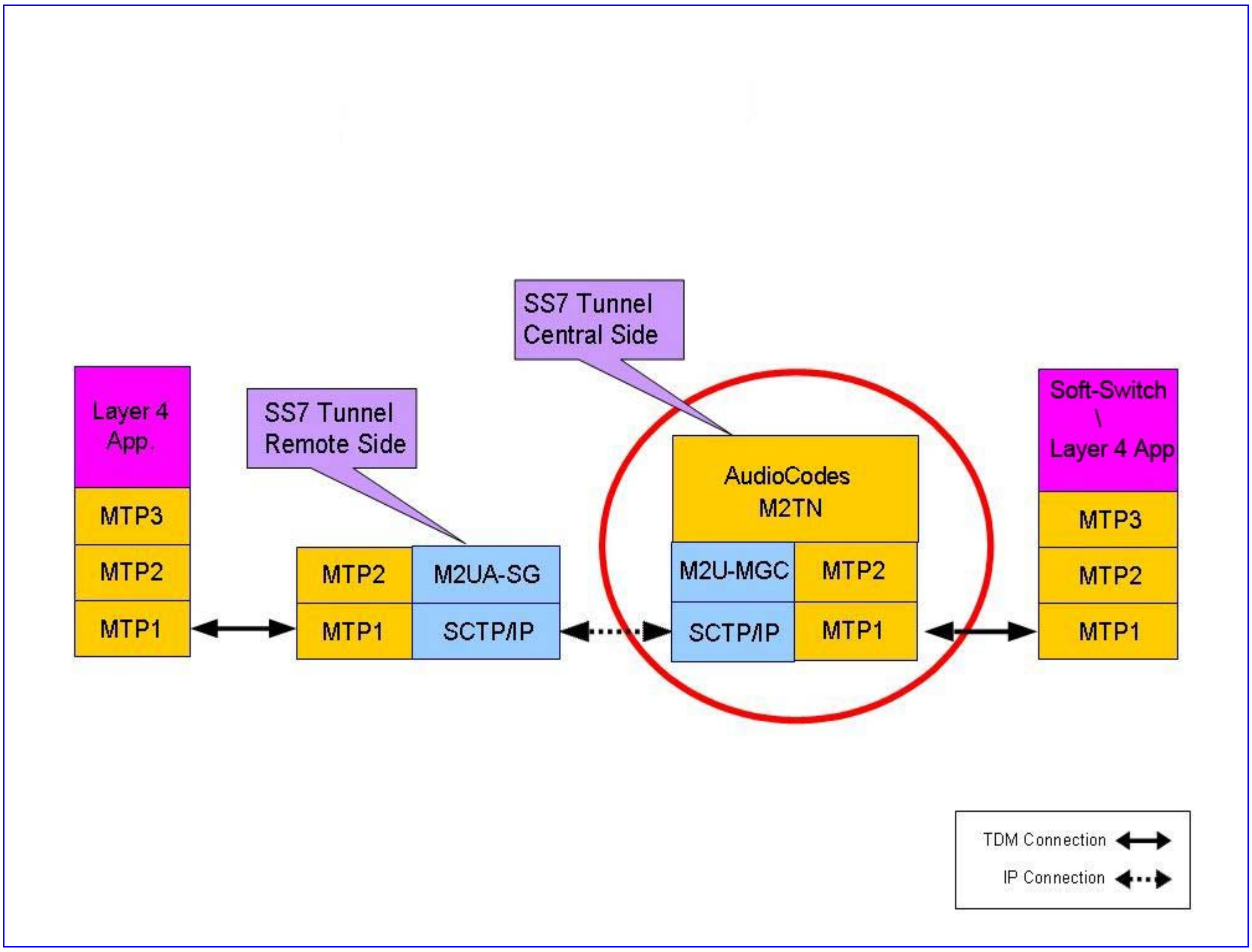


6.1.4 SS7 MTP2 Tunneling

For the SS7 MTP2 Tunneling configuration, the MTP2 SS7 link payload is sent across long distances (over the IP network). Both of its termination ends have SS7 MTP2 interfaces, which are unaware of the MTP2 Tunneling between them.

MTP2 Tunneling is a proprietary solution, based on SS7 and SigTran standards.

Figure 6-4: SS7 MTP2 Tunneling



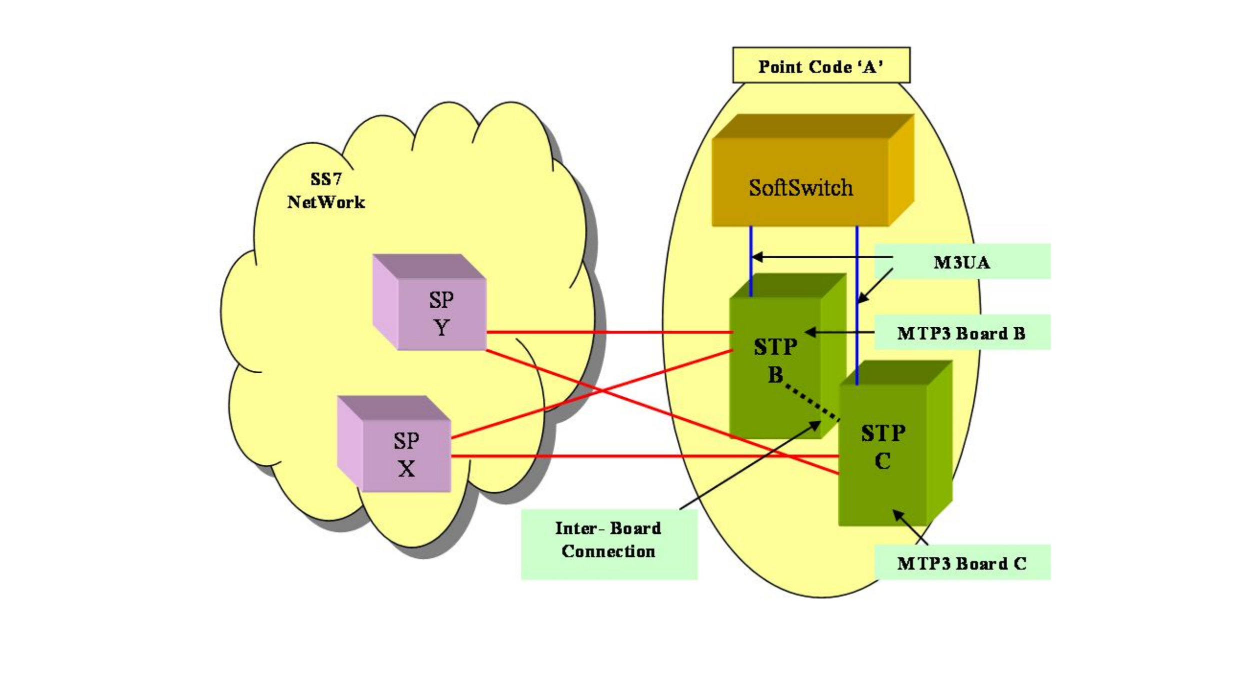
6.1.5 SS7 SN Redundancy - MTP3 Shared Point Code

In the MTP3 Shared Point Code configuration point code **A** is implemented by two devices (i.e., CPUs): **B** and **C**.

In the figure below, there are 2 link-sets from point-code **B**: one to point-code **Y**, and the other to point-code **X**. The same link-sets are also from point-code **C**.

=> Both link-sets are distributed across the two devices.

Figure 6-5: MTP3 Shared Point Code Configuration Diagram



6.1.6 Configuration Extensions:

In addition to the basic SS7 configurations described above, the extensions below provide more options:

1. 2 SS7 Nodes (SP/STP) can be configured per TPM
2. Device supports mixed SS7 link types, i.e. one device can have few MTP2 links and a number of M2UA links
3. Supported SS7 variants: ITU-T, ANSI, CHINA
4. SS7 signaling links can be configured on any available timeslot of any trunk, so that several SS7 signaling links can be configured on one E1/T1
5. F-links are supported: any mixture of voice and signaling links can be configured on any trunk (providing that the trunk type supports SS7 signaling links - refer to the examples provided below).

6.1.7 Other Dependencies in ini File:

Trunk Protocol Types

Trunks that carry SS7 Link(s) must be configured with protocol type: T1_TRANSPARENT=4, E1_TRANSPARENT31=5 or E1_TRANSPARENT30=6

Note : The Trunks Protocol definition must appear in the INI file before the SS7/SIGTRAN Table definition.

6.2 Examples of SS7 ini Files

This section provides examples of ***ini*** files for each of the SS7 network elements described previously. Each example can be modified to fit the user's field configuration is accompanied by loading instructions for a testing/Lab mini-network environment.

6.2.1 SS7 M2UA - SG Side ini File Example

For the SS7 M2UA - SG Side ***ini*** file example, take into account the following notes:

- There are 4 SS7 links of type: MTP2->M2UA SG.
- There is 1 interface group (only 1 remote Media Gateway Controller).
- There are 4 interface IDs defined: 1 per link.
- This file is intended for ITU link variant (E1 trunks).

➢ **To load the SS7 M2UA - SG Side *ini* file example, take these 2 steps:**

1. This ***ini*** file is a configuration of an MTP2 SG device. An MTP2 Media Gateway Controller device should be connected (using SCTP over IP) to the MTP2 SG device.
2. Change the value of the SyslogServerIP parameter to your own Syslog Server IP.

The following is an example of SS7 M2UA - SG Side ***ini*** File

```
[TDM BUS configuration]

; 1=aLaw  3=ulaw
PCMLawSelect=    1

; EXT BUS=5   H110=4   QSLAC=3   FRAMERS=2   SC BUS=1   MVIP BUS=0
TDMBusType=        2

; 0=2048kbps, 2=4096kbps, 3=8192kbps
TDMBusSpeed=   3

;1 - internal, 3 - mvip, 4 - Network, 8 - h110a, 9 - h110b, 10 -
Netref
TDMBUSCLOCKSOURCE= 1

[Trunk Configuration]

;e1 euro isdn=1 t1 isdn=2 ;e1 cas r2=8 (8 for fcd); e1 trans 31=5
ProtocolType = 5

TraceLevel   = 0

; acCLOCK MASTER ON =1
CLOCKMASTER= 1

;acUSER TERMINATION SIDE = 0
TerminationSide = 1

;acEXTENDED SUPER FRAME=0
FramingMethod = 0

;acB8ZS = 0   2 for E1 CAS - FCD
LineCode = 0

; 0=Internal Clock, 1=rx signal derived clk
PHYCLKSOURCE=0
```

```
[syslog]
SYSLOGSERVERIP = 168.100.0.1
ENABLESYSLOG = 1

WATCHDOGSTATUS = 0

[ SS7 LINK TABLE ]
FORMAT SS7 LINK INDEX = SS7 LINK NAME, SS7 LINK TRACE LEVEL,
SS7 LINK ADMINISTRATIVE STATE,SS7 LINK L2 TYPE, SS7 LINK L3 TYPE,
SS7 LINK TRUNK NUMBER,SS7 LINK TIMESLOT NUMBER,SS7 LINK M2UA IF ID,
SS 7 LINK GROUP ID;

SS7 LINK TABLE 0 = new link 0, 0, 2, 1, 1, 1, 15, 50, 4;
SS7 LINK TABLE 1 = new link 1, 0, 2, 1, 1, 2, 12, 12, 4;
SS7 LINK TABLE 2 = new link 2, 0, 2, 1, 1, 4,  7, 18, 4;
SS7 LINK TABLE 3 = new link 3, 0, 2, 1, 1, 5,  3,  1, 4;

[\SS7 LINK TABLE]


[ SS7 SIG IF GROUP TABLE ]
FORMAT SS7 SIG IF GR INDEX = SS7 IF GR ID,SS7 SIG SG MGC,
SS7_SIG_LAYER, SS7_SIG_TRAF_MODE, SS7_SIG_T_REC, SS7_SIG_T_ACK,
SS7 SIG T HB, SS7 SIG MIN ASP, SS7 SIG BEHAVIOUR,
SS7 LOCAL SCTP PORT, SS7 SIG NETWORK;
SS7_SIG_IF_GROUP_TABLE 4 =  4,83, 2, 1, 2000, 2000, 30000, 1, 0,
2904, 1;
[ \SS7 SIG IF GROUP TABLE ]

 [ SS7 SIG INT ID TABLE ]
FORMAT SS7 SIG IF ID INDEX = SS7 SIG IF ID VALUE,
SS7_SIG_IF_ID_NAME, SS7_SIG_IF_ID_OWNER_GROUP, SS7_SIG_IF_ID_LAYER,
SS7 SIG IF ID NAI, SS7 SIG M3UA SPC;
SS7 SIG INT ID TABLE 7 = 50, BELFAST12, 4, 2, 0, 0;
SS7_SIG_INT_ID_TABLE 8 = 12, AMSTERDAM, 4, 2, 1, 0;
SS7 SIG INT ID TABLE 9 = 18, ROTERDAM , 4, 2, 2, 0;
SS7 SIG INT ID TABLE 10 = 1, GAUDA    , 4, 2, 3, 0;
[ \SS7_SIG_INT_ID_TABLE ]
```

6.2.2 SS7 M2UA - Media Gateway Controller Side ini File Example

For the SS7 M2UA - Media Gateway Controller Side ***ini*** file example, take into account the following notes:

- This ***ini*** file is a configuration of the M2UA Media Gateway Controller (toward the remote MTP2 side) and M3UA SG (toward the layer 4 application, e.g., Soft-Switch).
- There are 4 SS7 links of type: MTP2 Media Gateway Controller->MTP3.
- There is 1 SN (Signaling Node). Modify its point-code according to your own network point code.
- There is 1 LinkSet with 4 links.
- There is 1 RouteSet.
- The DPC of the RouteSet and LinkSet is the point-code of the remote end (to which the MTP2 link on the MTP2 SG side is connected). Modify it on both LinkSet and RouteSet tables.
- There are 2 interface groups: 1 interface group is used for the M2UA SG <=> M2UA Media Gateway Controller connection, and the other one is used for the M3UA SG <=> M3UA Media Gateway Controller connection.

- There are 4 interface IDs defined: 1 per link (M2UA Media Gateway Controller side) and one more interface Id for M3UA SG. The connection between the interface ID and the Interface group is determined by the SS7_SIG_IF_ID_OWNER_GROUP parameter.
- This file is intended for ITU link variant (E1 trunks).

➢ To load the SS7 M2UA - Media Gateway Controller Side *ini* file example, take these 4 steps:

1. Load this ***ini*** file on an MTP2 Media Gateway Controller device. An MTP2 SG device should be connected (over IP) to the MTP2Media Gateway Controller device.
2. Change the value of the **SyslogServerIP** parameter to your own Syslog Server IP.
3. Change the **SS7_DEST_IP** parameter according to the actual IP address of the M2UA SG device.
4. Change the **SS7_SIG_M3UA_SPC** parameter of line 0 (M3UA related interface ID line) according to the local SN point code.

The following is an example of SS7 M2UA - Media Gateway Controller Side ***ini*** File

```
[TDM BUS configuration]

; 1=aLaw  3=ulaw
PCMLawSelect=    1

; EXT BUS=5   H110=4   QSLAC=3   FRAMERS=2   SC BUS=1   MVIP BUS=0
TDMBusType=        2

; 0=2048kbps, 2=4096kbps, 3=8192kbps
TDMBusSpeed=   3

;1 - internal, 3 - mvip, 4 - Network, 8 - h110a, 9 - h110b, 10 -
Netref
TDMBUSCLOCKSOURCE= 1

[Trunk Configuration]

;e1 euro isdn=1 t1 isdn=2 ;e1 cas r2=8 (8 for fcd); e1 trans 62=5
ProtocolType = 5

; acCLOCK MASTER ON =1
CLOCKMASTER= 1

;acUSER TERMINATION SIDE = 0
TerminationSide = 1

;acEXTENDED SUPER FRAME=0
FramingMethod = 0

;acB8ZS = 0   2 for E1 CAS - FCD
LineCode = 0

[SS7]

SS7 MTP2 PARAM TIMER T7 0=2000

; 0=Internal Clock, 1=rx signal derived clk
PHYCLKSOURCE=0

[syslog]
SYSLOGSERVERIP = 168.100.0.1
ENABLESYSLOG = 1
```

```
; ****************************************
; SS7 TIMERS - ITU
; ****************************************

[SS7 SN TIMERS TABLE]
FORMAT SS7 SNTIMERS INDEX = SS7 SNTIMERS NAME, SS7 SNTIMERS T6,
SS7 SNTIMERS T8, SS7 SNTIMERS T10, SS7 SNTIMERS T11,
SS7 SNTIMERS T15, SS7 SNTIMERS T16, SS7 SNTIMERS T18 ITU,
SS7 SNTIMERS T19 ITU, SS7 SNTIMERS T20 ITU, SS7 SNTIMERS T21 ITU,
SS7 SNTIMERS T24 ITU;
SS7 SN TIMERS TABLE 0 = TENERIFF 0, 800, 1000, 30000, 30000, 2000,
1400, 5000, 4000, 15000, 10000, 500;
[\SS7 SN TIMERS TABLE]

[SS7 LINKSET TIMERS TABLE]
FORMAT SS7 LKSETTIMERS INDEX = SS7 LKSETTIMERS NAME,
SS7 LKSETTIMERS T1SLT, SS7 LKSETTIMERS T2SLT, SS7 LKSETTIMERS T1,
SS7_LKSETTIMERS_T2, SS7_LKSETTIMERS_T3, SS7_LKSETTIMERS_T4,
SS7 LKSETTIMERS T5, SS7 LKSETTIMERS T7, SS7 LKSETTIMERS T12,
SS7 LKSETTIMERS T13, SS7 LKSETTIMERS T14, SS7 LKSETTIMERS T17,
SS7_LKSETTIMERS_T22_ITU, SS7_LKSETTIMERS_T23_ITU;
SS7 LINKSET TIMERS TABLE 0 = DELHI 0, 8000, 30000, 800, 1400, 800,
800, 800, 1000, 1500, 800, 2000, 1500, 180000, 180000;
[\SS7_LINKSET_TIMERS_TABLE]

; ****************************************
; SS7 TIMERS - ANSI
; ****************************************

 [SS7_SN_TIMERS_TABLE]
FORMAT SS7 SNTIMERS INDEX = SS7 SNTIMERS NAME, SS7 SNTIMERS T6,
SS7 SNTIMERS T8, SS7 SNTIMERS T10, SS7 SNTIMERS T11,
SS7_SNTIMERS_T15, SS7_SNTIMERS_T16, SS7_SNTIMERS_T22_ANSI,
SS7 SNTIMERS T23 ANSI, SS7 SNTIMERS T24 ANSI,
SS7 SNTIMERS T25 ANSI, SS7 SNTIMERS T26 ANSI,
SS7_SNTIMERS_T28_ANSI, SS7_SNTIMERS_T29_ANSI,
SS7 SNTIMERS T30 ANSI;
SS7 SN TIMERS TABLE 1 = BABILON 0, 800, 1000, 30000, 30000, 2000,
1400, 180000, 180000, 5000, 30000, 12000, 3000, 60000, 30000;
[\SS7 SN TIMERS TABLE]

[SS7_LINKSET_TIMERS_TABLE]
FORMAT SS7 LKSETTIMERS INDEX = SS7 LKSETTIMERS NAME,
SS7 LKSETTIMERS T1SLT, SS7 LKSETTIMERS T2SLT, SS7 LKSETTIMERS T1,
SS7_LKSETTIMERS_T2, SS7_LKSETTIMERS_T3, SS7_LKSETTIMERS_T4,
SS7 LKSETTIMERS T5, SS7 LKSETTIMERS T12, SS7 LKSETTIMERS T13,
SS7 LKSETTIMERS T14, SS7 LKSETTIMERS T17, SS7 LKSETTIMERS T20 ANSI,
SS7_LKSETTIMERS_T21_ANSI;
SS7 LINKSET TIMERS TABLE 1 = HANOI 0, 8000, 30000, 800, 1400, 800,
800, 800, 1000, 1500, 2000, 1500, 90000, 90000;
[\SS7_LINKSET_TIMERS_TABLE]

[SS7 SN TABLE]
FORMAT SS7_SN_INDEX = SS7_SN_NAME, SS7_SN_TRACE_LEVEL,
SS7 SN ADMINISTRATIVE STATE, SS7 SN VARIANT, SS7 SN NI,
SS7 SN SP STP, SS7 SN OPC, SS7 SN ROUTESET CONGESTION WINSIZE,
SS7_SN_TIMERS_INDEX, SS7_SN_ISUP_APP, SS7_SN_SCCP_APP,
SS7 SN BISUP APP, SS7 SN ALCAP APP;
SS7 SN TABLE 0 = SN 0, 0, 2, 1, 0, 0, 11, 8, 0, 4, 4, 4, 4;
[\SS7 SN TABLE]

 [ SS7 LINK TABLE ]
FORMAT SS7 LINK INDEX = SS7 LINK NAME, SS7 LINK TRACE LEVEL,
SS7 LINK ADMINISTRATIVE STATE,SS7 LINK L2 TYPE,
SS7 LINK L3 TYPE,SS7 LINK GROUP ID, SS7 LINK M2UA IF ID;
SS7 LINK TABLE 0 = link 0 SP A, 0, 2, 2, 2, 4, 50;
SS7_LINK_TABLE 1 = link_1_SP_B, 0, 2, 2, 2, 4, 12;
```

```
SS7 LINK TABLE 2 = link 2 SP C, 0, 2, 2, 2, 4, 18;
SS7 LINK TABLE 3 = link 3 SP D, 0, 2, 2, 2, 4, 1;
[\SS7 LINK TABLE]

[ SS7 LINKSET TABLE ]
FORMAT SS7 LINKSET SN INDEX, SS7 LINKSET LINKSET INDEX =
SS7 LINKSET NAME, SS7 LINKSET ADMINISTRATIVE STATE,
SS7 LINKSET DPC, SS7 LINKSET TIMERS INDEX;
SS7 LINKSET TABLE 0, 0 = lkset0 sp A, 2, 10, 0;
[ \SS7 LINKSET TABLE ]

[ SS7 LINKSETLINK TABLE ]
FORMAT SS7 LINKSETLINK SN INDEX, SS7 LINKSETLINK LINKSET INDEX,
SS7 LINKSETLINK INNER LINK INDEX = SS7 LINKSETLINK LINK NUMBER,
SS7 LINKSETLINK LINK SLC;
SS7 LINKSETLINK TABLE 0, 0, 0 = 0, 0;
SS7 LINKSETLINK TABLE 0, 0, 1 = 1, 1;
SS7 LINKSETLINK TABLE 0, 0, 2 = 2, 2;
SS7 LINKSETLINK TABLE 0, 0, 3 = 3, 3;
[ \SS7 LINKSETLINK TABLE ]


[ SS7 ROUTESET TABLE ]
FORMAT SS7 ROUTESET SN INDEX, SS7 ROUTESET INDEX =
SS7 ROUTESET NAME, SS7 ROUTESET ADMINISTRATIVE STATE,
SS7 ROUTESET DPC;
SS7 ROUTESET TABLE 0, 0 = RTESET0 SP A, 2, 10;
[ \SS7 ROUTESET TABLE ]


[ SS7 ROUTESETROUTE TABLE ]
FORMAT SS7 ROUTESETROUTE SN INDEX,
SS7 ROUTESETROUTE ROUTESET INDEX,
SS7 ROUTESETROUTE INNER ROUTE INDEX =
SS7 ROUTESETROUTE LINKSET NUMBER, SS7 ROUTESETROUTE PRIORITY;
SS7 ROUTESETROUTE TABLE 0, 0, 0 = 0, 0;
[ \SS7 ROUTESETROUTE TABLE ]

[ SS7 ROUTING CONTEXT TABLE ]
FORMAT SS7 RC INDEX, SS7 RC INNER INDEX = SS7 RC SN INDEX,
SS7 RC OPC1, SS7 RC OPC2, SS7 RC OPC3, SS7 RC OPC4, SS7 RC SI1,
SS7 RC SI2, SS7 RC SI3, SS7 RC SI4;
SS7 ROUTING CONTEXT TABLE 0, 0 = 0, -1, -1, -1, -1, -1, -1, -1, -1;
[ \SS7 ROUTING CONTEXT TABLE ]

[ SS7 SIG IF GROUP TABLE ]
FORMAT SS7 SIG IF GR INDEX = SS7 IF GR ID,SS7 SIG SG MGC,
SS7 SIG LAYER, SS7 SIG TRAF MODE, SS7 SIG T REC, SS7 SIG T ACK,
SS7 SIG T HB, SS7 SIG MIN ASP, SS7 SIG BEHAVIOUR,
SS7 LOCAL SCTP PORT, SS7 SIG NETWORK, SS7 SIG RC INDEX,
SS7 SIG RC VALUE, SS7 SIG NETWORK APPEARANCE;
;
; M3UA SG SIDE DEFINITION:
;
SS7 SIG IF GROUP TABLE 2 =  2, 83, 3, 1, 2000, 2000, 30000, 1, 0,
2905, 1, 0, 1, 1;
[ \SS7 SIG IF GROUP TABLE ]

[ SS7 SIG IF GROUP TABLE ]
FORMAT SS7 SIG IF GR INDEX = SS7 IF GR ID,SS7 SIG SG MGC,
SS7 SIG LAYER, SS7 SIG TRAF MODE, SS7 SIG T REC, SS7 SIG T ACK,
SS7 SIG T HB, SS7 SIG MIN ASP, SS7 SIG BEHAVIOUR,
SS7 LOCAL SCTP PORT, SS7 SIG NETWORK, SS7 DEST SCTP PORT,
SS7 DEST IP, SS7 MGC MX IN STREAM, SS7 MGC NUM OUT STREAM;
;
; M2UA MGC SIDE DEFINITION:
;
```

```
SS7 SIG IF GROUP TABLE 4 =  4, 77, 2, 1, 2000, 2000, 30000, 1, 0,
2904, 1,2904,168.100.0.2,3,3;
[ \SS7 SIG IF GROUP TABLE ]


[ SS7 SIG INT ID TABLE ]
FORMAT SS7 SIG IF ID INDEX = SS7 SIG IF ID VALUE,
SS7 SIG IF ID NAME, SS7 SIG IF ID OWNER GROUP, SS7 SIG IF ID LAYER,
SS7 SIG IF ID NAI, SS7 SIG M3UA SPC;
SS7 SIG INT ID TABLE 0 = 100, BELFAST12, 2, 3, 0, 11;
SS7 SIG INT ID TABLE 1 = 50, AMSTERDAM1, 4, 2, 0,  0;
SS7 SIG INT ID TABLE 2 = 12, GAUDA,      4, 2, 1,  0;
SS7 SIG INT ID TABLE 3 = 18, PRAGUE,     4, 2, 2,  0;
SS7 SIG INT ID TABLE 4 = 1, PARIS ,      4, 2, 3,  0;
[ \SS7_SIG_INT_ID_TABLE ]
```

6.2.3 SS7 MTP3 Node ini File Example

For the SS7 MTP3 Node *ini* file example, take into account the following notes:

- This ini file defines 2 MTP3 SNs (signaling nodes). These nodes are connected to each other in an external loop, using E1 trunks.
- There are 4 SS7 links of type: MTP2->MTP3.
- There is 1 LinkSet per SN with 2 links.
- There is 1 RouteSet per SN with 1 LinkSet.
- There is 1 Interface group - Both SNs are using it.
- There are 2 Interface IDs defined: 1 per SN.
- This file is intended for ITU link variant (E1 trunks).

To load the SS7 MTP3 Node *ini* file example, take these 2 steps:

1. Load this *ini* file on an MTP3+M3UA blade. A layer 4 application (such as Soft-Switch) should be connected (over IP) to the blade.
2. Change the value of the SyslogServerIP parameter to your own Syslog Server IP.

The following is an example of SS7 MTP3 Node *ini* File.

```
[TDM BUS configuration]

; 1=aLaw  3=ulaw
PCMLawSelect=    1

; EXT_BUS=5   H110=4   QSLAC=3   FRAMERS=2   SC_BUS=1   MVIP_BUS=0
TDMBusType=        2

; 0=2048kbps, 2=4096kbps, 3=8192kbps
TDMBusSpeed=   3

;1 - internal, 3 - mvip, 4 - Network, 8 - h110a, 9 - h110b, 10 -
Netref
TDMBUSCLOCKSOURCE= 1

[Bit Configuration]
[Trunk Configuration]

;e1 euro isdn=1 t1 isdn=2 ;e1 cas r2=8 (8 for fcd); e1 trans 62=5
ProtocolType = 5

TraceLevel   = 0
```

```
; acCLOCK MASTER ON =1
CLOCKMASTER= 1

;acUSER TERMINATION SIDE = 0
TerminationSide = 1

;acEXTENDED SUPER FRAME=0
FramingMethod = 0

;acB8ZS = 0   2 for E1 CAS - FCD
LineCode = 0

[megaco conference support]

MGControlprotocoltype =2

; 0=Internal Clock, 1=rx signal derived clk
PHYCLKSOURCE=0

[syslog]
SYSLOGSERVERIP = 168.100.0.1
ENABLESYSLOG = 1

; ****************************************
; SS7 TIMERS - ITU
; ****************************************

[SS7 SN TIMERS TABLE]

FORMAT SS7 SNTIMERS INDEX = SS7 SNTIMERS NAME, SS7 SNTIMERS T6,
SS7 SNTIMERS T8, SS7 SNTIMERS T10, SS7 SNTIMERS T11,
SS7 SNTIMERS T15, SS7 SNTIMERS T16, SS7 SNTIMERS T18 ITU,
SS7 SNTIMERS T19 ITU, SS7 SNTIMERS T20 ITU, SS7 SNTIMERS T21 ITU,
SS7 SNTIMERS T24 ITU;

SS7 SN TIMERS TABLE 0 = TENERIFF 0, 800, 1000, 30000, 30000, 2000,
1400, 5000, 4000, 15000, 10000, 500;

[\SS7 SN TIMERS TABLE]

[SS7 LINKSET TIMERS TABLE]

FORMAT SS7 LKSETTIMERS INDEX = SS7 LKSETTIMERS NAME,
SS7 LKSETTIMERS T1SLT, SS7 LKSETTIMERS T2SLT, SS7 LKSETTIMERS T1,
SS7 LKSETTIMERS T2, SS7 LKSETTIMERS T3, SS7 LKSETTIMERS T4,
SS7 LKSETTIMERS T5, SS7 LKSETTIMERS T7, SS7 LKSETTIMERS T12,
SS7 LKSETTIMERS T13, SS7 LKSETTIMERS T14, SS7 LKSETTIMERS T17,
SS7 LKSETTIMERS T22 ITU, SS7 LKSETTIMERS T23 ITU;

SS7 LINKSET TIMERS TABLE 0 = DELHI 0, 8000, 30000, 800, 1400, 800,
800, 800, 1000, 1500, 800, 2000, 1500, 180000, 180000;

[\SS7 LINKSET TIMERS TABLE]

; ****************************************
; SS7 TIMERS - ANSI
; ****************************************

[SS7 SN TIMERS TABLE]

FORMAT SS7 SNTIMERS INDEX = SS7 SNTIMERS NAME, SS7 SNTIMERS T6,
SS7 SNTIMERS T8, SS7 SNTIMERS T10, SS7 SNTIMERS T11,
SS7 SNTIMERS T15, SS7 SNTIMERS T16, SS7 SNTIMERS T22 ANSI,
SS7 SNTIMERS T23 ANSI, SS7 SNTIMERS T24 ANSI,
SS7_SNTIMERS_T25_ANSI, SS7_SNTIMERS_T26_ANSI,
SS7 SNTIMERS T28 ANSI, SS7 SNTIMERS T29 ANSI,
SS7 SNTIMERS T30 ANSI;

SS7 SN TIMERS TABLE 1 = BABILON 0, 800, 1000, 30000, 30000, 2000,
1400, 180000, 180000, 5000, 30000, 12000, 3000, 60000, 30000;

[\SS7 SN TIMERS TABLE]

[SS7 LINKSET TIMERS TABLE]

FORMAT SS7 LKSETTIMERS INDEX = SS7 LKSETTIMERS NAME,
SS7 LKSETTIMERS T1SLT, SS7 LKSETTIMERS T2SLT, SS7 LKSETTIMERS T1,
SS7 LKSETTIMERS T2, SS7 LKSETTIMERS T3, SS7 LKSETTIMERS T4,
SS7 LKSETTIMERS T5, SS7 LKSETTIMERS T12, SS7 LKSETTIMERS T13,
SS7 LKSETTIMERS T14, SS7 LKSETTIMERS T17, SS7 LKSETTIMERS T20 ANSI,
SS7 LKSETTIMERS T21 ANSI;

SS7_LINKSET_TIMERS_TABLE 1 = HANOI_0, 8000, 30000, 800, 1400, 800,
800, 800, 1000, 1500, 2000, 1500, 90000, 90000;

[\SS7_LINKSET_TIMERS_TABLE]

[SS7 SN TABLE]
FORMAT SS7_SN_INDEX = SS7_SN_NAME, SS7_SN_TRACE_LEVEL,
SS7 SN ADMINISTRATIVE STATE, SS7 SN VARIANT, SS7 SN NI,
SS7 SN SP STP, SS7 SN OPC, SS7 SN ROUTESET CONGESTION WINSIZE,
SS7_SN_TIMERS_INDEX, SS7_SN_ISUP_APP, SS7_SN_SCCP_APP,
SS7 SN BISUP APP, SS7 SN ALCAP APP;

SS7_SN_TABLE 0 = SN_0, 0, 2, 1, 0, 0, 11, 8, 0, 4, 4, 4, 54;

SS7 SN TABLE 1 = SN 1, 0, 2, 1, 0, 0,  4, 8, 0, 4, 4, 4, 4;
[\SS7_SN_TABLE]

[ SS7 LINK TABLE ]

FORMAT SS7 LINK INDEX = SS7 LINK NAME, SS7 LINK TRACE LEVEL,
SS7 LINK ADMINISTRATIVE STATE, SS7 LINK L2 TYPE, SS7 LINK L3 TYPE,
SS7_LINK_TRUNK_NUMBER, SS7_LINK_TIMESLOT_NUMBER,
SS7 LINK LAYER2 VARIANT,  SS7 LINK MTP2 ATTRIBUTES,
SS7 CONGESTION LOW MARK, SS7 CONGESTION HIGH MARK;

SS7 LINK TABLE 0 = link 0 SP A, 0, 2, 1, 2, 0, 16, 1, 0, 5, 20;
SS7 LINK TABLE 1 = link 1 SP B, 0, 2, 1, 2, 1, 16, 1, 0, 5, 20;
SS7_LINK_TABLE 2 = link_2_SP_A, 0, 2, 1, 2, 0, 17, 1, 0, 5, 20;
SS7 LINK TABLE 3 = link 3 SP B, 0, 2, 1, 2, 1, 17, 1, 0, 5, 20;

[\SS7_LINK_TABLE]

[ SS7 LINKSET TABLE ]
FORMAT SS7_LINKSET_SN_INDEX, SS7_LINKSET_LINKSET_INDEX =
SS7 LINKSET NAME, SS7 LINKSET ADMINISTRATIVE STATE,
SS7 LINKSET DPC, SS7 LINKSET TIMERS INDEX;

;; for SN 0:
SS7 LINKSET TABLE 0, 0 = lkset0 sp A, 2, 4, 0;
;SS7_LINKSET_TABLE 0, 6 = lkset1, 2, 444, 0;

;; for SN 1:
SS7 LINKSET TABLE 1, 0 = lkset0 sp1, 2, 11, 0;

[ \SS7 LINKSET TABLE ]

[ SS7 LINKSETLINK TABLE ]
FORMAT SS7 LINKSETLINK SN INDEX, SS7 LINKSETLINK LINKSET INDEX,
SS7 LINKSETLINK INNER LINK INDEX = SS7 LINKSETLINK LINK NUMBER,
SS7_LINKSETLINK_LINK_SLC;
```

```
;; for SN 0:
SS7 LINKSETLINK TABLE 0, 0, 0 = 0, 0;
SS7 LINKSETLINK TABLE 0, 0, 1 = 2, 1;

;; for SN 1:
SS7 LINKSETLINK TABLE 1, 0, 0 = 1, 0;
SS7 LINKSETLINK TABLE 1, 0, 1 = 3, 1;
[ \SS7 LINKSETLINK TABLE ]

[ SS7 ROUTESET TABLE ]
FORMAT SS7 ROUTESET SN INDEX, SS7 ROUTESET INDEX =
SS7 ROUTESET NAME, SS7 ROUTESET ADMINISTRATIVE STATE,
SS7 ROUTESET DPC;

; for SN 0:
SS7 ROUTESET TABLE 0, 0 = RTESET0 SP A, 2, 4;

; for SN 1:
SS7 ROUTESET TABLE 1, 0 = RTESET0 SN1, 2, 11;

[ \SS7 ROUTESET TABLE ]

[ SS7 ROUTESETROUTE TABLE ]
FORMAT SS7 ROUTESETROUTE SN INDEX,
SS7 ROUTESETROUTE ROUTESET INDEX,
SS7 ROUTESETROUTE INNER ROUTE INDEX =
SS7 ROUTESETROUTE LINKSET NUMBER, SS7 ROUTESETROUTE PRIORITY;

; for SN 0:
SS7 ROUTESETROUTE TABLE 0, 0, 0 = 0, 0;

; for SN 1:
SS7 ROUTESETROUTE TABLE 1, 0, 0 = 0, 0;

[ \SS7 ROUTESETROUTE TABLE ]

[ SS7 ROUTING CONTEXT TABLE ]
FORMAT SS7 RC INDEX, SS7 RC INNER INDEX = SS7 RC SN INDEX,
SS7 RC OPC1, SS7 RC OPC2, SS7 RC OPC3, SS7 RC OPC4, SS7 RC SI1,
SS7 RC SI2, SS7 RC SI3, SS7 RC SI4;
SS7 ROUTING CONTEXT TABLE 0, 0 = 0, -1, -1, -1, -1, -1, -1, -1, -1;
SS7 ROUTING CONTEXT TABLE 0, 1 = 1, -1, -1, -1, -1, -1, -1, -1, -1;
[ \SS7 ROUTING CONTEXT TABLE ]

[ SS7 SIG IF GROUP TABLE ]
FORMAT SS7 SIG IF GR INDEX = SS7 IF GR ID,SS7 SIG SG MGC,
SS7 SIG LAYER, SS7 SIG TRAF MODE, SS7 SIG T REC, SS7 SIG T ACK,
SS7 SIG T HB, SS7 SIG MIN ASP, SS7 SIG BEHAVIOUR,
SS7 LOCAL SCTP PORT, SS7 SIG NETWORK, SS7 SIG RC INDEX,
SS7 SIG RC VALUE, SS7 SIG NETWORK APPEARANCE;
SS7 SIG IF GROUP TABLE 0 = 0, 83, 3, 1, 2000, 2000, 30000, 1, 0,
2905, 1, 0, 1, 1;

[ \SS7 SIG IF GROUP TABLE ]
```

6.2.4 SS7 MTP2 Tunneling ini File Example

'For the SS7 MTP2 Tunneling ***ini*** file example, take into account the following notes:

- This ***ini*** file is a configuration of the MTP2 tunneling central side (M2UA Media Gateway Controller links).
- There are 8 SS7 links - 4 links of type M2UA Media Gateway Controller, and 4 links of type MTP2. Each pair of links (1 M2UA Media Gateway Controller and 1 MTP2) defines an MTP2 tunnel.

- There is 1 interface that is used for the M2UA Media Gateway Controller <=> M2UA SG connection.
- There are 4 interface IDs defined: 1 per link (M2UA Media Gateway Controller side).
- This file is intended for ITU link variant (E1 trunks).

➢ To load the example of an SS7 MTP2 Tunneling *ini* file, take these 4 steps:

1. Load this ***ini*** file (as shown below, 'SS7 MTP2 Tunneling ***ini*** File Example - Media Gateway Controller') on a Tunnel central gateway (MTP2 Media Gateway Controller).
2. Load the ***ini*** file as shown in 'SS7 MTP2 Tunneling ***ini*** File Example - SG' on a tunnel remote gateway (MTP2 SG). The Media Gateway Controller gateway connects (over IP) to the SG gateway. For more information on loading an ***ini*** file, refer to **Software Upgrade Wizard** in the product's User's Manual.
3. In the Media Gateway Controller gateway, change the 'SS7_DEST_IP' parameter to the actual IP address of the M2UA SG gateway.
4. Change the value of the 'SyslogServerIP' parameter in the Media Gateway Controller and SG gateways to your Syslog server IP address.

The following is an example of SS7 MTP2 Tunneling ***ini*** File.

```
MGCONTROLPROTOCOLTYPE = 2
[TDM BUS configuration]

; 1=aLaw  3=ulaw
PCMLawSelect=    1

; EXT BUS=5   H110=4   QSLAC=3   FRAMERS=2   SC BUS=1   MVIP BUS=0
TDMBusType=        2

; 0=2048kbps, 2=4096kbps, 3=8192kbps
TDMBusSpeed=   3

;1 - internal, 3 - mvip, 4 - Network, 8 - h110a, 9 - h110b, 10 -
Netref
TDMBUSCLOCKSOURCE= 1

MGCONTROLPROTOCOLTYPE = 2
PROVISIONEDCALLAGENTS = 10.10.2.77
[Trunk Configuration]

;e1 euro isdn=1 t1 isdn=2 ;e1 cas r2=8 (8 for fcd); e1 trans 62=5
ProtocolType = 5

TraceLevel   = 0
; acCLOCK MASTER ON =1
CLOCKMASTER= 1

;acUSER TERMINATION SIDE = 0
TerminationSide = 1

;acEXTENDED SUPER FRAME=0
FramingMethod = 0

;acB8ZS = 0   2 for E1 CAS - FCD
LineCode = 0


[SS7]

SS7_MTP2_PARAM_TIMER_T1_0=50000
```

```
SS7 MTP2 PARAM TIMER T2 0=150000
SS7 MTP2 PARAM TIMER T3 0=1000
SS7 MTP2 PARAM TIMER T4E 0=500
SS7 MTP2 PARAM TIMER T4N 0=8200
SS7 MTP2 PARAM TIMER T5 0=100
SS7 MTP2 PARAM TIMER T6 0=3000
SS7 MTP2 PARAM TIMER T7 0=2000
;
[syslog]
;SYSLOGSERVERIP = 168.100.0.1
ENABLESYSLOG = 1

WATCHDOGSTATUS = 0

[ SS7 LINK TABLE ]
FORMAT SS7 LINK INDEX = SS7 LINK NAME, SS7 LINK TRACE LEVEL,
SS7 LINK ADMINISTRATIVE STATE,SS7 LINK L2 TYPE, SS7 LINK L3 TYPE,
SS7 LINK GROUP ID, SS7 LINK M2UA IF ID;
SS7 LINK TABLE 1 = new link 1, 0, 2, 2, 3, 4, 50;
SS7 LINK TABLE 3 = new link 3, 0, 2, 2, 3, 4, 12;
SS7 LINK TABLE 5 = new link 5, 0, 2, 2, 3, 4, 18;
SS7 LINK TABLE 7 = new link 7, 0, 2, 2, 3, 4,  1;

[ \SS7 LINK TABLE ]

[ SS7 LINK TABLE ]
FORMAT SS7 LINK INDEX = SS7 LINK NAME, SS7 LINK TRACE LEVEL,
SS7 LINK ADMINISTRATIVE STATE,SS7 LINK L2 TYPE, SS7 LINK L3 TYPE,
SS7 LINK TRUNK NUMBER,SS7 LINK TIMESLOT NUMBER,
SS7 LINK LAYER2 VARIANT,SS7 LINK MTP2 ATTRIBUTES,SS7 CONGESTION LOW
 M ARK, SS7 CONGESTION HIGH MARK, SS7 LINK TNL MGC LINK NUMBER,
SS7 LINK TNL ALIGNMENT MODE, SS7 LINK TNL CONGESTION MODE,
SS7 LINK TNL WAIT START COMPLETE TIMER,
SS7 LINK TNL OOS START DELAY TIMER,
SS7 LINK TNL WAIT OTHER SIDE INSV TIMER;


SS7 LINK TABLE 0 = new link 0, 0, 2, 1, 3, 0, 15, 1, 0, 5, 50, 1,
1,  0, 30000, 5000, 30000;
SS7 LINK TABLE 2 = new link 2, 0, 2, 1, 3, 3, 12, 1, 0, 5, 50, 3,
1, 0, 30000, 5000, 30000;
SS7 LINK TABLE 4 = new link 4, 0, 2, 1, 3, 6,  7, 1, 0, 5, 50, 5,
1, 0, 30000, 5000, 30000;
SS7 LINK TABLE 6 = new link 6, 0, 2, 1, 3, 7,  3, 1, 0, 5, 50, 7,
1, 0, 30000, 5000, 30000;
[ \SS7 LINK TABLE ]

[ SS7 SIG IF GROUP TABLE ]
FORMAT SS7 SIG IF GR INDEX = SS7 IF GR ID,SS7 SIG SG MGC,
SS7 SIG LAYER, SS7 SIG TRAF MODE, SS7 SIG T REC, SS7 SIG T ACK,
SS7 SIG T HB, SS7 SIG MIN ASP, SS7 SIG BEHAVIOUR,
SS7 LOCAL SCTP PORT, SS7 SIG NETWORK, SS7 DEST SCTP PORT,
SS7 DEST IP, SS7 MGC MX IN STREAM, SS7 MGC NUM OUT STREAM;
SS7 SIG IF GROUP TABLE 4 = 4, 77, 4, 1, 2000, 2000, 30000, 1, 0,
2904, 1,2904,168.100.0.2,3,3;

[ \SS7 SIG IF GROUP TABLE ]

[ SS7 SIG INT ID TABLE ]
FORMAT SS7 SIG IF ID INDEX = SS7 SIG IF ID VALUE,
SS7 SIG IF ID NAME, SS7 SIG IF ID OWNER GROUP, SS7 SIG IF ID LAYER,
SS7 SIG IF ID NAI, SS7 SIG M3UA SPC;
SS7 SIG INT ID TABLE 7 = 50, BELFAST12, 4, 4, 1, 0;
SS7 SIG INT ID TABLE 8 = 12, AMSTERDAM, 4, 4, 3, 0;
SS7 SIG INT ID TABLE 9 = 18, ROTERDAM , 4, 4, 5, 0;
SS7 SIG INT ID TABLE 10 = 1, GAUDA    , 4, 4, 7, 0;
[ \SS7_SIG_INT_ID_TABLE ]
```

For the SS7 MTP3 Redundancy ini file example, take into account the following notes:

- This *ini* file acts as a local end-point in Shared Point Code configuration. In this configuration, all local end-points should have the same *ini* file except of one parameter: SS7MTP3RdcyBoardNum. This parameter should be unique in each device *ini* file.
- There are 2 blades with MTP3 shared point code.
- There are 2 physical links in each redundant blade.
- There is 1 Linkset that is distributed across the 2 blades.
- This file is intended for ITU link variant (E1 trunks).

➢ To load the example of an SS7 SN Redundancy ini file, take these 4 steps:

1. Change the value of the 'SyslogServerIP' parameter in the Media Gateway Controller and SG gateways, to your Syslog server IP address.
2. Change the value of the ' SS7_RDCYSN_BOARD_IP' parameters in the redundancy table to your redundant devices IP address.
3. Create a new *ini* file that is duplicated to the *ini* file below (SS7 SN Redundancy *ini* file example) and change the SS7MTP3RdcyBoardNum parameter in this *ini* file to 1.
4. Load the 2 *ini* files you received from the 2 TP devices you have.

The following is an example of SS7 MTP3 Redundancy *ini* file:

```
[TDM BUS configuration]
;   1=aLaw  3=ulaw
PCMLawSelect=    1

;   EXT BUS=5   H110=4   QSLAC=3   FRAMERS=2   SC BUS=1
MVIP BUS=0
TDMBusType=        2

;   0=2048kbps, 2=4096kbps, 3=8192kbps
TDMBusSpeed=   3

;  1 - internal, 3 - mvip, 4 - Network, 8 - h110a, 9 - h110b, 10 -
Netref
TDMBUSCLOCKSOURCE= 1

[Trunk Configuration]
;E1 EURO ISDN=1, T1 CAS=2, T1 RAW CAS=3, T1 TRANS=4, E1 TRANS 62=5,
E1 TRANS 60=6,
;E1 MFCR2=7, E1 CAS R2=8, E1 RAW CAS=9, T1 NI2 ISDN=10,
T1 4ESS ISDN=11, T1 DMS100 ISDN=14
ProtocolType = 5

CLOCKMASTER= 1

TerminationSide = 1

[syslog]
SYSLOGSERVERIP = x.x.x.x
ENABLESYSLOG = 1

TraceLevel   = 1

;; SS7 MTP3 REDUNDANCY PARAMS:
```

```
;; 0 = DISABLED, 1 = ENABLED
SS7MTP3RdundancyMode = 1

;; 0 = NONE, 2 = TCP
SS7MTP3RdundancyTransferType = 2

SS7MTP3RdcyBoardNum = 0

; TABLES

; ******************************************
; SS7 TIMERS - ITU
; ******************************************
[SS7 SN TIMERS TABLE]
FORMAT SS7 SNTIMERS INDEX = SS7 SNTIMERS NAME, SS7 SNTIMERS T6,
SS7 SNTIMERS T8, SS7 SNTIMERS T10, SS7 SNTIMERS T11,
SS7 SNTIMERS T15, SS7 SNTIMERS T16, SS7 SNTIMERS T18 ITU,
SS7 SNTIMERS T19 ITU, SS7 SNTIMERS T20 ITU, SS7 SNTIMERS T21 ITU,
SS7 SNTIMERS T24 ITU;

SS7 SN TIMERS TABLE 0 = TENERIFF 0, 800, 1000, 30000, 30000, 2000,
1400, 5000, 4000, 15000, 10000, 500;

[\SS7 SN TIMERS TABLE]

[SS7 LINKSET TIMERS TABLE]
FORMAT SS7 LKSETTIMERS INDEX = SS7 LKSETTIMERS NAME,
SS7 LKSETTIMERS T1SLT, SS7 LKSETTIMERS T2SLT, SS7 LKSETTIMERS T1,
SS7 LKSETTIMERS T2, SS7 LKSETTIMERS T3, SS7 LKSETTIMERS T4,
SS7 LKSETTIMERS T5, SS7 LKSETTIMERS T7, SS7 LKSETTIMERS T12,
SS7 LKSETTIMERS T13, SS7 LKSETTIMERS T14, SS7 LKSETTIMERS T17,
SS7 LKSETTIMERS T22 ITU, SS7 LKSETTIMERS T23 ITU;

SS7 LINKSET TIMERS TABLE 0 = DELHI 0, 8000, 30000, 800, 1400, 800,
800, 800, 1000, 1500, 800, 2000, 1500, 180000, 180000;
[\SS7 LINKSET TIMERS TABLE]


SS7 SN TIMERS TABLE 1 = BABILON 0, 800, 1000, 30000, 30000, 2000,
1400, 5000, 4000, 5000, 10000, 5000, 5000, 30000, 30000;

[\SS7 SN TIMERS TABLE]

[SS7 LINKSET TIMERS TABLE]
FORMAT SS7 LKSETTIMERS INDEX = SS7 LKSETTIMERS NAME,
SS7 LKSETTIMERS T1SLT, SS7 LKSETTIMERS T2SLT, SS7 LKSETTIMERS T1,
SS7 LKSETTIMERS T2, SS7 LKSETTIMERS T3, SS7 LKSETTIMERS T4,
SS7 LKSETTIMERS T5, SS7 LKSETTIMERS T12, SS7 LKSETTIMERS T13,
SS7 LKSETTIMERS T14, SS7 LKSETTIMERS T17, SS7 LKSETTIMERS T20 ANSI,
SS7 LKSETTIMERS T21 ANSI;

SS7 LINKSET TIMERS TABLE 1 = HANOI 0, 8000, 30000, 800, 1400, 800,
800, 800, 1000, 1500, 1500, 1500, 90000, 90000;
[\SS7 LINKSET TIMERS TABLE]

[SS7 SN TABLE]
FORMAT SS7 SN INDEX = SS7 SN NAME, SS7 SN TRACE LEVEL,
SS7 SN ADMINISTRATIVE STATE, SS7 SN VARIANT, SS7 SN NI,
SS7 SN SP STP, SS7 SN OPC,  SS7 SN ROUTESET CONGESTION WINSIZE,
SS7 SN TIMERS INDEX, SS7 SN ISUP APP, SS7 SN SCCP APP,
SS7 SN BISUP APP, SS7 SN ALCAP APP;

SS7 SN TABLE 0 = PLOP 0, 1, 2, 1, 0, 1, 3,  8, 0, 4, 4, 4, 4;

[\SS7 SN TABLE]
```

```
; SS7 LINKS - MTP3 LINKS!!!
[ SS7 LINK TABLE ]
FORMAT SS7 LINK INDEX = SS7 LINK NAME, SS7 LINK RDCY BOARD,
SS7 LINK TRACE LEVEL, SS7 LINK ADMINISTRATIVE STATE,
SS7 LINK L2 TYPE, SS7 LINK L3 TYPE, SS7 LINK TRUNK NUMBER,
SS7 LINK TIMESLOT NUMBER, SS7 LINK LAYER2 VARIANT,
SS7 LINK MTP2 ATTRIBUTES, SS7 CONGESTION LOW MARK,
SS7 CONGESTION HIGH MARK;

SS7 LINK TABLE 0 = LINK 0 SP 0, 0, 1, 2, 1, 2,  0, 14,  1, 0, 5,
50;
SS7 LINK TABLE 1 = LINK 1 SP 0, 1, 1, 2, 1, 2,  0, 14,  1, 0, 5,
50;
SS7 LINK TABLE 2 = LINK 3 SP 0, 0, 1, 2, 1, 2,  0, 18,  1, 0, 5,
50;
SS7 LINK TABLE 3 = LINK 2 SP 0, 1, 1, 2, 1, 2,  0, 18,  1, 0, 5,
50;


; SS7 LINKSET TABLE
[ SS7_LINKSET_TABLE ]
FORMAT SS7 LINKSET SN INDEX, SS7 LINKSET LINKSET INDEX =
SS7 LINKSET NAME, SS7 LINKSET ADMINISTRATIVE STATE,
SS7_LINKSET_DPC, SS7_LINKSET_TIMERS_INDEX;

SS7 LINKSET TABLE 0, 0 = PLOP 0 0, 2, 4, 0;

[ \SS7 LINKSET TABLE ]


; SS7 LINKS in LINKSETS
[ SS7 LINKSETLINK TABLE ]
FORMAT SS7_LINKSETLINK_SN_INDEX, SS7_LINKSETLINK_LINKSET_INDEX,
SS7 LINKSETLINK INNER LINK INDEX = SS7 LINKSETLINK LINK NUMBER,
SS7 LINKSETLINK LINK SLC;

SS7 LINKSETLINK TABLE 0, 0, 0 = 0, 0;
SS7 LINKSETLINK TABLE 0, 0, 1 = 1, 1;
SS7_LINKSETLINK_TABLE 0, 0, 2 = 2, 2;
SS7 LINKSETLINK TABLE 0, 0, 3 = 3, 3;

[ \SS7_LINKSETLINK_TABLE ]


; SS7 ROUTESET TABLE
[ SS7 ROUTESET TABLE ]
FORMAT SS7 ROUTESET SN INDEX, SS7 ROUTESET INDEX =
SS7_ROUTESET_NAME, SS7_ROUTESET_ADMINISTRATIVE_STATE,
SS7 ROUTESET DPC;

SS7_ROUTESET_TABLE 0, 0 = ROUTESET_0_SP_0, 2, 4;

[ \SS7 ROUTESET TABLE ]



; SS7 ROUTES IN ROUTESETS
[ SS7 ROUTESETROUTE TABLE ]
FORMAT SS7 ROUTESETROUTE SN INDEX,
SS7 ROUTESETROUTE ROUTESET INDEX,
SS7 ROUTESETROUTE INNER ROUTE INDEX =
SS7 ROUTESETROUTE LINKSET NUMBER, SS7 ROUTESETROUTE PRIORITY;

; for SN 0:
SS7 ROUTESETROUTE TABLE 0, 0, 0 = 0, 0;

[ \SS7_ROUTESETROUTE_TABLE ]
[ SS7 ROUTING CONTEXT TABLE ]
FORMAT SS7 RC INDEX, SS7 RC INNER INDEX = SS7 RC SN INDEX,
SS7 RC OPC1, SS7 RC OPC2, SS7 RC OPC3, SS7 RC OPC4, SS7 RC SI1,
SS7 RC SI2, SS7 RC SI3, SS7 RC SI4;
SS7 ROUTING CONTEXT TABLE 0, 0 = 0, -1, -1, -1, -1, -1, -1, -1, -1;

[ \SS7 ROUTING CONTEXT TABLE ]


[ SS7 SIG IF GROUP TABLE ]
FORMAT SS7 SIG IF GR INDEX = SS7 IF GR ID,SS7 SIG SG MGC,
SS7 SIG LAYER, SS7 SIG TRAF MODE, SS7 SIG T REC, SS7 SIG T ACK,
SS7 SIG T HB, SS7 SIG MIN ASP, SS7 SIG BEHAVIOUR,
SS7 LOCAL SCTP PORT, SS7 SIG NETWORK, SS7 SIG RC INDEX,
SS7 SIG RC VALUE, SS7 SIG NETWORK APPEARANCE;
SS7 SIG IF GROUP TABLE 6 =  6,83, 3, 1, 2000, 2000, 30000, 1, 1024,
2905, 1, 0, 1, 1;
[ \SS7 SIG IF GROUP TABLE ]


[ \SS7 SIG INT ID TABLE ]

; ****************************************
;
;           SS7 REDUNDANCY TABLES
;
; ****************************************

[ SS7 REDUNDANCYSN TABLE ]
FORMAT SS7 RDCYSN BOARD INDEX, SS7 RDCYSN SN INDEX =
SS7 RDCYSN BOARD IP, SS7 RDCYSN OPC;

;; Board 0 SN 0 (IP y.y.y.y)
SS7 REDUNDANCYSN TABLE 0, 0 = y.y.y.y, 1;

;; Board 1 SN 0 (IP z.z.z.z)
SS7 REDUNDANCYSN TABLE 1, 0 = z.z.z.z,  2;

[ \SS7_REDUNDANCYSN_TABLE ]
```

6.3 SS7 Tunneling: Feature Description

The SS7 tunneling feature facilitates peer-to-peer transport of SS7 links between gateways that support this unique MTP2 Tunneling application (M2TN) for transferring SS7 MTP2 link data over IP. In this scenario, both sides of the link are pure TDM switches and are unaware of the IP tandem that is utilized between them. Using M2TN, the network operator can support SS7 connections over IP, carrying MTP level 3, as well as higher level SS7 layers (e.g., user parts and application protocols, such as TUP, ISUP, SCCP, TCAP, etc.).

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

Figure 6-6: M2UA Architecture

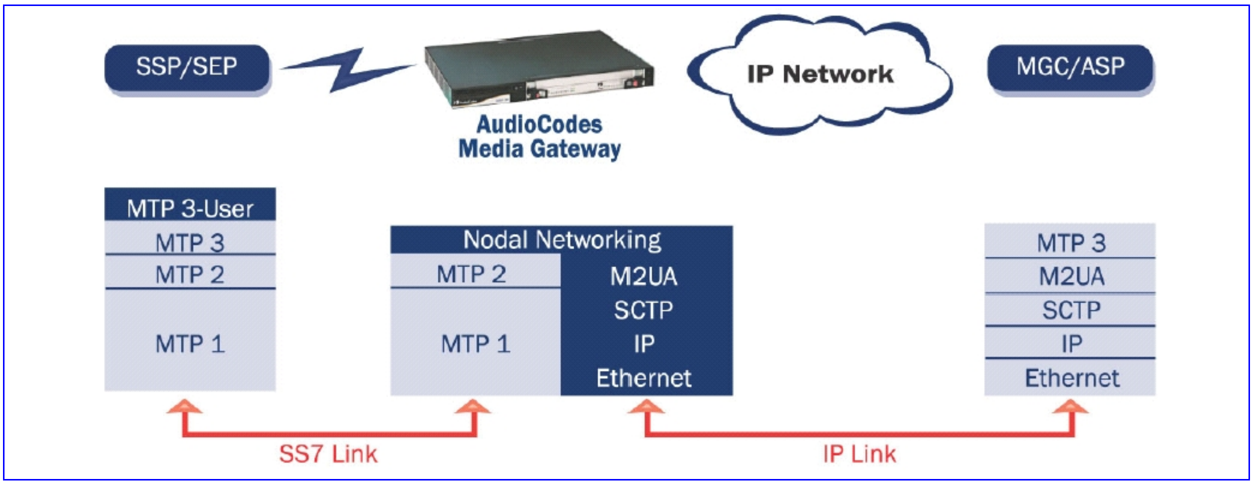
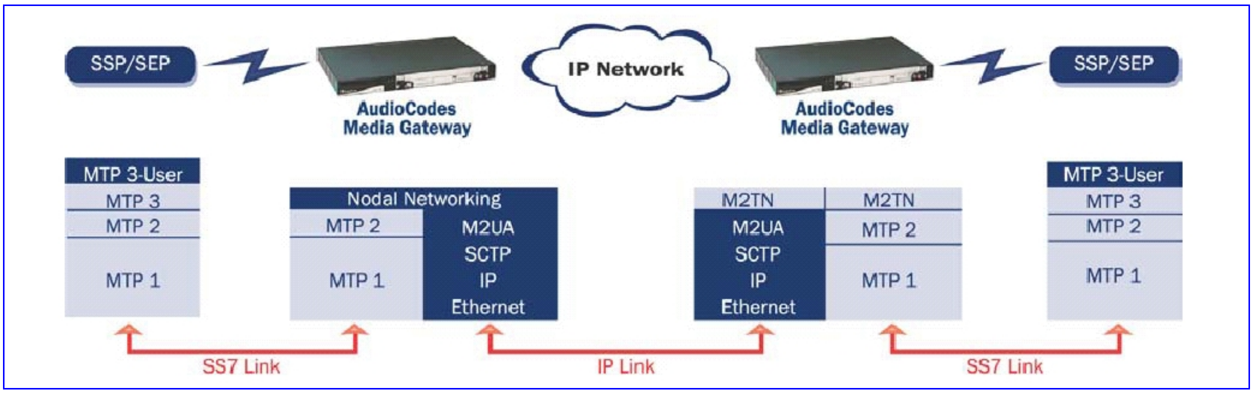


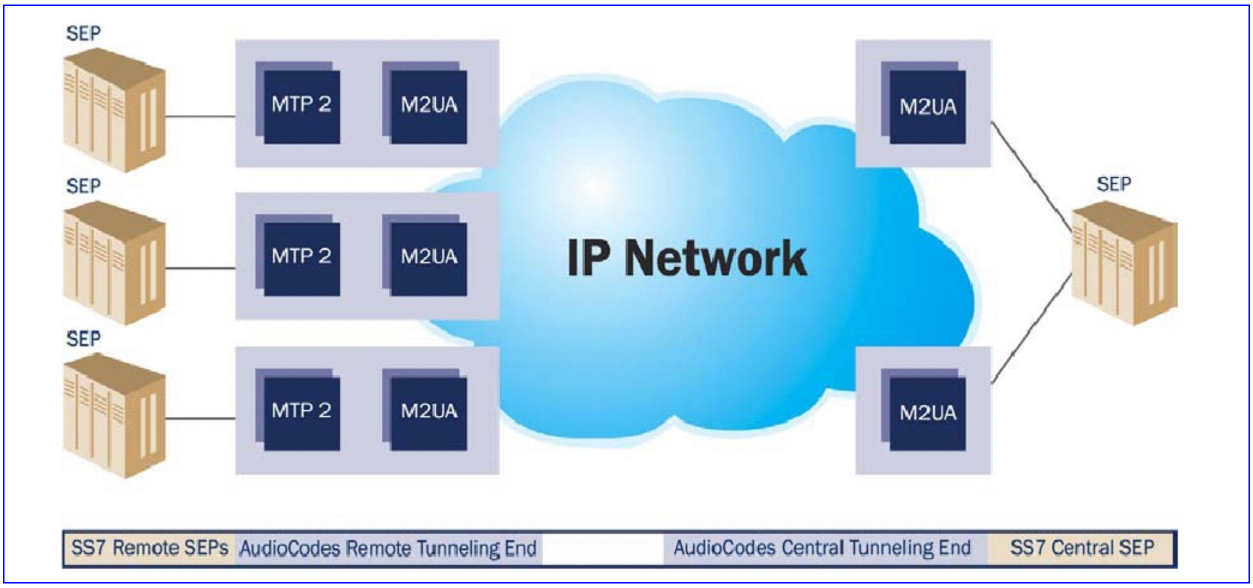
Figure 6-7: M2TN Architecture



6.3.1 MTP2 Tunneling Technology

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

Figure 6-8: Protocol Architecture for MTP2 Tunneling



6.3.2 SS7 Tunneling Application Characteristics

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

6.4 IUA/DUA

6.4.1 IUA /DUA Behind NAT Support

To be able to support IUA Signaling Gateway (SG) functionality behind a NAT, SG needs to initiate SCTP (by sending SCTP init to the MGC Side). After SCTP association establishment, the SG waits for ASP commands from MGC. This is done via a new configuration of the following line in the SS7_SIG_IF_GROUP table. Note: This line is valid also for DUA.

```
[ SS7_SIG_IF_GROUP_TABLE ]
FORMAT SS7_SIG_IF_GR_INDEX = SS7_IF_GR_ID,SS7_SIG_SG_MGC,
SS7_SIG_LAYER, SS7_SIG_TRAF_MODE, SS7_SIG_T_REC, SS7_SIG_T_ACK,
SS7_SIG_T_HB, SS7_SIG_MIN_ASP, SS7_SIG_BEHAVIOUR,
SS7_LOCAL_SCTP_PORT, SS7_SIG_NETWORK, SS7_DEST_SCTP_PORT,
SS7_DEST_IP, SS7_MGC_MX_IN_STREAM, SS7_MGC_NUM_OUT_STREAM;

SS7_SIG_IF_GROUP_TABLE 1 =  1,1, 1, 1, 2000, 2000, 30000, 1, 0,
9900, 1,9900,10.31.4.100,3,3;

[ \SS7_SIG_IF_GROUP_TABLE ]
```

- SS7_SIG_SG_MGC parameter value for NAT is 1.
- SS7_DEST_IP is MGC's IP.
- SS7_MGC_MX_IN_STREAM and SS7_MGC_NUM_OUT_STREAM values need to be coordinated with the value that is configured in the MGC side.

6.4.2 DASS2 Support in DUA

The device supports the DASS2 protocol via DUA (RFC 4129). DUA supports the DASS2 protocol according to BTNR 190 (June 1992). To configure the device for DASS2, use the E1_DUA protocol and the DPNSSBehavior parameter, with a value of 16.

6.5 M3UA Routing Context

The following has been taken from RFC 4666 – Section 1.4.2.1:

"The distribution of SS7 messages between the SGP and the Application Servers is determined by the Routing Keys and their associated Routing Contexts.

A Routing Key is essentially a set of SS7 parameters used to filter SS7 messages, whereas the Routing Context parameter is a 4-octet value (integer) that is associated to that Routing Key in a 1:1 relationship.

The Routing Context therefore can be viewed as an index into a sending node's Message Distribution Table containing the Routing Key entries.

Possible SS7 address/routing information that comprise a Routing Key entry includes, for example, the OPC, DPC, and SIO found in the MTP3 routing label.

Some example Routing Keys are: the DPC alone, the DPC/OPC combination, or the DPC/OPC/SI combination…"

- Up to 16 Static Routing Contexts are supported. The static Routing Context parameters should be configured using the "SS7_ROUTING_CONTEXT_TABLE" table. (Please refer to SS7 Static Routing Context Table in the Web Interface and in the product's User Manual.)
- In the Interface Group table, the user MUST configure for each group, the index in Routing Context table that relates to this group. The relationship between these 2 tables is 1:1.

- Users can configure Routing Context values and Network appearance in the Interface Group table.

Note: The “M3UAROUTINGCLIST” table in no longer supported. Users MUST use the new table and fields, as described above.

Note: There is no need to configure M3UA in the Interface table. The SN index should be configured in the Routing Context Table.

6.6 SS7 MTP3 Redundancy

SS7 MTP3 is the network layer of SS7. It defines and manages the behavior of signaling nodes (point-codes). Each signaling node may use several signaling links to communicate with the rest of the SS7 network.

AudioCodes has a working MTP3 layer in one CPU. It is an important goal to be able to manage a point code which is distributed over several CPUs. The major reasons for doing this are:

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

6.6.1 General Architecture

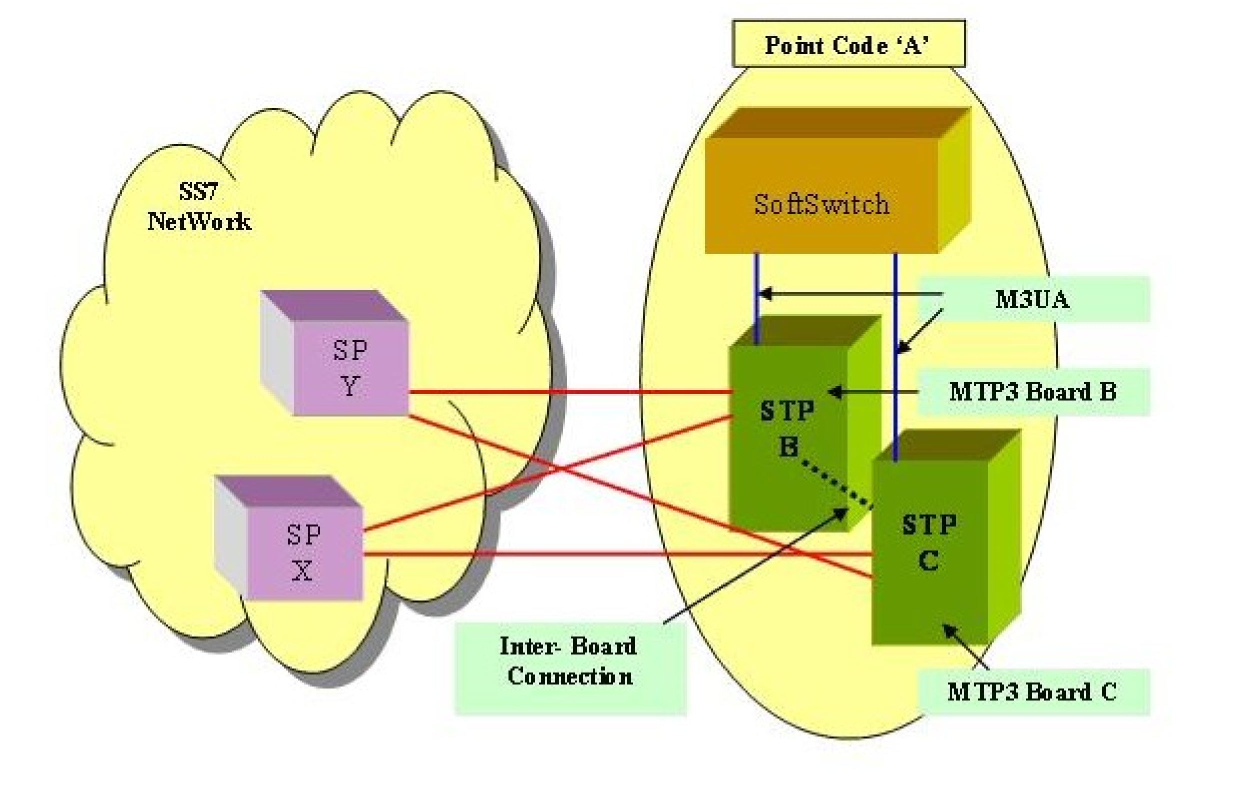
There are two working modes for MTP3:

1. Regular mode - This mode behaves as MTP3 did so far: all links are handled by a single CPU. In this mode, all that is listed in this document will have no effect.
2. Redundancy mode - In this mode two devices (i.e. CPUs) may participate in a distributed point-code.

6.6.2 SS7 MTP3 Redundancy

Note: The figure below is a private case of Point-Codes-Sharing between 2 devices.

Figure 6-9: SS7 MTP3 Redundancy



The point code 'A' is implemented by two devices (i.e. CPUs): 'B' and 'C'.

In the figure above, there are 2 link-sets from point-code 'A': one to point-code 'Y', and the other to point-code 'X'. Both link-sets are distributed across the two devices (B and C).

6.6.2.1 SS7 Redundancy X-Connection: Traffic Diversion Policy

In case links from same link-set are distributed across devices, a failure of a link will be handled by all devices according to the following rules:

- If there are alternative ways (i.e. another link from same link-set, another route-set) from the given device - the transmitted MSUs will be sent using that option. No inter-device traffic is required.
- If not - the MSUs will be sent to another device using the inter-device connection, and then sent over the active links there.
- If no links are available on all devices, the MSUs arriving from the Soft-Switch will be discarded (the Soft-Switch will get an immediate indication via M3UA from MTP3 in that case, and should avoid sending MSUs to the relevant destinations).

Traffic will be sent over X-Connection only if there are no available links on the receiving device.

6.6.2.2 SS7 Redundancy - Events Policy

SS7 shared point-code will generate all events that has been supported prior to that feature. All devices will generate all events. This means that there may be multiple events for same issue from different devices. For example, if a link fails, all devices will generate an event regarding that.

6.6.3 Configuration of Shared Point-Code

Configuration of a shared point-code is simply an extension to the current configuration of SS7 MTP3 single-device point-code.

The general principle is that the same SS7 configuration will be loaded on all devices that shares the same point code. It means that the following tables' configuration must be the same: tables of links, SNs, SN timers, linksets, linksetlinks, linksets timers, routesets, routesetroutes, SN-RDCY and SS7M3UATrafficBehavior flag.

This way, all gateways will be aware of the global view of the shared point-code.

⚠ **Note:** This is a mandatory requirement. Gateways must be co-ordinated in order to function properly.

Note that this requires a basic knowledge of configuring an AudioCodes single device SS7 MTP3. All relevant details related to this issue can be found in the formal documentation of the GA version.

Shared point-code must be configured as STP.

A new configuration was added for the redundancy feature and is used only when the device is in MTP3 Redundancy mode.
The following sections describe the new configuration.

6.6.4 New Device Parameters in INI File

These are the new device parameters that must be configured in the ***ini*** file for MTP3 redundancy:

Table 6-1: New Device Parameters for MTP3 Redundancy

Parameter	Description / Modification	Default	Note
SS7MTP3RdcyTransferType	SS7 MTP3 Cross-Device Connection media type used for redundancy feature. Should set to: TCP (2)	None (0)	Can't be changed on-the-fly.
SS7MTP3RdcyMode	SS7 MTP3 redundancy mode. Determines redundancy flavor: Disable (No redundancy) and Enable. Disable = 0, Enable = 1	Disable	Can't be changed on-the-fly.
SS7MTP3RdcyBoardNum	Device number for SS7 MTP3 redundancy mode - Each device is given a unique number since they all share a common redundancy table.	0	Must be set. Can't be changed on-the-fly.
SS7MTP3RdcyKeepAliveInterval	Defines redundancy X-link keep-alive interval in seconds.	1	Can't be changed

Table 6-1: New Device Parameters for MTP3 Redundancy

Parameter	Description / Modification	Default	Note
	(x-link between devices in Signaling System 7 (SS7) MTP3-User Adaptation Layer redundancy mode). Range [0 .. 100] 0 = no keep-alive mechanism is activated.		on-the-fly.
SS7MTP3RdcyKeepAliveWindow	Defines redundancy X-link keep-alive tolerance window. (x-link between devices in Signaling System 7 (SS7) MTP3-User Adaptation Layer redundancy mode). Range [1 .. 15]	2	Can't be changed on-the-fly.
SS7MTP3RdcyTblSyncInterval	Define the interval between SS7 tables automatic synchronization process, in minutes. This process compares the current device SS7 tables with the next “device number” (i.e.: device #1 will compare with device #2). 0 = synchronization mechanism is disabled.	0	Can be changed on-the-fly.

6.6.5 Parameter in Links Table

The parameter, SS7_LINK_RDCY_BOARD specifies the device number on which the physical link actually resides. It is used to distinguish real links from 'stand-by' links.

Stand-by links are real links that are physically active on another device.

If the link's device number equals actual device number (see 'SS7MTP3RdcyBoardNum' above), then the link in active on that device.

6.6.6 Parameter in Sigtran Interface Group Table

The parameter RdcyBoardNum specifies the device number on which the physical SCTP connection should be opened.

6.6.7 MTP3 Redundancy SNs Table

One new table (MTP3 Redundancy SNs Table) has been added to support issues such as inter-device connectivity and used only when the device is in redundancy mode.

This table includes details required for cross-connectivity between devices. It describes which SN is distributed across devices, and other required details regarding it.

6.6.8 Adding a New Gateway

In case there are already one or more gateways running in Shared-Point-Code configuration and a new gateway is to be added to this configuration, take the following steps:

- Configure the new gateway "RDCY SN-Board table" parameters on an active gateway
- Upload SS7 and MTP3 RDCY parameters from the active gateway
- Take the uploaded parameters, change the SS7MTP3RdcyBoardNum to the new device number and initiate the new gateway with these INI parameters.

Reader's Notes

7 IPmedia Functionality & Configuration

Note: This IPmedia section is applicable to **IPM** devices.

This section describes the IPmedia capabilities for VoPLib users. The IPmedia supports a variety of media-processing functions on one device, such as conferencing and message record/playback.

Note:

The device supports 240 media channels that transfer and process media between Endpoints using a large variety of coders and algorithms, such as G.168-compliant echo cancelation, G.723.1, G.729A, Vox ADPCM and MS-GSM (Microsoft). The media-processing functions enjoy full flexibility in the mixture of Endpoints and functions that can be selected: mix and match of DS0, RTP and MVIP/SCbus/H.100 conferencing, host-based RTP recording in PCM coding, etc. The device is truly designed for any-to-any architecture.

The basic configuration includes an MVIP/SCbus/H.100 interface for adjacent Network Interface cards. An optional E1/T1/J1 trunk interface is available for up to 8 T1/E1\J1 trunks.

Note:

The device supports 240 (480 in STR) media channels that transfer and process media between Endpoints using a large variety of coders and algorithms, such as G.168-compliant echo cancelation, G.723.1, G.729A, Vox ADPCM and MS-GSM (Microsoft). The media-processing functions enjoy full flexibility in the mixture of Endpoints and functions that can be selected: mix and match of DS0, RTP and H.110 conferencing, host-based RTP recording in PCM coding, etc. The device is truly designed for any-to-any architecture.

The basic configuration includes an H.110 interface for adjacent Network Interface cards. An E1/T1/J1 trunk interface is available for up to 8 T1/E1/J1 trunks.

Supported by the same VoIP Library and control protocols as the other devices in the series, the device protects the investment of AudioCodes users in creating enhanced devices and services.

For detailed documentation on the IPmedia library API, refer to AudioCodes' “VoPLib API Reference Manual”, Document #: LTRT-840xx.

7.1 Basic Media Server

The IPmedia features are:

- IP Record/Play to/from PCI
- Conferencing
  - Whisper Coach
  - 3 to 64 participants (full-duplex) in a single conference
- Time Slots Summation
- Barge in - Stops Playback when voice is detected
- IPmedia Detectors
  - Pattern Detector
  - Answer Detector (AD)
  - Energy Detector
  - Answering Machine Detector (AMD)
- AGC (Automatic Gain Control)
- Network side - In Band Signaling (IBS) detection
- Fast/Slow playback

The following subsections provide a general description of the operation of each of the features listed above.

7.1.1 Conferencing

7.1.1.1 Introduction

The conferencing functionality enables users to create several simultaneous multi-party conferences. The conference participants can be either PSTN/TDM participants or IP participants (in any mix). Participants can be either regular participants or listen-only participants. Users can control various parameters both at the conference level (such as number of simultaneous speakers, and participant addition/deletion audible tone (beep) enabling) and at the participant level (such as input/output gains).

The following sections detail use of this new conferencing functionality.

7.1.1.2 Available Conference Resources

Conferencing is based on two resources: conferences and conference participants.

- Conference - a group of parties connected together simultaneously.
- Conference Participants - the total number of parties that can participate in all of the simultaneous conferences together.

Various devices support different conference resource densities:

The device, supporting 240 channels, running IPmedia software, supports up to 240 conference participants. Each conference can accommodate between 3 to 64 participants. The maximum number of conferences is 40 (**for 260/UNI**) and 80 (**for 1610/2000**).

7.1.1.3 Whisper Coach Function

In a Conference Call scenario, instead of a user specifying many mute commands, a single Whisper Coach command can be used. For example, if participant A wants to whisper to participant E without participants B, C and D overhearing, then instead of sending commands to mute A to B, mute A to C and mute A to D, a single Whisper Coach Command for A to E achieves this.

7.1.1.4 Active Speaker Notification Service

The device supports Active Speaker Notification service (ASN) which enables receiving a report of the current active speakers in the conference. When enabled, a report will be issued for every change in the conference active speakers. The minimum interval between reports can be configured.

7.1.2 Additional Time Slot Summation

7.1.2.1 General Description

Note: Users can configure the Additional Time Slot (ATS) and an output slot to be from the PSTN, H.100, MVIP, or SC Bus. This is applicable to **260/UNI only.**

Note: Time Slot (ATS) and an output slot to be from the PSTN, H.110 Bus. **This is applicable to 1610/2000 only.**

Note: When a channel is in Summation state, the channel's output to the TDM side is disabled.

7.1.3 Barge-In Function

This function is used to stop the playback operation prematurely (that is, for example, before reaching the end of the file to play).

An application for this function can be to predict dialing when an IVR machine performs playback and the user dials a DTMF tone before the playback is completed. When detecting the DTMF while playing back with the Barge-In function, the IVR playback can be terminated.

VoPLib users can stop the playback using the "Barge In" parameter of the acPlay() function. By using this parameter, users can cause the playback to stop automatically when a DTMF and/or speech event is detected.

7.1.4 IPmedia Detectors

The following is a description of the IPmedia special detectors:

- Pattern Detector
- Answering Machine Detector (AMD)
- Answer Detector (AD)
- Energy Detector
- Automatic Gain Control (AGC)

Note: When the above DSP Detectors are employed, the maximum number of DSP channels supported may be reduced. For more information please refer to the relevant product's Release Notes document.

7.1.4.1 Pattern Detector

The device supports detection of a specific one-byte data pattern (with a repetition of user-defined N times) transmitted over digital E1/T/J1 or TDM time slots. A typical application can be to generate and transmit this pattern to pause or resume recording specific E1/T1/J1 time slots. On initialization, the device can be configured to detect up to four different one-byte data patterns. When the defined data pattern is detected, an event (with time information) is generated. The event includes the pattern identity and the input channel number through which the data pattern was received. This event can be read by the recording application through the device's API. Based on this event, the application can pause/resume recording. Users can define any four one-byte data patterns to be detected (AA, A5, 7B…etc.).

An event is generated indicating the appearance as well as the disappearance of the pattern.

7.1.4.2 Answering Machine Detector (AMD)

Answering Machine Detection can be useful in automatic dialing applications. In some of these applications, it is important to detect if a human voice or answering machine is answering the call.

Answering Machine Detection can be activated and de-activated only after a channel is already open. The direction of the detection (TDM or Network) can be configured, as well as the detector detection sensitivity.

Upon every Answering Machine Detection activation, an event is being issued. The event holds the detection results. The result can be one of the following values:

- acAmdLiveCall - human voice has been detected
- acAmdAnsweringMachine - answering machine has been detected
- acAmdSilenceCall - there was no voice input (silence)

7.1.4.3 Answer Detector (AD)

Note: A typical application of the AD functionality is to enable the pause or resumption of the recording of a specific input channel (PSTN, IP or H.110 bus) according to the presence of a speech signal on specific input channels.

When the AD is activated after a speech signal is detected, an event is generated. This event can be read by the recording application through the device API. The event is generated when the speech signal is detected, and again when the speech signal disappears. The event includes the channel number, the reason (speech on/off), and time information.

7.1.4.4 Energy Detector (ED)

Note: The Energy Detector notifies the application when the signal on a specific channel (PSTN, IP or H.110 bus) crosses a pre-defined energy threshold. **This is applicable to IPM-1610 / IPmedia 2000 devices only.**

When the signal crosses this threshold, an event is generated that can be read by the recording application via the API. The event includes the channel number and the reason that the signal exceeds or falls below the threshold. Using the API, users can enable the Energy Detector per channel and set up the threshold level. The Energy Detector is designed with hysteresis.

7.1.5 Automatic Gain Control (AGC) Settings

Note: Instead of manually specifying the gain of the input channel (PSTN, IP or MVIP/SC/H.100 bus), users can enable or disable AGC per input channel through the API. **This is applicable to 260 devices only.**

Users can use the AGC feature to control the gain of a signal entered from the PSTN, IP, or MVIP/SC/H.100 bus to the device. This feature enables users to compensate for near-far gain differences:

- Activate: When equal to 1, activates the AGC
- Redirection: Determines the AGC direction
  When equal to 0, the AGC works on signals from the TDM side
  When equal to 1, the AGC works on signals coming from the Network side
- Target Energy: Determines the signal energy value [-dBm] that the AGC tries to reach

7.1.6 In-Band Signaling (IBS) Detection - Network Side

When streaming playback to and recording from the IP side, users can “manually” change the direction of IBS detection to detect IBS from the network by changing IBSDRedirection to 1 (struct acTIBSDetectorsSettings) to control the play/record on an IBS event such as DTMFs or CallProgressTones.

7.2 Advanced Media Server (AMS) Features

Note: The AMS Features chapter is only applicable to **IPmedia.**

The device incorporates the Advanced Media Server (AMS) features, and currently provides the following capabilities:

- Interactive Voice Response (IVR)
- Bearer Channel Tandeming (BCT)
- Conferencing
- Test Trunk Support

7.2.1 Interactive Voice Response (IVR)

As part of the AMS feature package, the device provides basic IVR control capability that includes:

- Playing an announcement (Play)
- Playing an announcement and collecting DTMF digits (PlayCollect)
- Playing an announcement and performing media recording (PlayRecord)

Announcements can be stored in the local memory of the device, or on a remote (HTTP or NFS) server.

The device supports the H.248.9 Advanced Audio Server Package (AASP) as well as TD-51, its precursor internal draft, as well as the PacketCable™ Audio Server Protocol Specification. (For the compliance tables of these protocols, refer to 'H.248.9 Compliance Matrix'.) The 'AASPackagesProfile' configuration parameter (***ini*** file) is used for selecting the required protocol variant.

These protocols provide a rich set of announcement specification capabilities. In addition to allowing a single clip to be played, they also provide for Sets and Audio Variables.

- Audio Variables are where the application passes the media server a value and type, and the media server assembles the correct fragments. More detail on Audio Variables is provided below.

- Sets are groups of clips identified with a single audio reference qualified by an index. For example, there might be a daily greeting specified by audio-id 12 that is refined by day-of-week - Play 12:Tuesday.

Additional capabilities in this area provided by IPmedia are Sequences and Aliases.

- Sequences are audio-ids that refer to a 'sequence' of other audio references. For instance audio-id 11 might refer to audio-id 7, 8, and 9.

- Aliases are alphanumeric references that can be used as an alternative to the local storage index.

IPmedia provides a rich set of Audio Variable capabilities. Audio variables are items such as directory numbers, dates, time, currency amounts that are constructed by the media server at execution time. The protocol passes in a numerical value (e.g., 032199) as well as the type of variable (in this case, date) and the media server composes the phrase "March twenty-first, nineteen ninety-nine".

An example of semantic differences in languages is numbers. In French, for example, '21' is spoken as "twenty-one" (vingt-et-un). The equivalent English spoken number does not include the word 'and'. Similar construction occurs in French for the number '22' through '29'. In German, the construction is "one and twenty". Another example is the differing use of plurals in currency units. The semantic structure for variable announcements is captured as 'rules' within the media server software on a per language basis. The media server uses these semantic rules and the value of the variable to construct and insert the correct phrase segments.

Even when digits are spoken correctly from a semantic perspective, if the inflection of all the digits is identical, the resulting phrase can sound very unnatural or robotic. While it may be impossible to capture all the stress and inflection nuances of audio variables, especially without knowing the content of any preceding or following context, some relatively basic treatment of inflection produces a much more natural sounding phrase. In many cases, the inflection rules scope is fully contained with the variable. For example, in English, when a string of digits is spoken, there is typically a rising inflection on the first digit, followed by flat inflections on the intermediate digits, and a falling inflection on the final digit. Again, these rules are language specific. For example, in French, a string of digits would be spoken with a flat inflection for the initial and intermediate digits and a rising inflection on the final digit.

Audio variable support is provided for the following 25 languages: Basque, Cantonese, Catalan, Czech, Dutch (two forms: Netherlands and Belgian), English, French, Gallegan, German, Greek, Hebrew, Italian, Japanese, Korean, Malay, Mandarin, Portuguese, Spanish, Tagalog, Thai, Turkish, Vietnamese, Hindi, and Russian.

The following audio variable constructs are provided for:

- Dates: The input form for a date is "yyyymmdd".
- Times: The input form for a time is "hhmm", and is based on a 24 hour clock. The output of a time can be in either a 12 or 24 hour format.
- Durations: The input for a duration is in seconds, and the valid range is from 0 to ~4 billion (2 to the 32nd power).
- Cardinal Numbers: The valid range is +/-2147483647.
- Ordinal Numbers: The valid range is +/-2147483647 where available for the specific language. Many languages don't identify ordinal numbers beyond 50 or a 100.
- Currency: The valid range is +/-2147483647. The value should be in form of the lowest sub-currency, such as cents for U.S. currency.
- Weekdays:
- Months:
- DNs: Valid digits are 0-9. Supported directory number formats are North American dns and strings of digits up to 32 digits. Note: The string of digits is not formatted. In other words, (919) 555-1212 is played as "9195551212" (no pauses).
- Strings: Valid characters are 0-9,a-z, A-Z, *, and #.
- Silence: The input number specifies the duration of the silence in tenths of seconds. The valid range is from 0 to ~4 billion (2 to the 32nd power).

IPmedia can play prompts residing on external HTTP or NFS servers, or resident in local memory. 32MB of resident memory is provided for local storage of audio prompts. Portions of this local memory area can also be allocated to a dynamic cache function for optimization of HTTP or NFS file access from remote servers.

The local storage area can be provisioned in a number of ways. The primary audio provisioning tool is the Audio Provisioning Server (APS). This tool provides for uploading audio from a web browser, storage and categorization, per media server assignment of audio, and manual or automatic delivery of audio to the media server. Delivery of new audio can be made to the media server without service interruption if both the new and old audio bundles fit into the 32MB storage area simultaneously. If a new bundle does not fit in the remaining space left by the existing bundle, a reset is required to update the audio.

In addition to the APS, audio bundles can be built with a supplied PC based tool called DConvert. However, DConvert does not support the ability to create various abstract audio references, including Sequences, Sets, Aliases, and Audio Variables. As a result, requests to play packaged prompts are made by their index only.

Bundles built with either DConvert or the APS can also be downloaded via "Automatic Update", where an audio bundle can be downloaded from a remote web, NFS, or ftp server based upon configuration. Refer to the appropriate section of this user's manual for a discussion regarding the use of Automatic Update.

Additionally, as directed by a Media Gateway Controller, the IPmedia can also play or record audio files located on remote HTTP or NFS servers.

7.2.1.1 Segment Description Matrix

Table 7-1: Supported Segment Descriptor Elements

Segment Descriptor Element	Supported
http://localhost/ URI	Yes
file:///	Yes
file:////	Yes
file://localhost/	Yes
http:// URI	Yes
file:// URI	Yes
ftp:// URI	No
standalone variables	Yes
embedded variables	Yes
nfs://URI	Yes

Table 7-2: Segment Descriptor Variables

Variables			
Name(TD51/H.2489)	**Subtype**	**Definition**	**Supported**
dat/date	Note – The 'dat' variable does not support subtypes for date. However, if a subtype is specified in the play request it will be gracefully ignored and will instead produce a date announcement according to the rules of each supported language.	Date	Yes
	mdy	Month-Day-Year	No (see note above)
	dym	Day-Year-Month	No (see note above)
dig/digits		Digits	Yes
	gen	Generic	Yes
	ndn	North American DN	Yes(Only in TD-51)

Table 7-2: Segment Descriptor Variables

Variables			
dur	none	Duration	Yes
mth/month	none	Month	Yes
mny/money	<ISO 4217 three letter codes>	Money	Yes
num/int		Number	Yes
	crd	Cardinal	Yes
	ord	Ordinal	Yes
sil	none	Silence	Yes
str/chars	none	String	Yes
tme/tod		Time	Yes
	t12	Twelve hour format	Yes
	t24	Twenty four hour format	Yes
wkd/dow	none	Weekday	Yes

7.2.2 Working with Audio Bundles

7.2.2.1 Audio Bundles

As described in the previous section, voice prompts can be played from the local memory where they are stored as Audio Bundles. An audio bundle is composed of a *.dat* file and an *.xml* file containing the information to properly parse the *.dat* file. Audio bundles are created through the APS and are then stored on a server supporting NFS or HTTP.

7.2.2.2 Configuring the Blade

The audio bundle can be uploaded using either FTP, NFS or HTTP. For more information see Automatic Update Facility on page 35.

In order to upload a voice bundle to the blade, the following *ini* file parameters should be set:

```
APSEnable = 1
AMSProfile = 1
VpFileUrl = 'url-dat-file/dat-file'
APSSegmentsFileUrl = ‘url-xml-file/xml-file’'
```

Where *url-dat-file* / *url-xml-file* relate to the location of the relevant *.dat* and *.xml* files and *dat-file / xml-file* relate to the actual files.

For example:

```
VpFileUrl = 'http://10.50.2.1/dat files/vp.dat'
APSSegmentsFileUrl = 'http://10.4.2.5/segments/segments.xml'
```

For more information, refer to the System Parameters tables.

7.2.2.3 Uploading Methods

A bundle can be uploaded to the blade using three different methods:

1. Setting relevant parameters as described in “Configuring the Blade” above and resetting the blade (hard reset). Optionally, a user may configure parameters via the Web or SNMP interface, burning parameters to Flash and then resetting the blade via the Web or SNMP interfaces (soft reset).
2. Adding the following *ini* file to periodically upload the *.dat* and *.xml* files:

```
AutoUpdateFrequency = 100 //
```

In this case updating will be done every 100 minutes. For more information, refer to System Parameters on page 134 and Automatic Update Facility on page 35.

3. Using SNMP to trigger an immediate upload of the files by setting *“acSysActionSetAutoUpdate”* to true. For more information refer to Using SNMP-based Management on page 64.

Note: When uploading files via HTTP, if the names of the file already loaded and the file intended to be uploaded are the same, time stamps of the old file and the new file should be different.

7.2.2.4 Force Repository Update

Assuming a user has loaded two audio bundles to the blade, a long voice prompt is being played from the first audio bundle (the old audio bundle). At the same time the user wants to upload a new audio bundle. Since two audio bundles have already been uploaded to the blade, the old audio bundle should be replaced with the audio bundle the user wants to upload. Since the voice prompt is being played from the old audio bundle, such an operation will normally fail. Only after the voice prompt has finished playing, may the user upload a new voice bundle.

A new *ini* file parameter has been added to enforce a replacement of a bundle regardless of voice prompts being played from it:

```
AMSForceRepositoryUpdateEnabled = 1
```

For more information, refer to Advanced Audio Server Parameters on page 215.

7.2.2.5 Notifying the Users

Users can be notified on the outcome of an operation in two ways:

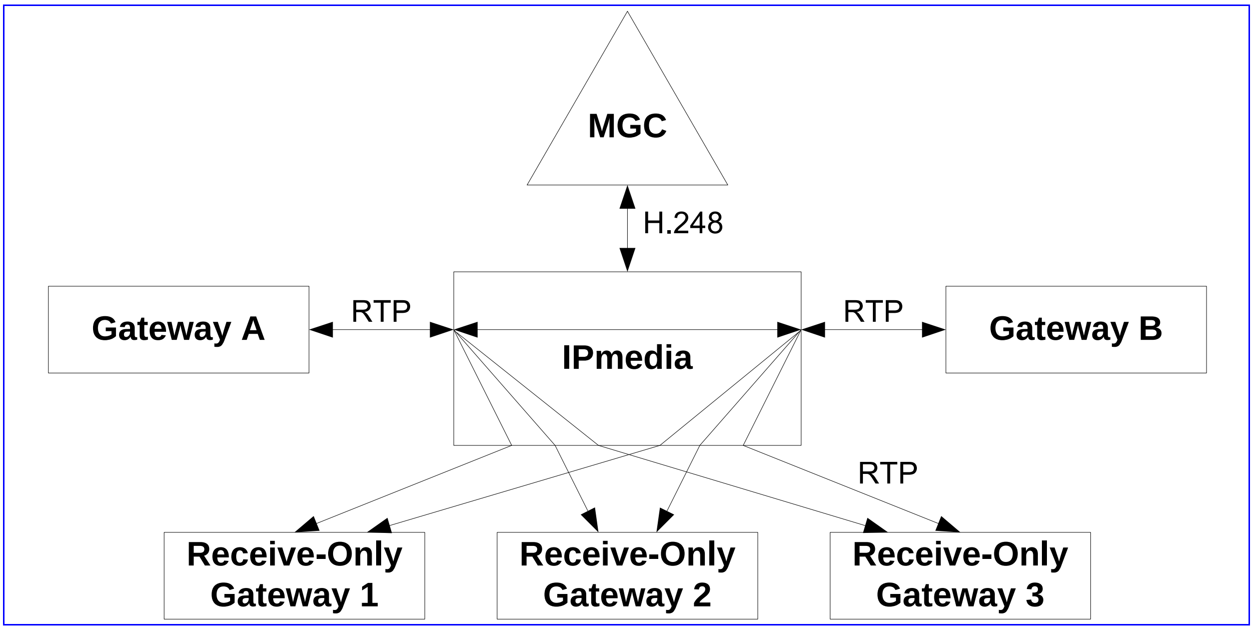
1. Syslog messages – Informative Syslog messages are supplied when the operation has succeeded or failed. On operation failure, the user should always resort to first analyzing those messages.
2. SNMP traps - Similar messages are also supplied via SNMP traps. For more information refer to SNMP Traps on page 95.

7.2.3 Bearer Channel Tandeming

The Bearer Channel Tandeming (BCT) function provided by the Advanced Media Server (AMS) allows for the contents of a VoIP bearer channel to be replicated to multiple additional receive-only destinations. A key attribute of this function is to have it be transparent to the monitored parties. To this end, no jitter or transcode processing is done to eliminate any latency or signal distortion.

Through MEGACO directives from a Media Gateway Controller (MGC), each link in the BCT configuration is created or destroyed. For illustration, suppose that three additional (receive-only) gateways are intended to receive the RTP packet stream established between Gateway A and Gateway B. Each of these three gateways is provided two RTP streams that consist of the audio stream (i.e. RTP packets) from Gateway A and the audio stream from Gateway B. This can be shown in the figure below.

Figure 7-1: Basic BCT Call



To implement this basic BCT call the following H.248 commands are sent by the Media Gateway Controller (MGC) to the device.

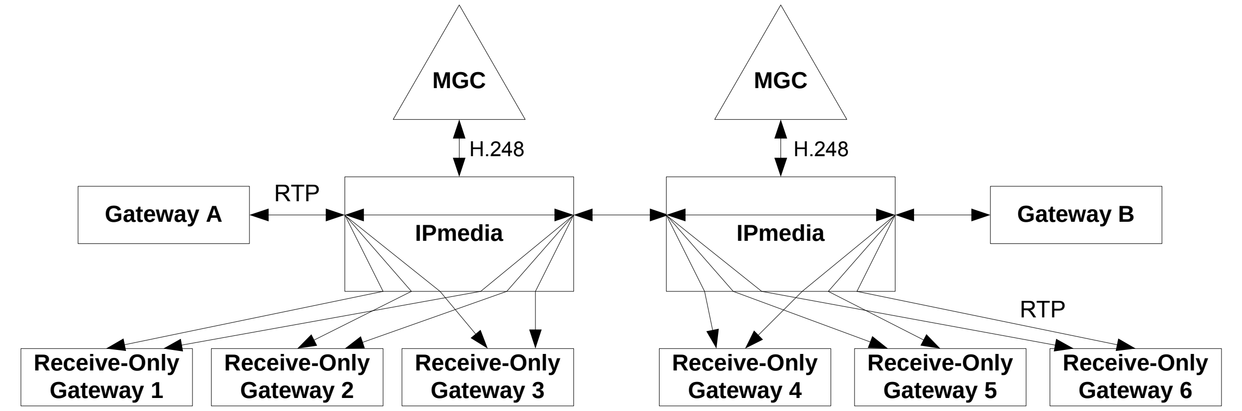
- Reserve an H.248 context for the BCT call.
- Perform eight ephemeral add operations for BCT endpoints. These commands establish all of the external endpoints and set the topology for each of them as 'isolated'. (The endpoints created consist of one endpoint for Gateway A, one endpoint for Gateway B, and six endpoints (send-only) for the three receive-only gateways.)

Note that the eight ephemeral endpoints may not contain the remote SDP. If the remote SDP is not contained within the command to add the ephemeral, it is followed by a modified one that contains the remote SDP.

- One 'two-way' link command to bridge Gateway A and Gateway B together.
- Three 'one-way' link commands for enabling the sending of packets from Gateway A to each of the receive-only gateways.
- Three 'one-way' link commands for enabling the sending of packets from Gateway B to each of the receive-only gateways.

It is possible that the interface connection on both Gateway A and Gateway B may independently require BCT directed towards separate sets of receive-only gateways. In this case, there is two BCT session established on the device involved in the call. In this scenario, one BCT session is unaware of existence of the other BCT session. From the standpoint of the device, this example resembles the following diagram shown in the figure below. Note that the Media Gateway Controller contains all intelligence of how the BCT sessions are interconnected (i.e. the call topology). There is nothing maintained on the device.

Figure 7-2: Independent BCT Support Required by both Call Ends

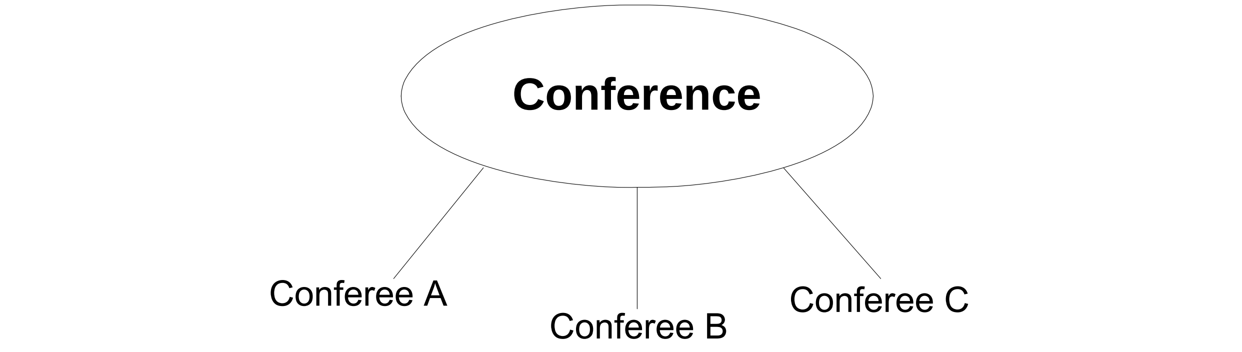


7.2.4 Conferencing

As part of the AMS feature package, the device supports conferencing in which a Media Gateway Controller can dynamically create, delete, or modify conferences up to the conference and conferee limits. Using conference resources that are DSP-based means that conference capacity is deterministic in terms of the amount of traffic that can be handled by the media server as well as the number of ports that can be supported.

Audio input is taken from each conferee on the conference and is mixed with that from the other members of the conference such that the conference audio output to each conferee includes everyone but the conferee. As an example, assume for simplicity the conference model of the following diagram below.

Figure 7-3: Conference Mixing



Conferees A, B, and C represent different audio streams into the conference. The returned audio output streams to each conferee in this example can be defined as follows:

- Output stream to conferee A = input from conferee B + input from conferee C.

- Output stream to conferee B = input from conferee A + input from conferee C.
- Output stream to conferee C = input from conferee A + input from conferee B.

The Device supports Active Speaker Notification service (ASN) which enables receiving a report of the current active speakers in the conference. When enabled , the report will be issued for every change in the conference active speakers. The minimum interval between reports can be configured.

Through MEGACO based directives a conference is established. In the successful establishment of a conference, a unique identifier is returned to the Media Gateway Controller, which allows for the unique identification of the conference in subsequent directives. Through additional directives the conference is controlled (e.g., delete the conference, modify the size of the conference, play audio into the conference, or audit the conference) as well as provide individual conferee operations, such as add to a conference, delete from the conference.

7.2.5 Test Trunk Support

As part of the AMS feature package, the device supports GR-822 based test trunks when inter-working with TDM trunk gateways as deployed in the North American market. This support is through the DSP-based functionality that is resident on the media server. This functionality is directed by a MEGACO-based Media Gateway Controller based upon a proprietary package developed in collaboration with Nortel Networks.

Test Line Tests (TLTs) are used to test PSTN (Public Switch Telephone Network) trunk connections to adjacent switching offices, both local and toll. TLTs are run under the control of the originating office, often without human intervention at the terminating office, and can be used to test both the originating and terminating ends of a TDM trunk. A number of standard tests are documented in the Telcordia (formerly BellCore) document GR-822 (Network Maintenance: Access and Testing - Switched Circuits and Public Packet Switched Network). The following is a brief description of the TLTs as provided by the media server.

- TL100

The TL100 Test Line, also known as the quiet or balanced termination, provides for far-to-near end transmission loss and noise measurements.

- TL102

The TL102 Test Line, also known as the Milliwatt test line, provides far-to-near end transmission loss measurements.

- TL105

The TL105 Test Line is a group of tests that provide for two-way trunk testing controlled through the originating side office that allows for the measurement of transmission loss, noise, and loss with self check. A subset of TL105 trunk tests is supported. The subset includes two-way loss measurement as well as noise measurement from both directions. Tests are executed under the direction of the originating office, also known as the test director.

- T904

The T904 test is a trunk test specific for Israel. IPmedia provides support only for the responder portion of the trunk test.

- TSWAP

The tone swap test allows for the continuous exchange and validation of Milliwatt tone on a trunk facility between two offices. Note that a transmission loss measurement is taken against the received Milliwatt tone with the results being relayed to the Media Gateway Controller.

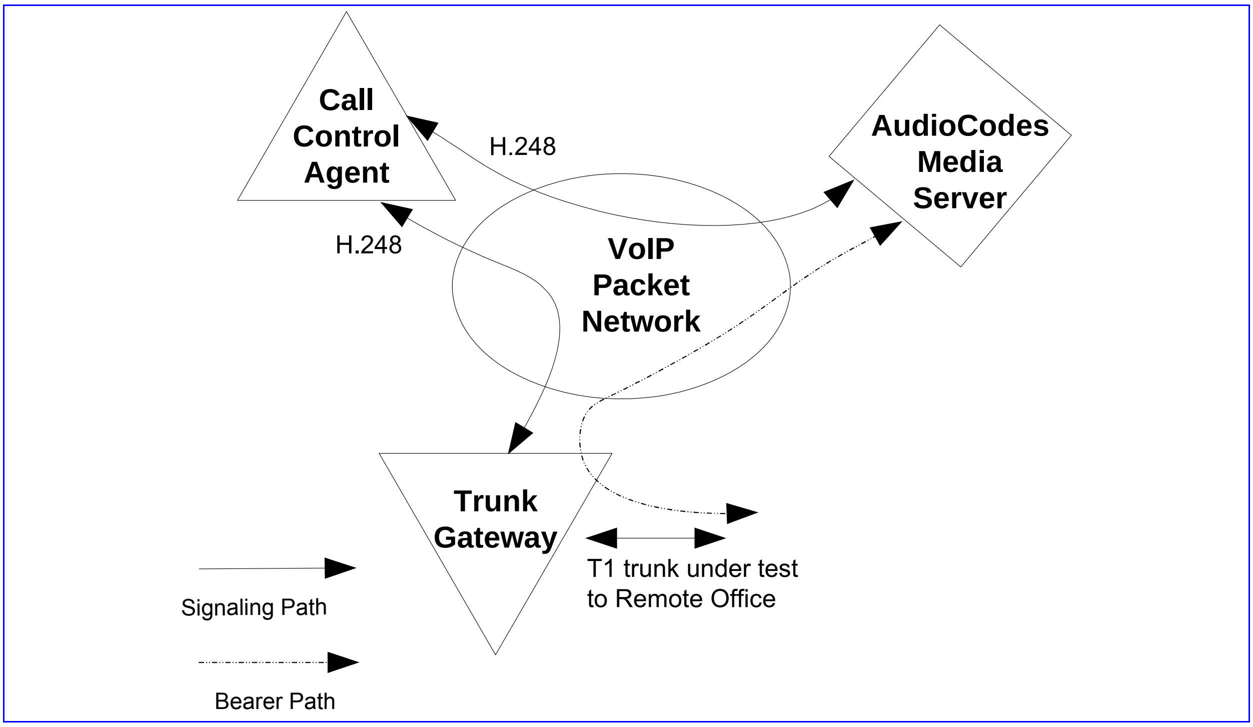
Digital trunks interconnect a trunk gateway and the PSTN and consist of multiple timeslots carrying digitized voice traffic, multiplexed at a T1 or higher rate. Individual timeslots correspond to 'trunk circuits' in the trunk database of the call agent. These circuits may connect directly to a Telco end office, to an access tandem switch, or over inter-machine trunks to other offices with further interconnects to long distance carriers and/or end offices. They may or may not transit a digital cross connect system (DCS) with T1 or DS0 access. Signaling on these trunks may be in-band (using seizure, wink and MF or DTMF digits) or out-of-band (using ISDN or the SS7 packet signaling network for call setup).

The AMS provides a single, consistent methodology for the testing of all trunks, whether at the T1 or higher rate, whether or not a transit through a DCS occurs, and whether call setup occurs in-band or out-of-band.

The Telcordia Remote Office Test Line (ROTL) tests are performed using automated test equipment at the Tandem Office (Class IV), which connects to a test trunk on the co-located switch matrix or on a trunk gateway. For Class IV applications, the test equipment is directed to select individual timeslots for testing using a maintenance dial plan specific to the interfacing switch or gateway. This allows a consistent test method for all interconnect trunks, regardless of rate (T1, etc.), signaling type (in-band or out-of-band), or presence or type of DCS. For the end office application (Class V), the test equipment must terminate the tandem switch originated trunk test with a Telcordia/Bellcore test line.

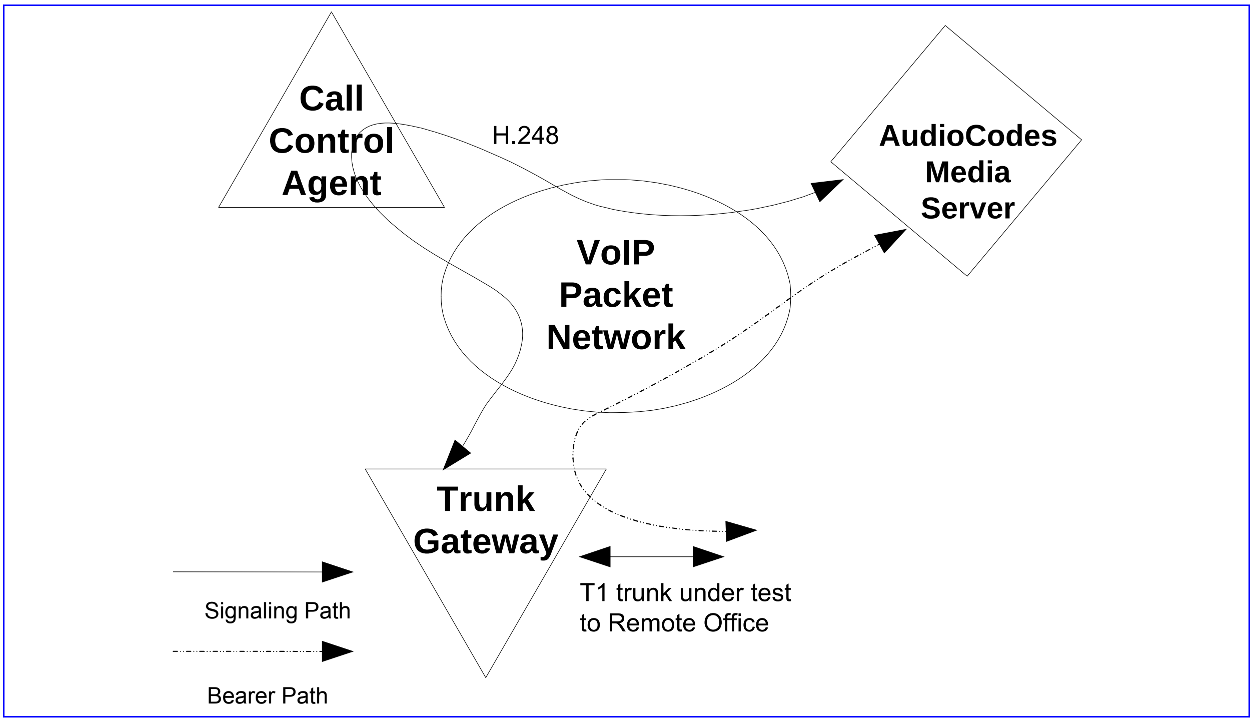
The following two diagrams below show the network connections plus the signaling and bearer paths for the device based TDM trunk testing. Note that in this configuration the network supports the exchange of call control messaging between the Media Gateway Controller of the call control agent and the device as well as the exchange of bearer messaging between the device and the remote trunk gateway. Note that the diagrams illustrate the two types of trunk tests configurations, originating (shown in the figure, ‘Originating Test Trunk Operation’ below) and terminating, better known as responder tests, (shown in the figure, ‘Terminating Test Trunk Operation’ below).

Figure 7-4: Originating Test Trunk Operation



An originating trunk test can be invoked in one of two different ways in terms of the call control agent; either as a manual action by a craft person seated at a maintenance console or in the form of an Automatic Test Trunk feature, which executes periodic trunk tests usually during the maintenance window associated with the office. In either case, the device provides the trunk test results back to the call control agent as an H.248 message.

Figure 7-5: Terminating Test Trunk Operation



A trunk test can be the result of a request by a far end office. In this situation, the local office simply acts as a terminator (or responder) to the requested trunk test. Unlike the originating trunk test, test results are not available for collection from the terminator for a requested trunk test; hence there is no H.248 message from the device to the call control agent containing any results.

7.2.5.1 General Operation

- Bearer Setup

  The test trunk capability on the device is managed through a VXML-based script. This script is not involved in connection setup or tear down. The script is invoked by the MEGACO task on the device only after the Media Gateway Controller establishes the bearer path. The Media Gateway Controller takes down the bearer connection after the VXML script has completed.

- Invocation

  The Media Gateway Controller invokes an originating test by telling the device to apply a proprietary H.248 'nnttrk' package signal to a termination. As the device does not provide an auto respond mode, a terminating signal in the 'nnttrk' package is used.

- Operation

  The AMS Test Trunk Feature consists of a single VXML script containing all the originating and terminating tests. As the script runs, it invokes on-device APIs to perform actions such as playing a tone or taking an energy measurement. After the test has finished, the VXML Script exits. The MEGACO task formats the results as required by its controlling Media Gateway Controller and sends the results to the Media Gateway Controller. Terminating tests do not report results to it controlling Media Gateway Controller; they exist solely to support the originating test.

- Unexpected Termination

  If the Media Gateway Controller tells the device to subtract the termination on which the test is running, the MEGACO task terminates the VXML script before it has completed.

- Download to Device

  Much like other applications on the device, the Test Trunk VXML script is converted to binary using the DConvert utility, and the resulting file is downloaded to the device and burned to flash via BootP/TFTP. All trunk tests are implemented in a single file for simpler management and control through the conversion and download process.

7.3 Video Functionality

AudioCodes' platform provides rich video services. Based on field-proven Audio technology, the video platform with its new video mezzanine, provides a full multimedia processing platform. The platform provides capabilities in three different areas:

- Transcoding - which is used to overcome differences between two video streams
- Conferencing - which provides a mixing of several video streams into one combined stream
- RTP Streaming - which enables playback of multimedia clips that reside on a remote HTTP server.

All these capabilities are combined to create a variety of user applications.

The basic building block for all user applications is the video channel. The fundamental video channel properties provided by the video platform, are summarized in the following table:

Table 7-3: Video Channel Properties

Property	Description
Video Codecs	MPEG-4 Simple profile H.263 Baseline profile H.263-1998 Baseline Profile H.264 Baseline profile
Video RTP Packetization	RFC 2190, RTP Payload Format for H.263 Video Streams RFC 2429, RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+) RFC 3016, RTP Payload Format for MPEG-4 Audio/Visual Streams RFC 3984, RTP Payload Format for H.264 Video
Video Resolutions	CIF (352 x 288) QCIF (176 x 144)
Frame Rates	Up to 30 frames per second (fps)
Bit Rates	Up to 384 kilobits per second (kbps)

7.3.1 SDP (Session Description Protocol) Video

7.3.1.1 Video Channel Configuration

The video channel is configured using SDP. The SDP body should conform to the following:

1. The media type for video streams should be video.
2. Multiple media lines should be supported as one line for audio and one for video.
3. The format of the video must be one of the platform's supported video coders (H.263, H.263-1998, H.264, MPEG-4).
4. a=fmtp should be supported according to the standards presented below.

7.3.1.2 H.263 & H.263-1998 Coder

H.263 packetization layer can be defined with either RFC 2429 or RFC 2190. The MIME media types "H263-1998" or "H263" string should be mapped to fields in the SDP as follows:

1. The MIME type "video " is entered in the SDP media line ("m=") as the media name.
2. The MIME subtype "H263-1998" or "H263" is entered in the SDP "a=rtpmap" line as the encoding name.
3. The optional parameter "rate" is entered in the "a=rtpmap" line as the clock rate and must be set to 90000.
4. The optional parameters "picture size"="MPI" are entered in the "a=rtpmap" line as the resolution, and their corresponding minimal picture interval, the terminal is willing to receive.

   The Minimum Picture Interval (MPI) is equal to 30 divided by the frame rate.

   When several combinations of picture size and MPI are offered they should be separated by a semicolon. Parameters offered first are the most preferred picture mode to be received.

H.263-1998 SDP Sample

```
m=video 49170/2 RTP/AVP 80
a=rtpmap:80 H263-1998/90000
a=fmtp:80 CIF=2; QCIF=1
```

H.263 SDP Sample

```
m=video 49170/2 RTP/AVP 34
a=rtpmap:34 H263/90000
a=fmtp:34 CIF=2; QCIF=1;
```

7.3.1.3 MPEG-4 Coder

MPEG-4 packetization layer is defined using RFC 3016. The MIME media type "MP4V-ES" string should be mapped to fields in the SDP as follows:

1. The MIME type video is entered in the SDP "m=" line as the media name.
2. The MIME subtype MP4V-ES is entered in the SDP "a=rtpmap" line as the encoding name.
3. The optional parameter "rate" is entered in the "a=rtpmap" line as the clock rate and must be set to 90000.
4. The optional parameter "profile-level-id" is entered in the "a=fmtp" line to indicate the coder capability. The simple profile is only one profile supported so the offer

answer SDP model should verify the profile-level-id value, which should be in the range of <1-4>.

Translation between simple profile levels to resolution, frame rate and bit rate conform to the MPEG-4 standard and is described in the table below.

Table 7-4: Profile Levels Translation

Level	Resolution	Frame Rate (fps)	Bit Rate (kbps)
0	QCIF	15	64
1	QCIF	15	64
2	CIF	15	128
3	CIF	30	384

MPEG-4 SDP Sample

```
m=video 49170/2 RTP/AVP 81
a=rtpmap:81 MP4V-ES/90000
a=fmtp:81 profile-level-id=0
```

7.3.1.4 H.264 Coder

The H.264 packetization layer is defined using RFC 3984. The MIME media type video/H.264 string is mapped to fields in the SDP as follows:

- The MIME type video is entered in the SDP "m=" line as the media name.
- The MIME subtype H.264 is entered in the SDP "a=rtpmap" line as the encoding name.
- The optional parameter "rate" is entered in the "a=rtpmap" line as the clock rate and must be set to 90000.
- The optional parameter "profile-level-id" is entered in the "a=fmtp" line to indicate the coder capability. Profile-level-id is a hexadecimal representation of 3 fields: profile_idc, profile-iop, level_idc. The following table specifies the valid configuration for the supported profile levels.

Table 7-5: Supported Profile Levels

Profile level	profile_idc / profile-iop	level_idc
Baseline level 1	profile_idc = 0x42 or constraint_set0_flag = 1	0x0A
Baseline level 1b	profile_idc = 0x42 and constraint_set3_flag = 1	0x0B
Baseline level 1.1	(profile_idc = 0x42 or constraint_set0_flag = 1) and constraint_set3_flag == 0	0x0B
Baseline level 1.2	profile_idc = 0x42 or constraint_set0_flag = 1	0x0C

Translation between baseline profile levels to resolution, frame rate and bit rate conform to the H.264 standard and is described in the table below.

Table 7-6: Baseline Profile Levels Translation

Level	Resolution	Frame Rate (fps)	Bit Rate (kbps)
1	QCIF	15	64
1b	QCIF	15	128
1.1	QCIF	30	192
1.2	CIF	15	384

Note: The specified translation describes the video platform's transmitted channel configuration for Level 1.1. According to the H.264 specification, when working with Level 1.1 the video channel is able to receive streams configured up to QCIF 30 fps or configured with CIF 7.5 fps.

Note: The specified translation describes the video platform's transmitted channel configuration for Level 1.2. According to the H.264 specification when working with Level 1.2, the video channel is able to receive streams configured up to QCIF 30 fps or CIF 15 fps.

The optional parameter "packetization-mode" is entered in the "a=fmtp" line to indicate the properties of an RTP payload type or the capabilities of a receiver implementation. Possible values are:

- 0 – When packetization-mode is not present – The single NAL mode.
- 1 – The non-interleaved mode.

H.264 SDP sample

```
m=video 49170 RTP/AVP 82
a=rtpmap:82 H264/90000
a=fmtp:82 profile-level-id=42A00b; packetization-mode=0
```

7.3.2 Bandwidth Control

Bandwidth control is performed using the *b* SDP parameter. This parameter is applicable to each of the media streams and defines the maximum bit rate used in this stream. As defined in RFC 2327, *b* line is made up of the modifier and bandwidth values. The video platform only supports the Application Specific Maximum (ASM) bandwidth modifier.

```
m=video $ rtp/avp 80
a=rtpmap:80 H263-1998/90000
a=fmtp:80 CIF=1
b=AS:256
```

When working with video coders that use profile levels to define their configuration (e.g. H.264 and MPEG-4), the bandwidth is calculated from the profile level according to the above tables. There is no need to use the *b* parameter in the SDP unless the required bandwidth is less than the bandwidth defined by the profile level.

For video coders that don't have a profile level mechanism, it is important to use the bandwidth parameter. Otherwise the video platform's default will be selected.

7.3.3 Video Transcoding

Video transcoding is used to bridge video configuration differences between two terminations. Under the umbrella of video transcoding, it is possible to find conversions with respect to four configuration aspects. In all four cases, both streams are fully decoded and re-encoded with the new configuration.

- Coder transcoding - This is used when two terminations are using two different video coders - e.g. H.264 and MPEG-4.
- Resizing - The picture generated by each termination should either be stretched or squeezed to adapt to the other termination required resolution.
- Trans-rating (bit rate reduction) - Reducing video stream bandwidth is possible if one termination would like to send multimedia to a termination with a lower bandwidth. Please note that only bandwidth reduction is possible. Bandwidth cannot be raised.
- Frame rate reduction - Reducing frames is useful when one termination is generating a stream with more frames needed by the other termination.

The following example demonstrates all four transcoding possibilities. Termination A is configured to work with H.263, CIF, 30FPS and 384BPS. Termination B is configured to work with H.264, QCIF, 15FPS and 128BPS.

```
MEGACO/1 [10.4.153.200]:2944
Transaction = 140 {
Context = $ {
Add = $ {
          Media {
                ST=1{
                      LocalControl {Mode = SendReceive},
                      Local {
                            v=0
                            c=IN IP4 $
                            m=video $ rtp/avp 80
                            a=rtpmap:80 H263-1998/90000
                            a=fmtp:80 CIF=1
                            b=AS:384
                      },
                      Remote {
                            v=0
                            c=IN IP4 <TERMINATION A IP>
                            m=video 4004 RTP/AVP 80
                            a=rtpmap:80 H263-1998/90000
                            a=fmtp:80 CIF=1
                            b=AS:384
                      }
                }
          }
},
Add = $ {
          Media {
                ST=1{
                      LocalControl {Mode = SendReceive},
                      Local {                            v=0
                            c=IN IP4 $
                            m=video $ rtp/avp 82
                            a=rtpmap:82 H264/90000
                            a=fmtp:82 profile-level-id=42F00A;
                             packetization-mode=1
                      },
                      Remote {
                            v=0
                            c=IN IP4 <TERMINATION B IP>
                            m=video 4014 RTP/AVP 82
```

```
                           a=rtpmap:82 H264/90000
                           a=fmtp:82 profile-level-id=42F00A;
                           packetization-mode=1
                     }
               }
         }
}
}
}
```

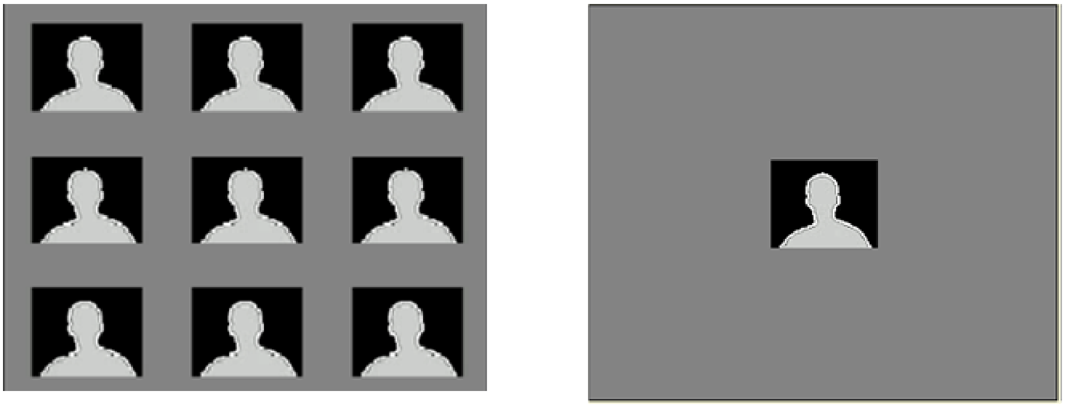
7.3.3.1 Audio Video Synchronization

Audio and video synchronization, also known as lips-sync, is maintained for all multimedia calls that contain audio and video streams. Synchronization is achieved using the RTCP sender report mechanism. RTCP streams must be active to create synchronization.

7.3.4 Video Conference

Video Conferencing can be divided into three basic types of operation:

- Voice Activated Switching (VAS) allows each participant in the conference sees a full view of the current active speaker. The current active speaker sees a full view of the previous active speaker.
- Continuous Presence (CP) allows the conference participants to see more than one participant in the view.
- Hybrid mode is a combination of the above two modes. Depending on the layout a participant selected, the participant will see the last X active speakers. For example, when using a 2x2 tiles layout with a hybrid mode, the participants sees the last four active speakers. Hybrid mode can be used for conferences with more participants than the required layout.



Continuous Presence / Hybrid

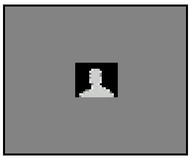
Voice Activated Switching

AudioCodes' video platform defines a number of basic building blocks that enable the controller to create a required conference view for each participant.

7.3.4.1 Participant Screen Layout

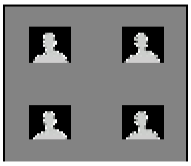
- Full screen layout

  When selecting a full screen layout, only one participant is seen. This layout is best suited for voice activated, triggered switching or timed triggered switching.



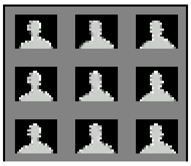
- 2x2 layout

  When a participant would like to see up to 4 other participants, a 2x2 layout should be selected. This layout is suitable for conference with up to 5 participants.



- 3x3 layout

  When a participant would like to see up to 9 other participants a 3x3 layout should be selected. This layout is suitable for conferencing with up to 10 participants.



Selecting screen layouts that are different from the full screen layout, will cause the output picture generated by the conference mixer to be with CIF resolution. Even though the 2x2 layout output is constructed of four combined QCIF pictures, it is preferred that the conference participants input stream will be using the QCIF resolution, thereby economizing on bandwidth. A participant's terminal equipment must be able to send in one resolution and receive in another. If the participant's terminal equipment does not support different receive / transmit resolutions, the stream resolution for both receive and transmit must be CIF.

7.3.4.2 Participant View Switching Triggers

A video conference participant may configure the conference output view, depending on the view switching trigger. Switching triggers can be either dynamic (like Active speaker and Time-triggered) or static (like Fixed mode).

- Active speaker triggered switching

  A participant's view will be changed according to the active speakers' detection. Depending on the number of users a participant wants to view (see screen layout), the user sees the last active speakers. Active speaker detection is done using the Audio Processing DSPs on the video platform. Therefore, a participant must also have an audio stream to be able to be selected as an active speaker. The active speaker detection must also be enabled either through control or management interfaces.

  The most classical example for this triggering method is the VAS described above. In VAS, a participant selects to view only one participant which is changed according to the current active speaker. Another possible use for active speaker triggering is having more participants than the layout supports. For example, a participant screen layout might be 2x2 layout while the number of participants in the conference is six. In that case the participant view will contain the last four active speakers.

- Time-triggered switching

  Working with this method requires the selection of a full screen layout. A participant's view is rotated every configured time among the different video participants in the conference. The configured time must not be zero and should correspond to the maximum intra-frame resolution of all the participants in the conference.

- Fixed mode – no trigger

  A participant may select which participants are to be viewed in each tile of their screen layout. The view is maintained throughout the whole conference session. In order to be able to use this mode, each conference participant must have a unique identity.

For all triggers that denote a view change, it should be clear that for the conference bridge to be able to switch into a new view which includes a participant stream, the conference bridge has to wait for an intra-frame from the included participant stream.

7.3.4.3 Additional Participant View Configurations

The video platform also provides some additional methods to configure a participant's view, besides setting the layout and switching triggers.

- Self-View

  Selecting self view is possible for non-full screen layouts. The self-view can be seen in the top left tile of its layout.

- Highlight active speaker

  Selecting highlight active speaker is possible for non-full screen layouts. When highlight is enabled, a pink frame will enclose the current active speaker tile.

- Hide topology

  Video participants can select which participants are to be hidden from their view. Using the hide configuration, the required topology can be selected.

7.3.4.4 H.248.19 Support

H.248.19 recommendation defines a standard way to create different video conference scenarios. The supported video packages are:

- Voice Activated Video Switch Package (vavsp)

  This package defines the functionality that allows the media processor to determine the mix of a video stream in a conference dependent on the active speaker.

  The volume level for video switching parameter defines decibels ranging from 0 to 100. Though the volume configuration should be done in DBMs, the following conversion should be applied:

```
0 – Master,
[1 - 100] will be mapped to [-39 - …] DBM
```

  All values in all termination should have the same values expect of the master (value 0).

- Lecture Video Mode Package (lvmp)

  This package defines functionality that allows a media processor to change the output video image from a mix of N input video sources every X seconds. For example, a lecture scenario where one user represented by a termination (the lecturer) will see a view of a participant for X seconds, then the next participant for X seconds, etc.

- Contributing Video Source Package (cvsp)

  This package describes a property that allows a media controller to identify the contributing video sources, for a particular video stream. This allows a media processor to mix the input video stream appropriately for output on a particular termination.

The package defines two parameters - input video source in local SDP and the contributing source to output in remote SDP. As with other local SDP parameters, like port and IP address, only $ will be supported for the input video source parameter. In addition, when using the contributing source to output parameter all input video sources must already be known. Therefore these parameters should be configured in different commands.

- Tiled Window Package (vwp)

  This package allows the media controller to order the media processor to display a number of tiled video windows with the same dimensions. Only the above screen layouts are supported.

  This package cannot be used with the voice activated video switch package or the lecture video mode package.

7.3.4.5 H.248 Example

The following scenario demonstrates a conference call with three participants.

Assume the first participant is using active speaker triggered switching with full screen layout, with an output resolution of QCIF.

The second participant is using timed trigger switching with full screen layout. The Participant view change will occur every 5 seconds with an output resolution of CIF.

The third participant is using the fixed 2x2 screen layout, with an output resolution of CIF.

Note that for simplicity reasons only the video parameters are included. Conferences should also include audio (especially if using VAS).

```
MEGACO/1 [10.4.153.200]:2944
Transaction = 140 {
Context = $ {
Add = $ {
          Media {
                ST=1{
                      LocalControl {
                            Mode = SendReceive,
                            vavsp/audsts = 1,
                            vavsp/vidmixbeh = aspasa,
                            vavsp/vollevvidsw = 20
                      },
                      Local {                         v=0
                            c=IN IP4 $
                            m=video $ rtp/avp 82
                            a=rtpmap:82 H264/90000
                            a=fmtp:82 profile-level-id=42E00A;
                             packetization-mode=1
                            a=h248item:cvsp/ivs=$
                      },
                      Remote {
                            v=0
                            c=IN IP4 <FIRST FRIEND IP>
                            m=video 4004 RTP/AVP 82
                            a=rtpmap:82 H264/90000
                            a=fmtp:82 profile-level-id=42E00A;
                            packetization-mode=1
                      }
                }
          }
},
Add = $ {
          Media {
                ST=1{
                      LocalControl {
                            Mode = SendReceive,
                            lvmp/vidswitchint = 50
                      },
                      Local {
                            v=0
                            c=IN IP4 $
                            m=video $ rtp/avp 80
                            a=rtpmap:80 H263-1998/90000
                            a=fmtp:80 QCIF=2
                            a=h248item:cvsp/ivs=$
                      },
                      Remote {
                            v=0
                            c=IN IP4 <SECOND FRIEND IP>
                            m=video 4014 RTP/AVP 80
                            a=rtpmap:80 H263-1998/90000
                            a=fmtp:80 CIF=2
                      }
                }
          }
},
Add = $ {
          Media {
                ST=1{
                      LocalControl {Mode = SendReceive},
                      Local {
                            v=0
                            c=IN IP4 $
                            m=video $ rtp/avp 34
                            a=rtpmap:34 H263/90000
                            a=fmtp:34 QCIF=2
                            a=h248item:cvsp/ivs=$
          },
                      Remote {
                            v=0
                            c=IN IP4 <THIRD PARTICIPANT IP>
                            m=video 4024 RTP/AVP 34
                            a=rtpmap:34 H263/90000
                            a=fmtp:34 CIF=2
                      }
                }
          }
    }
}
}
```

```
MEGACO/1 [10.4.153.200]:2944
Transaction = 140 {
Context = 1 {
    MF = gwrtp/2 {
          Media {
                ST=2{
                      Remote
                      {
                         v=0
                         c=IN IP4 <THIRD PARTICIPANT IP>
                         m=video 4024 RTP/AVP 34
                         a=rtpmap:34 H263/90000
                         a=fmtp:34 CIF=2
                         a=h248item:tilwin/tiledet=2,2,1,1,2,2
                      }
                }
          }
    }
}
}
```

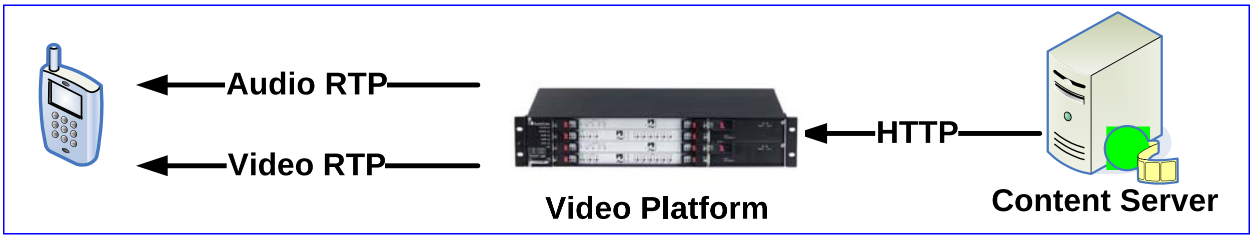
7.3.5 Interactive Voice and Video Response (IVVR)

Extending the regular IVR control, the video platform enables playing of remote multimedia announcements. IVVR available control capabilities include:

- Playing a multimedia announcement (Play)

  The video platform downloads the multimedia announcements from a remote HTTP server and generates two media streams - one for audio and one for video. Users can select to play only one of the streams or both. When playing both audio and video media streams, synchronization (also known as lips synchronization) is maintained between the audio and video streams.

Figure 7-6: Streams Synchronization



Multimedia announcements reside on the content server inside container files. Each container file contains the encoder media data of both the audio and the video streams accompanied with meta-data, which describes the media-data properties. The video platform is capable of playing announcements contained in either *.3gp or *.mp4 file format container types.

7.3.5.1 Audio and Video Supported Coders

Currently the video platform supports only the following streaming coders:
Audio Coders - AMR-NB.
Video Coders - H.263, H.264 and MPEG-4.

7.3.5.2 Streaming and Transcoding

The video platform is capable of transcoding audio and video coders. The user can open a channel with audio or video coders that are different from the played file coders. In this case, the platform will identify the need for transcoding and will send the RTP containing the user’s required coders to the remote side.
Also, the platform is capable of adjusting all fundamental parameters such as resolution, frame-rate and bit-rate between the user’s handset and the played file.

7.3.5.3 Streaming Audio Only or Video Only

The audio and the video streams are independent. Playback of one media type is possible using one of the following:

- Opening an audio channel only or a video channel only. The video platform will play only the relevant media stream.
- Sending a “play” command only to one of the media streams.
- Playing a multimedia file which contains audio only or video only.

7.3.5.4 Play Actions Example

The user application can play a file and stop it. The following commands are MEGACO standard compliant with the H.248.9 recommendation:

Open Video Channel command example:

```
MEGACO/1 [10.31.10.135]:2944
Transaction = 140 {
     Context = $ {
          Add = $ {
               Media {
                    ST=1{
                         LocalControl {Mode = SendReceive},
                         Local {
                              v=0
                              c=IN IP4 $
                              m=audio $ RTP/AVP 64
                         },
                         remote {
                              v=0
                              c=IN IP4 10.31.2.31
                              m=audio 4000 RTP/AVP 64
                              a=rtpmap:64 AMR/8000/1
                              a=fmtp:64 mode-set=7
                         }
                    },
                    ST=2{
                         LocalControl {Mode = SendReceive},
                         Local {
                              v=0
                              c=IN IP4 $
                              m=video $ rtp/avp 81
                              a=rtpmap:81 MP4V-ES
                              a=fmtp:81 PROFILE-LEVEL-ID=3
                         },
                         remote {
                              v=0
                              c=IN IP4 10.31.2.31
                              m=video 4004 RTP/AVP 81
                              a=rtpmap:81 MP4V-ES/90000
                              a=fmtp:81 PROFILE-LEVEL-ID=3
                              a=ptime:20
                         }
                    }
               }
          }
     }
}
```

Play command example:

```
MEGACO/1 [10.31.10.135]:2944 Transaction = 140 {
     Context = 1 {
          Modify = gwRTP/0 {
               Signals {
                    aasb/play {
                      an="sid=<http://10.0.0.1/example.3gp>",
                      ni=TRUE,it=1,iv=0,nc={TO,IBS,IBE,OR}
                    }
               },
               Events = 34 {
                    aasb/audcomp,aasb/audfail,g/sc
               }
          }
     }
}
```

Stop Play command example:

```
MEGACO/1 [10.31.10.135]:2944 Transaction = 140 {
     Context = 1 {
          Modify = gwRTP/0 {
               Signals {
               },
               Events = 34 {
                    aasb/audcomp,aasb/audfail,g/sc
               }
          }
      }
}
```

7.4 Using Push-to-Talk over Cellular (PoC) Media Server

Note: This section on Push-to-Talk over Cellular is only applicable to IPM-6310, IPM-8410 and IPmedia 3000.

7.4.1 PoC Media Server (PMS) Interface Description

The PMS is a PTT User Plane server. Its purpose is to establish half-duplex one-to-many connections for PTT applications. The PMS implementation guideline is in accordance with the OMA PoC Version 1.0 User Plane standard. Yet, the TBCP management is outside the scope of the PMS implementation and should be handled by the PMS controller (CA/MGC). The control over the PMS PTT sessions is made by using the H.248 control protocol. The MGC should use an H.248 Context called a PoC context for each PTT session. The terminations added to the PoC context are regarded as the PoC session participants.

The PMS Media protocol is RTP (RFC 3550). Each participant in the PoC session may either receive or transmit RTP frames but never both at the same time. A PoC participant may also be idle. In this mode the participant is neither receiving nor sending any media packets.

7.4.2 The PoC Context

The PoC Context is a proprietary H.248 Context that supports PoC capabilities. The PoC context characteristic is to make a dynamic one-to-many session with the ability to control the active speaker in the session. This is done in accordance with the OMA PoC User Plane Controlling Server guidelines.

When a PoC session is to be created, the PoC context needs to be allocated. This is done using the Context Reservation Package which is described in details on a section 5.

Once allocated, PoC terminations can be added to the PoC Context. Each participant can be added as either a listener or an active speaker. There can be no more than one active speaker on a PoC Context. A context having all its participants set as Listeners is a valid configuration in between talk bursts silence gaps or pre-established sessions.

Once the floor-speaker changes, the PoC context can be set to have a new active speaker by setting the former active speaker to be a listener and setting the new PoC termination granted by the floor to be the active speaker.

7.4.3 The PoC Termination

The PoC termination is an H.248 termination being added to a PoC Context.

The PoC Termination Stream Mode defines the role of the PTT participant in the PoC session. If the PoC termination's Mode is set to "Receive Only", the PoC termination is regarded as the Active Speaker of the PoC session. If the PoC termination's Mode is set to "Send Only", the PoC termination is regarded as a Listener at the PoC session. The mode can also be set to "Inactive" in order to support Participating Server PoC sessions as described at the PoC Context section. The PoC termination's mode cannot be configured to "Send/Receive". This is due to the half-duplex nature of the PTT application. No two terminations can be configured to "Receive Only" mode in the same PoC Context.

7.4.4 The PoC Events

7.4.4.1 Notification of Last Media Packet

The Media Gateway Controller (MGC) may request the PMS to notify the receipt of a media packet with a specified sequence number. The event will be requested using the *acpoc* proprietary package with Last Media Packet (lmp) Event Id and Sequential Event Descriptor parameter, to specify the requested sequence number. The event will be requested on an ephemeral termination in a PoC context.

The PMS will send a notification upon reception of the RTP packet with the requested RTP sequence number. If the RTP packet with the requested sequence number was already received before the event request from the MGC, the PMS will send the notification immediately.

7.4.4.2 Notification of Unexpected Media Packets

When PMS termination is set to "SendOnly" or "Inactive" mode, the MGC may request the PMS to detect incoming RTP media packets.

In case the PMS termination is in "SendOnly" or "Inactive" mode and receives RTP packets, it will send a notification with the Unexpected Media Packets Event Id, to the MGC.

The event will be requested using the *acpoc* proprietary package with the Unexpected Media Packet (ump) Event Id. The event will be requested on an ephemeral termination in a PoC context.

The event is only reported once. If the MGC wants to check if RTP packets are still coming from this PoC client, it should send the event request to the PMS again.

7.4.4.3 Notification of Stopped RTP Stream (T1 Timer Support)

The MGC may request the PMS to send a notification when RTP packets stopped reaching the Active Speaker termination. This will enable the support of the T1 timer in the MGC,

The event will be requested using the standard H.248's *nt/netfail* event. The event will be requested on the active speaker ephemeral termination in the PoC context.

When a request to detect the network failure event has been received, a timer is set.

The timeout for that timer is configurable by the *BrokenConnectionEventTimeout* provisioning parameter.

The timer is reset each time a media packet arrives at the termination. When the timer expires, the network failure event is notified.

7.4.4.4 Notification of the First RTP Packet (T2 timer support)

The MGC may request the PMS to send a notification when the first RTP reaches the Active Speaker termination. This will enable the support of the T2 timer in the MGC.

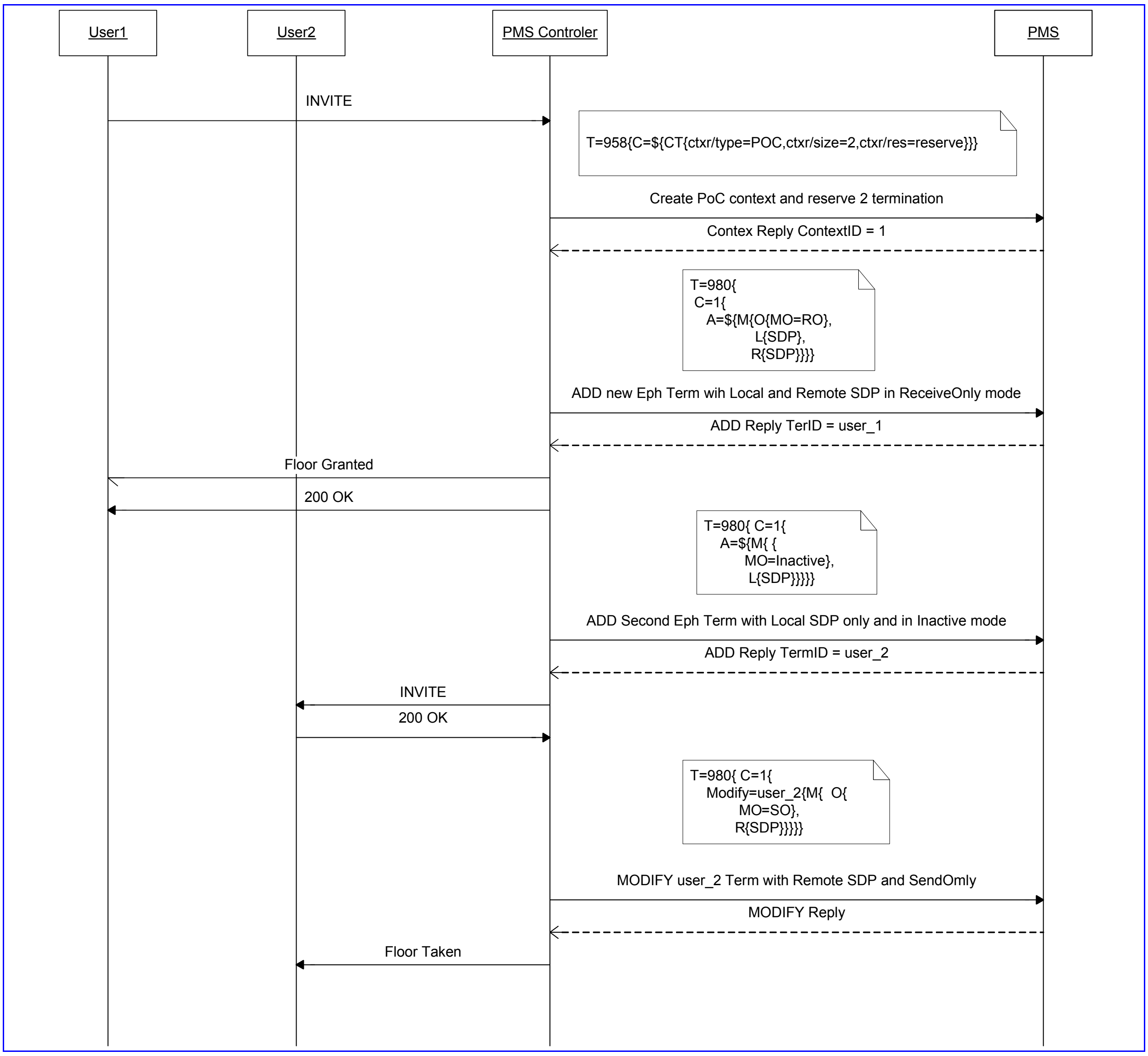
The event will be requested using the *acpoc* proprietary package with the First Media Packet (fmp) Event Id. The event should be requested on the active speaker ephemeral termination (termination in receive only mode) in the PoC context.

Upon receipt of a first RTP packet, the *acpoclfmp* event will be sent to the MGC. The PSC can set the T2 timer upon receipt of the *acpoclfmp* event.

7.4.5 Call Flows

7.4.5.1 Call Establishment with 2 Participants

Figure 7-7: Call Establishment With 2 Participants


User1
User2
PMS Controler
PMS
INVITE
T=958{C=${CT{ctxr/type=POC,ctxr/size=2,ctxr/res=reserve}}}
Create PoC context and reserve 2 termination
Contex Reply ContextID = 1
T=980{ C=1{ A=${M{O{MO=RO}, L{SDP}, R{SDP}}}}
ADD new Eph Term wih Local and Remote SDP in ReceiveOnly mode
ADD Reply TerID = user_1
Floor Granted
200 OK
T=980{ C=1{ A=${M{ { MO=Inactive}, L{SDP}}}}}
ADD Second Eph Term with Local SDP only and in Inactive mode
ADD Reply TermID = user_2
INVITE
200 OK
T=980{ C=1{ Modify=user_2{M{ O{ MO=SO}, R{SDP}}}}}
MODIFY user_2 Term with Remote SDP and SendOmly
MODIFY Reply
Floor Taken

Detailed H.248 messages for above example:

➢ Create a new PoC context and reserve 2 termination resources for this call

```
;; Command
MEGACO/2 [10.4.229.18]:2944
T=958{C=${CT{ctxr/type=PoC,ctxr/size=2,ctxr/res=reserve}}}

;;Reply
MEGACO/2 [10.4.4.32]:2944
P=363{
C=2{
CT {
ctxr/type=PoC,ctxr/size=2,ctxr/res=reserve}}}
```

➢ Add first user with Local and Remote SDP, Receive Only Mode

```
; Command
MEGACO/2 [10.4.229.18]:2944
T=980{
    C=1{
    A=${
    M{  O{
            MO=ReceiveOnly
        },
        L{
            v=0
            c=IN IP4 $
            m=audio $ RTP/AVP 0
            a=ptime:20
        },R{v=0
            c=IN IP4 10.4.4.34
            m=audio 4000 RTP/AVP 0
            a=ptime:20
        }
    },
}}}

;; Reply
MEGACO/2 [10.4.4.32]:2944
P=365{
C=1{
A = te/tepool1/0{
M{
L{

v=0
c=IN IP4 10.4.4.32
m=audio 4000 RTP/AVP 0
a=ptime:20
a=silencesupp:off - - - -
}}}}}
```

➢ Add second user with Local SDP only, Inactive mode

```
;; Command
MEGACO/2 [10.4.229.18]:2944
T=980{
    C=1{
    A=${M{
        O{
            MO=Inactive
        },
        L{
            v=0
            c=IN IP4 $
            m=audio $ RTP/AVP 18
            a=ptime:10
        }
    }
}}}

;; Reply
MEGACO/2 [10.4.4.32]:2944
P=366{
C=1{
A = te/tepool1/1{
M{
L{

v=0
c=IN IP4 10.4.4.32
m=audio 4010 RTP/AVP 18
a=fmtp:18 annexb=yes
a=ptime:10
a=silencesupp:off - - - -
}}}}}
```

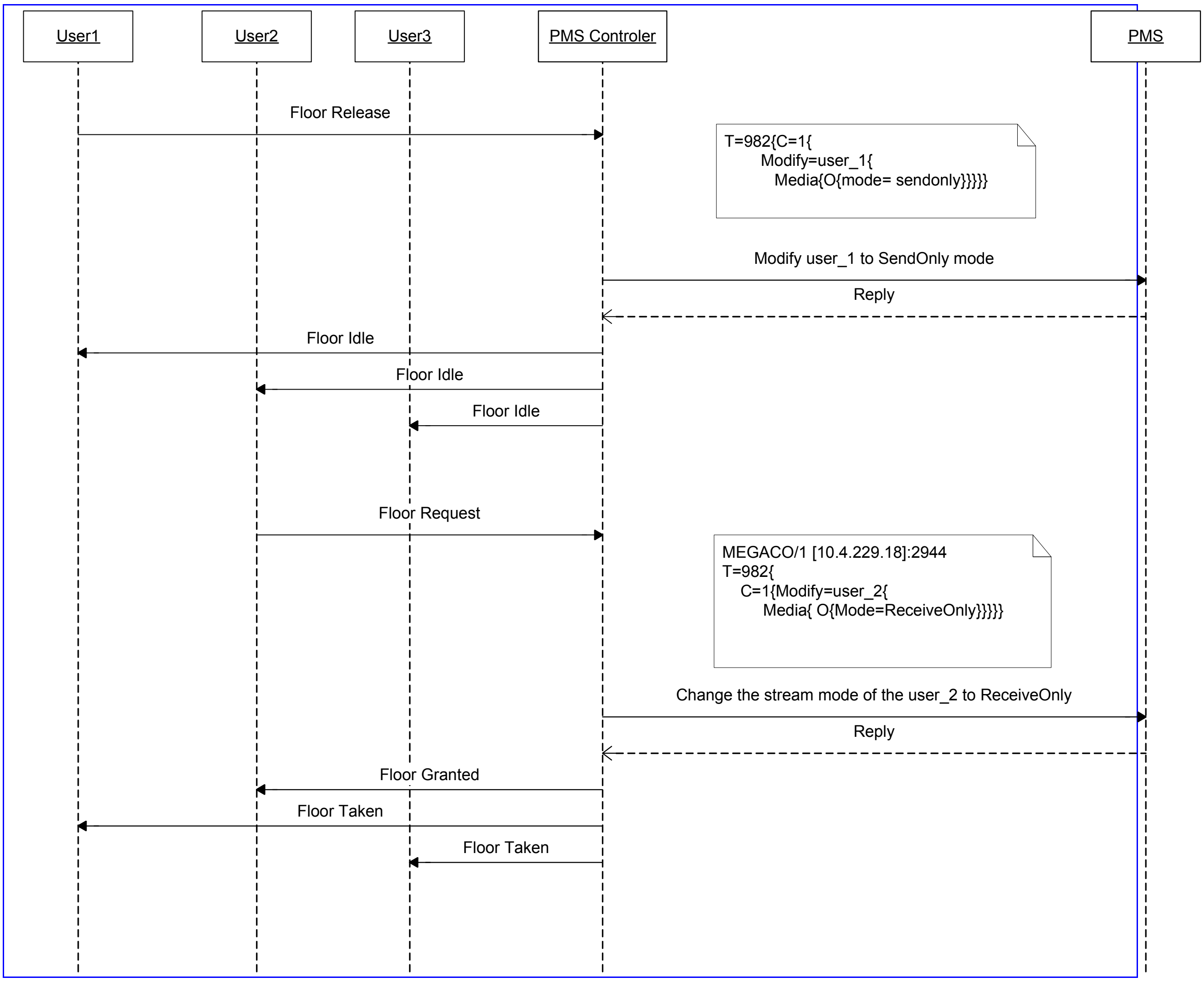
➢ Modify second termination with Remote SDP, Mode Send Only

```
;;Command
MEGACO/2 [10.4.10.226]:2944
T=980{
    C=1{
    Modify=te/tepool1/1{
        Media{
            O{MO=SendOnly},
            R{v=0
                c=IN IP4 10.4.4.34
                m=audio 4010 RTP/ AVP 18
a=fmtp:18 annexb=yes
a=ptime:10
a=silencesupp:off - - - -
            }
        }
    }
}}

;;Reply
MEGACO/2 [10.4.4.32]:2944
P=369{
C=1{
MF = te/tepool1/1
}}
```

7.4.6 Floor Changing

Figure 7-8: Floor Changing



```
T=982{C=1{
      Modify=user_1{
        Media{O{mode= sendonly}}}}}
```

```
MEGACO/1 [10.4.229.18]:2944
T=982{
   C=1{Modify=user_2{
       Media{ O{Mode=ReceiveOnly}}}}}
```

Detailed H.248 messages for above example:

Modify the stream mode of the user_1 to SendOnly

```
;;Command
MEGACO/2 [10.4.10.226]:2944
T=980{
    C=1{
    Modify=te/tepool1/1{
        Media{
            O{MO=SendOnly}
        }
    }
}}

;;Reply
MEGACO/2 [10.4.4.32]:2944
P=369{
C=1{
```

```
MF = te/tepool1/1
}}
```

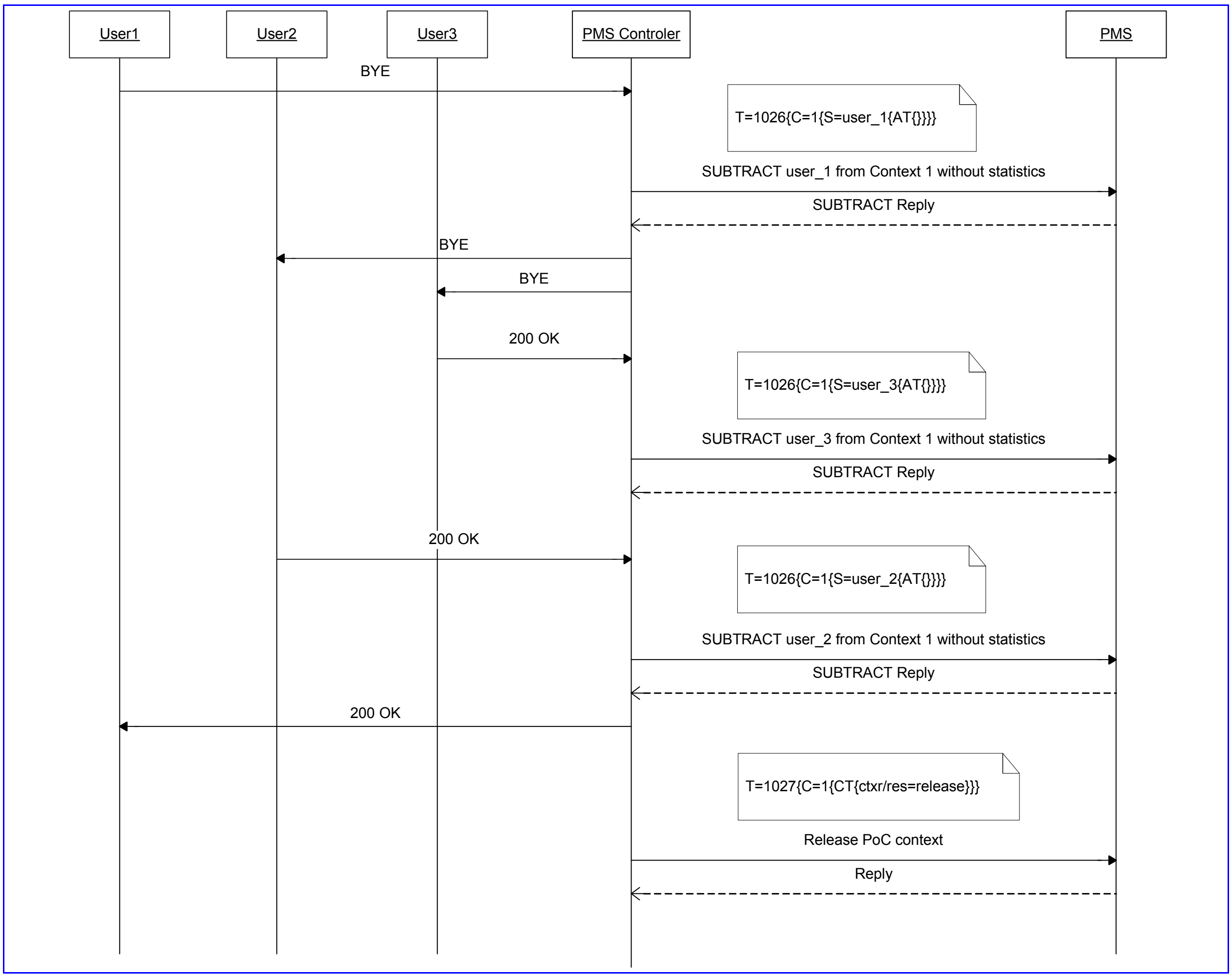
Modify the stream mode of the user_2 to ReceiveOnly

```
;;Command
MEGACO/2 [10.4.10.226]:2944
T=980{
    C=1{
    Modify=te/tepool1/2{
        Media{
            O{MO=ReceiveOnly}
        }
    }
}}

;;Reply
MEGACO/2 [10.4.4.32]:2944
P=369{
C=1{
MF = te/tepool1/2
}}C
```

7.4.7 Call Release

Figure 7-9: Call Release


User1
User2
User3
PMS Controler
PMS
BYE
T=1026{C=1{S=user_1{AT{}}}}
SUBTRACT user_1 from Context 1 without statistics
SUBTRACT Reply
BYE
BYE
200 OK
T=1026{C=1{S=user_3{AT{}}}}
SUBTRACT user_3 from Context 1 without statistics
SUBTRACT Reply
200 OK
T=1026{C=1{S=user_2{AT{}}}}
SUBTRACT user_2 from Context 1 without statistics
SUBTRACT Reply
200 OK
T=1027{C=1{CT{ctxr/res=release}}}
Release PoC context
Reply

Note 1: The subtraction should be done without statistics.

Note 2: Only after release reserved termination from the context (CT{ctxr/res=release}) will the context be freed.

Detailed H.248 messaging example:

➢ Subtract first RTP termination without Statistics

```
;; Command
MEGACO/2 [10.4.229.18]:2944 T=1026{C=1{S=te/tepool1/0{AT{}}}}

;;Reply
MEGACO/2 [10.4.4.32]:2944
P=371{
C=1{
S = te/tepool1/0}}
```

➢ Subtract second RTP termination without Statistics

```
;; Command
MEGACO/2 [10.4.229.18]:2944 T=1026{C=1{S=te/tepool1/1{AT{}}}}

;;Reply
MEGACO/2 [10.4.4.32]:2944
P=371{
C=1{
S = te/tepool1/1}}
```

➢ Release reserved termination from the context only after this command the context will be freed

```
;; Command
MEGACO/2 [10.4.229.18]:2944 T=1027{C=1{CT{ctxr/res=release}}}

;;Reply
MEGACO/2 [10.4.4.32]:2944
P=372{
C=1{
CT {
ctxr/res=release}}}
```

7.4.8 Context Reservation Package

PackageID: ctxr (0x00xx)

Version: 1

Extends: none

This package defines the mechanism by which contexts can be reserved and the context type can be defined.

7.4.8.1 Properties

Reserve

PropertyId: res (0x0001)

Can be set by the MGC to indicate that the context (and associated resources) should be reserved or released.

Type: Enumerated
Possible Values: Reserve, Release
Defined in: ContextAttribute Descriptor
Characteristics: Read/Write

Reservation Size
PropertyId: size (0x0002)

Can be set by the MGC to indicate the number of terminations that the reserved context should be able to hold.

Type: Integer
Possible Values: 0-255
Defined in: ContextAttribute Descriptor
Characteristics: Read/Write

Reservation Type
PropertyId: type (0x0003)

Can be set by the MGC to indicate what type of context resources are to be reserved.

Type: Enumerated
Possible Values: Conf, BCT, PoC
Defined in: ContextAttribute Descriptor
Characteristics: Read/Write

7.4.8.2 Events

None

7.4.8.3 Signals

None

7.4.8.4 Statistics

None

7.4.8.5 Procedures

The MGC sets the reserve property to "Reserve" to indicate that the context identified in the transaction, and any associated resources should be held in reserve until the property is set to "Release". As long as the reserve property is set to "Reserve", the context is preserved, even if the last termination is subtracted. Once the MGC sets the reserve property to "Release" the context and its associated resources are released once the last termination is subtracted from the context. If there are no terminations left in the context, then the MG destroys the context and releases any resources immediately. In the instant that the property is set on the CHOOSE ($) context, the MG returns the ID of the context it has reserved for the MGC. The reserve property defaults to "Release".

The size property indicates how many terminations the MG should be prepared to place in the context. The MGC may alter the reserved size at any time by resetting the property. The size property is of no relevance unless the reserve property is set to "Reserve". The size property defaults to 2.

This package is realized on the root termination, and is applicable to all contexts in the Media Gateway.

7.4.9 PoC Proprietary Package

Package Name: ACL proprietary PoC Package

PackageID: ACPoC

Description: This package defines the interaction required between the PSC and the PMN through the MEGACO protocol. It defines the properties, signals and events. It will enable the PMN to detect and report events on the MEDIA passing through it. All the events and signals defined in this package are applied to ephemeral (RTP) terminations.

Version: 1

Designed to be extended only: No

Extends: None

7.4.9.1 Events

7.4.9.1.1 First Media Packet

Event Name: First Media Packet Detection

Event ID: fmp,(0x0000)

Description: PSC requesting the termination to detect the first incoming RTP media packet.

Events Descriptor Parameters: None

Observed Events Descriptor Parameters: None

7.4.9.1.2 Last Media Packet

Event Name: Last Media Packet Detection

Event ID: lmp,(0x0001)

Description: PSC requesting the termination to detect a particular incoming RTP media packet based on the sequence number. If the RTP packet with the requested sequence number was already received by the PMN, the PMN should report immediately.

Events Descriptor Parameters:

Parameter Name: RTP sequence number

Parameter ID: seq, (0x0002)

Type: Integer

Possible Values: Unsigned integer

Description: incoming RTP media packet sequence number to detect.

Observed Events
Descriptor Parameters: None

7.4.9.1.3 Unexpected Media Packets

Event Name: Unexpected Media Packets

Event ID: ump,(0x0003)

Description: PSC requesting the termination to detect incoming RTP media packets when the mode of the termination is SendOnly

Events Descriptor
Parameters: None

Observed Events
Descriptor Parameters: None

7.5 Using Voice Streaming

The voice streaming layer provides the user with the ability to play and record different types of files towards IP or TDM while using an NFS or HTTP server.

For a detailed explanation on the **.ini* file parameters refer to the "NFS parameters", "NFS Servers Table parameters" and "Voice Streaming parameters" in Individual ini File Parameters on page 133.

> **Note:**. The following sub-section on Voice Streaming Features is applicable to **ALL** devices.

7.5.1 Voice Streaming Features

The following summarizes the Voice Streaming features which are supported both on HTTP servers and NFS servers unless stated differently:

7.5.1.1 Basic Streaming Play

A user may play a **.wav, *.au or *.raw* file from a remote server using G.711 coders.

7.5.1.2 Play from Offset

A user may play a **.wav, *.au or *.raw* file from a given offset within the file. Offset can be both positive and negative relative to the files length. A negative offset relates to an offset from the end of the file.

7.5.1.3 Working with Remote File Systems

A user may configure up to 16 remote file systems working with the device through NFS mounting.

7.5.1.4 Using Proprietary Scripts

A user may use cgi or servlet scripts released with the version for recording to a remote HTTP server using the POST or PUT method.

7.5.1.5 Combining HTTP and NFS Play / Record

A client may use any combination of HTTP/NFS play and HTTP/NFS record on the same channel.

7.5.1.6 Supporting Dynamic HTTP URLs

Voice streaming supports dynamic HTTP URLs. The following terminology is used:

1. Static audio content - traditional audio file URLs containing references to specific files (*.wav, *.au or *.raw). For example: **http://10.50.0.2/qa/GOSSIP_ENG.wav**
2. Dynamic audio content - URLs referencing to cgi scripts or servlets. For example: **http://10.50.0.2/cgi/getaudio.cgi?filename=DEFAULT_GREETING.raw&offset=0**

In the case of dynamic URLs, the device performs the GET command with the supplied URL and as a result the servlet or cgi script on the Web server gets invoked. The Web server responds by sending a GET response containing the audio.

The URL can be of this form (RFC 1738 URLs, section 3.3)

```
http://<host>:<port>/<path>?<searchpart>
```

where:

:<port> is optional.

<path> is a path to a server-side script.

<searchpart> is of the form: key=value[&key=value]*

Note: At least one key=value pair is required.

Here is another example of a dynamic URL:

```
http://MyServer:8080/prompts/servlet?action=play&language=eng&file=
welcome.raw&format=1
```

See also RFC 2396 URI: Generic Syntax.

The servlet or cgi script can respond by sending a complete audio file or a portion of an audio file. The device will skip any **.wav* or **.au* file header that it encountered at the beginning of the response. The device will not attempt to use any information in the header. For example, the device does not use the coder from the header. Note however that the coder may be supplied through Web or **.ini* file parameters. For further information, refer to Individual ini File Parameters on page 133.

Note: The following features are only applicable to **TP-260/UNI, TP-1610, IPM-260/UNI, IPM-1610 and IPmedia 2000**.

Note: The following features are relevant for both NFS and HTTP.

7.5.1.7 Play LBR Audio File

A user may play a file using low bit rate coders for **.wav* and **.raw* files.

7.5.1.8 Basic Record

A user may record a *.wav, *.au or *.raw files to a remote server using G.711 coders.

7.5.1.9 Remove DTMF Digits at End of Recording

A user may configure a recording to remove the DTMF receive at the end indicating an end of a recording.

7.5.1.10 Record Files Using LBR

A user may record a file using low bit rate coders for **.wav* and **.raw* files.

Note: The following features are only applicable to **TP-6310, TP-8410, IPM-6310, IPM-8410 and IPmedia 3000**.

Note: The following features are relevant only for working with NFS servers.

7.5.1.11 Basic Record

A user can record a **.wav, *.au* or **.raw* files to a remote server using G.711 coders.

7.5.1.12 Play file Under Construction

The device can be used to play an **.au / *.raw* file that is still under construction. For example, one can be recording to an **.au / *.raw* file while one is playing the same file. There must be a delay between the start of the record and the start of the play. That delay is dependent on the coder used to encode the recorded audio.

For G.711, the delay should be 2 seconds or greater. For other coders, the delay should be at least 10000/avgBytesPerSec*1.2 [add 20%] rounded up to the nearest second. For G.723Low, the delay would be 10000/667*1.2 or 18 seconds.

Note: The device does not support play of a **.wav* file still under construction.

In some instances, caching of streamed audio files might interfere with the play-file-under-construction feature. Consider the following scenario:

1. Perform record to *x.raw*.
2. Play *x.raw* (the device begins caching the file while it is being constructed - it won't cache the whole file).
3. Recording completes.
4. Play *x.raw* again. The device plays the cached, incomplete *x.raw*.

To avoid this scenario, it is recommended that stream caching be disabled in cases where the call agent will replay a file that was first played when the file was under construction.

7.5.2 Dynamic Caching Mechanism

Dynamic caching is a mechanism for optimization of HTTP and NFS file access from remote servers.

By using this mechanism, the traffic to/from remote HTTP and NFS servers can be significantly decreased. The mechanism is based on the Least Frequently Used (LFU) algorithm - i.e., frequently played files will be stored in the internal cache memory, and will be played from the cache memory rather than being retrieved from a remote server and then played.

The mechanism uses a portion of the 32 MB of the resident local memory area in order to cache the remotely played files. The *StreamingCacheSize ini* file parameter sets the number of MB that will be used for the cache mechanism out of the 32MB local memory. The remainder of the memory is used for saving and playing local IVR (e.g. voice prompts). When the *StreamingCacheSize* is greater than 0 (default), the mechanism is automatically enabled and active.

In order to avoid a situation where files that are being played from the cache, are updated/changed on server, the cache mechanism offers a refresh timeout. The *StreamingCacheRefreshTime ini* file parameter sets the mechanism's refresh rate in term of minutes. At every refresh, the files saved in the cache memory will be re-retrieved from their remote servers.

In order to monitor the cache mechanism statistics and performance, the following performance-monitoring parameters are supported:

StreamingCacheHitRate - defines the number of cache hits in the last second.

StreamingCacheMissRate - defines the number of cache misses in the last second.

StreamingCacheServerRequestsRate - defines number of server request for files in the last second.

All these parameters should be monitored using their averages - e.g., 'what is the average hit rate per second?'.

Note: One should be aware of the following "stale cache" problem when playing a recorded file when stream caching is enabled:

1. Record to file x.
2. After the record is complete, play file x.
3. Device caches file x.
4. Record to file x again.
5. After the record is complete, play file x. The file cached in step 3 will be played, not the new file.

One way to avoid the stale cache problem is to use a unique file name on each record operation. This assumes that there is a server-side audit that cleans up old recorded files.

7.5.3 Using File Coders with Different Channel Coders

The following tables describe the support for different combinations of file coders (used for recording or playing a file) and channel coders (used for opening the channel).

The following abbreviations are used:

LBR	Low Bit Rate Coder
PCMU	G.711 µ-law coder
PCMA	G.711 A-law coder
IP	The direction is to the network
TDM	The direction is to the TDM

Note: When recording with an LBR type coder, it is assumed that the same coder is used both as the file coder and the channel coder. Combinations of different LBR coders are not supported at this time.

Note: The following three tables are only applicable to **TP-260/UNI, TP-1610, IPM-260/UNI, IPM-1610 and IPmedia 2000**.

7.5.3.1 Playing a File to TDM/IP

Note: For IPM devices, a file may also be played towards IP.

Table 7-7: Coder Combinations - Playing a file to TDM/IP

File Coder	File Type								
	*.wav			*.au			*.raw		
	Channel Coder			Channel Coder			Channel Coder		
	PCMA	PCMU	LBR	PCMA	PCMU	LBR	PCMA	PCMU	LBR
PCMA	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCMU	✓	✓	✓	✓	✓	✓	✓	✓	✓
LBR	×	×	✓	×	×	×	×	×	✓

7.5.3.2 Recording a file from IP

Table 7-8: Coder Combinations - Recording a file from IP (for IPM devices only)

File Coder	File Type		
	*.wav	*.au	*.raw
	Channel Coder	Channel Coder	Channel Coder

	PCMA	PCMU	LBR	PCMA	PCMU	LBR	PCMA	PCMU	LBR
PCMA	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCMU	✓	✓	✓	✓	✓	✓	✓	✓	✓
LBR	✗	✗	✓	✗	✗	✗	✗	✗	✗

7.5.3.3 Recording a file from TDM

Table 7-9: Coder Combinations - Recording a file from TDM

File Coder	File Type								
	*.wav			*.au			*.raw		
	Channel Coder			Channel Coder			Channel Coder		
	PCMA	PCMU	LBR	PCMA	PCMU	LBR	PCMA	PCMU	LBR
PCMA	✓	✗	✗	✓	✗	✗	✓	✗	✗
PCMU	✗	✓	✗	✗	✓	✗	✗	✓	✗
LBR	✗	✗	✓	✗	✗	✗	✗	✗	✓

Constraint

When opening a channel for TDM recording, the user should open the channel with the following parameter settings:

```
DTMFTransportType = TransparentDTMF
MFTransportType = TransparentMF
FaxTransportMode = Disable
V34FaxTransportType = Disable
CNGDetectorMode = Disable
VXXModemTransportType = Disable
BellModemTransportType = Disable
CallerIDTransportType = Disable
TTYTransportType = Disable
Coder = [File's Coder]
SCE = 0
```

Note: The following two tables are only applicable to **TP-6310, TP-8410 IPM-6310 IPM-8410 and IPmedia 3000**.

7.5.3.4 Playing a File to TDM/IP

Note: For IPM devices, a file may also be played towards IP.

Table 7-10: Coder Combinations - Playing a file to TDM/IP

File Coder	File Type								
	*.wav			*.au			*.raw		
	Channel Coder			Channel Coder			Channel Coder		
	PCMA	PCMU	LBR	PCMA	PCMU	LBR	PCMA	PCMU	LBR
PCMA	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCMU	✓	✓	✓	✓	✓	✓	✓	✓	✓
LBR	×	×	×	×	×	×	×	×	×

7.5.3.5 Recording a file from IP/TDM (only NFS supported)

Note: For IPM devices, a file may also be recorded from IP.

1. For **.raw* files, when recording with PCMA and PCMU, the file coder should be identical to the **.ini* file PcmLawSelect value. Note that the PcmLawSelect value may also be configured through Web/SNMP.
2. Recording is supported only for NFS.

Table 7-11: Coder Combinations - Recording a file from IP/TDM

File Coder	File Type								
	*.wav			*.au			*.raw		
	Channel Coder			Channel Coder			Channel Coder		
	PCMA	PCMU	LBR	PCMA	PCMU	LBR	PCMA	PCMU	LBR
PCMA	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCMU	✓	✓	✓	✓	✓	✓	✓	✓	✓
LBR	×	×	✓	×	×	×	×	×	✓

7.5.4 Maximum Concurrent Playing and Recording

For more details, refer to the relevant Release Notes relating to the device.

7.5.5 Supporting LBR Coders

The following list describes the DSP template needed for using different low bit rate coders and their support for **.wav, *.au* and **.raw* files:

Table 7-12: DSP Templates Applicable to TP-260/UNI, TP-1610, IPM-260/UNI, IPM-1610, Mediant 2000 and IPmedia 2000

Coder	DSP Templates		
	*.wav file	*.raw file	*.au file
G726_16	0-3	0-5	×
G726_24	0-3	0-5	×
G726_32	0-3	0-5	×
G726_40	0-3	0-5	×
G727_16	×	0-5	×
G727_24_16	×	0-5	×
G727_24	×	0-5	×
G727_32_16	×	0-5	×
G727_32_24	×	0-5	×
G727_32	×	0-5	×
G727_40_16	×	0-5	×
G727_40_24	×	0-5	×
G727_40_32	×	0-5	×
G723_LOW	0	0	×
G723_HIGH	0	0	×
G729	0-3	0-3	×
GSM610	0,1	0-1	×
GSM610MS	0,1	0-1	×
GSM_EFR	0,1	1	×
G728	3	3	×
NET_CODER_6_4	×	0	×
NET_CODER_7_2	×	0	×
NET_CODER_8	×	0	×
NET_CODER_8_8	×	0	×
NET_CODER_9_6	×	0	×
VOX_ADPCM	0	0,4,5	×
G729E	×	3	×
LINEAR_PCM	×	0-5	×

Table 7-12: DSP Templates Applicable to TP-260/UNI, TP-1610, IPM-260/UNI, IPM-1610, Mediant 2000 and IPmedia 2000

Coder	DSP Templates		
	*.wav file	*.raw file	*.au file
AMR_4_75	×	1	×
AMR_5_15	×	1	×
AMR_5_9	×	1	×
AMR_6_7	×	1	×
AMR_7_4	×	1	×
AMR_7_95	×	1	×
AMR_10_2	×	1	×
AMR_12_2	×	1	×
QCELP_8	×	2	×
QCELP_13	×	2	×

Table 7-13: DSP Templates
Applicable to TP-6310, TP-8410 IPM-6310 IPM-8410, Mediant 3000 and IPmedia 3000

Coder	DSP Templates		
	*.wav file	*.raw file	*.au file
G726_16	0-6	0-8	×
G726_24	0-6	0-8	×
G726_32	0-6	0-8	×
G726_40	0-6	0-8	×
G727_16	×	0-8	×
G727_24_16	×	0-8	×
G727_24	×	0-8	×
G727_32_16	×	0-8	×
G727_32_24	×	0-8	×
G727_32	×	0-8	×
G727_40_16	×	0-8	×
G727_40_24	×	0-8	×
G727_40_32	×	0-8	×
G723_LOW	0	0	×
G723_HIGH	0	0	×

Table 7-13: DSP Templates
Applicable to TP-6310, TP-8410 IPM-6310 IPM-8410, Mediant 3000 and IPmedia 3000

G729	0-6	0-8	×
GSM610	×	1,4,5	×
GSM610MS (see note below)	1,4	1,4,5	×
GSM_EFR	×	1,4	×
G729E	×	3	×
G722_64K	4	4	×

Note: GSM610MS may only be used to record from TDM. Recording from IP is currently **not** supported.

7.5.6 Basic Voice Streaming Configuration

Note: The following *.ini*. file parameters can be configured via SNMP or Web interface. For further details, refer to the relevant sections in the manual.

For enabling the voice streaming, the following offline *.*ini** file parameter should appear in the *.*ini** file:

```
EnableVoiceStreaming = 1
```

7.5.7 HTTP Recording Configuration

Note: The following *.ini*. file parameters can be configured via SNMP or Web interface. For further details, refer to the relevant sections in the manual.

The HTTP record method (PUT or POST) is configured via the following offline *.*ini** parameter:

```
// 0=post (default), 1=put
VoiceStreamUploadMethod = 1
```

The default value is:

```
VoiceStreamUploadPostUri =
"/audioupload/servlet/AcAudioUploadServlet"
```

Note: The PUT method disregards this string.

7.5.8 NFS Configuration via **.ini* File

Note: The following *.ini*. file parameters can be configured via SNMP or Web interface. For further details, refer to the relevant sections in the manual.

The following is a sample NFS configuration. It shows that the NFS server at 192.168.20.26 is sharing 2 file systems, one rooted at /PROV_data, and the other rooted at /opt/uas. NFSv3 is used for both remote file systems. The defaults for UID(0) and GID(1) are used.

```
[ NFSServers ]

FORMAT NFSServers Index = NFSServers HostOrIP, NFSServers RootPath,
NFSServers NfsVersion;
NFSServers          0                 = 192.168.20.26             ,
/PROV data                 ,    3;
NFSServers          1                 = 192.168.20.26             ,
/opt/uas                         ,    3;

[ \NFSServers ]
```

For further details, refer to NFS Parameters on page 196 and NFS Servers Table Parameters on page 235.

The following are some general notes on NFS configuration:

1. The combination of Host/IP and Root Path should be unique for each row in the table. For example, there should be only one row in the table with a Host/IP of 192.168.1.1 and Root Path of /audio.
2. To avoid terminating calls in progress, a row should not be deleted or modified while the device is currently accessing files on that remote NFS file system.
3. An NFS file server can share multiple file systems. There should be a separate row in this table for each remote file system shared by the NFS file server that needs to be accessed by this device.

7.5.9 Supporting HTTP Servers

The following is a list of HTTP servers that are known to be compatible with AudioCodes™ voice streaming under Linux™:

- Apache - cgi scripts are used for recording and supporting dynamic URLs.
- Jetty - servlets scripts are used for recording and supporting dynamic URLs.
- Apache tomcat - also using servlets.

7.5.9.1 Tuning the Apache Server

It is recommended to perform the following changes in the **http.conf** file located in the **apache conf/** directory:

- Defining PUT script location - Assuming you have the put.cgi file included in this package, add the following line for defining the PUT script to be used (script should be placed in the cgi-bin/ directory:

```
Script PUT /cgi-bin/put.cgi
```

- Create the directory /the-apache-dir/perl (for example /var/www/perl) and copy the CGI script to that directory. In the script, change the first line from c:/perl/bin/perl to your perl executable file (this step is required only if mod_perl is not included in your Apache installation).
- Keep-alive parameters - the following parameters should be set for correct working with multiple POST requests:

```
KeepAlive On
MaxKeepAliveRequests 0 (unlimited amount)
```

- Using **mode perl** - fix the mod_perl to:

```
<IfModule mod perl.c>
<Location /cgi-bin>
   SetHandler perl-script
   PerlResponseHandler ModPerl::Registry
   Options +ExecCGI
   PerlOptions +ParseHeaders
   Order allow,deny
   Allow from all
</Location>
</IfModule>
```

- Apache MPM worker - we recommend using the Multi-Processing Module implementing a hybrid multi-threaded multi-process Web server. The following configuration is recommended:

```
<IfModule worker.c>
ThreadLimit       64
StartServers         2
ServerLimit       20000
MaxClients        16384
MinSpareThreads    100
MaxSpareThreads     250
ThreadsPerChild     64
MaxRequestsPerChild  16384
</IfModule>
```

7.5.10 Supporting NFS Servers

The following is a list of NFS servers that are known to be compatible with AudioCodes Voice Streaming functionality.

Table 7-14: Compatible NFS Servers

Operating System	Server	Versions
Solaris™ 5.8 and 5.9	nfsd	2, 3
Fedora™ Linux™ 2.6.5-1.358	nfsd	2, 3
Mandrake™ Linux™ v2.4.22	nfsd	2, 3
Windows™ 2000	Services For Unix™ (SFU)	2, 3
Windows™ 2000	winnfsd	2 (Note 1)
SCO UnixWare™ 7.1.1	nfsd	2, 3

Table 7-14: Compatible NFS Servers

Operating System	Server	Versions
Windows™ 2000	Cygwin nfsd	2 (Note 2)

Note: Cygwin and winnfsd support only NFSv2.

7.5.10.1 Solaris-based NFS Servers

If you are using a Solaris™-based NFS server, then the following nfsd configuration change is recommended, especially if you are planning to support voice recording:

Edit the file /etc/default/nfs and set the value of NFSD_SERVERS to N*2, where N is the max number of record and play sessions that you expect to have in progress at any one time.

This parameter controls the number of worker threads that the NFS daemon will use to satisfy requests. When a request comes in, a check is made for an idle worker thread. If an idle worker thread is available, then the request is passed to it. If an idle worker thread is not available, then a new one is created and the request is passed to it. If the limit in worker threads is reached, the request is queued until one of the existing worker threads is available. Queuing of NFS requests from a real-time application such as the media server should be avoided. Therefore, the NFSD_SERVERS parameter should be used to ensure there is an adequate number of worker threads.

The default value for NFSD_SERVERS is 1, though typically the /etc/default/nfs file will contain NFSD_SERVERS=16.

To determine how many worker threads are running on the NFS server, invoke the following command:

```
pstack `pgrep nfsd` | grep nfssys | wc -l
```

An idle NFS daemon process will show 1 nfsd thread.

Directories are shared by placing an entry in the /etc/dfs/dfstab file. See the share(1M) and share_nfs(1M) man pages for information on the format of entries in the dfstab file. Note that read-write (rw) is the default behavior. If you are planning to record to the file system, ensure that the directory is shared as rw. Also ensure that the recording directory has 777 (rwxrwxrwx) permissions.

Here is an example /etc/dfs/dfstab file. Note that /audio1 is shared as read-only, and /audio2 is shared as read-write.

```
> cat /etc/dfs/dfstab
share -F nfs -o ro /audio1
share -F nfs /audio2
```

Ensure that the /etc/nfssec.conf file is configured so that "sys" is the default security mode. You should see the following.

```
> cat /etc/nfssec.conf
…
none          0       -       -       -       # AUTH NONE
sys             1       -       -       -       # AUTH SYS
dh              3       -       -       -       # AUTH DH
…
default       1       -       -       -       # default is AUTH SYS
```

If the systems administrator wishes to use a default of other than AUTH_SYS in the nfssec.conf file, then you should add "sec=sys" to each line in the dfstab file that is to be shared with an AudioCodes device. For example:

```
> cat /etc/dfs/dfstab
share -F nfs -o sec=sys,ro /audio1
share -F nfs -o sec=sys /audio2
```

To restart the nfs daemon on Solaris, issue the following two commands:

```
> /etc/init.d/nfs.server stop
> /etc/init.d/nfs.server start
```

To see a log of which directories were shared on the previous restart of the nfs daemon, type out the sharetab file. For example:

```
> cat /etc/dfs/sharetab
/audio1      -       nfs     ro
/audio2      -       nfs     rw
```

Other useful Solaris™ commands are:

- dfmounts - displays shared directories, including a list of clients that have those resources mounted
- dfshares - displays a list of shared directories

7.5.10.2 Linux-based NFS Servers

The AudioCodes products uses local UDP ports that are outside of the range of 0..IPPORT_RESERVED(1024). Therefore, when configuring a remote file system to be accessed by an AudioCodes product, use the insecure option in the /etc/exports file. The insecure option allows the nfs daemon to accept mount requests from ports outside of that range. Without the insecure option, you will receive this nfs daemon log:

```
rpc.mountd: refused mount request from <ip> for <dir> illegal port
28000
```

and this Syslog:

```
NFS mount failed, reason=permission denied IP=<ip> path=<dir>
state=waitForMountReply numRetries=0
```

For more information, see the exports(5) man page on your Linux server.

Here is a sample /etc/exports entry:

```
/nfsshare *(rw,insecure,no_root_squash,no_all_squash,sync)
```

7.5.11 Common Problems and Solutions

Be sure to inspect the Syslog for any problem you encounter; in many cases the cause will appear there.

7.5.11.1 General Voice Streaming Problems

Problem: Attempts to perform voice streaming operations results in each Syslog containing this string: VS_STACK_NOT_ACTIVE.

Probable Cause: Voice streaming is not enabled.

Corrective Action: Enable voice streaming by loading an **.ini* file containing this entry:

```
EnableVoiceStreaming = 1
```

7.5.11.2 HTTP Voice Streaming Problems

Problem: The last half-second of an announcement is not played, or a record operation terminates abnormally and a "VSReceiveFromNetwork: VS_CONNECTION_WITH_SERVER_LOST" Syslog is generated. The problem has been seen with Apache version 2.0.50 on Solaris 9.

Probable Cause: The Web server is closing the virtual circuit at unexpected times.

Corrective Action: Increase the Apache KeepAliveTimeout config parameter. Try to increase it to be 30 seconds or so longer than the longest announcement or expected record session.

7.5.11.3 NFS Voice Streaming Problems

Problem: Announcement is terminated prematurely. Syslog shows this log:

"NFS request aborted … networkError"

Probable Cause: The AudioCodes media server lost communications with the NFS server. There was a network problem or some problem with the NFS server.

Corrective Action: Fix the network problem or NFS server problem. Ensure that the NFS server is not over-loaded.

Problem: Unable to play announcements from an NFS server. Each Syslog is shown:

"Unable to create new request, file system not mounted"

"NFS mount error …"

Probable Cause: Either there is a problem with the NFS server, the network, or configuration of the media server or NFS server.

Corrective Action: Fix the network problem or NFS server problem. Check the configuration on both the media server and the NFS server.

Problem: Record is terminated prematurely. Syslog shows these logs:

"VeData: no free buffers, req=16"

"Unable to play announNFS request aborted, reqid=16 cid=16 error=noRecordBufferError reqtype=vsHostRecord state=recTransfer"

Probable Cause: This occurs when the media server is receiving audio faster than it can save it to the remote NFS server. Either there is a problem with the NFS server, the network, or configuration of the media server or NFS server.

Corrective Action: Fix the network problem or NFS server problem. Check the configuration on both the media server and the NFS server.

Problem: Remote file system is not being mounted and you receive this Syslog:

"NFS mount failed, reason=permission denied IP=<ip> path=<dir> state=waitForMountReply numRetries=0"

Probable Cause: The NFS server is not configured to accept requests on ports outside of the range of 0..1024.

Corrective Action: On a Linux NFS server, use the insecure option in the /etc/exports file. See the NFS Server Configuration section for more info.

Problem: All recording sessions are aborted at the same time with these Syslogs:

NFS request aborted, reqid=209 cid=-1 error=writeReplyError reqtype=writeFile state=writeWait [File:NfsStateMachine.cpp …]

```
NFS request aborted, reqid=186 cid=-1 error=writeReplyError reqtype=writeFile
state=writeWait [File:NfsStateMachine.cpp …]
```

Probable Cause: The file system on the NFS server is full.

Corrective Action: Remove unwanted files on the file system.

Reader's Notes

8 Security

This chapter describes the device's implementation of security protocols.

The following list specifies the available security protocols and their purposes:

- **IKE**
- **IPSec**

The IKE and IPSec protocols are part of the IETF standards for security issues. IKE and IPSec are used together on the media gateway to provide security for control and management protocols.

The IKE protocol (Internet Key Exchange) is responsible for obtaining the IPSec encryption keys and encryption profile (known as IPSec Security Association).

The IPSec protocol is responsible for securing the data streams. IPSec is used by device to assure confidentiality, authentication and integrity for the following media types:

- Control traffic, such as H.248 and MGCP
- Management traffic, such as SNMP and HTTP

Note: Some Security features are optional and can be ordered or upgraded at a future time.

Note: The RTP and RTCP streams cannot be secured by IPSec.

Important

Using IPSec may reduce the channel capacity of the device. Contact your AudioCodes sales representative for capacity information.

-
- **SSL/TLS** - Secures Web access (HTTPS) and Telnet access.
- **Internal Firewall** – Allows filtering unwanted inbound traffic.
- **RADIUS** - Is utilized by the Web interface and Telnet server for authentication.
- **Media Security** - Allows encryption of voice traffic on the IP network.

This section also contains network port usage information (useful for firewall administrators) and recommended practices for keeping your network secure.

8.1 IKE (Internet Key Exchange) and IPSec (IP Security)

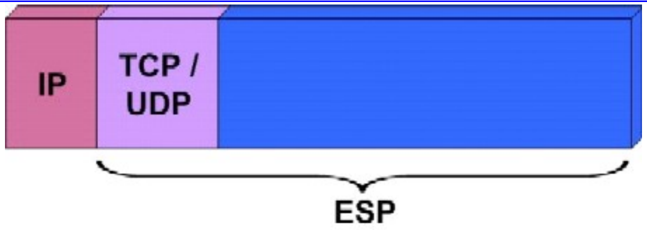
IKE and IPSec protocols are part of the IETF standards for establishing a secured IP connection between two applications (also referred to as peers). Providing security services at the IP layer, IKE and IPSec and are transparent to IP applications.

IKE and IPSec are used in conjunction to provide security for control and management (e.g., SNMP and Web) protocols but not for media (i.e., RTP, RTCP and T.38).

The IKE protocol is responsible for obtaining the IPSec encryption keys and encryption profile (known as IPSec Security Association (SA)).

IPSec is responsible for securing the IP traffic. This is accomplished by using the Encapsulation Security Payload (ESP) protocol to encrypt the IP payload (illustrated in the figure below).

Figure 8-1: IPSec Encryption using Encapsulation Security Payload (ESP) Protocol



8.1.1 IKE

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

The IKE negotiation is separated into two phases: main mode and quick mode. The main mode employs the Diffie-Hellman (DH) protocol to obtain an encryption key (without any prior keys), and uses a pre-shared key/X.509 certificate to authenticate the peers. The created channel secures the messages of the following phase (quick mode) in which the IPSec SA properties are negotiated.

The IKE negotiation is as follows:

- Main mode (the main mode creates a secured channel for the quick mode)
  - SA negotiation – The peers negotiate their capabilities using four proposals. Each proposal includes three parameters: Encryption method, Authentication protocol and the length of the key created by the DH protocol. The key's lifetime is also negotiated in this stage. For detailed information on configuring the four-main mode proposals, refer to IKE Configuration on page 494.
  - Key exchange (DH) – The DH protocol is used to create a phase-1 key.
  - Authentication – The two peers authenticate one another using the pre-shared key (configured by the parameter 'IKEPolicySharedKey') or by using certificate-based authentication. (For information regarding authentication methods, refer to
    IKE Configuration on page 494.)
- Quick mode (quick mode negotiation is secured by the phase-1 SA)

- SA negotiation – The peers negotiate their capabilities using up to four proposals. Each of the proposals includes two parameters: Encryption method and Authentication protocol. The quick mode SA lifetime is also negotiated in this stage. For detailed information on configuring the four-quick mode proposals, refer to the SPD table under IPSec Configuration on page 496.
- Key exchange – a symmetrical key is created using and secured by the previously negotiated main mode SA.

IKE Specifications:

- Authentication mode - pre-shared key only / certificate based authentication
- Main mode is supported for IKE Phase 1
- Supported IKE SA encryption algorithms - AES, DES and 3DES
- Hash types for IKE SA - SHA1 and MD5

8.1.2 IPSec

IPSec is responsible for encrypting and decrypting the IP streams.

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

Each outgoing packet is analyzed and compared to the SPD table. The packet's destination IP address (and optionally, destination port, source port and protocol type) are compared to each entry in the table. If a match is found, the gateway checks if an SA already exists for this entry. If it doesn't, the IKE protocol is invoked (refer to Section 1.1.1 above) and an IPSec SA is established. The packet is encrypted and transmitted. If a match is not found, the packet is transmitted un-encrypted.

Note 1: An incoming packet whose parameters match one of the entries of the SPD table but is received un-encrypted, is dropped.

Note 2: IPSec does not function properly if the gateway's IP address is changed on-the-fly. Therefore, reset the gateway after you change its IP address.

IPSec Specifications:

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

8.1.3 Configuring IKE and IPSec

Note: To enable IKE and IPSec on the device set the ***ini*** file parameter 'EnableIPSec' to 1. Note that when this parameter is defined, even if no table entries exist, the device channel capacity is reduced (refer to the IMPORTANT Note at the beginning of the chapter).

8.1.3.1 IKE Configuration

The parameters described in the table below are used to configure the first phase (main mode) of the IKE negotiation for a specific peer. A different set of parameters can be configured for each of the 20 available peers.

Table 8-1: IKE Table Configuration Parameters

Parameter Name	Description
Shared Key [IKEPolicySharedKey]	Determines the pre-shared key (in textual format). Both peers must register the same pre-shared key for the authentication process to succeed. The parameter will only be valid if ikePolicyAuthenticationMethod is set to pre-shared key authentication. Note 1: The pre-shared key forms the basis of IPSec security and should therefore be handled cautiously (in the same way as sensitive passwords). It is not recommended to use the same pre-shared key for several connections. Note 2: Since the ini file is in plain text format, loading it to the device over a secure network connection is recommended, preferably over a direct crossed-cable connection from a management PC. Note 3: After it is configured, the value of the pre-shared key cannot be obtained via Web, ini file or SNMP.
First to Fourth Proposal Encryption Type [IKEPolicyProposal Encryption_X]	Determines the encryption type used in the main mode negotiation for up to four proposals. X stands for the proposal number (0 to 3). The valid encryption values are: Not Defined (default)
	DES-CBC [1] Triple DES-CBC [2] AES [3]
First to Fourth Proposal Authentication Type [IKEPolicyProposal Authentication_X]	Determines the authentication protocol used in the main mode negotiation for up to four proposals. X stands for the proposal number (0 to 3). The valid authentication values are:

Table 8-1: IKE Table Configuration Parameters

Parameter Name	Description
	Not Defined (default) HMAC-SHA1-96) [2] HMAC-MD5-96 [4]
First to Fourth Proposal DH Group [IKEPolicyProposal DHGroup_X]	Determines the length of the key created by the DH protocol for up to four proposals. X stands for the proposal number (0 to 3). The valid DH Group values are:
	Not Defined (default) DH-786-Bit [0] DH-1024-Bit [1]
IKE Policy Authentication Method **[IkePolicyAuthenticationMethod]**	Determines the authentication method for IKE. The valid authentication method values are:
	PRE_SHARED_KEY (default) [0] RSA_SIGNATURE [1]
	Note 1: for pre-shared key based authentication, peers participating in an IKE exchange must have a prior (out of band) knowledge of the common key (see IKEPolicySharedKey parameter). Note 2: for RSA signature based authentication peers must be loaded with a certificate signed by a common CA, for more info on certificates see the Server Certificate Replacement on page 505.
IKE SA LifeTime (sec) [IKEPolicyLifeInSec]	Determines the duration (in seconds) for which the negotiated SA is valid. After the time expires, the SA is re-negotiated. The default value is 28800 (8 hours).
IKE SA LifeTime (KB) [IKEPolicyLifeInKB]	Determines the duration (in kilobytes) for which the negotiated SA is valid. After the time expires, the SA is re-negotiated. The default value is 0 (this parameter is ignored).
The lifetime parameters (IKEPolicyLifeInSec and IKEPolicyLifeInKB) determine the duration the SA created in the main mode phase is valid. When the lifetime of the SA expires, it is automatically renewed by performing the IKE first phase negotiations. To refrain from a situation where the SA expires, a new SA is being negotiated while the old one is still valid. As soon as the new SA is created, it replaces the old one. This procedure occurs whenever an SA is about to expire.	

If no IKE methods are defined (Encryption / Authentication / DH Group), the default settings (shown in the table below) are applied.

Table 8-2: Default IKE First Phase Proposals

	Encryption	Authentication	DH Group
Proposal 0	3DES	SHA1	1024
Proposal 1	3DES	MD5	1024
Proposal 2	3DES	SHA1	786
Proposal 3	3DES	MD5	786

➢ To configure the IKE table using the *ini* file:

The IKE parameters are configured using ***ini*** file tables (described in Tables in the Uploaded ***ini*** File on page 33). Each line in the table refers to a different IKE peer.

The Format line (IKE_DB_INDEX in the example below) specifies the order in which the actual data lines are written. The order of the parameters is irrelevant. Parameters are not mandatory unless stated otherwise. To support more than one Encryption / Authentication / DH Group proposals, for *each* proposal specify the relevant parameters in the Format line. Note that the proposal list must be contiguous. The following is an example of an IKE Table.

```
[IPSec_IKEDB_Table]

FORMAT IKE_DB_INDEX = IKEPolicySharedKey,
IKEPolicyProposalEncryption_0, IKEPolicyProposalAuthentication_0,
IKEPolicyProposalDHGroup_0, IKEPolicyProposalEncryption_1,
IKEPolicyProposalAuthentication_1, IKEPolicyProposalDHGroup_1,
IKEPolicyLifeInSec;
IkePolicyAuthenticationMethod;

IPSEC_IKEDB_TABLE 0 = 123456789, 1, 2, 0, 2, 2, 1, 28800, 0;

[\IPSEC_IKEDB_TABLE]
```

In the above example, a single IKE peer is configured. Pre-shared key authentication is selected. Its pre-shared key is 123456789. Two security proposals are configured: DES/SHA1/786DH and 3DES/SHA1/1024DH. In addition, a lifetime of 28800 seconds is applied.

8.1.3.2 IPSec Configuration

The parameters described in the table below are used to configure the SPD table. A different set of parameters can be configured for each of the 20 available IP destinations.

Table 8-3: SPD Table Configuration Parameters

Parameter Name	Description	
Mode [IPSecMode]	Defines the IPSec mode of operation. (0) Transport (Default) (1) Tunneling	IPSec is applied to outgoing packets whose IP address, destination port, source port and protocol type match the values defined for these four parameters.
Remote Tunnel IP Address [IPSecPolicyRemoteTunnelIPAddress]	Defines the IP address of the remote IPSec tunneling device. Note: This parameter is only available if IPSecMode is set to Tunneling (1).	
Remote Tunnel Subnet Mask [IPsecPolicyRemoteSubnetMask]	Defines the Subnet mask of the remote IPSec tunneling device. The default value is host-to-host IPSec tunnel. (i.e.255.255.255.255 - No Subnet) Note: This parameter is only available if IPSecMode is set to Tunneling (1).	
Remote IP Address [IPSecPolicyRemoteIPAddress]	Defines the destination IP address (or a FQDN) the IPSec mechanism is applied to. This parameter is mandatory. Note: When a FQDN is used, a DNS server must be configured (DNSPriServerIP).	
Local IP Address Type [IPSecPolicyLocalIPAddressType]	If multiple IP's/VLANs are configured, the user should define the required local interface on which to apply encryption. Valid Address Type values are: (0) OAMP interface (2) Control interface Default value is 2 (Control interface)	
Source Port [IPSecPolicySrcPort]	Defines the source port the IPSec mechanism is applied to. The default value is 0 (any port).	
Destination Port [IPSecPolicyDstPort]	Defines the destination port the IPSec mechanism is applied to. The default value is 0 (any port).	
Protocol [IPSecPolicyProtocol]	Defines the protocol type the IPSec mechanism is applied to. 0 = Any protocol (default) 17 = UDP 6 = TCP Or any other protocol type defined by	

Table 8-3: SPD Table Configuration Parameters

Parameter Name	Description
	IANA (Internet Assigned Numbers Authority).
Related Key Exchange Method Index [IPsecPolicyKeyExchangeMethod Index]	Determines the index for the corresponding IKE entry. Note that several policies can be associated with a single IKE entry. The valid range is 0 to 19. The default value is 0.
IKE Second Phase Parameters (Quick Mode)	
SA Lifetime (sec) [IPsecPolicyLifeInSec]	Determines the duration (in seconds) for which the negotiated SA in the second IKE session (quick mode) is valid. After the time expires, the SA is re-negotiated. The default value is 28800 (8 hours).
SA Lifetime (KB) [IPSecPolicyLifeInKB]	Determines the duration (in kilobytes) for which the negotiated SA in the second IKE session (quick mode) is valid. After this size is reached, the SA is re-negotiated. The default value is 0 (this parameter is ignored).
The lifetime parameters (IPsecPolicyLifeInSec and IPSecPolicyLifeInKB) determine the duration an SA is valid. When the lifetime of the SA expires, it is automatically renewed by performing the IKE second phase negotiations. To refrain from a situation where the SA expires, a new SA is being negotiated while the old one is still valid. As soon as the new SA is created, it replaces the old one. This procedure occurs whenever an SA is about to expire.	
First to Fourth Proposal Encryption Type [IPSecPolicyProposalEncryption _X]	Determines the encryption type used in the quick mode negotiation for up to four proposals. X stands for the proposal number (0 to 3). The valid encryption values are: Not Defined (default)
	(0) None (No encryption) (1) DES-CBC (2)Triple DES-CBC (3) AES
First to Fourth Proposal Authentication Type [IPSecPolicyProposal Authentication_X]	Determines the authentication protocol used in the quick mode negotiation for up to four proposals. X stands for the proposal number (0 to 3). The valid authentication values are:
	Not Defined (default) (2) HMAC-SHA-1-96 (4) HMAC-MD5-96

If no IPsec methods are defined (Encryption / Authentication), the default settings (shown in the table below) are applied.

Table 8-4: Default IKE Second Phase Proposals

	Encryption	Authentication
Proposal 0	3DES	SHA1
Proposal 1	3DES	MD5
Proposal 2	DES	SHA1
Proposal 3	DES	MD5

8.1.3.2.1 Configuring the SPD table using the *ini* file

The SPD table is configured using ***ini*** file tables (described in ***ini*** File Structure on page 29). Each line in the table refers to a different peer/traffic type combination.

The Format line (SPD_INDEX in the example below) specifies the order in which the actual data lines are written. The order of the parameters is irrelevant. Parameters are not mandatory unless stated otherwise. To support more than one Encryption / Authentication proposals, for *each* proposal specify the relevant parameters in the Format line. Note that the proposal list must be contiguous. The following is an example of an SPD Table

```
[ IPSEC_SPD_TABLE ]

FORMAT SPD_INDEX = IPSecMode, IPSecPolicyRemoteIPAddress,
IpsecPolicySrcPort, IPSecPolicyDStPort,IPSecPolicyProtocol,
IPSecPolicyLifeInSec,
IPSecPolicyProposalEncryption_0,
IPSecPolicyProposalAuthentication_0,
IPSecPolicyProposalEncryption_1,
IPSecPolicyProposalAuthentication_1,
IPSecPolicyKeyExchangeMethodIndex, IPSecPolicyLocalIPAddressType;

IPSEC_SPD_TABLE 0 = 0, 10.11.2.21, 0, 0, 17, 900, 1,2, 2,2 ,1,0;

[ \IPSEC_SPD_TABLE ]
```

In the SPD example above, all packets designated to IP address 10.11.2.21 and originating from the OAMP interface (regardless to their destination and source ports) and whose protocol is UDP, are encrypted. The SPD also defines an SA lifetime of 900 seconds and two security proposals: DES/SHA1 and 3DES/SHA1. IPSec is performed using Transport mode.

8.1.3.3 IKE and IPSec Configuration Table's Confidentiality

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

- **Hidden IKE and IPSec tables -** When uploading the ***ini*** file from the gateway the IKE and IPSec tables are not available. Instead, the notifications (shown in the example below) are displayed. The following is an example of an ***ini*** File

Notification of Missing Tables.

```
;
;  *** TABLE IPSEC_IKEDB_TABLE ***
; This table contains hidden elements and will not be exposed.
; This table exists on board and will be saved during restarts
;

;
;  *** TABLE IPSEC_SPD_TABLE ***
; This table contains hidden elements and will not be exposed.
; This table exists on board and will be saved during restarts
;
```

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

```
i.e.
[IPSec_IKEDB_Table]
[\IPSec_IKEDB_Table]

[IPSEC_SPD_TABLE]
[\IPSEC_SPD_TABLE]
```

8.1.4 Dead Peer Detection (DPD) - RFC 3706

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

This is achieved by performing message exchanges between the peers. When no reply is received, the sender will assume the SAs are no longer valid on the remote peer and attempt to renegotiate.

DPD is auto negotiated. In order to activate DPD functionality, the IPSecDPDMode configuration parameter must be set to one of the following values:

0 - Disabled (default)

1 - Periodic - Message exchanges will be occur at regular intervals

2 - On-Demand - Message exchanges will occur as needed (for data transfers)

8.2 Secure Shell

The device command-line interface is used primarily for configuration and status and may be accessed using Telnet. Unless configured for TLS mode, Telnet is not secure, as it requires that passwords are transmitted in clear text. To overcome this, SSH (Secure SHell) is used, which is the de-facto standard for secure command-line interface. SSH 2.0 is a protocol built above TCP, and it provides methods for key exchange, authentication, encryption, and authorization.

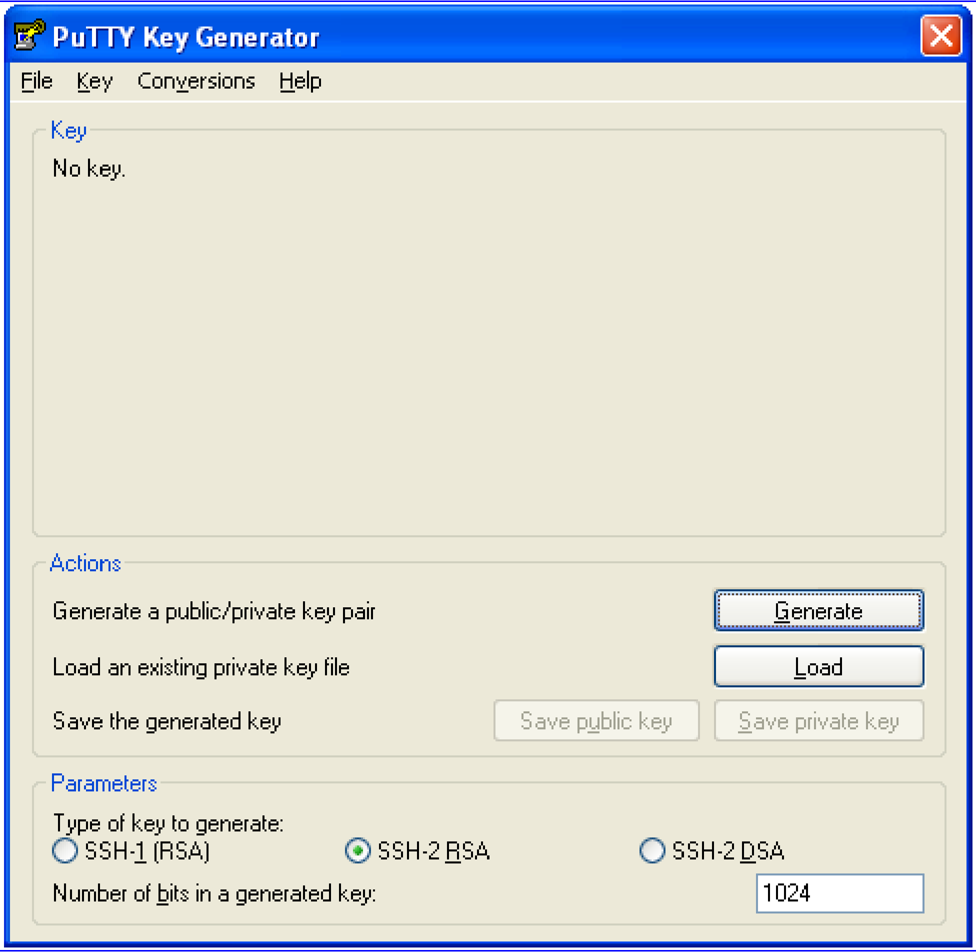
SSH requires appropriate client software for the management PC. Most Linux distributions have OpenSSH pre-installed; Windows-based PCs require SSH client software such as PuTTY, which can be downloaded from: http://www.chiark.greenend.org.uk/~sgtatham/putty/

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

➢ To configure RSA public keys on Windows (using PuTTY SSH software), take these steps:

1. Run **PuTTYgen.exe**. The PuTTY Key Generator screen appears.

Figure 8-2: PuTTY Key Generator



No key.

Generate a public/private key pair

Load an existing private key file

Save the generated key

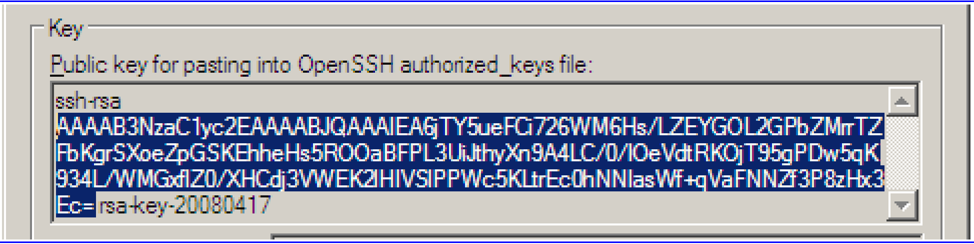
Type of key to generate:
○ SSH-1 (RSA) ● SSH-2 RSA ○ SSH-2 DSA

Number of bits in a generated key: 1024

2. Select **SSH-2 RSA** as the type of key to generate.
3. Select **1024** as the Number of bits in a generated key.
4. Click on the **Generate** button and follow the on-screen instructions.
5. Save the new private key to a file.

6. Copy the encoded text displayed on the top of the PuTTYgen window, between “ssh-rsa” and “rsa-key-….”. Refer to the example below.

Figure 8-3: PuTTY Example



Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAIEA6jTY5ueFCi726WM6Hs/LZEYGOL2GPbZMrrTZ
FbKgrSXoeZpGSKEhheHs5ROOaBFPL3UiJthyXn9A4LC/0/lOeVdtRKOjT95gPDw5qK
934L/WMGxflZ0/XHCdj3VWEK2lHIVSlPPWc5KLtrEc0hNNlasWf+qVaFNNZf3P8zHx3
Ec= rsa-key-20080417
```

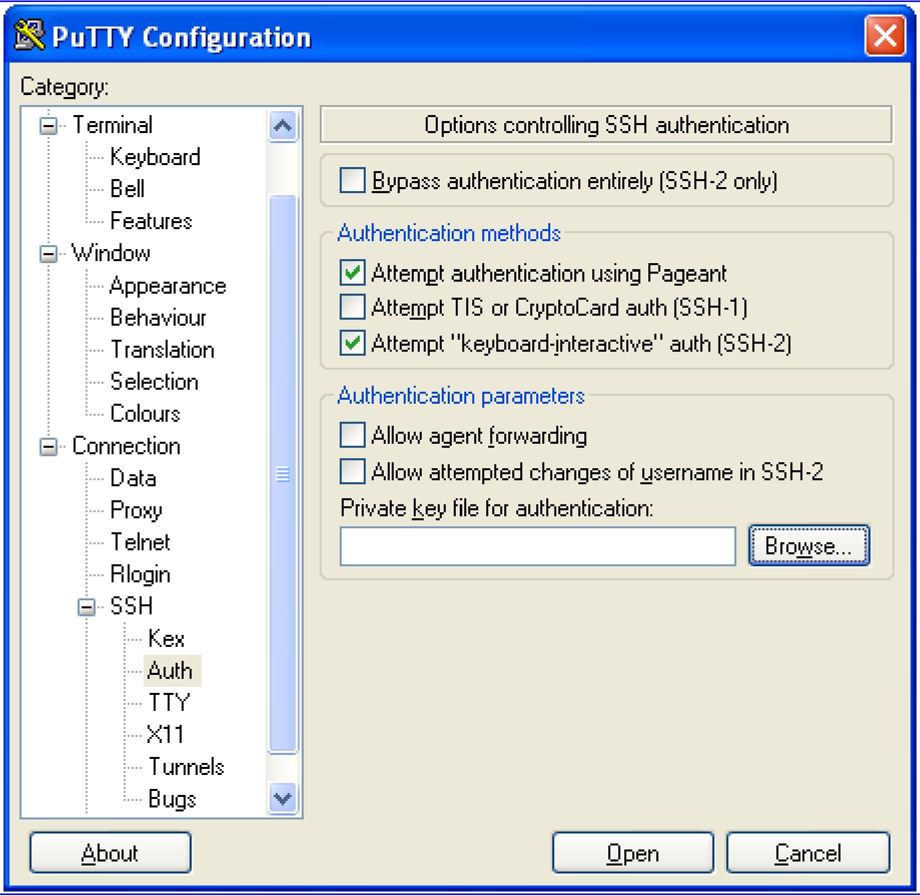
7. Edit the device's *ini* file and set the **SSHAdminKey** to the value copied above, e.g.:

```
SSHAdminKey = AAAAB3NzaC1yc2EAAAABJQ
```

8. Load the *ini* file to the device.

9. Open **PuTTY.exe**. Select **Connection** > **SSH** > **Auth** and press the **Browse** button to locate the private key file created in Step 5 above. The following screen appears.

Figure 8-4: PuTTY Configuration



Options controlling SSH authentication

□ Bypass authentication entirely (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant

□ Attempt TIS or CryptoCard auth (SSH-1)

☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

□ Allow agent forwarding

□ Allow attempted changes of username in SSH-2

Private key file for authentication:

10. Connect to the device using the **Admin** username. RSA key negotiation will take place automatically and no password will be required.

➢ To configure RSA public keys on Linux (using OpenSSH 4.3), take the following steps:

1. Run the following command:

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

A new key will be created in the **admin.key** file. The public section will be saved to the **admin.key.pub** file.

2. Open the **admin.key.pub** file and copy the long encoded string following "***ssh-rsa***" up to the blank space.
3. Edit the device's *ini* file and set the **SSHAdminKey** to the value copied above, e.g.:

```
SSHAdminKey = AAAAB3NzaC1yc2EAAAABJQ…
```

4. Load the *ini* file to the device.
5. Connect to the device using the ssh -i admin.key xx.xx.xx.xx command, where "xx.xx.xx.xx" is the IP address of the device. RSA key negotiation will take place automatically and no password will be required.

For additional security, set the SSHRequirePublicKey *ini* file parameter to 1. This will ensure SSH access is only possible using the RSA key, and not via username and password.

8.3 SSL/TLS

SSL (the Secure Socket Layer), also known as TLS (Transport Layer Security), is the method used to secure the device's Web interface and Telnet server. The SSL protocol provides confidentiality, integrity and authenticity of the Web server.

Specifications for the SSL/TLS implementation:

- Supported transports: SSL 2.0, SSL 3.0, TLS 1.0
- Supported ciphers: DES, aRC4, 3DES, AES
- Authentication: X.509 certificates; CRLs are not supported

Note: A common security practice is to disable SSLv2/SSLv3 and use only TLSv1. This can be achieved by setting the configuration parameter TLSVersion to 1. If using Microsoft Internet Explorer, make sure to disable SSL 2.0 / SSL 3.0 and enable TLS 1.0 under Tools > Internet Options > Advanced.

8.3.1 Web Server Configuration

For additional security, you can configure the Web server to accept only secure (HTTPS) connections. This is done by changing the HTTPSOnly ***ini*** file parameter, or via the Web interface, Network Settings screen (refer to the Network Settings section of the Web Interface in the product's User's Manual). You can also change the port number used for the secure Web server (by default 443) by changing the ***ini*** file parameter, HTTPSPort.

8.3.2 Using the Secure Web Server

➢ **To use the secure Web server, take these 3 Steps:**

1. Navigate your browser to the following URL:

https://[*hostname] or [ip address]*

Depending on the browser's configuration, a security warning dialog may be displayed. The reason for the warning is that the device's initial certificate is not trusted by your PC. The browser may allow you to install the certificate, thus skipping the warning dialog the next time you connect to the device

2. If you are using Internet Explorer 6, click **View Certificate** and then **Install Certificate**.
3. The browser also warns you if the host name used in the URL is not identical to the one listed in the certificate. To overcome this, add the IP address and host name (ACL_nnnnnn where nnnnnn is the serial number of the device) to your hosts file, located at /etc/hosts on UNIX or C:\WINDOWS\SYSTEM32\DRIVERS\ETC\hosts on Windows; then use the host name in the URL, e.g., https://ACL_280152 .Below is an example of a host file:

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

8.3.3 Secure Telnet

The device has an embedded Telnet server allowing easy command-line access to the device configuration and management interface. The Telnet server is disabled by default. To enable it, set the parameter, TELNETServerEnable to 1 (standard mode) or 2 (SSL mode).

No information is transmitted in the clear when using SSL mode.

If the Telnet server is set to SSL mode, a special Telnet client is required on your PC to connect to the Telnet interface over a secure connection; examples include C-Kermit for UNIX, Kermit-95 for Windows, and AudioCodes' acSSLTelnet utility for Windows (which requires prior installation of the free OpenSSL toolkit).

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

```
[ WelcomeMessage ]
FORMAT WelcomeMessage Index = WelcomeMessage Text ;
WelcomeMessage 01 = "WARNING! This computer system and network is
PRIVATE and PROPRIETARY and may" ;
WelcomeMessage 02 = "only be accessed by authorized users.
Unauthorized use of this computer" ;
WelcomeMessage 03 = "system or network is strictly prohibited and
may be subject to criminal" ;
WelcomeMessage 04 = "prosecution, employee discipline up to and
including discharge, or the" ;
WelcomeMessage 05 = "termination of vendor/service contracts. The
owner, or its agents, may" ;
WelcomeMessage 06 = "monitor any activity or communication on the
computer system or network." ;
```

```
WelcomeMessage 07 = "The owner, or its agents, may retrieve any
information stored within the" ;
WelcomeMessage 08 = "computer system or network. By accessing and
using this computer system or" ;
WelcomeMessage 09 = "network, you are consenting to such monitoring
and information retrieval for" ;
WelcomeMessage 10 = "law enforcement and other purposes. Users
should have no expectation of" ;
WelcomeMessage 11 = "privacy as to any communication on or
information stored within the computer" ;
WelcomeMessage 12 = "system or network, including information
stored locally or remotely on a hard" ;
WelcomeMessage 13 = "drive or other media in use with this computer
system or network." ;
[ /WelcomeMessage ]
```

8.3.4 Server Certificate Replacement

The device is shipped with a working SSL configuration consisting of a unique self-signed server certificate. If an organizational PKI (public key infrastructure) is in place, you may wish to replace this certificate with one provided by your security administrator.

➢ To replace this certificate, take these 9 steps:

1. Your network administrator should allocate a unique DNS name for the device (e.g., dns_name.corp.customer.com). This name is used to access the device, and should therefore be listed in the server certificate.
2. Navigate your browser to the device's Web interface. Select **Advanced Configuration> Security settings> Certificates**. The Certificate Signing Request Web page is displayed.
3. Enter the DNS name as the certificate subject (in the input box), and click **Generate CSR**. The Web page displays a textual certificate signing request, which contains the SSL device identifier
4. Copy this text and send it to your security provider. The security provider (also known as Certification Authority or CA) signs this request and will send you a Server Certificate for the device.
5. Save the certificate in a file (e.g., cert.txt) and make sure it is a plain-text file with the "BEGIN CERTIFICATE" header. Below is an example of a Base64-Encoded X.509 Certificate.

```
-----BEGIN CERTIFICATE-----
MIIDkzCCAnugAwIBAgIEAgAAADANBgkqhkiG9w0BAQQFADA/MQswCQYDVQQGEwJG
UjETMBEGA1UEChMKQ2VydGlwb3N0ZTEbMBkGA1UEAxMSQ2VydGlwb3N0ZSBTZXJ2
ZXVyMB4XDTk4MDYyNDA4MDAwMFoXDTE4MDYyNDA4MDAwMFowPzELMAkGA1UEBhMC
RlIxEzARBgNVBAoTCkNlcnRpcG9zdGUxGzAZBgNVBAMTEkNlcnRpcG9zdGUgU2Vy
dmV1cjCCASEwDQYJKoZIhvcNAQEBBQADggEOADCCAQkCggEAPqd4MziR4spWldGR
x8bQrhZkonWnNm+Yhb7+4Q67ecf1janH7GcN/SXsfx7jJpreWULf7v7Cvpr4R7qI
JcmdHIntmf7JPM5n6cDBv17uSW63er7NkVnMFHwK1QaGFLMybFkzaeGrvFm4k3lR
efiXDmuOe+FhJgHYezYHf44LvPRPwhSrzi9+Aq3o8pWDguJuZDIUP1F1jMa+LPwv
REXfFcUW+w==
-----END CERTIFICATE-----
```

6. Before continuing, set the parameter HTTPSOnly = 0 to make sure you have a method of accessing the device in case the new certificate is not working. Restore the previous setting after testing the configuration.
7. In the Certificates Web page, locate the server certificate upload section.

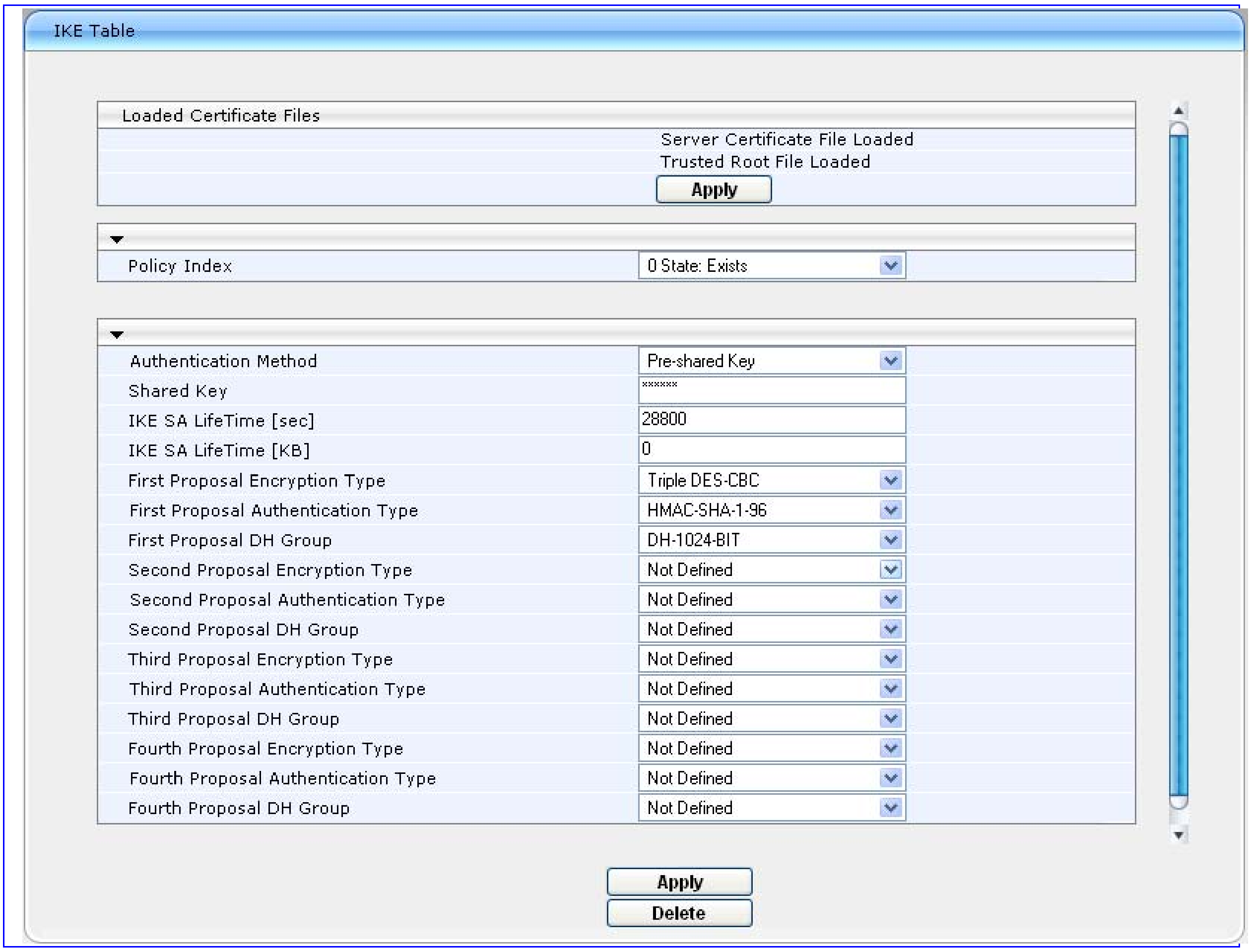
8. Click **Browse** and locate the *cert.txt* file, then click **Send File**.
9. When the operation is complete, save the configuration and restart the device. The Web server now uses the provided certificate.

> **Note 1:** The certificate replacement process may be repeated as necessary, e.g., when the new certificate expires.
>
> **Note 2:** It is possible to set the subject name to the IP address of the device (e.g., "10.3.3.1") instead of a qualified DNS name. This practice is not recommended, since the IP address is subject to changes and may not uniquely identify the device.

➢ To apply loaded certificates for IPsec negotiations, take these 2 steps:

1. Navigate to the **IKE Table** screen (Configuration>Security Settings>IKE Table). A new table, listing the newly-uploaded certificates should appear at the top of the screen, as shown below.

Figure 8-5: IKE Table



IKE Table

Loaded Certificate Files

Server Certificate File Loaded
Trusted Root File Loaded

Policy Index	0 State: Exists

Authentication Method	Pre-shared Key
Shared Key	******
IKE SA LifeTime [sec]	28800
IKE SA LifeTime [KB]	0
First Proposal Encryption Type	Triple DES-CBC
First Proposal Authentication Type	HMAC-SHA-1-96
First Proposal DH Group	DH-1024-BIT
Second Proposal Encryption Type	Not Defined
Second Proposal Authentication Type	Not Defined
Second Proposal DH Group	Not Defined
Third Proposal Encryption Type	Not Defined
Third Proposal Authentication Type	Not Defined
Third Proposal DH Group	Not Defined
Fourth Proposal Encryption Type	Not Defined
Fourth Proposal Authentication Type	Not Defined
Fourth Proposal DH Group	Not Defined

2. Press the **Apply** button to load certificates. Future IKE negotiations will be performed using the new certificates.

8.3.5 Using Self-Signed Certificates

As noted above, the device is shipped with a working self-signed server certificate. The subject name for this default certificate is "ACL_nnnnnnn" where nnnnnnn is the serial number of the device. This name may not be appropriate for your network.

This section describes how to change the subject name while still using self-signed certificates.

➢ To change the subject name and regenerate the self-signed certificate, take these 5 steps:

1. Your network administrator should allocate a unique DNS name for the device (e.g., dns_name.corp.customer.com). This name is used to access the device, and should therefore be listed in the server certificate.
2. Make sure the device is not processing any traffic. The certificate generation process is disruptive and should be executed during maintenance time.
3. Navigate your browser to the device's Web interface. Click on **Configuration** and then **Security Settings.** Select **Certificates**. The Certificate Signing Request Web page is displayed.

Enter the fully-qualified DNS name (FQDN) as the certificate subject (in the input box), and click Generate Self-signed. The web page will display a text message with the new subject name.

4. Save the configuration and restart the device for the new certificate to take effect.

Alternatively, the certificate may be re-generated using the CLI command CertificateMgmt (CM) in the Configuration directory:

```
/> /CONF/CM GENERATE dns_name.corp.customer.com
```

to generate a self-signed server certificate using the subject name "dns_name.corp.customer.com".

8.3.6 Client Certificates

By default, Web servers using SSL provide one-way authentication. The client is certain that the information provided by the Web server is authentic. When an organizational PKI is in place, two-way authentication may be required: both client and server should be authenticated using X.509 certificates. This is achieved by installing a client certificate on the management PC, and uploading the same certificate (in base64-encoded X.509 format) to the device's Trusted Root Certificate Store. The Trusted Root Certificate file should contain both the certificate of the authorized user, and the certificate of the CA.

Since X.509 certificates have an expiration date and time, the device must be configured to use NTP (Network Time Protocol) to obtain the current date and time. Without a correct date and time, client certificates cannot work.

➢ To enable two-way certificates, take these 7 steps:

1. Before continuing, set HTTPSONLY=0 to make sure you have a method of accessing the device in case the client certificate is not working. Restore the previous setting after testing the configuration.
2. To upload the Trusted Root Certificate file, go to the Certificates Web page as shown above and locate the trusted root certificate upload section.
3. Access the Certificates screen (Advanced Configuration -> Security Settings -> Certificates).

4. In the Certificates Web page, locate the server certificate upload section.
5. Click **Browse** and locate the file, then click **Send File**.
6. When the operation is complete, set the ***ini*** file parameter, HTTPSRequireClientCertificates = 1.
7. Save the configuration and restart the device.

   When a user connects to the secure Web server:

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

Note : The process of installing a client certificate on your PC is beyond the scope of this document. For more information, refer to your Web browser or operating system documentation, and/or consult your security administrator.

8.3.7 Certificate Revocation Checking

Some Public-Key Infrastructures support an ability to revoke a certificate after it is issued. AudioCodes devices which employ SSL/TLS and IPSec may be configured to check whether a peer's certificate has been revoked, using the Online Certificate Status Protocol (OCSP).

To enable OCSP, the following configuration parameters should be set:

```
OcspEnable
OcspServerIP
OcspServerPort
OcspDefaultResponse
```

See the individual ini file parameter documentation for these parameters for syntax and possible values.

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

The device will not query OCSP for its own certificate.

Note: Some PKIs do not support OCSP but generate Certificate Revocation Lists (CRLs). In this case, set up an OCSP server such as OCSPD.

8.3.8 Enhancing SSL/TLS Performance

Note: Applicable to all devices other than **Mediant 1000** and **MediaPack**.

In cases where cryptographic operation imposes an unacceptable computational overhead, you can enable the off-loading of cryptographic operations to a hardware crypto-processor.

To enable this feature, use the EnableTLSHW parameter (EnableTLSHW = 1).

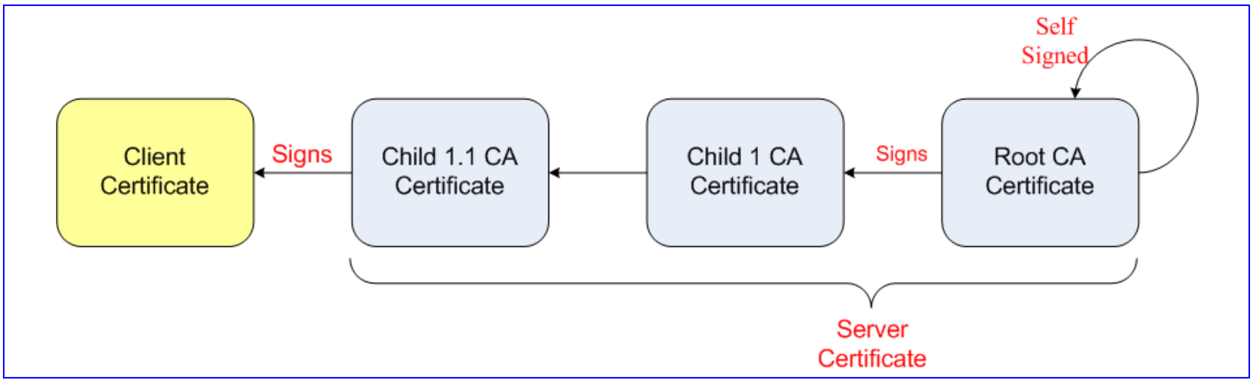
The supported Encryption algorithms are DES, 3DES & AES 128 bit.

Note: Enabling this parameter may reduce channel capacity (refer to Configuring IKE and IPSec on page 493).

8.3.9 Certificate Chain

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

Figure 8-6: Certificate Chain Hierarchy



Note: The chained certificate is limited to up to 9000 characters (including the certificate's headers).

8.4 RADIUS Support

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

When accessing the Web and Telnet servers, users must provide a valid username and password of up to 128 Unicode characters. When RADIUS authentication isn't used, the username and password are authenticated with the Web interface's usernames and passwords of the primary or secondary accounts or with the Telnet server's username and password stored internally in the gateway's memory. When RADIUS authentication is used, the gateway doesn't store the username and password but simply forwards them to the pre-configured RADIUS server for authentication (acceptance or rejection). The internal Web / Telnet passwords can be used as a fallback mechanism in case the RADIUS server doesn't respond. Note that when RADIUS authentication is performed, the Web / Telnet servers are blocked until a response is received (with a timeout of 5 seconds).

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

8.4.1 Setting Up a RADIUS Server

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

➢ **To set up a RADIUS server, take these 5 steps:**

1. Define the gateway as an authorized client of the RADIUS server, with a predefined 'shared secret' (a password used to secure communication) and a vendor ID. The figure below displays an example of the file clients.conf (FreeRADIUS client configuration).

Example of the File clients.conf (FreeRADIUS Client Configuration)

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

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

Example of a Dictionary File for FreeRADIUS (FreeRADIUS Client Configuration)

```
#
# AudioCodes VSA dictionary
```

```
#
VENDOR AudioCodes 5003
ATTRIBUTE ACL-Auth-Level 35 integer AudioCodes
VALUE ACL-Auth-Level ACL-Auth-UserLevel 50
VALUE ACL-Auth-Level ACL-Auth-AdminLevel 100
```

VALUE ACL-Auth-Level ACL-Auth-SecurityAdminLevel 200

```
#
# AudioCodes VSA dictionary
#
VENDOR AudioCodes 5003
ATTRIBUTE ACL-Auth-Level 35 integer AudioCodes
VALUE ACL-Auth-Level ACL-Auth-UserLevel 50
VALUE ACL-Auth-Level ACL-Auth-AdminLevel 100
```

3. In the RADIUS server, define the list of users authorized to use the gateway, using one of the password authentication methods supported by the server implementation. The following example shows a user configuration file for FreeRADIUS using a plain-text password.

Example of a User Configuration File for FreeRADIUS Using a Plain-Text Password

```
# users - local user configuration database

john    Auth-Type := Local, User-Password == "qwerty"
        Service-Type = Login-User,
        ACL-Auth-Level = ACL-Auth-SecurityAdminLevel

larry   Auth-Type := Local, User-Password == "123456"
        Service-Type = Login-User,
        ACL-Auth-Level = ACL-Auth-UserLevel
```

4. Record and retain the IP address, port number, 'shared secret', vendor ID and VSA access level identifier (if access levels are used) used by the RADIUS server.
5. Configure the gateway's relevant parameters according to the section below.

8.4.2 Configuring RADIUS Support

➢ **To configure RADIUS support on the gateway via the Web interface, take these 13 steps:**

1. Access the Web interface (refer to the Web Interface section in the product's User's Manual).
2. Open the 'General Security Settings' screen (Configuration > Security Settings > General Security Settings option); the 'General Security Settings' screen is displayed.
3. Under section 'General RADIUS Settings', in the field 'Enable RADIUS Access Control', select 'Enable'; the RADIUS application is enabled.
4. In the field 'Use RADIUS for Web / Telnet Login', select 'Enable'; RADIUS authentication is enabled for Web and Telnet login.
5. Enter the RADIUS server IP address, port number and shared secret in the relevant fields.
6. Under section 'RADIUS Authentication Settings', in the field 'Device Behavior Upon RADIUS Timeout', select the gateway's operation if a response isn't received from the RADIUS server after the 5 seconds timeout expires:
   - Deny Access – the gateway denies access to the Web and Telnet embedded servers.

- Verify Access Locally – the gateway checks the local username and password.

7. In the field 'Local RADIUS Password Cache Timeout', enter a time (in seconds); when this time expires, the username and password verified by the RADIUS server becomes invalid and a username and password must be re-validated with the RADIUS server.
8. In the field 'Local RADIUS Password Cache Mode', select the gateway's mode of operation regarding the above-mentioned 'Local RADIUS Password Cache Timer' option:
   - Reset Timer Upon Access – upon each access to a Web screen, the timer resets (reverts to the initial value configured in the previous step).
   - Absolute Expiry Timer - when you access a Web screen, the timer doesn't reset but rather continues decreasing.
9. In the field 'RADIUS VSA Vendor ID', enter the vendor ID you configured in the RADIUS server.
10. When using the Web access-level mechanism, perform one of the following options:
    - When RADIUS responses include the access level attribute:
In the field 'RADIUS VSA Access Level Attribute', enter the code that indicates the access level attribute in the Vendor Specific Attributes (VSA) section of the received RADIUS packet.
    - When RADIUS responses don't include the access level attribute:
In the field 'Default Access Level', enter the default access level that is applied to all users authenticated by the RADIUS server.
11. In the field 'Require Secured Web Connection (HTTPS)', select 'HTTPS only'.
    It is important you use HTTPS (secure Web server) when connecting to the gateway over an open network, since the password is transmitted in clear text.
    For Telnet, use SSL (TelnetServerEnable = 2) or SSH.
12. Save the changes so they are available in case of a power failure.
13. Reset the gateway. Click the Reset button on the main menu bar; the Reset screen is displayed. Click the button Reset.

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

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

1. Add the following parameters to the ***ini*** file. For information on modifying the ***ini*** file, refer to Modifying ***ini*** File Parameters via the AdminPage.
   - EnableRADIUS = 1
   - WebRADIUSLogin = 1

RADIUSAuthServerIP = IP address of RADIUS server

RADIUSAuthPort = port number of RADIUS server, usually 1812

SharedSecret = your shared secret'

- HTTPSOnly = 1
- RadiusLocalCacheMode = 1
- RadiusLocalCacheTimeout = 300
- RadiusVSAVendorID = your vendor's ID
- RadiusVSAAccessAttribute = code that indicates the access level attribute
- DefaultAccessLevel = default access level (0 to 200)

The following table lists the different RADIUS Authentication Settings.

Table 8-5: RADIUS Authentication Settings

Local RADIUS Password Cache Mode [RadiusLocalCacheMode]	Defines the gateway's mode of operation regarding the timer (configured by the parameter RadiusLocalCacheTimeout) that determines the validity of the username and password (verified by the RADIUS server). 0 (Absolute Expiry Timer) = when you access a Web screen, the timeout doesn't reset but rather continues decreasing. 1 (Reset Timer Upon Access) = upon each access to a Web screen, the timeout always resets (reverts to the initial value configured by RadiusLocalCacheTimeout).
Local RADIUS Password Cache Timeout [RadiusLocalCacheTimeout]	Defines the time (in seconds) the locally stored username and password (verified by the RADIUS server) are valid. When this time expires, the username and password becomes invalid and must be re-verified with the RADIUS server. The valid range is 1 to 0xFFFFFF. -1 = Never expires. 0 = Each request requires RADIUS authentication. The default value is 300 (5 minutes).
RADIUS VSA Vendor ID [RadiusVSAVendorID]	Defines the vendor ID the gateway accepts when parsing a RADIUS response packet. The valid range is 0 to 0xFFFFFFFF. The default value is 5003.
RADIUS VSA Access Level Attribute [RadiusVSAAccessAttribute]	Defines the code that indicates the access level attribute in the Vendor Specific Attributes (VSA) section of the received RADIUS packet. The valid range is 0 to 255. The default value is 35.
Default Access Level [DefaultAccessLevel]	Defines the default access level for the gateway when the RADIUS (authentication) response doesn't include an access level attribute. The valid range is 0 to 255. The default value is 200 (Security Administrator').

2. Authenticating via RADIUS with credentials in the URL:
   - The device is capable of authenticating via RADIUS server when the UserName/Password are in the URL, e.g.:
   - http://10.4.4.112/Forms/RadiusAuthentication?WSBackUserName=Guyy&WSBackPassword=1234
   - This method is applicable when using RADIUS server with HTTP basic authentication. Note that only one connection is possible at a time.
3. To set this feature, use Radius with Basic authentication settings:
   - **a.** Default settings - You are prompted for your login every time you connect to the device.
   - **b.** Enable Radius configuration as described above.
   - **c.** Enable Basic HTTP authentication settings.
   - **d.** Connect to the device using a URL as in the example.

Note: This feature is restricted to 5 users simultaneously only.

8.5 Internal Firewall

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

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

The access list consists of a table with up to 50 ordered lines. **(For TP-6310, Mediant 3000 and IPmedia 3000, up to 25 ordered lines.)**

For each packet received on the network interface, the table is scanned from the top until a matching rule is found (or the table end is reached). This rule can either block the packet or allow it; however it is important to note that subsequent rules will not be scanned. If the table end is reached without a match, the packet is accepted.

Each rule is made up of the following fields:

Table 8-6: Internal Firewall Fields

Parameter	Description
Source IP [AccessList_Source_IP]	IP address (or DNS name) of source network, or a specific host
Mask [AccessList_Net_Mask]	IP network mask. 255.255.255.255 for a single host, or the appropriate value for the source IP addresses The IP address of the sender of the incoming packet is bitwise ANDed with this mask and then compared to the field 'Source IP'.
Local Port Range [AccessList_Start_Port] [AccessList_End_Port]	The destination UDP/TCP ports (on this device) to which packets are sent. The valid range is 0 to 65535. Note: When the protocol type is not TCP or UDP, the entire range must be provided.
Protocol [AccessList_Protocol]	The protocol type (e.g., UDP, TCP, ICMP, ESP or 'Any'), or the IANA protocol number (in the range of 0 (Any) to 255). Note: The protocol field also accepts the abbreviated strings 'SIP', 'MGCP', 'MEGACO', 'HTTP'. Specifying these strings imply selection of the TCP or UDP protocols, and the appropriate port numbers as defined on the device.
Packet Size [AccessList_Packet_Size]	Maximum allowed packet size. The valid range is 0 to 65535. Note: When filtering fragmented IP packets, the Packet Size field relates to the overall (reassembled) packet size, not to the size of each fragment.

Table 8-6: Internal Firewall Fields

Parameter	Description
Byte Rate [AccessList_Byte_Rate]	Expected traffic rate (bytes per second).
Burst Bytes [AccessList_Byte_Burst]	Tolerance of traffic rate limit (number of bytes)
Action Upon Match [AccessList_Allow_Type]	Action upon match (allow or block)
Match Count [ACCESSLIST_MatchCount]	A read-only field that provides the number of packets accepted / rejected by a specific rule.

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

```
[ ACCESSLIST ]
FORMAT ACCESSLIST Index = ACCESSLIST Source IP,
ACCESSLIST Net Mask, ACCESSLIST Start Port, ACCESSLIST End Port,
ACCESSLIST_Protocol, ACCESSLIST_Packet_Size, ACCESSLIST_Byte_Rate,
ACCESSLIST Byte Burst, ACCESSLIST Allow Type;

ACCESSLIST 10 = mgmt.customer.com, 255.255.255.255, 0, 80, tcp, 0,
0, 0, allow ;
ACCESSLIST 15 = 192.0.0.0, 255.0.0.0, 0, 65535, any, 0, 40000,
50000, block ;
ACCESSLIST 20 = 10.31.4.0, 255.255.255.0, 4000, 9000, any, 0, 0, 0,
block ;
ACCESSLIST 22 = 10.4.0.0, 255.255.0.0, 4000, 9000, any, 0, 0, 0,
block ;
[ \ACCESSLIST ]
```

The following is an explanation of the example access list:

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

More complex rules may be defined, relying on the "single-match" process described below:

The following is an advanced example of an access list definition via ***ini*** file:

```
[ ACCESSLIST ]
FORMAT ACCESSLIST Index = ACCESSLIST Source IP,
ACCESSLIST Net Mask, ACCESSLIST Start Port, ACCESSLIST End Port,
ACCESSLIST Protocol, ACCESSLIST Packet Size, ACCESSLIST Byte Rate,
ACCESSLIST Byte Burst, ACCESSLIST Allow Type;

ACCESSLIST 10 = 10.0.0.0, 255.0.0.0, 0, 65535, any, 0, 40000,
50000, allow ;
```

```
ACCESSLIST 15 = 10.31.4.0, 255.255.255.0, 4000, 9000, any, 0, 0, 0,
allow ;
ACCESSLIST 20 = 0.0.0.0, 0.0.0.0, 0, 65535, any, 0, 0, 0, block;
 [ \ACCESSLIST ]
```

The following is an explanation of the example access list:

This access list consists of three rules:

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

The internal firewall can also be configured via the Web interface (refer to the Firewall Settings section of the Web Interface in the product's User's Manual). Note that when creating access rules via the Web interface, it is necessary to click the Activate button after reviewing the rule's fields.

8.6 Network Port Usage

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

Table 8-7: Default TCP/UDP Network Port Numbers

Port number	Peer port	Application	Notes
2	2	Debugging interface	Always ignored
4	4	EtherDiscover	Open only on unconfigured devices
22	-	SSH	Disabled by default (SSHServerEnable). Configurable (SSHServerPort), access controlled by WebAccessList.
23	-	Telnet	Disabled by default (TELNETSERVERENABLE). Configurable (TELNETSERVERPORT), access controlled by WebAccessList
68	67	DHCP	Active only if DHCPENABLE=1
80	-	Web server (HTTP)	Configurable (HTTPPORT), may be disabled (DISABLEWEBTASK or HTTPSONLY). Access controlled by WEBACCESSLIST
161	-	SNMP GET/SET	Configurable (SNMPPORT), may be disabled (DISABLESNMP). Access controlled by SNMPTRUSTEDMGR
443	-	Web server (HTTPS)	Configurable (HTTPSPORT), may be disabled (DISABLEWEBTASK). Access

Table 8-7: Default TCP/UDP Network Port Numbers

Port number	Peer port	Application	Notes
			controlled by WEBACCESSLIST
500	-	IPSec IKE	May be disabled (ENABLEIPSEC)
2422	2422	TPM LinkLayer	Used for internal synchronization between the two TPMs on a device **(Applicable to 1610 blades only)**
2423-2424	2423 and up	TPNCP	Proprietary control protocol. Access controlled by ENABLETPNCPSECURITY and AUTHORIZEDTPNCPSERVERS
2427 or 2944	2427 or 2944	MGCP / Megaco	Configurable (GATEWAYMGCPPORT), Access controlled by PROVISIONEDCALLAGENTS and MEGACOCHECKLEGALITYOFMGC
4000, 4010 and up	-	RTP traffic	Base port number configurable (BASEUDPPORT), fixed increments of 10. The number of ports used depends on the channel capacity of the device.
4001, 4011 and up	-	RTCP traffic	Always adjacent to the RTP port number
4002, 4012 and up	-	T.38 traffic	Always adjacent to the RTCP port number
32767	-	SCTP	If SCTP/IUA is available on the device
(random) > 32767	514	Syslog	Configurable (SyslogServerPort) May be disabled (ENABLESYSLOG).
(random) > 32767	-	Syslog ICMP	May be disabled (ENABLESYSLOG).
(random) > 32767	-	ARP listener	
(random) > 32767	162	SNMP Traps	May be disabled (DISABLESNMP)
(random) > 32767	-	DNS client	

8.7 Media Security

Note :

This sub-section on Packet Cable Security is not applicable to **MediaPack** and **Mediant 1000**.

8.7.1 Packet Cable Security

The device supports media encryption via TGCP (PacketCable extensions to MGCP protocol) and via the proprietary VoPLib API. With media security, IP voice traffic for some or all channels is encrypted using predefined session keys. No key negotiation is performed for media security. Instead, the device assumes higher-level protocols handle key management.

Encryption specifications:

- AES (Rijndael) cipher algorithm, in CBC mode
- Key strength - 128 bit
- Encryption key supplied by TGCP or manually via VoPLib API

The VoPLib API may be used over the network (TPNCP protocol). Media security over TPNCP should be used with caution, since the TPNCP connection itself is not encrypted, and sniffing techniques may be used to obtain the session key. The same is applicable for TGCP connections. Physical security is required to make sure the softswitch connection is protected from unauthorized sniffing.

Note : Using media security reduces the channel capacity of the device. Refer to the relevant product's Release Notes document for more information.

For further information regarding the VoPLib API, consult the "VoPLib API Reference Manual", Document #: LTRT-840xx.

8.7.2 Secure RTP

The device supports Secure RTP (SRTP) as defined in RFC 3711. SRTP provides confidentiality, message authentication, and replay protection to the RTP & RTCP traffic.

Key negotiation is not part of SRTP. Instead, the device assumes higher-level protocols handle key management.

Specifications:

- Encryption - AES 128 in Counter Mode
- Authentication - HMAC-SHA1
- Support of Key Derivation
- Key management is provided via VoPLib API, MGCP and MECAGO

The VoPLib API may be used over the network (TPNCP protocol). Media security over TPNCP should be used with caution, since the TPNCP connection itself is not encrypted, and sniffing techniques may be used to obtain the session key. Physical security is required to make sure the softswitch connection is protected from unauthorized sniffing.

Note : Using media security reduces the channel capacity of the device.

For further information regarding the VoPLib API, consult the "VoPLib API Reference Manual", Document #: LTRT-840xx.

8.8 Recommended Practices

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

- Set the management password to a unique, hard-to-guess string. Do not use the same password for several devices, as a compromise of one may lead to the compromise of others. Keep this password safe at all times, and change it frequently.
- If possible, use a RADIUS server for authentication. RADIUS allows you to set different passwords for different users of the device, with centralized management of the password database. Both Web and Telnet interfaces support RADIUS authentication.
- Use IPSec to secure traffic to all management and control hosts. Since IPSec encrypts all traffic, hackers cannot capture sensitive data transmitted on the network, and malicious intrusions are severely limited.
- Use HTTPS when accessing the Web interface. Set HTTPSONLY=1 to allow only HTTPS traffic (and block port 80). If you don't need the Web interface, disable the Web server.
- If you use Telnet, do not use the default port (23). Use SSL mode to protect Telnet traffic from network sniffing.
- If you use SNMP, do not leave the community strings at their default values, as they can be easily discovered by hackers. See the SNMP configuration chapter for further details.
- Use a firewall to protect your VoIP network from external attacks. Robustness of the network may be compromised if the network is exposed to "denial of service" (DoS) attacks; such attacks are mitigated by statefull firewalls. Do not allow unauthorized traffic to reach the device.

8.9 Legal Notice

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

This device includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. Refer to www.openssl.org/.

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

Reader’s Notes

9 Diagnostics & Troubleshooting

9.1 Diagnostics Overview

A wide range of diagnostic tools are provided to enable the user to easily identify an error condition and to provide a solution or work-around when working with the device.

- LED Indication of channel activity status, data, control and LAN status
- MediaPack Self-Testing on hardware initialization
- Error/Notification Messages via the following interfaces:
    - RS-232 terminal
    - Syslog
    - Control Protocols:
        - MGCP
    - SNMP
- Solutions to Common Problems

They are described in the following pages.

9.2 Troubleshooting MediaPack Devices via the RS-232 Port

To troubleshoot initialization problems and view the status and error messages of the MediaPack, use serial communication software (e.g., HyperTerminal^{TM}) to connect to the MediaPack via the RS-232 port. You can also use this connection to change the network settings (IP address, subnet mask and default gateway IP address) of the MediaPack.

To connect the MP-11x RS-232 port to your PC, refer to Connecting the MP-11x RS-232 Port to Your PC. To connect the MP-124 RS-232 port to your PC, refer to Connecting the MP-124 RS-232 Port to Your PC.

9.2.1 Viewing the Gateway's Information

After applying power to or resetting the gateway, the information, shown in the example below, appears on the terminal screen. This information is used to determine possible MediaPack initialization problems, such as incorrectly defined (or undefined) Local IP address, subnet mask, default router IP address, TFTP server IP address, BootFile name, ***ini*** file name and Full/Half duplex network state. Below is an example of Status and Error Messages.

```
MAC address = 00-90-8F-01-00-9E
Local IP address = 10.1.37.6
Subnet mask = 255.255.0.0
Default gateway IP address = 10.1.1.5
TFTP server IP address = 10.1.1.167
Boot file name = ram35136.cmp
INI file name = mp108.ini
Call agent IP address = 10.1.1.18
Log server IP address = 0.0.0.0
Full/Half Duplex state = HALF DUPLEX
Flash Software Burning state = OFF
```

```
Serial Debug Mode = OFF
Lan Debug Mode = OFF

BootLoad Version 1.75
Starting TFTP download... Done.
MP108 Version 3.80.00
```

9.2.2 Changing the Networking Parameters

You can use the serial connection to change the network settings (IP address, subnet mask and default gateway IP address) of the MediaPack.

➢ To change the network settings via RS-232, take these 4 steps:

1. At the prompt type **conf** and press **Enter**. The configuration command shell is activated.
2. To check the current network parameters, at the prompt, type **GCP IP** and press **Enter**. The current network settings are displayed.
3. To change the network settings, type **SCP IP [ip_address] [subnet_mask][{default_gateway]** (e.g., "SCP IP 10.13.77.7 255.255.0.0 10.13.0.1"). The new settings take effect immediately. Connectivity is active at the new IP address.

Note 1: This command requires you to enter all three network parameters.

Note 2: Consult your network administrator before setting these parameters.

4. To save the configuration, at the prompt, type SAR. And press Enter. The MediaPack restarts with the new network settings.

9.2.3 Determining MediaPack Initialization Problems

Possible initialization problems encountered with the MediaPack can be determined by viewing the HyperTerminal screen after performing a hot hardware reset. Possible initialization problems are listed in the table below. (LED indicators located on the front view of the MediaPack provide first indication that the device has an initialization problem. Refer to LED Indicators on page 526 for a description of the LED visual indicators.)

Table 9-1: Possible Initialization Problems

Parameter	Problem Definition
Local IP address	Undefined/incorrectly defined
Subnet Mask	Undefined/incorrectly defined
Default gateway IP address	Undefined/incorrectly defined
TFTP server IP address	Undefined/incorrectly defined
Boot file name	Undefined/incorrectly defined/missing
ini file name	Undefined/incorrectly defined/missing

Table 9-1: Possible Initialization Problems

Parameter	Problem Definition
Call Agent IP address	Undefined/incorrectly defined
Log server IP address	Undefined/incorrectly defined
Full/Half Duplex state	Undefined/incorrectly defined
Flash Software Burning state	Undefined/incorrectly defined
Serial Debug Mode	Undefined/incorrectly defined
BootLoad version	Incorrect

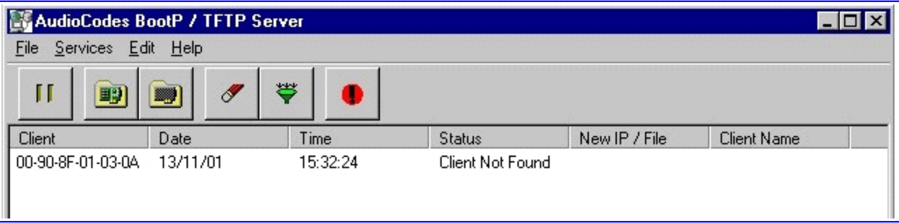
9.2.4 Reinitializing the MediaPack

If an initialization problem is encountered, reinitialize the MediaPack. To reinitialize the MediaPack, a BootP/TFTP Server application must be installed in your management PC. Reinitializing the MediaPack using the BootP/TFTP Server enables you to quickly get started with the MediaPack. For a detailed description of the BootP/TFTP Server Configuration Tool, including installation and configuration, refer to BootP Server.

➢ **To reinitialize the MediaPack, take the next 13 steps:**

1. Install the BootP/TFTP Server Configuration Tool from the Software CD, Document # LSTC00005 (MediaPack Series), refer to BootP Server.
2. Open the BootP/TFTP Server from Start>Programs>BootP. The BootP/TFTP Server main screen opens:

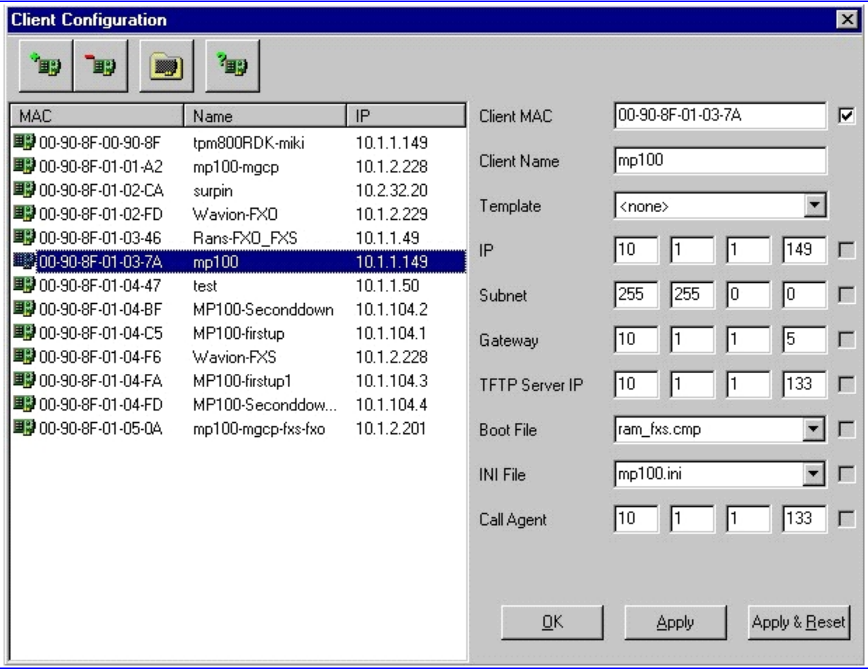
Figure 9-1: BootP/TFTP Server Main Screen



Client	Date	Time	Status	New IP / File	Client Name
00-90-8F-01-03-0A	13/11/01	15:32:24	Client Not Found		

3. In the Services menu, choose Edit Clients. Alternately, double-click on the Client Not Found log entry. The Client Configuration screen appears. (Refer to the figure below). The parameter fields displayed on the right side of the screen constitute the MediaPack software profile configuration. For a Client Not Found, the parameter fields are all blank.

Figure 9-2: Client Configuration



Client Configuration

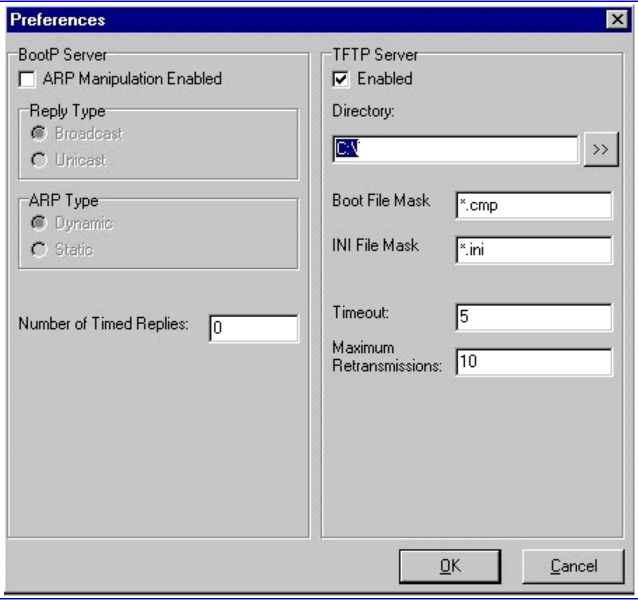
MAC	Name	IP
00-90-8F-00-90-8F	tpm800RDK-miki	10.1.1.149
00-90-8F-01-01-A2	mp100-mgcp	10.1.2.228
00-90-8F-01-02-CA	surpin	10.2.32.20
00-90-8F-01-02-FD	Wavion-FXO	10.1.2.229
00-90-8F-01-03-46	Rans-FXO_FXS	10.1.1.49
00-90-8F-01-03-7A	mp100	10.1.1.149
00-90-8F-01-04-47	test	10.1.1.50
00-90-8F-01-04-BF	MP100-Seconddown	10.1.104.2
00-90-8F-01-04-C5	MP100-firstup	10.1.104.1
00-90-8F-01-04-F6	Wavion-FXS	10.1.2.228
00-90-8F-01-04-FA	MP100-firstup1	10.1.104.3
00-90-8F-01-04-FD	MP100-Seconddow...	10.1.104.4
00-90-8F-01-05-0A	mp100-mgcp-fxs-fxo	10.1.2.201

Client MAC: 00-90-8F-01-03-7A ☑
Client Name: mp100
Template: <none>
IP: 10 1 1 149 □
Subnet: 255 255 0 0 □
Gateway: 10 1 1 5 □
TFTP Server IP: 10 1 1 133 □
Boot File: ram_fxs.cmp □
INI File: mp100.ini □
Call Agent: 10 1 1 133 □

4. Enter the reported MediaPack MAC address (labeled on the underside of the device) in the Client MAC field.
5. Enter the Client Name.
6. Enter the IP address (such as 10.1.1.33).
7. Enter the Subnet (such as 255.255.255.0) and set the Subnet to a valid value in accordance with the IP address. (That is, class C IP addresses can only have subnet starting with 255.255.255.X, while class B IP addresses can only have subnet starting with 255.255.X.X, and class A IP addresses can only have subnet starting with 255.X.X.X).)
8. Enter the IP address of the default Gateway. It can be any address within the subnet.
9. Enter the Call Agent IP address.

10. Upload the *ram_fxs.cmp* and the *mp_fxs.ini* configuration files by opening the Edit menu and choosing Preferences. The Preferences screen appears.

Figure 9-3: 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:\

Boot File Mask: *.cmp

INI File Mask: *.ini

Timeout: 5

Maximum Retransmissions: 10

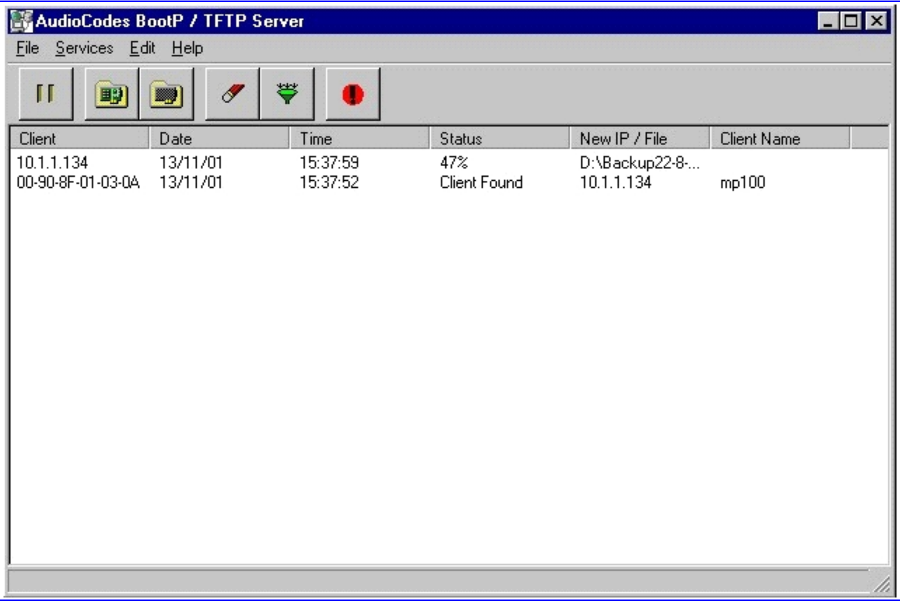
11. In the Directory field, click on the >> button and navigate to the directory of the source *cmp* and *ini* files.

If they are not already on your hard disk (C:), copy them to it (under a directory you should create called C:\\AudioCodes\). If you do not have the MediaPack Software CD from which to copy the *cmp* and ***ini*** files, contact support@audiocodes.com.

12. Click OK. The *cmp* and ***ini*** files are uploaded.

13. Perform a hot hardware reset or cold reset. The MediaPack initializes and the following status messages should be displayed in the BootP/TFTP Server main screen:

Figure 9-4: BootP/TFTP Server - Client Found



AudioCodes BootP / TFTP Server

Client	Date	Time	Status	New IP / File	Client Name
10.1.1.134	13/11/01	15:37:59	47%	D:\Backup22-8-...	
00-90-8F-01-03-0A	13/11/01	15:37:52	Client Found	10.1.1.134	mp100

9.2.5 LED Indicators

All LED indicators are described in the tables in Front LED Indicators and Rear LED Indicators.

9.2.5.1 MediaPack Front View LED Indicators

The full range of the MediaPack includes a front view displaying LED Indications of channel activity status, data, control and LAN status.

9.3 Syslog

The Syslog server (refer to the figure below) enables filtering of messages according to priority, IP sender address, time, date, etc. Users may choose to download and use the following examples of the many Syslog servers available as shareware on the Internet:

- Kiwi Enterprises: http://www.kiwisyslog.com/downloads.php
- The US CMS Server: uscms.fnal.gov/hanlon/uscms_server/
- TriAction Software: www.triaction.nl/Products/SyslogDaemon.asp
- Netal SL4NT 2.1 Syslog Daemon: www.netal.com

Syslog protocol is an event notification protocol that allows a device to send event notification messages across IP networks to event message collectors - also known as Syslog servers. Syslog protocol is defined in the IETF 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

9.3.1 Operating the Syslog Server

9.3.1.1 Sending Syslog Messages

The 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
```

- Warning level message

```
<132>Release contains no h.225 Reason neither q.931 Cause
information stateMode: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
```

- Debug level message:

```
<135>SCTP port 2905 was initialized
```

9.3.1.2 Setting the Syslog Server IP Address and Port

➢ To set the address of the Syslog server:

- Use the Web interface or the 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 an ***ini*** file example showing this parameter, refer to 'Setting Syslog Server IP Address, Enabling Syslog, in an ***on page 527ini*** File' on page 527 and to the Example of Setting Syslog Server IP Address, Enabling Syslog, in an ***ini*** File below.

9.3.1.3 Activating the Syslog Client

➢ To activate the Syslog client:

- Use the Web interface. Refer to the Web interface in the product's User's Manual.
- Alternately, use the BootP/TFTP Server to send the ***ini*** configuration file containing the parameter EnableSyslog to the device. For an ***ini*** file example showing this parameter, refer to the example below.

The example below shows:

- an ***ini*** file section with an example configuration for the address parameter SyslogServerIP
- configuration for the client activation parameter EnableSyslog
- configuration for the Syslog Server Port parameter SyslogServerPort

```
[Syslog]
SyslogServerIP=10.2.0.136
EnableSyslog =1
SyslogServerPort =601
```

9.4 The Web Interface's 'Message Log' (Integral Syslog)

The Message Log screen in the Web interface, similar to a Syslog server only integral to the Web server, displays debug messages useful for debugging. For detailed information, refer to the Message Log sub-section under Management Functions in the Product Reference Manual. 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 'Syslog' on page 526.

9.5 Control Protocol Reports

9.5.1 TPNCP Error Report

When working with the AudioCodes proprietary TPNCP (TrunkPack Network Control Protocol), the device 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, “Blade Originated Error Codes,” in the “VoPLib API Reference Manual”, Document #: LTRT-844xx.

Examples of using the Log Event Report Mechanism are also shown in the “VoPLib API Reference Manual”, Document #: LTRT-844xx.

9.5.2 MGCP/MEGACO Error Conditions

When working with MGCP/MEGACO, the device reports error conditions via the Call Manager (or via a Call Manager of the user’s choice) using the standard MGCP/MEGACO facilities, through the network interface. For more information on MGCP/MEGACO error conditions, refer to RFC 3435/3661 for MGCP and RFC 3015 for MEGACO.

9.5.3 MEGACO Error Conditions

When working with MEGACO, the device reports error conditions via the Call Manager (or via a Call Manager of the user’s choice) using the standard MEGACO facilities, through the network interface. For more information on MEGACO error conditions, refer to the IETF Website at:

http://www.ietf.org/rfc/

Refer to RFC 3015.

9.5.4 SNMP Traps

Note: This sub-section on SNMP Traps is not applicable to **260** devices.

Devices support various SNMP traps via the SNMP Agent running on the device. Among these traps are Trunk MIB traps, acBoardStarted and acResetingBoard traps. Refer to Using SNMP on page 64 for more details on all SNMP traps available on the device.

9.6 Solutions to Possible Problems

9.6.1 Solutions to Possible Common Problems

Solutions to possible common problems are described in the table below.

Table 9-2: Solutions to Possible Common Problems

Problem	Probable Cause	Solutions
No communication	Software does not function in the device	Try to "ping" the device/module. If ping fails, check for network problems/definitions and try to reset the device/module.
	Network problem	Check the cables.
	Network definitions	Check if the default gateway can reach the IP of the device/module.
		Check if the device/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 device/module	Check if the BootP server replied to the device/module at restart by viewing the log of the BootP server.
		Try to restart the BootP server.
		Check the MAC address of the device/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. Check the "next server" configuration in the BootP server.
	Device didn't request the file from your TFTP	Check the "next server" configuration in the BootP server.
	TFTP server bug	Try to restart the TFTP server.
	BootP sent to a device 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

Table 9-2: Solutions to Possible Common Problems

Problem	Probable Cause	Solutions
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.
	ini file corrupted	Check the ***ini*** file syntax.
BootP reply from wrong BootP server	Other BootP servers contain the MAC address of the device/module	Check that only your BootP server contains the device's MAC address.

9.6.2 Solutions to Possible Voice Problems

Solutions to possible voice problems are described in the table below.

Table 9-3: Solutions to Possible Voice Problems

Problem	Probable 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 equal to local.
		Check that the correct Mu-law or A-law compression is in use.
No voice	There is no match in the codecs.	Change the codec definition.
Echo problems	Any increase of the dB value of the Voice Volume and/or Input Gain parameters may increase the echo level. The default setting of these parameters is 0 dB.	When changing these parameters, please be aware that it can increase the echo level. These changes must be avoided or done very carefully.
	Echo problems due to wrong synchronization of the PSTN E1/T1 interfaces.	If the TDM clocks are not synchronized, echo problems are observed. Check your clock configurations.
	Acoustic echo	Acoustic echo can occur when the echo cancelation test is done between two phones physically located in the same room. Any tests of echo cancellation must be done using phones in separate rooms.

Table 9-3: Solutions to Possible Voice Problems

Problem	Probable Cause	Solutions
	When the echo is heard by an FXS/FXO user that is connected to the gateway (to the PSTN side of the Gateway).	This does not indicate any problem in the gateway. The problem is in the echo canceler at the other side (the far side).
	Network delay	Decrease jitter buffer size.

9.6.3 User Error Messages

Error code: 0x22008

Message: CreateOfAuditService() Error during creation of Audit Packets Queue

Explanation: An internal error was detected creating buffers for PSTN traces.

System action: PSTN trace will not be available, but system startup continues normally.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x22009

Message: Error during create of SendAuditPackets_Task

Explanation: An internal error was detected creating buffers for PSTN traces.

System action: PSTN trace will not be available, but system startup continues normally.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x22010

Message: CreateOfAuditService() Error during start of SendAuditPackets_Task

Explanation: An internal error was detected creating buffers for PSTN traces.

System action: PSTN trace will not be available, but system startup continues normally.

User Response: Contact AudioCodes customer support.

Source: PSTN

Message: PSTNcheckBoardParamsOfIdlePatterns() - You Can`t use idle ABCD=0000 when you use E1. changed to 0XF.

Explanation: A CAS trunk (CAS protocol or RAW_CAS) was configured over E1 with wrong idle ABCD value (INI file parameter “IdleABCDPattern”). The ABCD bits value should be nonzero on E1.

System action: The idle ABCD pattern is changed to 1111 (0xF).

User Response: Check the ABCD configuration in case a different value is required. If you are using the Web interface, use the TDM settings page to reconfigure the idle ABCD pattern, save configuration and reset the device.

Source: PSTN

Error code: 0x23014

Message: PSTNcheckBoradParamsOfIdlePatterns() - Your idle ABCD=0000 while using E1. It can cause problems if you use CAS.

Explanation: This notice message warns that INI file parameter “IdleABCDPattern” has a value of zero. There is nothing wrong with the current configuration but it will cause problem in the future if trunk configuration is set to CAS (protocol type CAS or RAW_CAS). It is not possible to change the INI file parameter “IdleABCDPattern” online, only offline.

System action: In case the trunk protocol type is CAS or RAW_CAS the pattern is changed automatically, otherwise there is no impact, other than a future limitation on the possible PSTN protocols.

User Response: Change this value to allow on-the-fly configurations changes (no device reset) for CAS/RAW_CAS protocols in the future. If you are using the Web interface, use the TDM settings page to reconfigure the idle ABCD pattern, save configuration and reset the device.

Source: PSTN

Error code: 0x23014

Message: PSTNcheckBoradParamsOfIdlePatterns() - You Can`t use idle PCM=p when you use T1 with D4 (the 2nd bit is 0). changed to 0xFF.

Explanation: INI file parameter “IdlePCMPattern” was set with second bit zeroed while setting the trunk to T1 with D4 framing (acT1_FRAMING_F12=’B’ or acSUPER_FRAME=’1’). This bit can cause an RAI alarm in this configuration.

System action: Idle ABCD pattern to changed to 1111 (0xF).

User Response: Check the ABCD configuration in case a different value is required. If you are using the Web interface, use the TDM settings page to reconfigure the idle ABCD pattern, save configuration and reset the device.

Source: PSTN

Error code: 0x23014

Message: PSTNcheckBoradParamsOfIdlePatterns() - Your idle PCM=p while using T1. It can cause problems if you use D4 framing method.

Explanation: This notice message warns that INI file parameter “IdleABCDPattern” has the second bit zeroed. There is nothing wrong with the current configuration but it will cause problem in the future if trunk configuration is set to T1 to D4 framing method. It is not possible to change the INI file parameter “IdleABCDPattern” online, only offline.

System action: In case D4 framing is used the pattern is changed automatically, otherwise there is no impact, other than a future limitation on the possible PSTN protocols.

User Response: Change this value to allow on-the-fly configurations changes (no device reset) for framing methods in the future. If you are using the Web interface, use the TDM settings page to reconfigure the idle ABCD pattern, save configuration and reset the device.

Source: PSTN

Error code: 0x23014

Message: Warning: TDMBusClockSource is Internal and Framer is acCLOCK_MASTER_OFF (Recover clock). This is not a stable Clock Source.

Explanation: The device is generating the clock internally and the trunk configuration recovers another clock from the trunk (far end) when these two clock systems might not be synchronized.

System action: Processing continues normally.

User Response: If clock accuracy is important, reconfigure the trunk's clock source to a certified Stratum-1 source. If using an INI file for configuration, select a different value for TDMBusClockSource. If using the web interface, use the Trunk Settings web page under Advanced Configuration.

Source: PSTN

Error code: 0x23014

Message: Warning: PSTNAutoClockEnable is relevant only when TDMBusClockSource is Network (your setting is TDMBusClockSource = Internal).

Explanation: PSTNAutoClockEnable configuration parameter is enabled while the device generates the clock. This setting is irrelevant for such TDMBusClockSource configuration. The PSTNAutoClockEnable parameter is relevant only when a network clock source is chosen (on one of the trunks).

System action: The system ignores the parameter PSTNAutoClockEnable and continues processing normally.

User Response: Check the setting of PSTNAutoClockEnable. If using an INI file for configuration, select a different value for PSTNAutoClockEnable. If using the web interface, use the Trunk Settings web page under Advanced Configuration.

Source: PSTN

Error code: 0x23014

Message: Warning: TDMBusLocalReference was set to a negative value and has been changed to zero.

Explanation: The device was configured with TDMBusPSTNAutoClockEnable set to 1 (enabled), but no trunk was connected when configuration was saved. The local reference parameter was set to -1. This message is issued at the next device restart.

System action: The local reference is changed to zero.

User Response: Review the local reference setting and reset the device. If using an INI file for configuration, select a different value for TDMBusLocalReference. If using the web interface, use the Trunk Settings web page under Advanced Configuration.

Source: PSTN

Error code: 0x23014

Message: Warning: TDMBusClockSource is set to Network. At least one trunk must be acCLOCK_MASTER_OFF (recover clock).

Explanation: The parameter TDMBusClockSource is set to network (recover the clock from one of the trunks) therefore at least one of the trunks should be set to clock recover mode (ClockMaster parameter = 0).

System action: Processing continues normally.

User Response: If the parameter TDMBusPSTNAutoClockEnable is set to 1 (enabled), change at least one trunk to be the clock source and reset the device. If using an INI file for configuration, select a different value for ClockMaster (e.g. specify CLOCKMASTER_3 = 0 to set trunk 3 to clock recover mode). If using the web interface, use the Trunk Settings web page under Advanced Configuration.

Source: PSTN

Error code: 0x23014

Message: Note: Trunk number = t configuration was changed to Recover Mode because it was configured to be TDMBusLocalReference.

Explanation: The parameter TDMBusClockSource is set to network (recover the clock from one of the trunks). TDMBusLocalReference is set to trunk t. This trunk must be set to clock recovery mode (ClockMaster = off).

System action: Clock mode on trunk t is set to recovery.

User Response: Review the new configuration.

Source: PSTN

Error code: 0x23014

Message: Error: TDMBusClockSource is set to Network. All trunks set to be acCLOCK_MASTER_ON (master clock). Can't use other trunks as recover clock. Set one trunk to be acCLOCK_MASTER_OFF.

Explanation: The parameter TDMBusClockSource is set to network (recover the clock from one of the trunks), therefore at least one of the trunks' ClockMaster parameter should be set to clock recover (acCLOCK_MASTER_OFF). The device cannot complete the configuration because TDMBusPSTNAutoClockEnable is 1 (ON). This configuration will probably experience clock problems.

System action: Processing continues normally.

User Response: Change the TDMBusLocalReference trunk to be slave (recover) clock or disable the TDMBusPSTNAutoClockEnable. If using an INI file for configuration, select a different value for ClockMaster or TDMBusPSTNAutoClockEnable. If using the web interface, use the Trunk Settings web page under Advanced Configuration.

Source: PSTN

Error code: 0x23014

Message: Error: When configuring the device to recover the clock from PSTNTDMBusLocalReference, the trunk must be active. Trunk Number = t.

Explanation: The parameter TDMBusClockSource is set to network (recover the clock from one of the trunks). Trunk t is set as TDMBusLocalReference. Your configuration for the TDMBusLocalReference trunk t is protocol type NONE (no configuration).

System action: Processing continues normally.

User Response: Change TDMBusLocalReference to another trunk which is configured. If using an INI file for configuration, select a different value for

TDMBusLocalReference. If using the web interface, use the Trunk Settings web page under Advanced Configuration.

Source: PSTN

Error code: 0x23014

Message: Warning: The Framer clock is set to acCLOCK_MASTER_OFF (recover clock). This might be problematic in case the trunk gets the clock from a different clock source.

Explanation: Clocking configuration is very sensitive and sometimes it might be a problem if the clocks are not related to the same system, TDMBusClockSource configuration is highly recommended.

System action: Processing continues normally.

User Response: If clock accuracy is important, reconfigure the trunk's clock source to a certified Stratum-1 source. If using an INI file for configuration, select a different value for TDMBusClockSource. If using the web interface, use the Trunk Settings web page under Advanced Configuration.

Source: PSTN

Error code: 0x23014

Message: Warning: PSTNAutoClockEnable is relevant only when recovering the clock from PSTN and TDMBusClockSource is t.

Explanation: PSTNAutoClockEnable is set to 1 (ON) and this is irrelevant for the current TDMBusClockSource configuration. The PSTNAutoClockEnable parameter is relevant only when clock mode is set to Network (on one of the trunks).

System action: The system ignores the parameter PSTNAutoClockEnable and continues processing normally.

User Response: Check the setting of PSTNAutoClockEnable. If using an INI file for configuration, select a different value for PSTNAutoClockEnable. If using the web interface, use the Trunk Settings web page under Advanced Configuration.

Source: PSTN

Error code: 0x23319

Message: APS FST failed to set WTR param

Explanation: Wrong SDH APS (Automatic Protection Switch) WTR (WaitToRestore) INI-file parameter detected.

System action: Default parameter value is set and system startup continues normally.

User Response: Review the setting of "SDHFbrGrp_APS_WTR" parameter in the INI-file and restart the device.

Source: PSTN

Error code: 0x23319

Message: SDH APS FST API failure

Explanation: An internal error was detected during Automatic Protection Switch state machine initialization.

System action: Automatic Protection Switch functionality will not be available, but system startup continues normally.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23301

Message: SDH Wrong BoardParams parameter

Explanation: Wrong SDH INI-file parameter detected.

System action: PSTN functionality will be fully or partially disabled.

User Response: Review SDH configuration parameters in the INI-file and restart the device. Follow the directions in the SDH INI-file parameters chapter of the user manual.

Source: PSTN

Error code: 0x23301

Message: SDH/SONET PSTN Transm type, but FbrGrp n SdhSonMode wrong=m.

Explanation: INI-file parameter PSTNTransmissionType parameter was set to SDH/SONET, but SdhSonMode INI-file parameter wasn’t set.

System action: PSTN functionality will be fully disabled.

User Response: Review the setting of PSTNTransmissionType and SdhSonMode INI-file parameters, and restart the device.

Source: PSTN

Error code: 9901

Message: *** M3B ERROR: Table SS7_SIGTRAN_INTERFACE_GROUP - New Line Instantiation failed: Illegal index (out of range):

Explanation: The table SS7_SIGTRAN_INTERFACE_GROUP was defined in the INI file, but for one of the rows the line index is out of the valid range (greater than the maximum or a negative index).

System action: The table line is rejected and not processed.

User Response: Review the setting of the SS7_SIGTRAN_INTERFACE_GROUP table in the INI file. Correct the line index or remove the incorrect line.

Source: PSTN

Error code: 9902

Message: *** SS7SigIntGroup Validation Error - Interface group g cannot be configured with no_layer value (0) in SS7_SIG_LAYER parameter

Explanation: The table SS7_SIGTRAN_INTERFACE_GROUP was defined in the INI file. One of the lines had a wrong value in the SS7_SIG_LAYER column.

System action: The table line is rejected and not processed.

User Response: Review the setting of the SS7_SIGTRAN_INTERFACE_GROUP table in the INI file. Correct the SS7_SIG_LAYER column or remove the incorrect line.

Source: PSTN

Error code: 9903

Message: *** SS7SigIntGroup Validation Error - Interface group g cannot be configured with Traffic_mode other than Override value (1) in SS7_SIG_TRAF_MODE parameter

Explanation: The table SS7_SIGTRAN_INTERFACE_GROUP was defined in the INI file. One of the lines had a wrong value in the SS7_SIG_TRAF_MODE column: the only valid value is override (1). Override traffic mode is the only supported mode.

System action: The table line is rejected and not processed.

User Response: Review the setting of the SS7_SIGTRAN_INTERFACE_GROUP table in the INI file. Correct the SS7_SIG_TRAF_MODE column or remove the incorrect line.

Source: PSTN

Error code: 9904

Message: SS7 Configuration file ERROR: interface group wasn't defined due to blocking of M3UA by license Key.

Explanation: There is no valid M3UA license key.

System action: The table line is rejected and not processed.

User Response: Contact AudioCodes customer support for a new license key.

Source: PSTN

Error code: 9905

Message: SS7 Configuration file ERROR: interface group wasn't defined due to blocking of IUA by license Key.

Explanation: There is no valid IUA license key

System action: The table line is rejected and not processed.

User Response: Contact AudioCodes customer support for a new license key.

Source: PSTN

Error code: 9906

Message: SS7 Configuration file ERROR: interface group wasn't defined due to blocking of M2UA by license Key.

Explanation: There is no valid M2UA license key

System action: The table line is rejected and not processed.

User Response: Contact AudioCodes customer support for a new license key.

Source: PSTN

Error code: 9907

Message: *** SS7SigIntGroup Validation Error - Double Interface group value g exists in Sigtran interface group TABLE !

Explanation: In the INI-file table SS7_SIGTRAN_INTERFACE_GROUP, index g already exists.

System action: The duplicate table line is rejected and not processed.

User Response: Review the setting of the SS7_SIGTRAN_INTERFACE_GROUP table in the INI file. Correct the line index or remove the incorrect line.

Source: PSTN

Error code: 9908

Message: *** SS7SigIntGroup Validation Error - Local Port p is already used by group Id g !

Explanation: The table SS7_SIGTRAN_INTERFACE_GROUP was defined in the INI file. The SCTP local port specified on one of the lines is already in use by another Interface Group.

System action: The table line is rejected and not processed.

User Response: Review the setting of the SS7_SIGTRAN_INTERFACE_GROUP table in the INI file. Correct the SCTP local port number or remove the incorrect line.

Source: PSTN

Error code: 9909

Message: *** SS7SigIntGroup Validation Error - Interface group ID g uses IP address equals to our OWN IP

Explanation: The table SS7_SIGTRAN_INTERFACE_GROUP was defined in the INI file. The SCTP destination IP address specified is the same as the device's own IP.

System action: The table line is rejected and not processed.

User Response: Review the setting of the SS7_SIGTRAN_INTERFACE_GROUP table in the INI file. Correct the destination IP or remove the incorrect line.

Source: PSTN

Error code: 9910

Message: *** SS7_SIGTRAN_INTERFACE_GROUP table ERROR: interface group g cannot be deleted while interface id k still exists

Explanation: An attempt was made to delete an entry in the SS7_SIGTRAN_INTERFACE_GROUP table. The entry cannot be deleted while there are active interface ID related to it (in table SS7_SIGTRAN_INTERFACE_ID).

System action: The table delete-line action is rejected.

User Response: If the delete attempt was intentional, remove the appropriate lines from SS7_SIGTRAN_INTERFACE_ID and retry the operation.

Source: PSTN

Error code: 0x23124

Message: Warning: you have requested clock from the Internal, and ISDN is user side. pls confirm

Explanation: The clock configuration is unusual and can cause problems. Internal clock configuration is often used with ISDN network side trunk configuration. Conversely, ISDN user side trunk configuration is usually used with network clock configuration.

System action: Processing continues normally.

User Response: Verify the clock configuration, check the value of TDMBusClockSource.

Source: PSTN

Error code: 0x23124

Message: Warning: ISDN is Network side and you have requested clock from the PSTN Network.

Explanation: The clock configuration is unusual and can cause problems. Internal clock configuration is often used with ISDN network side trunk configuration.

Conversely, ISDN user side trunk configuration is usually used with network clock device configuration.

System action: Processing continues normally.

User Response: Verify the clock configuration, check the value of TDMBusClockSource.

Source: PSTN

Error code: 0x23014

Message: IsdnValidation - Cannot add another variant - configuration failed. Reached maximum ISDN variants number.

Explanation: Maximum of 4 different ISDN variants can be configured. If you got this error, it means that you have tried to add an ISDN trunk and this is the fifth ISDN variant on the device.

System action: Trunk validation fails, and hence the trunk is not configured.

User Response: Verify that all 4 ISDN variants are needed, and change the trunk configuration accordingly. Be aware that a maximum of 4 different ISDN variants may be configured concurrently.

Source: PSTN

Error code: 0x23014

Message: PSTN Validation Check failed. TDMBusLocalReference is configured to be trunk number = t and the trunk isn't configured to be Recover Mode.

Explanation: While attempting to reset a device, the trunk configuration validation check failed due to bad clock configuration. The parameter TDMBusClockSource is set to network and the TDMBusLocalReference trunk t is not set to recover-clock mode.

System action: The reset action is rejected.

User Response: Correct the trunk configuration and the TDMBusLocalReference parameters, and retry the reset command.

Source: PSTN

Error code: 0x23014

Message: PSTN Validation Check failed. When configuring the device to recover the clock from PSTNTDMBusLocalReference, the trunk must be active (protocol type should not be set to NONE).

Explanation: While attempting to reset a device, the trunk configuration validation check failed due to bad clock configuration. The parameter TDMBusClockSource is set to network and the TDMBusLocalReference trunk t is not configured (PROTOCOLTYPE_t = NONE). The reference trunk must be configured and set to recover-clock mode (acCLOCK_MASTER_OFF).

System action: The reset action is rejected.

User Response: Correct the trunk configuration, TDMBusLocalReference and ClockMaster parameters, and retry the reset command.

Source: PSTN

Error code: 0x23013

Message: acPSTNVCValidationCheck() - Vc Id (vc) is not in range (0-vmax).

Explanation: This message indicates a failure in the SDH envelope of the trunk (Virtual Container) validation. SDH VC (Virtual Container) validation failed (wrong VC number).

System action: The trunk will not configure but system startup continues normally.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23013

Message: acPSTNVcVsTrkValidationCheck() - Trunk Id (t) is not in range (0 - tmax).

Explanation: Failure of validation of SDH VC Container configuration parameters against trunk configuration parameters.

System action: The trunk will not configure but system startup continues normally.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23013

Message: PSTN Validation Check. LP Mapping type[lpm] doesn't suit Trunk Line type [tlt]. VCv Trkt.

Explanation: PSTN protocol type doesn’t match the SDHFbrGrp_LP_Mapping_Type INI-file parameter. E.g., protocol type is E1, but LP mapping type is ASYNC_TU11_DS1.

System action: The trunk will not configure but system startup continues normally.

User Response: Check ProtocolType and SDHFbrGrp_LP_Mapping_Type parameters, fix the problem and restart the system with the fixed INI-file. If using the web interface, use the Trunk Settings page under Advanced Configuration, and restart the system.

Source: PSTN

Error code: 0x23014

Message: PSTN Trunk Validation Check failed. Protocol type must be the same line type (E1\T1). Do not mix line types.

Explanation: One trunk is configured as E1 and another trunk is configured as T1. The device cannot handle E1 and T1 configuration concurrently.

System action: The trunk will not configure but system startup continues normally.

User Response: Change the ProtocolType configuration.

Source: PSTN

Error code: 0x23014

Message: PSTN Trunk Validation Check failed. Framing method f is not valid (Line type is lt)

Explanation: The FramingMethod f specified for this trunk is invalid for the selected line type lt (E1 or T1).

System action: The trunk will not configure but system startup continues normally.

User Response: Change the FramingMethod configuration parameter.

Source: PSTN

Error code: 0x23014

Message: Warning: Line Code does not fit T1 Line Type. Line Code is : lc. B8ZS will be used as default

Explanation: LineCode configuration cannot be used with T1 hardware, the configuration changed to acB8ZS.

System action: LineCode is changed and system startup continues normally.

User Response: Review the new LineCode and correct if necessary. If you are using the Web interface, use the trunk setting page to reconfigure the relevant LineCode, save configuration and reset the device.

Source: PSTN

Error code: 0x23014

Message: PSTN Trunk Validation Check failed due to configuration of CAS protocol with wrong HW or INI file parameter - CASProtocolEnable disabled.

Explanation: The hardware in use cannot handle CAS configuration.

System action: The trunk will not configure but system startup continues normally.

User Response: Contact AudioCodes customer support for if CAS configuration is needed.

Source: PSTN

Error code: 0x23014

Message: PSTN Trunk Validation Check failed due to wrong CASTableIndex parameter configuration.

Explanation: The CASTableIndex parameter for this trunk configuration is invalid (there is no such CAS state machine dat file).

System action: The trunk will not configure but system startup continues normally.

User Response: Download a CAS state machine dat file to the device, and change the CASTableIndex configuration parameter accordingly.

Source: PSTN

Error code: 0x23014

Message: PSTN Trunk Validation Check failed. Missing CAS table at TableIndex=t.

Explanation: The CASTableIndex parameter for this trunk configuration is invalid (there is no such CAS state machine – dat file).

System action: The trunk will not configure but system startup continues normally.

User Response: Download a CAS state machine dat file to the device, and change the CASTableIndex configuration parameter accordingly.

Source: PSTN

Error code: 0x23014

Message: Framing method is not supported in Ultra mapper.

Explanation: The requested FramingMethod parameter value in not supported on the current hardware.

System action: The trunk will not configure but system startup continues normally.

User Response: Change the FramingMethod configuration parameter.

Source: PSTN

Error code: 0x23356

Message: SPE_Init [n1] failed base [n2], umrc = n3

Explanation: An internal error. Initialization of framer internal SPE block failure. Framer SPE block number, base address and driver return code can be found in the n1, n2, n3 values.

System action: PSTN functionality is disabled. System startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23356

Message: STS1_Connect [n1] failed base [n2], umrc = n3

Explanation: An internal error. Initialization of framer internal STS1 cross-connect block failure. Framer STS1 block number, base address and driver return code can be found in the n1, n2, n3 values.

System action: PSTN functionality is disabled. System startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23356

Message: TMUX_Prov failed base [n1], umrc = n2

Explanation: An internal error. Initialization of framer internal TMUX block failure. Framer base address and driver return code can be found in the n1 and n2 values.

System action: PSTN functionality is disabled. System startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23356

Message: XC1_Connect FRM_TP_T-VTMPR [n1] failed base [n2], umrc = n3

Explanation: An internal error. Failure of framer-internal cross-connect on connection of VT Mapper and DS1 framer block transmit. Framer block, base address and driver return code can be found in the n1, n2 and n3 values.

System action: PSTN functionality is disabled. System startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23356

Message: XC1_Connect VTMPR-FRM_RP_R [n1] failed base [n2], umrc = n3

Explanation: An internal error. Failure of framer-internal cross-connect on connection of VT Mapper and DS1 framer block. Framer block, base address and driver return code can be found in the n1, n2 and n3 values.

System action: PSTN functionality is disabled. System startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23356

Message: XC1_Connect DJA-FRM_RP_R [n1] failed base [n2], umrc = n3

Explanation: An internal error. Failure of framer-internal cross-connect on connection of Digital Jitter attenuator and DS1 framer block. Framer block, base address and driver return code can be found in the n1, n2 and n3 values.

System action: PSTN functionality is disabled. System startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23356

Message: XC1_Connect VTMPR-DJA [n1] failed base [n2], umrc = n3

Explanation: An internal error. Failure of framer-internal cross-connect on connection of VT mapper and Digital Jitter attenuator. Framer block, base address and driver return code can be found in the n1, n2 and n3 values.

System action: PSTN functionality is disabled. System startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23353

Message: isDs3ClockSourceMaster: DS3 clock source parameter unavailable

Explanation: INI-file parameter DS3CONFIG_ClockSource is unavailable.

System action: DS3 clock will be set as a Master (LOCAL_BOARD) for the DS3 ID and system startup continues normally.

User Response: Review the DS3 configuration parameters in the INI-file, modify if required and restart the device.

Source: PSTN

Error code: 0x23356

Message: FRM_LinkEnable [n1] failed base [n2], umrc = n3

Explanation: An internal error. Initialization of framer DS1/E1 failure. Framer block, base address and driver return code can be found in the n1, n2 and n3 values.

System action: PSTN functionality is disabled. System startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23356

Message: FRM_LinkInit [n1] failed base [n2], umrc =n3

Explanation: An internal error. Initialization of framer DS1/E1 failure. Framer block, base address and driver return code can be found in the n1, n2 and n3 values.

System action: PSTN functionality is disabled. System startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23356

Message: FRM_Init [n1] failed base [n2], umrc = n3

Explanation: An internal error. Initialization of framer DS1/E1 failure. Framer block, base address and driver return code can be found in the n1, n2 and n3 values.

System action: PSTN functionality is disabled. System startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23356

Message: FRM_LinkDisable [n1] failed base [n2], umrc = n3

Explanation: An internal error. Initialization of framer DS1/E1 failure. Framer block, base address and driver return code can be found in the n1, n2 and n3 values.

System action: At least one PSTN trunk block (21 or 28 trunks) functionality is disabled. System startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23356

Message: XC1_ConnectInit failed base [n1], umrc = n2

Explanation: An internal error. Initialization of framer internal connection matrix failure. Framer base address and driver return code can be found in the n1 and n2 values.

System action: PSTN functionality is disabled. System startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23353

Message: UltraMapper Wrong parameter =p, Val2=v

Explanation: An internal error. One of the PSTN framer driver APIs received wrong parameters.

System action: In most cases part or all of the PSTN functionality is disabled and system startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23351

Message: UltraMapper init fail RC=n1, Val2=n2

Explanation: An internal error. PSTN Framer driver API returned failure return code. The error code and some complemented info can be found in the n1 and n2 values.

System action: PSTN functionality is disabled. System startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23357

Message: DS3 LIU Driver failed

Explanation: An internal error. DS3 LIU driver API returned failure return code. Return code of the API printed out. Probably indicates a problem with DS3 LIU hardware.

System action: PSTN functionality is disabled. System startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23314

Message: SDH ALPO queue rx fail RC=n1, Val2=n2

Explanation: An internal error. Queue-receive operation ended with an error. The error code and some complementing info can be found in the n1 and n2 values.

System action: System continues functioning normally.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23306

Message: SDH Driver fail RC=n1, Val2=n2

Explanation: An internal error. One of the PSTN SDH/DS3 routines reported that a HW driver routine returns error; the driver routine error code and some complementing data can be found in the n1 and n2 values.

System action: In most cases part of the SDH alarm functionality is disabled and system startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23307

Message: SDH Wrong alarm instance Val1=n2, Val2=n2

Explanation: An internal error. One of the PSTN SDH routines reported wrong alarm instance; this parameter and some complemented info can be found in the n1 and n2 values.

System action: In most cases part of the SDH alarm functionality is disabled and system startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23302

Message: SDH Wrong parameter =n1, Val2=n2

Explanation: An internal error. One of the PSTN SDH routines received wrong parameter; this parameter and some complemented info can be found in the n1 and n2 values.

System action: In most cases part or all of the PSTN functionality is disabled and system startup continues.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23122

Message: acPSTNNfasTrunkValidationCheck(): Cannot configure trunk - there is already another trunk in the group with different TerminationSide / FramingMethod / LineCode / ProtocolType .

Explanation: When configuring an NFAS group, all the trunks in the group must be configured with the same ISDN termination side (network or user), framing method, line code, and protocol type.

System action: Trunk configuration is aborted.

User Response: Configure all the trunks in the NFAS group with the same ISDN termination side, framing method, line code, and protocol type.

Source: PSTN

Error code: 0x23122

Message: acPSTNNfasTrunkValidationCheck(): Cannot configure trunk - there is already another trunk in the group with same InterfaceId.

Explanation: When configuring an NFAS group, every trunk in the group must be configured with a different interface id.

System action: Trunk configuration is aborted.

User Response: Select a different interface ID for the reported trunk (set "ISDNNFASInterfaceID_xx = y" in the INI file, where xx is the trunk number and y is the interface ID).

Source: PSTN

Error code: 0x23122

Message: acPSTNNfasTrunkValidationCheck(): NFAS cannot be defined on Protocol p on trk t.

Explanation: NFAS can be configured only on some PRI variants, such as NI2 or DMS or 5ESS.

System action: Trunk configuration is aborted.

User Response: Select a different ISDN variant for the trunk, or remove the trunk from the NFAS group.

Source: PSTN

Error code: 0x23122

Message: acPSTNNfasTrunkValidationCheck(): Cannot configure backup trunk, no primary trunk defined for this group (GroupId = g).

Explanation: When configuring an NFAS group using an ini file (and not using the web interface), a primary trunk for the group must be configured. The message indicates that an NFAS trunk was configured with a GroupId with which no primary trunk is associated.

System action: Trunk configuration is aborted.

User Response: Configure a primary trunk for the NFAS group g by setting both "DCHConfig_xx = 0" and "NFASGroupNumber_xx = g" in the INI file, where xx is the selected trunk number.

Source: PSTN

Error code: 0x23014

Message: acl_iua/dua_config() - InterfaceId not configured for iua/dua trunk = t.

Explanation: During IUA/DUA trunk configuration, the IUA/DUA layer uses the value of the IUAInterfaceID parameter to associate IUA/DUA interfaces with trunks. For one of the trunks, the IUAInterfaceID is missing. IUA configuration is not complete without the Interface ID.

System action: Trunk t can't be used for IUA/DUA.

User Response: Define the interface ID to this IUA/DUA trunk. If using the web interface, go to the "Advanced Configuration" / "Trunk Settings" page add the IUA/DUA interface ID in IUA/DUA Interface ID tab. If using an INI file for configuration, specify "IUAINTERFACEID_t = x " where t is the reported trunk and x is the selected interface ID.

Source: PSTN

Error code: 0x23014

Message: acl_iua/dua_config() - duplicate Int_id = x, on Trunk = t.

Explanation: During IUA/DUA trunk configuration, the IUA/DUA layer uses the value of the IUAInterfaceID parameter to associate IUA/DUA interfaces with trunks. Two or more trunks were defined with the same interface ID (which must be unique).

System action: Trunk t can't be used for IUA/DUA.

User Response: Define a different interface ID to this IUA/DUA trunk. If using the web interface, go to the "Advanced Configuration" / "Trunk Settings" page change the IUA/DUA interface ID in IUA Interface ID tab. If using an INI file for configuration, specify "IUAINTERFACEID_t = y " where t is the reported trunk and y is the selected interface ID.

Source: PSTN

Error code: 0x23001

Message: IuaDuaTrunkValidation ERROR: trunk wasn't defined due to blocking of IUA by license Key.

Explanation: IUA/DUA cannot be configured due to missing license key.

System action: Trunk configuration is aborted.

User Response: Contact AudioCodes customer support for a new license key.

Source: PSTN

Error code: 0x23001

Message: Interface ID is missing in this IUA/DUA trunk definition.

Explanation: During IUA/DUA trunk configuration, the IUA/DUA layer uses the value of the IUAInterfaceID parameter to associate IUA/DUA interfaces with trunks. For one of the trunks, the IUAInterfaceID is missing. IUA configuration is not complete without the Interface ID.

System action: Trunk t can't be used for IUA/DUA.

User Response: Define the interface ID to this IUA/DUA trunk. If using the web interface, go to the "Advanced Configuration" / "Trunk Settings" page add the IUA/DUA interface ID in IUA/DUA Interface ID tab. If using an INI file for configuration, specify "IUAINTERFACEID_t = x " where t is the reported trunk and x is the selected interface ID.

Source: PSTN

Error code: 0x23014

Message: Trunk t definition rejected. IUA interfaceId x is already defined.

Explanation: During IUA/DUA trunk configuration, the IUA/DUA layer uses the value of the IUAInterfaceID parameter to associate IUA/DUA interfaces with trunks. Two or more trunks were defined with the same interface ID (which must be unique).

System action: Trunk t can't be used for IUA/DUA.

User Response: Define a different interface ID to this IUA/DUA trunk. If using the web interface, go to the "Advanced Configuration" / "Trunk Settings" page change the IUA/DUA interface ID in IUA Interface ID tab. If using an INI file for configuration, specify "IUAINTERFACEID_t = y " where t is the reported trunk and y is the selected interface ID.

Source: PSTN

Error code: 0x23014

Message: Trunk t1 definition rejected. IUA interfaceId x is assigned to trunk number t2.

Explanation: During IUA/DUA trunk configuration, the IUA/DUA layer uses the value of the IUAInterfaceID parameter to associate IUA/DUA interfaces with trunks. Two or more trunks were defined with the same interface ID (which must be unique).

System action: Trunk t can't be used for IUA/DUA.

User Response: Define a different interface ID to this IUA/DUA trunk. If using the web interface, go to the "Advanced Configuration" / "Trunk Settings" page change the IUA/DUA interface ID in IUA Interface ID tab. If using an INI file for configuration, specify "IUAINTERFACEID_t1 = y " where t1 is the reported trunk and y is the selected interface ID.

Source: PSTN

Error code: 0x23014

Message: Trunk t definition rejected. Either IUA InterfaceId is missing or failed to generate If Group for this IUA trunk.

Explanation: An internal error has occurred while configuring IUA/DUA interfaces.

System action: Trunk configuration is aborted.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 0x23014

Message: Trunk t definition rejected. InterfaceId x is not defined.

Explanation: During IUA/DUA trunk configuration, interface group ID 1 is reserved for IUA/DUA default group configuration. In the current device configuration, this group already exists and it used for another SIGTRAN application.

System action: Trunk configuration is aborted.

User Response: Check SS7_SIG_IF_GROUP_TABLE configuration and change the application that uses SS7_IF_GR_ID = 1 to use a different interface group ID value (not 1).

Source: PSTN

Error code: 0x23014

Message: Trunk t definition rejected. Could not generate Interface for IUA.

Explanation: An internal error has occurred while configuring IUA/DUA interfaces.

System action: Trunk configuration is aborted.

User Response: Contact AudioCodes customer support.

Source: PSTN

Error code: 9911

Message: set Action: Unknown action

Explanation: During SS7 Sigtran Interface Group table configuration an ambiguous value was provided for the Action field.

System action: The requested action is rejected.

User Response: Contact AudioCodes customer support.

Source: \PSTN\SS7\Sigtran\ACL_UAL\SS7SigIfGroupParamSet.cpp

Error code: 9912

Message: *** acSS7SigIfGroup Validation Error - No entries in SS7_SN_TABLE

Explanation: During SS7 Sigtran Interface Group table configuration action for M3UA interface group, it was detected that MTP3 Signaling node (NAI parameter) is missing.

System action: The table line is rejected and not processed.

User Response: Define an NAI or remove the problematic line from the INI file.

Source: \PSTN\SS7\Sigtran\ACL_UAL\SS7SigIfGroupParamSet.cpp

Error code: 9913

Message: *** acSS7SigIfGroup Validation - Error while processing next line indices

Explanation: During SS7 Sigtran Interface Group table configuration, an internal problem in table mechanism had occurred.

System action: Table The table line is rejected and not processed.

User Response: Contact AudioCodes customer support.

Source: \PSTN\SS7\Sigtran\ACL_UAL\SS7SigIfGroupParamSet.cpp

Error code: 9914

Message: *** acSS7SigIfGroup Validation Error - variant = v not found in SS7_SN_TABLE!

Explanation: During SS7 Sigtran Interface Group table configuration action for M3UA interface group, an MTP3 variant mismatch was detected. Namely, the MTP2 layer variant in SN table and M3UA interface group variant are not identical.

System action: The table line is rejected and not processed.

User Response: Change SS7 interface group VARIANT parameter (v) value to be the same as SN table or remove the problematic line from the INI file.

Source: \PSTN\SS7\Sigtran\ACL_UAL\SS7SigIfGroupParamSet.cpp

Error code: 9915

Message: GroupID Table: Failed to Re-generate line

Explanation: During SS7 Sigtran Interface Group table configuration, an internal problem in the table mechanism had occurred.

System action: Configuration is aborted.

User Response: Contact AudioCodes customer support.

Source: \PSTN\SS7\Sigtran\ACL_UAL\SS7SigIfGroupParamSet.cpp

Error code: 9916

Message: " *** M3B ERROR: Table SS7_SIGTRAN_INT_ID - New Line Instantiation failed: Illegal index (out of range):

Explanation: During SS7 Sigtran Interface ID table configuration, an internal problem in table mechanism had occurred. The line index is out of range.

System action: The table line is rejected and not processed.

User Response: Change SS7_SIGTRAN_INT_ID table line index to be within the valid range or remove the line from the INI file.

Source: \PSTN\SS7\Sigtran\ACL_UAL\SS7SigIntIdParamSet.cpp

Error code: 9917

Message: *** SS7SigIntId Validation Error - No entries in SS7_SN_TABLE!

Explanation: During SS7 Sigtran Interface Id table configuration action for M3UA interface Id, it was detected that MTP3 Signaling node (NAI parameter) that this interface Id represents is missing.

System action: The table line is rejected and not processed.

User Response: Configure the MTP3 layer using the SS7_SN_TABLE table, or remove the reported line from the INI file.

Source: \PSTN\SS7\Sigtran\ACL_UAL\SS7SigIntIdParamSet.cpp

Error code: 9918

Message: " *** SS7SigIntId Validation - Error while processing next line indices!”

Explanation: During SS7 Sigtran Interface ID table configuration, an internal problem in the table mechanism had occurred.

System action: The table line is rejected and not processed.

User Response: Contact AudioCodes customer support.

Source: \PSTN\SS7\Sigtran\ACL_UAL\SS7SigIntIdParamSet.cpp

Error code: 9919

Message: " *** SS7SigIntId Validation Error - OPC = p was not found in SS7_SN_TABLE!”

Explanation: During SS7 Sigtran Interface Id table configuration action for M3UA interface Id, it was detected that MTP3 Signaling Point node (SPC parameter) that this interface Id represents is missing from the SN table.

System action: The table line is rejected and not processed.

User Response: Change the SPC parameter (p) to be equal to the equivalent point-code in the SS7_SN_TABLE table, or remove the reported line from the INI file.

Source: \PSTN\SS7\Sigtran\ACL_UAL\SS7SigIntIdParamSet.cpp

Error code: 9920

Message: " *** SS7SigIntId Validation Error - Interface ID = i cannot be configured with no_layer value (0) in SS7_SIG_IF_ID_LAYER parameter”

Explanation: During SS7 Sigtran Interface Id table configuration action for new interface Id, it was detected that the Sigtran layer (SS7_SIG_IF_ID_LAYER parameter) of this interface Id has an illegal value (‘0’ – no layer). An Interface ID line must be configured with a valid layer application code.

System action: The table line is rejected and not processed.

User Response: Change SS7_SIG_IF_ID_LAYER parameter value to be other than ‘0’ or remove the reported line from the INI file.

Source: \PSTN\SS7\Sigtran\ACL_UAL\SS7SigIntIdParamSet.cpp

Error code: 0x22016

Message: PSTNUlmFramerSetClockSourceFromTrkId() Trunk=n1 failed. Framer driver failure.

Explanation: An Internal error occurred on set of clock source from specific trunk (Trunk number can be found in the n1 value).

System action: Source Clock is not set and system startup continues normally.

User Response: Contact AudioCodes customer support.

Source: PSTN\TP6310PSTNHal.cpp

Error code: 0x23001

Message: PSTNTransmType parameter set NONE or SdhSonMode set UNKNOWN. No trunk will be available.

Explanation: PSTNTransmissionType INI-file parameter set to NONE or SDHFbrGrp_SDHSONETMode parameter is not set.

System action: PSTN functionality is disabled and system startup continues normally.

User Response: If no PSTN functionality necessary no action required, otherwise inspect your INI-file configuration parameters (PSTNTransmissionType and SDHFbrGrp_SDHSONETMode). To configure parameters using Web interface, go to “Advanced Configuration” \ ”PSTN Settings” \ “Transmission Settings” and select “Sonet/SDH” or “DS3” Transmission Type.

If selected “Sonet/SDH” go to “Advanced Configuration” \ ”PSTN Settings” \ “SDH Settings” and configure SDH parameters.

If selected Transmission Type “DS3”, go to “Advanced Configuration” \ ”PSTN Settings” \ “DS3 Settings” and configure DS3 parameters.

Source: PSTN\TP6310PSTNHal.cpp

Error code: 0x23315

Message: SDH DS3 LIU driver failure

Explanation: An internal error. Initialization of DS3 LIU hardware failure.

System action: PSTN functionality will try to recover but is probably disabled.

User Response: Internal error. Contact AudioCodes customer support.

Source: PSTN\TP6310PSTNHal.cpp

Error code: 0x23356

Message: xxx VT_Init [n1] failed base [n2], umrc = n3

Explanation: An internal error. Initialization of framer internal VTMPR block failure. Block number, base address and driver return code can be found in the n1, n2, n3 values.

System action: PSTN functionality is disabled and system startup continues normally.

User Response: Internal error. Contact AudioCodes customer support.

Source: PSTN\UlmSdhApi.c

Error code: 0x23356

Message: xxx SPE_Prov [n1] failed base [n2], umrc = n3

Explanation: An internal error. Initialization of framer internal SPE block failure. Block number, base address and driver return code can be found in the n1, n2, n3 values.

System action: PSTN functionality is disabled and system startup continues normally.

User Response: Internal error. Contact AudioCodes customer support.

Source: PSTN\UlmSdhApi.c

Error code: 0x23001

Message: SDH-DS3 params validation: PSTN Transmission type is not set.

Explanation: Relevant for remote management only. PSTNTransmissionType parameter is not set to any known value.

System action: Remote reset rejected.

User Response: Inspect the PSTN DS3 mode parameters (DS3CONFIG), fix them and try the reset again. To configure parameters using Web interface, go to "Advanced Configuration" \ "PSTN Settings" \ "Transmission Settings" and select "Sonet/SDH" or "DS3" Transmission Type.

If selected "Sonet/SDH" go to "Advanced Configuration" \ "PSTN Settings" \ "SDH Settings" and configure SDH parameters.

If selected Transmission Type "DS3", go to "Advanced Configuration" \ "PSTN Settings" \ "DS3 Settings" and configure DS3 parameters.

Source: PSTN\PstnHal.cpp

Error code: 0x23001

Message: SDH-DS3 params validation: PSTN Transmission type set to NONE.

Explanation: Relevant for remote management only. PSTNTransmissionType parameter is set to NONE.

System action: Remote reset performed, but after reset no PSTN is configured.

User Response: After the device start-up configure desirable PSTN configuration – SDH/SONET or DS3 – and reset device again. For this goal if using Web interface go to "Advanced Configuration" \ "PSTN Settings" \ "Transmission Settings" and select "Sonet/SDH" or "DS3" Transmission Type.

If selected "Sonet/SDH" go to "Advanced Configuration" \ "PSTN Settings" \ "SDH Settings" and configure SDH parameters.

If selected Transmission Type "DS3", go to "Advanced Configuration" \ "PSTN Settings" \ "DS3 Settings" and configure DS3 parameters.

Source: PSTN\PstnHal.cpp

Error code: 0x23001

Message: SDH-DS3 params validation: DS3 PSTN Transm type, but no DS3 configuration found.

Explanation: Relevant for remote management only. PSTNTransmissionType parameter defined as "DS3", but DS3 interfaces are not configured.

System action: Remote reset rejected.

User Response: Inspect the PSTN DS3 mode parameters, fix them and try the reset again. To fix parameters using Web interface go to "Advanced Configuration" \ "PSTN Settings" \ "DS3 Settings" and configure "DS3 Clock Source", "DS3 Framing Method", "DS3 Line Build Out" for at least one interface.

Source: PSTN\PstnHal.cpp

Error code: 0x23001

Message: SDH-DS3 params validation: SDH/SONET PSTN Transm type, but SdhSonMode or LP Mapping param is not set.

Explanation: Relevant for remote management only. PSTNTransmissionType parameter defined as “Optical Sonet/SDH”, but incorrect SDH configuration (SDHFbrGrp_SDHSONETMode and SDHFbrGrp_LP_Mapping_Type parameters) set.

System action: Remote reset rejected.

User Response: Inspect the Sdh/SONET mode parameters, fix them and try the reset again. To configure parameters using Web interface, go to “Advanced Configuration” \ ”PSTN Settings” \ “SDH Settings” and configure SDH/Sonet Mode and SDH LP Mapping Type.

Source: PSTN\PstnHal.cpp

Error code: 0x23001

Message: SDH-DS3 params validation: Unknown PSTN Transmission type.

Explanation: PSTNTransmissionType INI-file parameter has unknown value.

System action: PSTN functionality is disabled and system startup continues normally.

User Response: Inspect the PSTNTransmissionType parameter in the INI-file, fix them and start the device again. To configure parameters using Web interface, go to “Advanced Configuration” \ ”PSTN Settings” \ “Transmission Settings”.

Source: PSTN\PstnHal.cpp

Error code: 0x23001

Message: SDH-DS3 params validation: DS3 PSTN Transm type, but Line Type is not DS1. Please check ProtocolType.

Explanation: PSTNTransmissionType INI-file parameter defined as “COPPER_DS3” (2) but ProtocolType parameter in the INI-file doesn’t suit DS1 trunk Line Type (for example E1 Transparent).

System action: PSTN functionality is disabled and system startup continues normally.

User Response: Inspect the ProtocolType parameter in the INI-file, fix them and start the device again. To configure parameters using Web interface, go to “Quick Setup” and configure DS1 Protocol Type.

Source: PSTN\PstnHal.cpp

Error code: 0x23001

Message: SDH-DS3 params validation: DS3 PSTN Transm type, but no DS3 config lines found.

Explanation: PSTNTransmissionType INI-file parameter defined as “COPPER_DS3” (2) but no DS3 configuration found in the INI-file (DS3CONFIG lines are missing).

System action: PSTN functionality is disabled and system startup continues normally.

User Response: Inspect the DS3 configuration parameters in the INI-file, fix them and start the device again. To configure parameters using Web interface go to “Advanced Configuration” \ ”PSTN Settings” \ “DS3 Settings” and configure “DS3 Clock Source”, “DS3 Framing Method”, “DS3 Line Build Out” for at least one interface.

Source: PSTN\PstnHal.cpp

Error code: 0x23001

Message: SDH-DS3 params validation: SDH/SONET PSTN Transm type, but no SDH config defined.

Explanation: PSTNTransmissionType INI-file parameter defined as “Optical Sonet/SDH” (1) but no SDH configuration found in the INI-file (SDHFbrGrp_SDHSONETMode and SDHFbrGrp_LP_Mapping_Type parameters missing).

System action: PSTN functionality is disabled and system startup continues normally.

User Response: Inspect the SDH/SONET configuration parameters in the INI-file, fix them and start the device again. To configure parameters using Web interface, go to “Advanced Configuration” \ ”PSTN Settings” \ “SDH Settings” and configure SDH/Sonet Mode and SDH LP Mapping Type.

Source: PSTN\PstnHal.cpp

Error code: 9931

Message: *** SS7SigIntId Validation Error - Trunk i already used by interface ID j

Explanation: During SS7 Sigtran Interface ID table processing, a configuration action for new IUA/DUA ID i was requested, but IUA/DUA trunk j is already configured in another table line.

System action: The table line is rejected and not processed.

User Response: Change the INTID_UAL_NAI column or remove the reported line from the INI file.

Source: \PSTN\SS7\SIGTRAN\ACL_UAL\SS7SIGINTIDPARAMSET.CPP

Error code: 9932

Message: Sigtran InterfaceID ERROR - Remove InterfaceId i failed because it is attached to trunk j

Explanation: During SS7 Sigtran Interface ID table processing, a delete action for IUA/DUA ID i was requested, but IUA/DUA trunk j is still configured, therefore IUA/DUA ID i will not be deleted.

System action: The table action is rejected.

User Response: Delete trunk j and retry the delete IUA/DUA ID i action.

Source: \PSTN\SS7\SIGTRAN\ACL_UAL\SS7SIGINTIDPARAMSET.CPP

Error code: 9933

Message: Sigtran InterfaceID ERROR - Remove InterfaceId i failed because it is attached to linkId j

Explanation: During SS7 Sigtran Interface ID table processing, a delete action for M2UA ID i was requested, but M2UA SS7 link j is still configured; an M2UA interface ID cannot be removed when the SS7 link is still configured.

System action: The table action is rejected.

User Response: Delete M2UA SS7 link j and retry the delete M2UA ID i action.

Source: \PSTN\SS7\SIGTRAN\ACL_UAL\SS7SIGINTIDPARAMSET.CPP

Error code: 9934

Message: IF ID Table: Failed to Re-generate line i

Explanation: During SS7 Sigtran Interface ID table configuration, an internal problem in the table mechanism had occurred.

System action: Configuration is aborted.

User Response: Contact AudioCodes customer support.

Source: \PSTN\SS7\SIGTRAN\ACL_UAL\SS7SIGINTIDPARAMSET.CPP

Error code: 9935

Message: un initialized sg_mgc_side for instance i

Explanation: An internal error was detected in Sigtran application operation.

System action: Sigtran will not function correctly.

User Response: Contact AudioCodes customer support.

Source: \PSTN\SS7\Sigtran\ACL_UAL\acl_ual_config.c

Error code: 9936

Message: cannot add instance i owned by port j

Explanation: SCTP port j cannot added to the Sigtran data base.

System action: Sigtran will not function correctly.

User Response: Contact AudioCodes customer support.

Source: \PSTN\SS7\Sigtran\ACL_UAL\acl_ual_config.c

Error code: 9937

Message: NOTE: port i is already owned by instance j, group id k

Explanation: The SS7 Sigtran application detected that the SCTP port i is already owned by group k. This represents an internal error.

System action: Sigtran will not function correctly.

User Response: Contact AudioCodes customer support.

Source: \PSTN\SS7\Sigtran\ACL_UAL\acl_ual_config.c

Error code: 9938

Message: MGC - NAT SCTP layer init port i for interface group index j

Explanation: The SS7 Sigtran application informs that SCTP port i was initialized as either a Media Gateway Controller (MGC) or a NAT device, for SS7 Sigtran interface group j.

System action: None.

User Response: None.

Source: \PSTN\SS7\Sigtran\ACL_UAL\acl_ual_config.c

Error code: 9939

Message: SG SCTP layer init port i for interface group index j

Explanation: The SS7 Sigtran application informs that SCTP port i was initialized as a Signaling Gateway (SG), for SS7 Sigtran interface group j.

System action: None.

User Response: None.

Source: \PSTN\SS7\Sigtran\ACL_UAL\acl_ual_config.c

Error code: 9940

Message: SCTP layer cannot init port i for interface group index j

Explanation: SCTP port i wasn't initialized for SS7 Sigtran interface group j. This indicates an internal error in the Sigtran application.

System action: Sigtran will not function correctly.

User Response: Contact AudioCodes customer support.

Source: \PSTN\SS7\Sigtran\ACL_UAL\acl_ual_config.c

Error code: 0x23122

Message: acPSTNNfasTrunkValidationCheck() had failed due to invalid GroupID number (GroupId = g), GroupId must be in the range of 1 -9.

Explanation: NFAS GroupId must be in the range of 1 – 9.

System action: Trunk configuration is aborted.

User Response: Select a different group ID for the reported trunk (set "NFASGroupNumber_xx = y" in the INI file, or the NFAS Group Number field to y in the Trunk Settings page in the web, where xx is the trunk number and y is the group ID. If using the web interface, apply the changes using the Apply Trunk Settings button).

Source: PSTN

Error code: 0x23122

Message: acPSTNNfasTrunkValidationCheck(): Cannot configure NFAS trunk - GroupId g does not exist.

Explanation: When configuring NFAS trunk on-the-fly (not via ini file), a primary trunk with the same group ID must be configured first.

System action: Trunk configuration is aborted.

User Response: Select existing group ID for the reported (set the NFAS Group Number field to g in the Trunk Settings page in the web, where g is the group ID. Apply the changes using the Apply Trunk Settings button).

Source: PSTN

Error code: 0x23122

Message: acPSTNNfasTrunkValidationCheck(): Cannot configure trunk, the backup trunk was deleted - user must delete the primary trunk and re-configure the group (GroupId = g).

Explanation: After deleting a backup trunk of a group, the group must be deleted first by deleting also the primary trunk, and only then the group can be re-configured.

System action: Trunk configuration is aborted.

User Response: Delete the NFAS group by deleting the primary trunk of the group, and then re-configure the group. Deleting a trunk can be done in the Trunk Settings page in the web: stop the trunk using the Stop Trunk button, set the Protocol Type field to NONE, and then apply the changes using the Apply Trunk Settings button.

Source: PSTN

Error code: 0x23122

Message: acPSTNNfasTrunkValidationCheck(): Cannot configure backup trunk, there is already active primary trunk in this group (GroupId = g).

Explanation: When configuring NFAS group on-the-fly (not via ini file), a backup trunk must be configured before the primary trunk.

System action: Trunk configuration is aborted.

User Response: Delete the NFAS group by deleting the primary trunk of the group, and then re-configure the group starting from the backup trunk. Deleting a trunk can be done in the Trunk Settings page in the web: stop the trunk using the Stop Trunk button, set the Protocol Type field to NONE, and then apply the changes using the Apply Trunk Settings button.

Source: PSTN

Error code: 0x23122

Message: acPSTNNfasTrunkValidationCheck(): Cannot configure backup trunk, there is already backup trunk in this group (GroupId = g).

Explanation: Only one backup trunk can be defined in NFAS group.

System action: Trunk configuration is aborted.

User Response: Configure the trunk as NFAS trunk (set "DChConfig_xx = 2" in the INI file, or the D-channel Configuration field to NFAS in the Trunk Settings page in the web, where xx is the trunk number. If using the web interface, apply the changes using the Apply Trunk Settings button).

Source: PSTN

Error code: 0x23122

Message: acPSTNNfasTrunkValidationCheck(): Cannot configure primary trunk, there is already active primary trunk in this group (GroupId = g).

Explanation: Only one primary trunk can be defined in NFAS group.

System action: Trunk configuration is aborted.

User Response: Configure the trunk as NFAS trunk in the group (set "DChConfig_xx = 2" in the INI file, or the D-channel Configuration field to NFAS in the Trunk Settings page in the web, where xx is the trunk number) or as primary with different group ID (set "NFASGroupNumber_xx = y" in the INI file, or the NFAS Group Number field to y in the Trunk Settings page in the web, where xx is the trunk number and y is the group ID) . If using the web interface, apply the changes using the Apply Trunk Settings button.

Source: PSTN

Error code: 0x23122

Message: acPSTNNfasTrunkValidationCheck(): Cannot configure primary trunk, there is inactive backup trunk in this group (GroupId = g).

Explanation: When configuring on-the-fly (not via ini file) primary trunk for NFAS group that already have backup trunk, the backup trunk must be active before configuring the primary trunk.

System action: Trunk configuration is aborted.

User Response: Activate the backup trunk of the NFAS group or delete it before configuring the primary. Activating a trunk can be done in the Trunk Settings page in the web using the Apply Trunk Settings button. Deleting a trunk can be done in the Trunk Settings page in the web: stop the trunk using the Stop Trunk button, set the Protocol Type field to NONE, and then apply the changes using the Apply Trunk Settings button.

Source: PSTN

Error code: 0x23122

Message: acPSTNNfasTrunkValidationCheck(): Cannot configure NFAS trunk, there is no active primary trunk in this group (GroupId = g).

Explanation: When adding NFAS trunk on-the-fly (not via ini file) to existing NFAS group, the group must have active primary trunk.

System action: Trunk configuration is aborted.

User Response: Configure and activate the primary trunk of the NFAS group (set the NFAS Group Number field to g in the Trunk Settings page in the web. Apply the changes using the Apply Trunk Settings button) and then add the NFAS trunk to the group (set the NFAS Group Number field to g in the Trunk Settings page in the web. Apply the changes using the Apply Trunk Settings button).

Source: PSTN

Error code: 0x23122

Message: acPSTNNfasTrunkValidationCheck(): Cannot configure NFAS trunk, there is inactive backup trunk in this group (GroupId = g).

Explanation: When adding NFAS trunk on-the-fly (not via ini file) to existing NFAS group, both the primary and the backup (if exists) trunks must be active.

System action: Trunk configuration is aborted.

User Response: Activate the backup (if exists) and the primary trunks of the NFAS group (activating a trunk can be done in the Trunk Settings page in the web using the Apply Trunk Settings button) and then add the NFAS trunk to the group (set in the Trunk Settings page in the web the D-channel Configuration field to NFAS and the NFAS Group Number field to g. Apply the changes using the Apply Trunk Settings button)

Source: PSTN

Error code: 0x23122

Message: acPSTNNfasTrunkValidationCheck(): Cannot configure more than 10 trunks in NFAS group.

Explanation: Cannot configure more than 10 trunks in NFAS group.

System action: Trunk configuration is aborted.

User Response: Configure the trunk with non NFAS configuration (set "DChConfig_xx = 0" in the INI file, or the D-channel Configuration field to PRIMARY in the Trunk Settings page in the web, where xx is the trunk number) or delete another NFAS trunk from the group (deleting a trunk can be done in the Trunk Settings page in the web: stop the trunk using the Stop Trunk button, set the Protocol Type field to NONE, and then apply the changes using the Apply Trunk Settings button) and then add this trunk to the group (set "DChConfig_xx = 2" and "NFASGroupNumber_xx = y" in the INI file, or set in the Trunk Settings page in the web the D-channel Configuration field to NFAS and the NFAS Group Number field to y, where xx is the trunk number and y is the group ID. If using the web interface, apply the changes using the Apply Trunk Settings button).

Source: PSTN

Error code: 0x24105

Message: Failed allocating Contexts Pool

Explanation: Initialization process ran out of memory.

System action: The device cannot be used.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24105

Message: Failed initializing Contexts Pool

Explanation: The device’s setup initialization failed.

System action: The device cannot be used.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24105

Message: Allocation of ROOT termination failed

Explanation: The device’s memory is not sufficient.

System action: The device cannot be used.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24105

Message: Allocation of ROOT termination failed

Explanation: The device's memory is not sufficient.

System action: The device cannot be used.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24105

Message: Allocation of termination TerminationId failed

Explanation: An internal error was detected while creating termination pool. Termination database was not created.

System action: The device cannot be used.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24105

Message: Allocation of Termination List buffer failed

Explanation: An internal error was detected while creating termination pool. Termination database was not created.

System action: The device cannot be used.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24105

Message: Allocation of Temporary Termination List buffer failed

Explanation: An internal error was detected while creating termination pool. Termination database was not created.

System action: The device cannot be used.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24105

Message: Allocation of Trunk List buffer failed.

Explanation: An internal error was detected while creating internal buffers.

System action: The device cannot be used.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24105

Message: ActionReply== NULL,ErrCode Error Code ActionReplyList->NoOfactionReply number ActionReplyList->ActionReply[ActionReplyList->NoOfactionReply] : reply.

Explanation: Insufficient resources for building reply.

System action: A MEGACO error message will be sent, system processing continues normally.

User Response: Consult the device manual for MEGACO limitations.

Source: MEGACO

Error code: 0x24105

Message: Allocation of ContextProperties failed.

Explanation: Insufficient resources for building reply.

System action: A MEGACO error message will be sent, system processing continues normally.

User Response: Consult the device manual for MEGACO limitations.

Source: MEGACO

Error code: 0x24105

Message: Failed allocating MEGACO message structs Pool.

Explanation: The device’s memory is not sufficient.

System action: The device cannot be used.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24105

Message: Failed allocating Contexts Pool.

Explanation: The device’s memory is not sufficient.

System action: The device cannot be used.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24105

Message: !!!!MEGACO not running - not enough memory for Pools!!!!!!!!!!.

Explanation: The device’s memory is not sufficient.

System action: The device cannot be used.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24105

Message: Failed allocating Terminations Pool.

Explanation: The device’s memory is not sufficient.

System action: The device cannot be used.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24105

Message: Failed initializing Terminations Pool.

Explanation: The device’s memory is not sufficient.

System action: The device cannot be used.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24105

Message: Failed allocating Topology Pool.

Explanation: The device's memory is not sufficient.

System action: The device cannot be used.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24105

Message: Failed initializing Topology Pool.

Explanation: The device's memory is not sufficient.

System action: The device cannot be used.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24106

Message: eCNPM_UpdateCallBackFunc() returns error code.

Explanation: An internal error was detected during callback function registration.

System action: No Detailed Congestion Reports (H.248.32) will be generated, system processing continues normally.

User Response: in case Detailed Congestion Reports are required – contact AudioCodes customer support, otherwise – ignore the message.

Source: MEGACO

Error code: 0x24106

Message: Illegal priority P for context

Explanation: The context's priority is not in the valid range (0-15).

System action: The command will not be executed

User Response: Check call agent configuration.

Source: MEGACO

Error code: 0x24106

Message: Unexpected CAS line signal received, string CAS state is state. Current events: events

Explanation: State mismatch: the CAS event received does not appear in the current event descriptor. This message will be followed by a current event descriptor. This state might be caused by loss of command from the MGC.

System action: The event is ignored.

User Response: Analyze the network state and command flow in the network.

Source: MEGACO

Error code: 0x24106

Message: Got wrong notification reply TransactionID1, Wait for TransactionID2

Explanation: A mismatch was detected between the expected notification reply and received notification reply.

System action: The termination will keep waiting for the correct notification reply.

User Response: Analyze the network state and command flow in the network.

Source: MEGACO

Error code: 0x24106
Message: (BuildNotify) Failed allocating TunnelString from pool.
Explanation: Insufficient resources for building Observed Event.
System action: Notify will not be sent, system processing continues normally.
User Response: Contact AudioCodes customer support.
Source: MEGACO

Error code: 0x24106
Message: Failed to create BT/BIT Event parameter.
Explanation: Illegal information found in SDP.
System action: Notify will not be sent, system processing continues normally.
User Response: Ignore the message.
Source: MEGACO

Error code: 0x24106
Message: MEGACO UNSUPORTED Digit D!!!
Explanation: Unrecognized digit, digit is not in the valid range (0-15).
System action: Event will not be handled.
User Response: Check call agent configuration.
Source: MEGACO

Error code: 0x24106
Message: MEGACO: Illegal port P to free. BaseUDP=Base MaxPort Max
Explanation: An internal error was detected while trying to free a port.
System action: System processing continues normally.
User Response: Contact AudioCodes customer support.
Source: MEGACO

Error code: 0x24106
Message: MEGACO: Illegal port p to recover. BaseUDP=base MaxPort=max
Explanation: An internal error was detected while trying to recover a port from switch-over.
System action: System processing continues normally.
User Response: Contact AudioCodes customer support.
Source: MEGACO

Error code: 0x24106
Message: MEGACO: No Signal class is initiated. Can't Play Signal PackageName PackageId
Explanation: An unsupported package appeared in the signal request, no supported package can initiate the signal.
System action: The signal will not be played, system processing continues normally.
User Response: Contact AudioCodes customer support.
Source: MEGACO

Error code: 0x24106

Message: MEGACO: error Establishing Bearer. RemoteBindingID=Binding, CID=ChannelId

Explanation: An internal error was detected while generating Establishing Bearer Signal.

System action: The signal will not be played, system processing continues normally.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24106

Message: mcHAL_PlaySignals: Unknown signal to play.

Explanation: Unknown signal received in MEGACO command.

System action: The signal will not be played, system processing continues normally.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24106

Message: mcHAL_PlaySignals: Unknown signal to play.

Explanation: Unknown signal received in MEGACO command.

System action: The signal will not be played, system processing continues normally.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24106

Message: CID ChannelId Not connected to trunk and channel!!!!!!

Explanation: An internal error was detected while playing a CAS signal.

System action: The CAS signal will not be played, system processing continues normally.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24106

Message: CID ChannelId Not connected to trunk and channel!!!!!!

Explanation: An internal error was detected while playing a CAS signal.

System action: The CAS signal will not be played, system processing continues normally.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24106

Message: MEGACO: error Free Resources. NULL Termination

Explanation: An internal error was detected while clearing a call.

System action: Ignored, system processing continues normally.

User Response: Ignore the message.

Source: MEGACO

Error code: 0x24106

Message: MEGACO: error Free Resources. Termination=TerminationId, CID=ChannelId

Explanation: An internal error was detected while clearing a call.

System action: Ignored, system processing continues normally.

User Response: Ignore the message.

Source: MEGACO

Error code: 0x24106

Message: Pending response parse error.

Explanation: An internal error was detected while handling pending response.

System action: The response will not be sent, system processing continues normally.

User Response: Capture network call flow and contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24106

Message: Failed translating Pended transaction tid=TransactionId, err is error .

Explanation: An internal error was detected while creating a pending response.

System action: The response will not be created, system processing continues normally.

User Response: Capture network call flow and contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24106

Message: Failed translating Transaction Pending Cmd TransactionID TransactionId err is error .

Explanation: An internal error was detected while sending pending response.

System action: The response will not be sent, system continues normally.

User Response: Capture network call flow and contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24106

Message: Failed getting packet from network, CmdLen: length

Explanation: An internal error was encountered while reading from the network interface, or the command length exceeds maximal command length.

System action: The packet could not be read, command will not be executed.

User Response: Capture and analyze network call flow.

Source: MEGACO

Error code: 0x24106

Message: acStunUsrHandleResponse failed , err is ErrorCode

Explanation: Malformed Simple Traversal of UDP over NAT (STUN) response received.

System action: The lock state will be preserved.

User Response: Check the STUN server settings.

Source: MEGACO

Error code: 0x24106

Message: Failed translating transaction, err is error

Explanation: An internal error was detected while building a MEGACO response, possibly because the Response size exceeded the maximum transport PDU.

System action: A MEGACO error message will be sent, system processing continues normally.

User Response: Consult the device manual for maximum PDU length. If the needed length does not exceed the documented limits, capture network call flow and contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24102

Message: Could not activate MFCR2 digit map.

Explanation: Quick digit collection is disabled.

System action: System processing continues normally.

User Response: Collect Syslog messages and contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24102

Message: BT/BIT signal with empty signal parameter.

Explanation: A signal with no signal parameters was received.

System action: The signal will not be executed.

User Response: Check call agent configuration.

Source: MEGACO

Error code: 0x24102

Message: Q1990 Header with Error indicator.

Explanation: The tunnel signal according to Q.1990 has incorrect header.

System action: The signal will not be executed.

User Response: Check call agent configuration.

Source: MEGACO

Error code: 0x24102

Message: Q1990 Header with invalid information.

Explanation: The tunnel signal according to Q.1990 has incorrect header.

System action: The signal will not be executed.

User Response: Check call agent configuration.

Source: MEGACO

Error code: 0x24102

Message: MEGACO: Allocation of cpLineSideAnswerSignal fail.

Explanation: An internal error was detected while handling the line side answer signal, insufficient resources, the allocation failed.

System action: The signal will not be executed.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24102

Message: MEGACO: Allocation of cpFarSideNetworkDisconnectSignal fail.

Explanation: An internal error was detected while handling the far side network disconnects, insufficient resources, the allocation failed.

System action: The signal will not be executed.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24102

Message: MEGACO: Allocation of cpRingingWithDisplaySignal fail.

Explanation: An internal error was detected while handling the ringing signal, insufficient resources, and the allocation failed.

System action: The signal will not be executed.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24102

Message: MEGACO: Allocation of cpCallWaitingWithDisplaySignal fail.

Explanation: An internal error was detected while handling the call waiting signal, insufficient resources, and the allocation failed.

System action: The signal will not be executed.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24102

Message: Allocation of cpAnnouncementSignal fail.

Explanation: An internal error was detected while handling the announcement signal, insufficient resources, and the allocation failed.

System action: The signal will not be executed.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24102

Message: Allocation of cpAdvancedAudioPlaySignal fail.

Explanation: An internal error was detected while handling the advanced audio play signal, insufficient resources, and the allocation failed.

System action: The signal will not be executed.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24102

Message: Allocation of cpAdvancedAudioPlayCollectSignal fail.

Explanation: An internal error was detected while handling the advanced audio play collect signal, insufficient resources, and the allocation failed.

System action: The signal will not be executed.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24102

Message: Allocation of cpAdvancedAudioContDigitCollectSignal fail.

Explanation: An internal error was detected while handling the advanced audio cont digit collect signal, insufficient resources, and the allocation failed.

System action: The signal will not be executed.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24102

Message: MEGACO: Allocation of cpCDialDigitsSignal fail

Explanation: An internal error was detected while handling the BCASAddr signal, insufficient resources, and the allocation failed.

System action: The signal will not be executed.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24102

Message: MEGACO: Allocation of cpCPlayToneSignal fail

Explanation: An internal error was detected while handling the play tone signal, insufficient resources, the allocation failed.

System action: The signal will not be executed.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24102

Message: Allocation of Trunk Testing signal fail Name=Type Terminator

Explanation: An internal error was detected while handling the Trunk Testing (Terminator) signal, insufficient resources, the allocation failed.

System action: The signal will not be executed.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24102

Message: Allocation of Trunk Testing Orig. signal fail Name=name Direction=d

Explanation: An internal error was detected while handling the Trunk Testing (Originator) signal, insufficient resources, the allocation failed.

System action: The signal will not be executed.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24102

Message: MEGACO: Illegal parameter P for SigOther descriptor of SignalType signal.

Explanation: One of the SignalType signal's parameters is not recognized.

System action: The signal will not be executed.

User Response: Check call agent configuration.

Source: MEGACO

Error code: 0x24102

Message: MEGACO: Initialization of SignalType fail. ErrCode=code, ErrStr=error.

Explanation: One of the signal's parameters is not supported.

System action: The signal will not be executed.

User Response: Check call agent configuration.

Source: MEGACO

Error code: 0x24102

Message: Name and Direction are mandatory parameters

Explanation: Either the "Name" parameter or the "Direction" parameter is missing from the Trunk Testing signal .

System action: The signal will not be executed.

User Response: Check call agent configuration.

Source: MEGACO

Error code: 0x24102

Message: MEGACO : Parameters AV is not supported.

Explanation: The AV parameter of Announcement signal is not supported.

System action: The parameter is ignored, system processing continues normally.

User Response: Check call agent configuration.

Source: MEGACO

Error code: 0x24102

Message: CAD parameter of analog ringing signal is not support.

Explanation: The CAD parameter is not supported.

System action: The parameter is ignored, system processing continues normally.

User Response: Check call agent configuration.

Source: MEGACO

Error code: 0x24102

Message: FREQ parameter of analog ringing signal is not support.

Explanation: The FREQ parameter is not supported.

System action: The parameter is ignored, system processing continues normally.

User Response: Check call agent configuration.

Source: MEGACO

Error code: 0x24102

Message: MEGACO: Pattern parameter is not supported.

Explanation: The Pattern parameter of the call waiting signal is not supported.

System action: The parameter is ignored, system processing continues normally.

User Response: Check call agent configuration.

Source: MEGACO

Error code: 0x24104

Message: MEGACO already Locked. Cannot perform Graceful shutdown.

Explanation: Trying to perform GracefulLock on an already locked device.

System action: The command will not be executed.

User Response: Check call agent configuration.

Source: MEGACO

Error code: 0x24104

Message: mcAPI_StunInit: Add event to queue failed

Explanation: An internal error was detected upon Simple Traversal of UDP over NAT (STUN) initialization, could not build an internal event in order to complete STUN address resolution.

System action: The device will preserve its current lock state.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24104

Message: mcAPI_StunResponse: Add event to queue failed

Explanation: An internal error was detected upon Simple Traversal of UDP over NAT (STUN) response handling, could not build an internal event in order to complete STUN address resolution.

System action: The device will preserve its current lock state.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24104

Message: MEGACO: Stun response unlock failed

Explanation: An internal error was detected upon Simple Traversal of UDP over NAT (STUN) response handling, could not unlock the device.

System action: The device will preserve its current lock state.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24104

Message: MEGACO: Stun response invalid state

Explanation: An internal error was detected upon Simple Traversal of UDP over NAT (STUN) response handling. A response was received when it was not expected.

System action: The packet is ignored.

User Response:Ignore the message.

Source: MEGACO

Error code: 0x24104

Message: MEGACO: Stun response unlock failed

Explanation: Internal error was detected upon Simple Traversal of UDP over NAT (STUN) response handling or initialization, an attempt to unlock the device failed.

System action: The device will not enter a working state.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0x24107

Message: Unknown Error. Result=code.

Explanation: Execution of the command (either add, modify or move) failed due to an internal error.

System action: The command will not be executed.

User Response: Check call agent configuration.
Source: MEGACO

Error code: 0x24109
Message: MEGACO UNSUPORTED SYSTEM EVENT!!! event index: id
Explanation: An internal error was encountered while trying to parse an internal event.
System action: The internal event will not be handled.
User Response: Contact AudioCodes customer support.
Source: MEGACO

Error code: 0x24109
Message: (Notify) Failed allocating an ActionRequest from pool
Explanation: An internal error was detected while building the notify message.
System action: Failed to build a notify message, the notify will not be handled.
User Response: Contact AudioCodes customer support.
Source: MEGACO

Error code: 0x24109
Message: (BuildNotify) Failed allocating an EventParameter from pool
Explanation: An internal error was detected while building the notify message.
System action: Failed to build a notify message, the notify will not be handled.
User Response: Contact AudioCodes customer support.
Source: MEGACO

Error code: 0x2410A
Message: MEGACO: error changing channel params ErrorCode=code. Reopening is forbidden.
Explanation: Failed to modify the channel's parameters while performing an add/modify/move command.
System action: The command (add/modify/move) will fail.
User Response: Contact AudioCodes customer support.
Source: MEGACO

Error code: 0x2410A
Message: MEGACO: error changing channel params ErrorCode=code.
Explanation: Failed to modify the channel's parameters while performing an add/modify/move command.
System action: The command (add/modify/move) will fail.
User Response: Contact AudioCodes customer support.
Source: MEGACO

Error code: 0
Message: MM: About cpCAudioStream::cpCAudioStream... Buffer Length is too small to handle Audio Data. Length=length
Explanation: Failed to recover audio media stream after switch over.
System action: The redundant device will not have the audio media stream.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0

Message: MM: About cpCDataStream::cpCDataStream... Buffer Length is too small to handle Data Data. Length=length

Explanation: Failed to recover data media stream after switch over.

System action: The redundant device will not have the data media stream.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0

Message: MM: About cpCFaxStream::cpCFaxStream... Buffer Length is too small to handle Fax Data. Length=length

Explanation: Failed to recover fax media stream after switch over.

System action: The redundant device will not have the fax media stream.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0

Message: MMHAL: Fail to DeActivate stream-type stream. ErrorCode=code

Explanation: Failed to deactivate a media stream (audio, fax, or video).

System action: The requested media stream will remain active.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Error code: 0

Message: MMHAL: Fail to Activate stream-type stream. ErrorCode= code

Explanation: Failed to activate a media stream (audio, fax, or video).

System action: The requested stream will not be activated.

User Response: Contact AudioCodes customer support.

Source: MEGACO

Reader's Notes

10 Auxiliary Files

This chapter describes the following auxiliary files:

- Call Progress Tone and User-Defined Tone Auxiliary Files
- Call Progress Tones, User-Defined Tones and Distinctive Ringing
- Prerecorded Tones (PRT) Auxiliary File
- *coeff.dat* Configuration File
- Coder Table File
- Dial Plan file

10.1 Call Progress Tone and User-Defined Tone Auxiliary Files

The auxiliary source file for Call Progress Tones and User-Defined Tones contains the definitions of the Call Progress Tones and User-Defined Tones to be detected/generated by the device. The Call Progress Tones are mostly used for Telephony In-Band Signaling applications (e.g., Ring Back tone). Each tone can be configured as one of the following types:

- Continuous
- Cadence (up to 4 cadences)
- Burst

A tone can also be configured for Amplitude Modulated (AM) (only 8 of the Call Progress Tones can be AM tones). The Call Progress Tones frequency range is 300 Hz to 1890 Hz.

The User-Defined Tones are general purpose tones to be defined by the user. They can be set only as 'Continuous' and their frequency range is 300 Hz to 3800 Hz. The maximum number of tones that may be configured for the User Defined and Call Progress Tones together is 32. The maximum number of frequencies that may be configured in the User Defined and Call Progress Tones together is 64. The device sample configuration file supplied by AudioCodes can be used to construct your own file.

The Call Progress Tones and User-Defined Tones file used by the device 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 modifiable *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 the DConversion Utility supplied with the device's software package. (Refer to the Utilities chapter in the Product Reference Manual 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 device, correctly define their parameters in the device's ***ini*** file. (Refer to 'Initialization (*ini*) Files' on page 29 for the ***ini*** file structure rules and ***ini*** file example.)

10.1.1 Format of the Call Progress Tones Section in the Auxiliary Source File

The format of the Call Progress Tones section in the auxiliary source file starts from the following string:

[NUMBER OF CALL PROGRESS TONES] - containing the following key only:

- **Number of Call Progress Tones** - defines the number of Call Progress Tones to be defined in the file.

[CALL PROGRESS TONE #X] - containing the Xth tone definition (starting from 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 the *enum* definition in the “VoPLib API Reference Manual”, Document #: LTRT-840xx.

- **Tone Modulation Typ**e – The tone may be either Amplitude Modulated (1) or regular (0).
- **Tone Form** – The format of the tone may be one of the following indices:
  - Continuous
  - Cadence
  - Burst
- **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. This parameter is relevant only in case the tone is not Amplitude Modulated.
- **High Freq [Hz]** - Frequency in Hertz of the higher tone component for of a dual frequency tone, or zero (0) for a single tone. This parameter is relevant only in case the tone is not modulated.
- **Low Freq Level [-dBm]** - Generation level 0 dBm to -31 dBm. This parameter is relevant only in case the tone is not Amplitude Modulated.
- **High Freq Level [-dBm]** - Generation level. 0 to -31 dBm. The value should be zero (0) for a single tone. This parameter is relevant only in case the tone is not Amplitude Modulated.
- **First Signal On Time [10 msec]** - “Signal On” period (in 10 msec units) for the first cadence ON-OFF cycle, for cadence tone. When a tone is configured to be continuous, this parameter defines the tone On event detection time. When a tone is configured to be burst tone, it defines the tone’s duration.
- **First Signal Off Time [10 msec]** - “Signal Off” period (in 10 msec units) for the first cadence ON-OFF cycle, for cadence tone. In case of burst tone, this parameter defines the off time required after burst tone ended until the tone detection is reported. For a continuous tone, this parameter is ignored.

- **Second Signal On Time [10 msec]** - “Signal On” period (in 10 msec units) for the second cadence ON-OFF cycle. This may be omitted if there is no second cadence.
- **Second Signal Off Time [10 msec]** - “Signal Off” period (in 10 msec units) for the second cadence ON-OFF cycle. This may be omitted if there is no second cadence.
- **Third Signal On Time [10 msec]** - “Signal On” period (in 10 msec units) for the third cadence ON-OFF cycle. This may be omitted if there is no third cadence.
- **Third Signal Off Time [10 msec]** - “Signal Off” period (in 10 msec units) for the third cadence ON-OFF cycle. This may be omitted if there is no third cadence.
- **Fourth Signal On Time [10 msec]** - “Signal On” period (in 10 msec units) for the fourth cadence ON-OFF cycle. This may be omitted if there is no fourth cadence.
- **Fourth Signal Off Time [10 msec]** - “Signal Off” period (in 10 msec units) for the fourth cadence ON-OFF cycle. This may be omitted if there is no fourth cadence.
- **Carrier Freq [Hz]** – the Carrier signal frequency in case the tone is Amplitude Modulated.
- **Modulation Freq [Hz]** – The Modulated signal frequency in case the tone is Amplitude Modulated (valid range from 1 Hz to 128 Hz).
- **Signal Level [-dBm]** – the tone level in case the tone is Amplitude Modulated.
- **AM Factor [steps of 0.02]** – Amplitude modulation factor. Valid values: 1 to 50. Recommended values: 10 to 25.
- **Default Duration [msec]** - The default duration (in 1 msec units) of the generated tone.

Note 1: When defining the same frequencies for both a continuous tone and a cadence tone, the Signal On Time parameter of the continuous tone should have a value that is greater than the Signal On Time parameter of the cadence tone. Otherwise the continuous tone is detected instead of the cadence tone.

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

Note 3: For more information on generating the Call Progress Tones Configuration file, refer to 'Converting a CPT ***ini*** File to a Binary *dat* File' in 'Utilities' on page 619.

Note 4: When constructing a CPT *dat* file, the **Use dBm units for Tone levels** checkbox must be marked. This checkbox enables defining the levels in [-dBm] units.

10.1.2 Format of the User Defined Tones Section

The format of the User Defined Tones section of the Call Progress Tone source auxiliary file starts from the following string:

[NUMBER OF USER DEFINED TONES] - containing the following key only:

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 – User Defined 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 the *enum* definition in the "VoPLib API Reference Manual", Document #: LTRT-840xx.

- **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.

Note: The detection of a Call Progress or User Defined Tone will be according to the detector frequency deviation as configured in the **ini** file. (Refer to "Initialization ('ini') Files" for the **ini** file structure rules and **ini** file example.)

- **Low Freq Level [-dBm]** - Generation level 0 dBm to -31 dBm.
- **High Freq Level [-dBm]** - Generation level. 0 to -31 dBm. The value should be zero (0) for a single tone.
- **Default Duration [msec]** - The default duration (in 1 msec units) of the generated tone.

Note: The sub-section on 'Format of the Distinctive Ringing Section' is applicable to **MediaPack** only.

10.1.3 Format of the Distinctive Ringing Section

The distinctive ringing section of the ***ini*** file format starts from string:

[NUMBER OF DISTINCTIVE RINGING PATTERNS] - Contains the following key only:

- Number of Distinctive Ringing patterns - Defines the number of distinctive ringing tones to be defined in the file.
- *[Ringing Pattern #X]* - Contains the Xth ringing pattern definition (starting from 1 and not exceeding 16 using the following keys:
    - **Ring Type -** Ring type is equal to the Ringing Pattern number.
    - **Freq [Hz] -** Frequency in Hertz of the ringing tone.
    - **First Ring On Time [10 msec] - "**Ring On" period (in 10 msec units) for the first cadence ON-OFF cycle.
    - **First Ring Off Time [10 msec] - "**Ring Off" period (in 10 msec units) for the first cadence ON-OFF cycle.
    - **Second Ring On Time [10 msec] - "**Ring On" period (in 10 msec units) for the second cadence on-off cycle.

- **Second Ring Off Time [10 msec] - “**Ring Off” period (in 10 msec units) for the second cadence ON-OFF cycle.
- **Third Ring On Time [10 msec] - “**Ring On” period (in 10 msec units) for the third cadence ON-OFF cycle.
- **Third Ring Off Time [10 msec] - “**Ring Off” period (in 10 msec units) for the third cadence ON-OFF cycle.
- **Fourth Ring Off Time [10 msec] - “**Ring Off” period (in 10 msec units) for the fourth cadence ON-OFF cycle.
- **Fourth Ring Off Time [10 msec] - “**Ring Off” period (in 10 msec units) for the fourth cadence ON-OFF cycle.
- **Burst -** Configures the cadence to be a burst cadence in the entire ringing pattern. The burst relates to On time and the Off time of the same cadence. It must appear between “First/Second/Third/fourth” string and the “Ring On/Off Time”

Using this configuration file, you can create up to 16 different distinctive ringing patterns. Every ringing pattern configures the ringing tone frequency and up to 4 ringing cadences. The same ringing frequency is used for all the ringing pattern cadences. The ringing frequency can be configured in the range from 10 Hz up to 200 Hz with a 5 Hz resolution. Each of the ringing pattern cadences is specified by the following parameters:

- Burst cadence is specified by the “Burst” string. This cadence rings once during the ringing pattern. Otherwise, the cadence is interpreted as cyclic: it repeats for every ringing cycle.
- Ring On Time - specifies the duration of the ringing signal.
- Ring Off Time - specifies the silence period of the cadence.

10.1.3.1 Default Template for Call Progress Tones

The device is initialized with the default Call Progress Tones configuration. To change one of the tones, edit the default call *progress txt* file. The table below lists the default call progress tones.

Table 10-1: Default Call Progress Tones

[NUMBER OF CALL PROGRESS TONES] **Number of Call Progress Tones=9**	
#Dial tone [CALL PROGRESS TONE #0]	Tone Type=1 Tone Form = 1 (Continuous) 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

Table 10-1: Default Call Progress Tones

[NUMBER OF CALL PROGRESS TONES] **Number of Call Progress Tones=9**	
#Dial tone [CALL PROGRESS TONE #1]	Tone Type=1 Tone Form = 1 (Continuous) Low Freq [Hz]=440 High Freq [Hz]=0 Low Freq Level [-dBm]=10 (-10dBm) High Freq Level [-dBm]=0 First Signal On Time [10msec]=300
#Ringback [CALL PROGRESS TONE #2]	Tone Type=2 Tone Form = 2 (Cadence) Low Freq [Hz]=440 High Freq [Hz]=480 Low Freq Level [-dBm]=19 (-19dBm) High Freq Level [-dBm]=19 First Signal On Time [10msec]=200 First Signal Off Time [10msec]=400
#Ringback [CALL PROGRESS TONE #3]	Tone Type=2 Tone Form = 2 (Cadence) Low Freq [Hz]=440 High Freq [Hz]=0 Low Freq Level [-dBm]=16 (-16dBm) High Freq Level [-dBm]=0 First Signal On Time [10msec]=100 First Signal Off Time [10msec]=300
#Busy [CALL PROGRESS TONE #4]	Tone Type=3 Tone Form = 2 (Cadence) Low Freq [Hz]=480 High Freq [Hz]=620 Low Freq Level [-dBm]=24 (-24dBm) High Freq Level [-dBm]=24 First Signal On Time [10msec]=50 First Signal Off Time [10msec]=50

Table 10-1: Default Call Progress Tones

[NUMBER OF CALL PROGRESS TONES] **Number of Call Progress Tones=9**	
#Busy [CALL PROGRESS TONE #5]	Tone Type=3 Tone Form = 2 (Cadence) Low Freq [Hz]=440 High Freq [Hz]=0 Low Freq Level [-dBm]=20 (-20dBm) High Freq Level [-dBm]=0 First Signal On Time [10msec]=50 First Signal Off Time [10msec]=50
#Reorder tone [CALL PROGRESS TONE #6]	Tone Type=7 Tone Form = 2 (Cadence) Low Freq [Hz]=480 High Freq [Hz]=620 Low Freq Level [-dBm]=24 (-24dBm) High Freq Level [-dBm]=24 First Signal On Time [10msec]=25 First Signal Off Time [10msec]=25
#Confirmation tone [CALL PROGRESS TONE #7]	Tone Type=8 Tone Form = 2 (Cadence) Low Freq [Hz]=350 High Freq [Hz]=440 Low Freq Level [-dBm]=20 (-20dBm) High Freq Level [-dBm]=20 First Signal On Time [10msec]=10 First Signal Off Time [10msec]=10
#Call Waiting Tone [CALL PROGRESS TONE #8]	Tone Type=9 Tone Form = 2 (Cadence) Low Freq [Hz]=440 High Freq [Hz]=0 Low Freq Level [-dBm]=20 (-20dBm) High Freq Level [-dBm]=0 First Signal On Time [10msec]=30 First Signal Off Time [10msec]=900

Note: This "Default Template for Distinctive Ringing Patterns" section is applicable to **MediaPack** only.

10.1.4 Default Template for Distinctive Ringing Patterns

The MediaPack is initialized with the default Distinctive Ringing Patterns configuration (refer to the table below). To change one of the tones, copy the call progress *txt* file and edit the default distinctive ringing section.

For example: to change the Ringing Pattern 2 to frequency of 35 Hz with a burst initial ringing of 300 msec on and 300 msec off

- replace the ring Freq = 35

- add 2 new lines with First Burst Ring On/Off Time = 30

- Replace the previous “First Ring On/Off Time” to “Second Ring On/Off Time”

Table 10-2: Number Of Distinctive Ringing Patterns

[NUMBER OF DISTINCTIVE RINGING PATTERNS]
Number of Ringing Patterns=14
#Regular North American Ringing Pattern
[Ringing Pattern #0]
Ring Type=0
Freq [Hz]=20
First Ring On Time [10msec]=200
First Ring Off Time [10msec]=400
#GR-506-CORE Ringing Pattern 1
[Ringing Pattern #1]
Ring Type=1
Freq [Hz]=20
First Ring On Time [10msec]=200
First Ring Off Time [10msec]=400
#GR-506-CORE Ringing Pattern 2
[Ringing Pattern #2]
Ring Type=2
Freq [Hz]=20
First Ring On Time [10msec]=80
First Ring Off Time [10msec]=40
Second Ring On Time [10msec]=80

Table 10-2: Number Of Distinctive Ringing Patterns

[NUMBER OF DISTINCTIVE RINGING PATTERNS]
Second Ring Off Time [10msec]=400
#GR-506-CORE Ringing Pattern 3
[Ringing Pattern #3]
Ring Type=3
Freq [Hz]=20
First Ring On Time [10msec]=40
First Ring Off Time [10msec]=20
Second Ring On Time [10msec]=40
Second Ring Off Time [10msec]=20
Third Ring On Time [10msec]=80
Third Ring Off Time [10msec]=400
#GR-506-CORE Ringing Pattern 4
[Ringing Pattern #4]
Ring Type=4
Freq [Hz]=20
First Ring On Time [10msec]=30
First Ring Off Time [10msec]=20
Second Ring On Time [10msec]=100
Second Ring Off Time [10msec]=20
Third Ring On Time [10msec]=30
Third Ring Off Time [10msec]=400
#GR-506-CORE Ringing Pattern 5 - One single Burst of 500 ms
[Ringing Pattern #5]
Ring Type=5
Freq [Hz]=20
First Burst Ring On Time [10msec]=50
First Burst Ring Off Time [10msec]=50
#EN 300 001 Ring - Belgium
[Ringing Pattern #6]
Ring Type=6

Table 10-2: Number Of Distinctive Ringing Patterns

[NUMBER OF DISTINCTIVE RINGING PATTERNS]
Freq [Hz]=25
First Ring On Time [10msec]=100
First Ring Off Time [10msec]=300
#EN 300 001 Ring - Finland
[Ringing Pattern #7]
Ring Type=7
Freq [Hz]=25
First Ring On Time [10msec]=50
First Ring Off Time [10msec]=550
#EN 300 001 Ring - Germany
[Ringing Pattern #8]
Ring Type=8
Freq [Hz]=25
First Ring On Time [10msec]=95
First Ring Off Time [10msec]=450
#EN 300 001 Ring - Italy
[Ringing Pattern #9]
Ring Type=9
Freq [Hz]=35
First Ring On Time [10msec]=100
First Ring Off Time [10msec]=400
#EN 300 001 Ring - Netherlands & Norway
[Ringing Pattern #10]
Ring Type=10
Freq [Hz]=25
First Ring On Time [10msec]=100
First Ring Off Time [10msec]=400
#EN 300 001 Ring - Sweden
[Ringing Pattern #11]
Ring Type=11

Table 10-2: Number Of Distinctive Ringing Patterns

[NUMBER OF DISTINCTIVE RINGING PATTERNS]
Freq [Hz]=35
First Ring On Time [10msec]= 100
First Ring Off Time [10msec]=500
#EN 300 001 Ring - UK
[Ringing Pattern #12]
Ring Type=12
Freq [Hz]=20
First Ring On Time [10msec]= 40
First Ring Off Time [10msec]= 20
Second Ring On Time [10msec]=40
Second Ring Off Time [10msec]=200
#EN 300 001 Ring - Finland
(informative ringing nr. 3: three ringing bursts preceding cyclic ringing)
[Ringing Pattern #13]
Ring Type=13
Freq [Hz]=25
First Burst Ring On Time [10msec]=30
First Burst Ring Off Time [10msec]=30
Second Burst Ring On Time [10msec]=30
Second Burst Ring Off Time [10msec]=30
Third Burst Ring On Time [10msec]=30
Third Burst Ring Off Time [10msec]=30
Fourth Ring On Time [10msec]=100
Fourth Ring Off Time [10msec]=400

10.1.5 Modifying the Call Progress Tones File

Note: The "Modifying the Call Progress ones" section is NOT applicable to **MediaPack.**

Users are supplied with a modifiable Call Progress Tone auxiliary source file (with ***ini*** file extension) and a non-modifiable Call Progress Tone *dat* binary file in the software package under **Auxiliary_Files\Sample_Call_Progress_Files\**.

Only the binary *dat* file can be sent to the device.

In the auxiliary source file, users can modify Call Progress Tone levels, Call Progress Tone frequencies to be detected/generated by the device, to suit user-specific requirements. An example of a Call Progress Tone ***ini*** file name is *call_progress_defaults.dat*. Note that the word 'tones' is defined in the Call Progress Tone ***ini*** file name, to differentiate it from the device's ***ini*** file.

10.1.6 Modifying the Call Progress Tones File & Distinctive Ringing File (MediaPack only)

Users are supplied with a modifiable Call Progress Tone, Distinctive Ringing ***ini*** file and a non-modifiable Call Progress Tone, Distinctive Ringing *dat* binary file in the software package.

Only the binary *dat* file can be sent to the device.

In the ***ini*** file, users can modify Call Progress Tone levels, Call Progress Tone frequencies and the characteristics of the Distinctive Ringing signal to be detected/generated by the device, to suit user-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 and Distinctive Ringing ***ini*** file name, to differentiate it from the device's ***ini*** file.

The default call progress tones configuration is found on call_progress_defaults.***ini*** file. To change one of the tones, edit the default call progress txt file.

10.1.7 Modifying the Call Progress Tone

The default call progress tones configuration is found on *call_progress_defaults.ini* file. 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 the table below with the following text:

#Dial tone

[CALL PROGRESS TONE #1]

Tone Type=1

Tone Form = 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; the dial tone is detected after 3 sec

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 device reports dial tone detection if either one of the two tones is detected.

➢ To modify these *ini* files and send the *dat* file to the device, take these 4 steps:

1. Open the CPT ***ini*** file (it opens in **Notepad** or in a user-defined text file editor.)
2. Modify the file in the text file editor according to your specific requirements.

3. Save your modifications and close the file.
4. Convert the file with the DConvert Utility into a binary dat file (refer to "Converting a Modified CPT ***ini*** File to a *dat* File with the Download Conversion Utility" below).

10.1.8 Converting a Modified CPT ini File to a dat File with the Download Conversion Utility

After modifying the original CPT ***ini*** file (supplied with the device's software package), you can use the Download Conversion Utility to convert the modified file into a *dat* binary file. You can send only 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 'Utilities' on page 619.

After making the *dat* file, send it to the device using either:

- The Web interface GUI's Auxiliary Files. (Refer to 'Auxiliary Files Download'.)

  or
- The BootP/TFTP Server to send 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 'BootP/TFTP Server').

  or
- For cPCI blades, refer to the appropriate section in the VoPLib Application Developer's Manual, Document #: LTRT-844xx.

10.2 Playing the Prerecorded Tones (PRT) Auxiliary File

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.

10.2.1 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 created using a Signaling Editor) if the user intends to download them in separate PCM files. The PCM files should include the following characteristics:

Coder:	G.711 A-law, G.711 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 continuous 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 the DConvert.

Note: The maximum number of prerecorded tones that can be stored in one *dat* file is 40.

10.2.2 Downloading the PRT *dat* File

Downloading the PRT *dat* file into the device can be done using one of the following:

- HTTP
- TFTP
- VoPLib API (not applicable to 260 devices)

For HTTP and TFTP download, refer to **Software Upgrade Wizard** in the product's User's Manual.

For VoPLib API download, refer to the Playing Prerecorded Tones (PRT) section of the VoPLib Application Developer's Manual, Document #: LTRT-844xx.

Note 1: The maximum PRT buffer size is 100KB. (**MediaPack** only)

Note 2 The maximum PRT buffer size is 1MB (All other products).
For the AMS configuration, the maximum PRT buffer size is 2MB.

Note 3: 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.

10.3 Downloading the dat File to a Device

The purpose of the *coeff.dat* configuration file is to provide the best termination and transmission quality adaptation for different line types. The file consists of a set of parameters for the signal processor of the loop interface devices. This parameter set provides control of the following AC and DC interface parameters:

- DC (V / I curve and max current)
- AC impedance matching
- Transmit gain
- Receive gain
- Hybrid balance
- Frequency response in transmit and receive direction
- Hook thresholds (FXS only)
- Ringing generation and detection parameters
- Metering parameters

This means, for example, that changing impedance matching or hybrid balance requires no hardware modifications, so that a single device can meet user-specific requirements. The digital nature of the filters and gain stages also ensures high reliability, no drifts (over temperature or time) and simple variations between different line types.

The *.dat* configuration file is produced by AudioCodes for each market after comprehensive performance analysis and testing and can be modified on request. The current file supports US line type of 600 ohm AC impedance (and for FXS, 40 V RMS ringing voltage for REN = 2).

The following list describes which *coeff.dat* file is to be used with which MP device. The files are located in the Analog_Coefficients_Files folder:

For MP-11x and MP-124RevD FXS coefficients file types:

- *MP11x-02-1-FXS_16KHZ.dat* - supports generation of 16 KHz metering tone and complies with USA standard.
- *MP11x-02-2-FXS_16KHZ.dat* - supports generation of 16 KHz metering tone and complies with TBR21 standard (Pan European).
- *MP11x-02-1-FXS_12KHZ.dat* - supports generation of 12 KHz metering tone and complies with USA standard.
- *MP11x-02-2-FXS_12KHZ.dat* - supports generation of 12 KHz metering tone and complies with TBR21 standard (Pan European).

In a situation where the selection of the metering type (16Khz or 12 KHz) is not important, use *MP11x-02-1-FXS_16KHZ.dat.*

The *dat* configuration file is produced by AudioCodes for each market after comprehensive performance analysis and testing, and can be modified on request. The current file supports US line type of 600 ohm AC impedance and 40 V RMS ringing voltage for REN = 2.

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

To support different types of countries and markets, it is necessary to support loading of a new *Coefficients.ini* file. This file consists of AC and DC line parameters for the peripheral devices.

➢ To send the Coeff.dat file to the device, take this step:

Use either the Web interface GUI's Auxiliary Files. Refer to **Software Upgrade Wizard** in the product's User's Manual.

or

- The BootP/TFTP Server to send to the device the ***ini*** file (which simultaneously downloads the Call Progress Tone ***ini*** file, provided that the device's ***ini*** file parameter CallProgressTonesFilename is defined, and provided that both ***ini*** files are located in the same directory.) (Refer to 'BootP/TFTP Server').

10.4 Coder Table File

The Coder Table file defines which coders are to be supported by the device. It is limited to the supported coders according to the loaded DSP template. Other coders cannot be added.

The following is an example of an ***ini*** file that includes these Coder Table definitions.

This ***ini*** file is converted (using the DConvert utility) to a binary file, and loaded to the device. If no such file is loaded, the default settings are used.

```
[Internal name]    [Coder name] [Txpayload] [RxPayload] [Ptime]
           PCMU            PCMU           0           0      20
           PCMA            PCMA           8           8      20
        G726-16         G726-16          35          35      20
        G726-24         G726-24          36          36      20
        G726-32         G726-32           2           2      20
        G726-40         G726-40          38          38      20
      X-G727-16       X-G727-16          39          39      20
```

```
   X-G727-24-16    X-G727-24-16          40          40      20
      X-G727-24       X-G727-24          41          41      20
   X-G727-32-16    X-G727-32-16          42          42      20
   X-G727-32-24    X-G727-32-24          43          43      20
      X-G727-32       X-G727-32          44          44      20
   X-G727-40-16    X-G727-40-16          45          45      20
   X-G727-40-24    X-G727-40-24          46          46      20
   X-G727-40-32    X-G727-40-32          47          47      20
       G723HIGH            G723           4           4      30
        G723LOW            G723          80          80      30
           G729            G729          18          18      20
           G728            G728          15          15      20
            GSM             GSM           3           3      20
          X-CCD           X-CCD          56          56      20
          EVRC0           EVRC0          60          60      20
     X-EVRC-TFO      X-EVRC-TFO          81          81      20
     X-EVRC-TTY      X-EVRC-TTY          85          85      20
      X-QCELP-8       X-QCELP-8          61          61      20
  X-QCELP-8-TFO   X-QCELP-8-TFO          82          82      20
          QCELP           QCELP          62          62      20
    X-QCELP-TFO     X-QCELP-TFO          83          83      20
          G729E           G729E          63          63      20
       AMR 4 75             AMR          64          64      20
       AMR_5_15             AMR          65          65      20
        AMR 5 9             AMR          66          66      20
        AMR 6 7             AMR          67          67      20
        AMR_7_4             AMR          68          68      20
       AMR 7 95             AMR          69          69      20
       AMR 10 2             AMR          70          70      20
       AMR_12_2             AMR          71          71      20
        GSM-EFR         GSM-EFR          84          84      20
         iLBC13            iLBC         100         100      30
         iLBC15            iLBC         101         101      20
           BV16            BV16         102         102      20
           EVRC            EVRC         103         103      20
telephone-event telephone-event          96          96      20
            RED             RED         104         104      20
             CN              CN          13          13      20
          no-op           no-op         120         120      20
           G722            G722           9           9      20
          EVRCB           EVRCB         103         103      20
          EVRC1           EVRC1         103         103      20
         EVRCB1          EVRCB1         103         103      20
         AMR-WB          AMR-WB         103         103      20
```

The first field is a text representation of the internal coder name. The second field is free text, and contains the name that is to be used in the SDP. The two payload fields define the default payload for this coder. The PTIME field defines the default to be used for this coder. The maximal value is the basic packet size (i.e., 20) multiplied by 6.

10.4.1 Coder Aliases

As explained above, each coder is given a free text name, which should be used in the SDP and in the LCO for MGCP. However, in real life, more than one name for each coder needs to be supported. The aliases mechanism supplies a solution for this need.

Each coder in the coder table has up to 6 hard coded aliases (including the default name) attached to it. If one of the aliases is used in the command, it is used throughout the entire command. For example, if the local SDP in MEGACO defined the coder 'clearmode', the returned SDP also uses it.

The table below defines the aliases used for each of the currently supported coders:

Table 10-3: Aliases Used for Currently Supported Coders

Default Name	Aliases				
PCMU	G.711	G.711U	G.711MULAW	G.711	G711MULAW
PCMA	G.711A	G.711ALAW	G711ALAW		
G726-16	G726_16				
G726-24	G726_24				
G726-32	G726_32				
G726-40	G726_40				
X-G727-16	G727_16	G727-16			
X-G727-24-16	G727_24_16	G727-24-16			
X-G727-24	G727_24	G727-24			
X-G727-32-16	G727_32_16	G727-32-16			
X-G727-32-24	G727_32_24	G727-32-24			
X-G727-32	G727_32	G727-32			
X-G727-40-16	G727_40_16	G727-40-16			
X-G727-40-24	G727_40_24	G727-40-24			
X-G727-40-32	G727_40_32	G727-40-32			
G723	G.723	G723HIGH			
G723	G723LOW				
G729	G.729	G729A	G.729A		
G728					
GSM					
X-CCD	TRANSPARENT	CCD	clearmode		
EVRC0					
X-EVRC-TFO	EVRC_TFO	EVRC-TFO			
X-EVRC-TTY	EVRC_TTY	EVRC-TTY			
X-QCELP-8	QCELP_8	QCELP-8			
X-QCELP-8-	QCELP_8_TFO	QCELP-			

Table 10-3: Aliases Used for Currently Supported Coders

Default Name	Aliases				
TFO		8-TFO			
QCELP	QCELP_13	QCELP-13			
X-QCELP-TFO	QCELP_13_TFO	QCELP-13-TFO	QCELP-TFO		
G729E	G.729E				
AMR	AMR_4_75	AMR-4-75	AMR475	AMR 2	
AMR	AMR_5_15	AMR-5-15	AMR515	AMR 2	
AMR	AMR_5_9	AMR-5-9	AMR590	AMR 2	
AMR	AMR_6_7	AMR-6-7	AMR670	AMR 2	
AMR	AMR_7_4	AMR-7-4	AMR740	AMR 2	
AMR	AMR_7_95	AMR-7-95	AMR795	AMR 2	
AMR	AMR_10_2	AMR-10-2	AMR1020	AMR 2	
AMR	AMR_12_2	AMR-12-2	AMR1220	AMR 2	
GSM-EFR	GSM_EFR				
iLBC	iLBC13	iLBC_13	iLBC-13		
iLBC	iLBC15	iLBC_15	iLBC-15		
BV16	BV_16	BV-16			
EVRC					
telephone-event					
RED					
CN	COMFORT-NOISE				
no-op					
G722	G.722				
EVRCB					
EVRC1					
EVRCB1					
AMR-WB	AMR_WB				

10.4.2 Coder Support Level

The application defines the following support levels for coders:

- None - A coder with support level "None" is not supported. An error is generated if an attempt is made to use the coder.
- Full - A coder with support level "Full" is valid for all type of calls.
- BCT - A coder with support level "BCT" (a new feature) is valid ONLY for BCT calls. The coders iLBC and BV16 belong to this feature. Other coders that appear in the file, but are not supported in the current DSP template, also receive this support level.

The support level is defined internally by the device.

10.4.3 Converting a Modified CoderTable ini File to a dat File Using DConvert Utility

After modifying the original CoderTable (Tbl) ***ini*** file (originally supplied with the device's software package), you can use the DConvert Utility to convert the modified file into a *dat* binary file. (The ***ini*** file cannot be sent.) For more information, refer to Utilities on page 619. You can only send the *dat* file to the device.

After creating the *dat* file, send it to the device using one of the following:

- The Web interface GUI's Auxiliary Files (Refer to Auxiliary Files Download.)

or

- The BootP/TFTP Server - used to send the ***ini*** file (which simultaneously downloads the CoderTbl *dat* file, to the device, The ***ini*** file parameter CoderTblFilename must be enabled and both the ***ini*** file and CoderTbl *dat* file must be located in the same directory.)

10.4.4 Default Coder Table (Tbl) ini file

The following is the default file for building the Coder Table (Tbl) *dat* file:

```
[Internal name]    [Coder name] [Txpayload] [RxPayload] [Ptime]
           PCMA            PCMA           8           8      20
           PCMU            PCMU           0           0      20
        G726-16         G726-16          35          35      20
        G726-24         G726-24          36          36      20
        G726-32         G726-32           2           2      20
        G726-40         G726-40          38          38      20
      X-G727-16       X-G727-16          39          39      20
   X-G727-24-16    X-G727-24-16          40          40      20
      X-G727-24       X-G727-24          41          41      20
   X-G727-32-16    X-G727-32-16          42          42      20
   X-G727-32-24    X-G727-32-24          43          43      20
      X-G727-32       X-G727-32          44          44      20
   X-G727-40-16    X-G727-40-16          45          45      20
   X-G727-40-24    X-G727-40-24          46          46      20
   X-G727-40-32    X-G727-40-32          47          47      20
       G723HIGH            G723           4           4      30
        G723LOW            G723          80          80      30
           G729            G729          18          18      20
           G728            G728          15          15      20
            GSM             GSM           3           3      20
          X-CCD           X-CCD          56          56      20
           EVRC           EVRC0          60          60      20
     X-EVRC-TFO      X-EVRC-TFO          81          81      20
     X-EVRC-TTY      X-EVRC-TTY          85          85      20
      X-QCELP-8       X-QCELP-8          61          61      20
```

```
  X-QCELP-8-TFO   X-QCELP-8-TFO          82          82      20
          QCELP           QCELP          62          62      20
    X-QCELP-TFO     X-QCELP-TFO          83          83      20
          G729E           G729E          63          63      20
       AMR 4 75             AMR          64          64      20
       AMR 5 15             AMR          65          65      20
        AMR 5 9             AMR          66          66      20
        AMR 6 7             AMR          67          67      20
        AMR 7 4             AMR          68          68      20
       AMR 7 95             AMR          69          69      20
       AMR 10 2             AMR          70          70      20
       AMR 12 2             AMR          71          71      20
        GSM-EFR         GSM-EFR          84          84      20
         iLBC13            iLBC         100         100      30
         iLBC15            iLBC         101         101      20
           BV16            BV16         102         102      20
         EVRC C            EVRC         103         103      20
telephone-event telephone-event          96          96      20
            RED             RED         104         104      20
  X-MODEM-RELAY   X-MODEM-RELAY         254         254      20
             CN              CN          13          13      20
      Image/T38       Image/T38         254         254      20
```

10.5 Dial Plan File

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

The following is an example of an ***ini*** file that includes these definitions.

This ***ini*** file is converted (using the TrunkPack Conversion Utility) to a binary file, and loaded to the device.

```
; Example of dial-plan configuration.
; This file contains two dial plans: you may specify which
; one to use in CAS configuration.
[ PLAN1 ]

; Define the area codes 02, 03, 04.
; In these area codes, phone numbers have 7 digits.
02,7
03,7
04,7

; Define the cellular/VoIP area codes 052, 054, 050, and 077.
; In these area codes, phone numbers have 8 digits.
052,8
054,8
050,8
077,8

; Define the international prefixes 00, 012, 014.
; The number following these prefixes may
; be 7 to 14 digits in length.
00,7-14
012,7-14
014,7-14

; Define the emergency number 911.
; No additional digits are expected.
911,0
```

```
[ PLAN2 ]

; Define the area codes 02, 03, 04.
; In these area codes, phone numbers have 7 digits.
0[2-4],7

; Operator services starting with a star: *41, *42, *43.
; No additional digits are expected.
*4[1-3],0
```

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

- Every line in the file defines a known dialing prefix, and the number of digits expected to follow that prefix. The prefix should be separated from the number of additional digits by a comma.
- Empty lines are ignored.
- Lines beginning with a semicolon (";") are ignored.
- Multiple dial plans may be specified in one file. A name in square brackets on a separate line indicates the beginning of a new dial plan. Up to 8 dial plans may be defined.
- Asterisks ("*") and number-signs ("#") may be specified as part of the prefix.
- Numeric ranges are allowed in the prefix.
- A numeric range is allowed in the number of additional digits.

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

The device supports up to 8000 distinct prefixes in the dial-plan file.

10.6 Channel Associated Signaling (CAS) Functions

Note: This sub-section is only applicable to **TP and Mediant** devices.

Note: Conferencing functionality is disrupted by CAS. To assure proper conferencing functionality, disable the CAS parameters.

10.6.1 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 the AudioCodes defined rules.

➢ To generate the protocol file, take these 5 steps:

1. Learn the protocol text file rules the CAS state machine is built from (refer to Table Elements on page 594)
2. Refer to the AudioCodes-supplied CAS files as an example.
3. Build the specific protocol/script text file (for example, *xxx.txt*) file and its related numerical value *h* file (for example, UserProt_defines_xxx.h). Note that the *xxx.txt* file must include the following 'C include' (for example, #include 'UserProt_defines_xxx.h').Compile the *xxx.txt* with the “TrunkPack Downloadable conversion utility” to produce the *xxx.dat* file. Refer to API Demonstration Utilities on page 619 for a detailed description of the utility usage.
4. Compile the *xxx.txt* with the 'TrunkPack Downloadable Conversion Utility' to produce the *xxx.dat* file. Note that the files *xxx.txt, CASSetup.h, cpp.exe* and UserProt_defines_xxx.h must be located in the same folder (You should choose Dynamic Format at the list).
5. Download the *User_protocol.dat* file to the device via acOpenBoard() command at initialization phase.

10.6.2 Table Elements

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

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

10.6.2.1 INIT variables

INIT variables - Numeric values defined by users in UserProt_defines_xxx.h. These values can be used in the file xxx.txt.

For example, INIT_RC_IDLE_CAS defines the ABCD bits expected to be received in IDLE state. INIT_DTMF_DIAL defines the On-time and Off-time for the DTMF digits generated towards the PSTN. Refer to the detailed list in UserProt_defines_xxx.h and in the sample protocol text file (AudioCodes-supplied CAS files). Refer to the following ST_INIT detailed explanation.

10.6.2.2 Actions

Actions (i.e., protocol table events) - Actions are protocol table events activated either by the DSP (e.g., EV_CAS_01) or by users (e.g., EV_PLACE_CALL, EV_TIMER_EXPIRED1). The full list of available predefined events is located in the file CASSetup.h.

10.6.2.3 Functions

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

10.6.2.4 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.

The following table shows an examples taken from an E&M wink start table protocol file:

Table 10-4: ST_DIAL: Table Elements

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

When the state machine reaches the dial state, it sets timer number 2 and then waits for one of two possible actions to be triggered: Either timer 2 expiration or end of dial event. When timer 2 expires, the protocol table executes function SEND_DEST_NUM and remains in the same state (NEXT_STATE=NO_STATE). When the dial event ends, the protocol table sets timer 4 and moves to ST_DIAL_ENDED written in the field NEXT_STATE.

Although users can define their own states, there are two states defined in file CASSetup.h which must appear in every protocol table created. The two states are ST_INIT and ST_IDLE.

Global Parameter	Description
ST_INIT	When channels initialization is selected, the table goes into 'Init' state.
ST_IDLE	When no active call is established or is in the process of being established, the table resides in Idle state, allowing it to start the process of incoming or outgoing calls. When the call is cleared, the state machine table returns to its Idle state.

Process the incoming call detection event by declaring end of digit reception in the following ways (both for ADDRESS/destination number and ANI/source number):

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

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

The ST_INIT state contains functions that initialize the following global parameters:

Table 10-5: Global Parameters

Parameter	Description
INIT_RC_IDLE_CAS	Defines the ABCD bits expected to be received in the IDLE state in the specific protocol. The third parameter used to enable detection of 4 bits` CAS value (see below).
INIT_TX_IDLE_CAS	Defines the ABCD bits transmitted in IDLE state in the specific protocol.
INIT_DIAL_PLAN	A change regarding the issue of an incoming call dialed number. In version 4.2 and earlier, users were required to pre-define the expected number of digits to receive an incoming call. If a lower number of digits than expected was received, the call setup would have failed.
INIT_DTMF_DIAL	Defines the On-time and Off-time for the DTMF digits generated towards the PSTN.
INIT_COMMA_PAUSE_TIME	Defines the delay between each digit when a comma is used as part of the dialed number string (refer to acPSTNPlaceCall for details).
INIT_DTMF_DETECTION	Defines the minimum/maximum On-time for DTMF digit dialing detection.
INIT_PULSE_DIAL_TIME	Not supported by the current stack version. Defines the Break and Make time for pulse dialing.
INIT_PULSE_DIAL	Not supported by the current stack version. Defines the Break and Make ABCD bits for pulse dialing.
INIT_DEBOUNCE	Defines the interval time of CAS to be considered (a stable one).
INIT_COLLECT_ANI	Enables or Disables reception of ANI in a specific protocol.
INIT_DIGIT_TYPE	The #1 parameter defines the dialing method used (DTMF, MF). With MFC-R2 protocols, this parameter is inapplicable (digits are assumed to be R2 digits). The #2 parameter enabled to usage of SS5 tones (not used). The #3 parameter used to enable digits detection at the OutGoing side of the call (which needed at some protocols.
INIT_NUM_OF_EVENT_IN_STATE	Inserted for detection on TOTAL_NUMBER_OF_EVENTS_IN_STATE

Table 10-5: Global Parameters

Parameter	Description
	(CASSetup.h).
INIT_INIT_MGCP_REPORT	Enables the event for MGCP. These tables are specific and relevant to MGCP only. Do not use if otherwise.
INIT_INIT_GLOBAL_TIMERS	Initiates specific timers; it is used with Parameter#1 for metering pulse timer duration.
INIT_PULSE_DIAL_ADDITIONAL_PARAMS	unused
INIT_RINGING_TO_ANALOGUE	When using analogue gateway option - defines the CAS value of ringing (#1) CAS value of silence (#2) and CAS value of polarity reversal(#3).
INIT_DIGIT_TYPE_1	Defines the signaling system used to send operator service.
INIT_REJECT_COLLECT	Define the method for reject collect calls - can be disabled, using Line signaling or using register signaling.
INIT_VERSION	Defines the version number. The version number is relevant to the release version number and is a text information string (not related to the utility compilation version number).
INIT_SIZE_OF_TABLE_PARAM	Users must insert the definition of TOTAL_NUMBER_OF_EVENTS_IN_STATE from CASSetup.h.

10.6.3 Reserved Words

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

10.6.4 State's Line Structure

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

1. action/event
2. function
3. Parameters : #1, #2 etc (dependent on the function)
4. next state

10.6.5 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):

10.6.6 User Command Oriented Action/Event

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

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

Event	Description
EV_CAS_1_1	A new cas a, b bits received (a=1, b=1, was stable for the bouncing period).
EV_CAS_1_0	A new cas a, b bits received (a=1, b=0, was stable for the bouncing period).
EV_CAS_0_1	A new cas a, b bits received (a=0, b=1, was stable for the bouncing period).
EV_CAS_0_0	A new cas a, b bits received (a=0, b=0, was stable for the bouncing period).
EV_CAS_1_1_1_1	A new cas a, b bits received (a=1, b=1, c=1, d=1 was stable for the bouncing period). In order to get such detection (that is different from EV_CAS_1_1) you must put YES at the #3 parameter of INIT_RC_IDLE_CAS

10.6.7 Timer Oriented Events

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

10.6.8 Counter Oriented Events

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

10.6.9 IBS Oriented Events

Event	Description
EV_RB_TONE_STARTED	Ringback tone as defined in the Call Progress Tone ini file (type and index) is detected.
EV_RB_TONE_STOPPED	Ringback tone as defined in the Call Progress Tone ini file (type and index) is stopped after it was previously detected.
EV_BUSY_TONE	Unused
EV_BUSY_TONE_STOPPED	Unused
EV_FAST_BUSY_TONE	Unused
EV_FAST_BUSY_TONE_STOPPED	Unused
EV_ANI_REQ_TONE_DETECTED	R1.5 ANI-request tone as defined in the Call Progress Tone ini file (type and index) is detected.
EV_R15_ANI_DETECTED	R1.5 ANI digit-string was detected.
EV_DIAL_TONE_DETECTED	Dial tone as defined in the Call Progress Tone ini file (type and index) is detected.
EV_DIAL_TONE_STOPPED	Dial tone as defined in the Call Progress Tone ini file (type and index) is stopped after it was previously detected.

10.6.10 DTMF/MF Oriented Events

Event	Description
EV_MFRn_0	MF digit 0 is detected (only DTMF & MFr1)
EV_MFRn_1	MF digit 1 is detected.
EV_MFRn_2	MF digit 2 is detected.
EV_MFRn_3	MF digit 3 is detected.
EV_MFRn_4	MF digit 4 is detected.
EV_MFRn_5	MF digit 5 is detected.
EV_MFRn_6	MF digit 6 is detected.
EV_MFRn_7	MF digit 7 is detected.
EV_MFRn_8	MF digit 8 is detected.
EV_MFRn_9	MF digit 9 is detected.
EV_MFRn_10	MF digit 10 is detected.
EV_MFRn_11	MF digit 11 is detected.

Event	Description
EV_MFRn_0	MF digit 0 is detected (only DTMF & MFr1)
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	This is used when DialMF() is applied and no more dialed number digits are available (they already were sent). For example, the far side requests the next ANI digit but all digits already have been sent. This event usually appears in MFC-R2 tables
EV_FIRST_DIGIT	The first digit of the DNI / ANI number is detected.
EV_DIGIT_IN	An incoming digit (MFR1 or DTMF) is detected
EV_WRONG_MF_LENGTH	An incoming digit was detected, but its duration (ON-TIME) is too long or too short.
EV_DIALED_NUM_DETECTED	The whole destination number detected.
EV_ANI_NUM_DETECTED	The whole source number detected.
EV_DIAL_ENDED	The dialing process finished and all digits dialed.
EV_NO_ANI	When DialMF() is used and no ANI is specified by the outgoing user in function acPSTNPlaceCall(). MFC

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

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

Actions/Events	Description
EV_ACCEPT	When acCASAcceptCall is used (only in MFC-R2) with CALLED_IDLE as its reason parameter (for example, this sends MF backward B-6).
EV_ACCEPT_SPARE_MF1	When acCASAcceptCall is used with SPARE_MF1 as its reason parameter.
EV_ACCEPT_SPARE_MF9	When acCASAcceptCall is used with SPARE_MF9 as its reason parameter.
EV_ACCEPT_SPARE_MF10	When acCASAcceptCall is used with SPARE_MF10 as its reason parameter.
EV_ACCEPT_SPARE_MF11	When acCASAcceptCall is used with SPARE_MF11 as its reason parameter.
EV_ACCEPT_SPARE_MF12	When acCASAcceptCall is used with SPARE_MF12 as its reason parameter.
EV_ACCEPT_SPARE_MF13	When acCASAcceptCall is used with SPARE_MF13 as its reason parameter.
EV_ACCEPT_SPARE_MF14	When acCASAcceptCall is used with SPARE_MF14 as its reason parameter.
EV_ACCEPT_SPARE_MF15	When acCASAcceptCall is used with SPARE_MF 15 as its reason parameter.
EV_REJECT_BUSY	When acCASAcceptCall is used with CALLED_BUSY as its reason parameter.
EV_REJECT_CONGESTION	When acCASAcceptCall is used with CALLED_CONGESTION as its reason parameter.
EV_REJECT_UNALLOCATED	When acCASAcceptCall is used with CALLED_UNALLOCATED as its reason parameter.
EV_REJECT_SIT	When acCASAcceptCall is used with SIT as its reason parameter.
EV_REJECT_RESERVE1	When acCASAcceptCall is used with CALLED_RESERVE1 as its reason parameter.
EV_REJECT_RESERVE2	When acCASAcceptCall is used with CALLED_RESERVE2 as its reason parameter.

10.6.11 Operator Service Events (up to GR-506)

Event	Explanation
EV_SEND_LINE_OPERATOR_SERVICE1	Send operator service 1 (=Operator Released) using line signaling
EV_SEND_LINE_OPERATOR_SERVICE2	Send operator service 2 (=Operator Attached) using line signaling
EV_SEND_LINE_OPERATOR_SERVICE3	Send operator service 3 (=Coin Collect) using line signaling
EV_SEND_LINE_OPERATOR_SERVICE4	Send operator service 4 (=Coin Return) using line signaling
EV_SEND_LINE_OPERATOR_SERVICE5	Send operator service 5 (=Ring-back) using line signaling
EV_SEND_REGISTER_OPERATOR_SERVICE1	Send operator service 1 (=Operator Released) using register signaling
EV_SEND_REGISTER_OPERATOR_SERVICE2	Send operator service 2 (=Operator Attached) using register signaling
EV_SEND_REGISTER_OPERATOR_SERVICE3	Send operator service 3 (=Coin Collect) using register signaling
EV_SEND_REGISTER_OPERATOR_SERVICE4	Send operator service 4 (=Coin Return) using register signaling
EV_SEND_REGISTER_OPERATOR_SERVICE5	Send operator service 5 (=Ring-back) using register signaling
EV_SEND_REGISTER_OPERATOR_SERVICE6	Send operator service 6 (=Coin Collect/Operator Released) using register signaling

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

EV_INIT_CHANNEL

EV_TO_USER

EV_CLOSE_CHANNEL

EV_OPEN_CHANNEL

EV_FAIL_DIAL

EV_FAIL_SEND_CAS

EV_ALARM

10.6.12 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.

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

10.6.13 Parameters

Table 10-6: 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, it can also be written as NONE.
Parameter #2	

Note: In previous versions, 3 parameters were needed per function. From Version 5.2 and on, to enable the dynamic format of the CAS file and reduce memory usage, only the relevant parameters are necessary.

Table 10-7: List of available user functions and their parameters

User Function	User Function Parameters and Descriptions
SET_TIMER	(Timer number, timeout). Sets the timers managed per B-channel. Their expiration triggers the state machine table. Each protocol table/state machine can use up to 8 timers per B-channel/call (timeout in msec) when the timers have 25 msec resolution.
SEND_CAS	(AB value, CD value). ABCD bits are sent as line signaling for the specific channel when the call is setup.
GENERATE_CAS_EV	Check the ABCD bits value, and send a proper event to the state machine.
SEND_EVENT	(Event type, cause). The specific event type is sent to the host/user and retrieved by applying the function acGetEvent().
SEND_DEST_NUM	En-bloc dialing: refers to the digits string located in function acPSTNPlaceCall. Three types are available: (1) DestPhoneNum (2) InterExchangePrefixNum (3) SourcePhoneNum.
DEL_TIMER	(Timer number). Deletes a specific timer or all the timers (0 represents all the timers) for the B-channel.
START_COLLECT	Initiates the collection of address information, i.e., the dialed (destination) number for incoming calls where appropriate, according to the protocol. In the time between START_COLLECT and STOP_COLLECT, no digit is reported to users (EV_DIGIT is blocked) and the destination number is reported in event EV_INCOMING_CALL_DETECTED.

Table 10-7: List of available user functions and their parameters

User Function	User Function Parameters and Descriptions
STOP_COLLECT	Refer to START_COLLECT.
SET_COUNTER	(Counter number, counter value or NONE). Sets counters managed per B-channel. Their expiration triggers the state machine. The counter initialization value should be a non-negative number. To delete all timers, invoke this function with 0 in the counter number field.
DEC_COUNTER	(Counter number). Decreases the counter value by 1. When the counter value reaches 0, EV_COUNTERx_EXPIRES is sent to the table (where x represents the counter number).
RESTRICT_ANI	Indicate the incoming side to hide the ANI from the Far-end user.
SEND_MF	(MF type, MF digit or index or NONE, MF sending time). This function is used only with MFC-R2 protocols.

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

AddressVector and ANIDigitVector	These parameters are initialized when function PlaceCall is used. When the code reaches the dialing section, it sends the MF digit according to the MF type specified in the MF type cell (the types are defined in file CASSetup.h):

Parameter	Description
ADDRESS	Sends the digit from the address vector (destination number) according to the index requested. Refer to the Index definition.
ANI	Sends the digit from the ANI vector (source number) according to the requested index.
SPECIFIC	Sends the MF digit specified in the cell Parameter #2.
SOURCE_CATEGORY	Sends the predefined source category MF digit. The source category digit is set as the parameter SourceNumberingType when function PlaceCall is used. The second and third parameters are ignored when this type is used.
TRANSFER_CAPABILITY	Sends the predefined line category MF digit. The line category digit is set as the parameter TransferCapability when function PlaceCall is used. The second and third parameters are ignored when this type is used.

Index	Specifies the Offset of the next digit to be sent from the vector (ADDRESS or ANI types, described above):

Parameter	Description
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).

MF Send Time	This send time parameter specifies the maximum transmission time of the MF.

Parameter	Description
STOP_SEND_MF	Stops sending the current MF
SEND_PROG_TON	Operation, Tone or NONE.

Two operations are available.

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

CHANGE_COLLECT_TYPE	(Collect Type). Used by the incoming user to indicate that his waiting for receipt of the digit of the requested type. The type can be one of those listed in the following table.

Parameter	Description
ADDRESS	The user waits for receipt of address digits.
ANI	The user waits for receipt of ANI digits.
SOURCE_CATEGORY	The user waits for receipt of the source category.
TRANSFER_CAPABILITY	The user waits for receipt of the source transfer capability (line category).

10.6.14 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.

10.6.15 Changing the Script File

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

10.6.15.1 MFC R2 Protocol

- Use the SEND_MF script function to generate the outgoing call destination number. In this case, the first parameter should be ADDRESS (or ANI for source phone number) and the second parameter -3 to 1 (+1), indicating which digit is sent out of the number that the string conveyed by the user in acPSTNPlaceCall().

1 (+1) implies sending of the next digit.

0 implies a repeat of the last digit.

-1 implies the penultimate digit.

This parameter actually changes the pointer to the phone number string of digits. Thus, a one-to-one mapping with the MF backward signals of the R2 protocol exists.

- Using parameter SEND_FIRST_DIGIT initiates resending the string from the beginning, (change the pointer back to first digit and then proceed as above). This parameter is defined in CASSetup.h.
- When MFC-R2 protocol is used, the two detectors (opened by default) are the Call Progress Tones and MFC-R2 Forward MF. When the user invokes an outgoing call via acPSTNPlaceCall(), MFC-R2 Forward MF detector is replaced with MFC-R2 Backward MF detector, since only two detectors per DSP channel are permitted to operate simultaneously.
- The correct MF is automatically generated according to the call direction - Forward for outgoing calls and Backward for incoming calls.
- MFC-R2 protocol fault can cause a channel block. In this case, the script file provided by AudioCodes releases the call to enable the user to free the call

resources and be notified as to being in blocking state.

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

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

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

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

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

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

Note: Applicable only to MFC-R2 protocol type.

10.6.16 Changing the Values of the Default Parameters of the CAS file (state machine)

The interface to change the ST_INIT parameter values off line is used to define the initialization of the CAS state machine without changing the state machine itself. This interface gives you the flexibility to change some timers and other basic parameters as described below. (No compilation is required). The change is to the configuration and does not affect the state machine itself.

Refer to the section on State above for the ST_INIT parameters.

You can have access with the \Web \ EMS and the VoPLib ***ini*** file parameters.

Note: It is strongly recommended not to change any of the default values unless you understand the changes and know the default values. Every change will affect the configuration of the state machine parameters and the call process related to the trunk you are using with this state machine.

Table 10-8: ST_INIT Parameter Values

Parameter name	Legal Values	Description
CasStateMachineGenerateDigitOnTime	Int - timer value must be positive value Default is -1	Generates digit on-time. The value is in msec.
CasStateMachineGenerateInterDigitTime	Int - timer value must be positive value Default is -1	Generates digit off-time. The value is in msec.
CasStateMachineDTMFMinOnDetection Time	Int - timer value must be positive value Default is -1	Detects digit minimum on time (according to DSP detection information event). The value is in msec.
CasStateMachineDTMFMaxOnDetection Time	Int - timer value must be positive value Default is -1	Detects digit maximum on time (according to DSP detection information event). The

Table 10-8: ST_INIT Parameter Values

Parameter name	Legal Values	Description
		value is in msec.
CasStateMachineMaxNumOfIncomingAddressDigits	Int - default value is -1	Defines the limitation for the Maximum address digits we ever need to collect. After reaching the number of digits, we stop the collection of address.
CasStateMachineMaxNumOfIncomingANIDigits	Int - default value is -1	Defines the limitation for the Maximum ANI digits we ever need to collect. After reaching the number of digits we stop the collection of ANI.
CasStateMachineCollectANI	Char - -1, 0 or 1 Default value is -1, No - 0, Yes - 1.	In some cases, when the state machine handles the ANI collection (not related to MFCR2), you can control the state machine to collect ANI or discard ANI. Collect ANI or not (Yes\No).
CasStateMachineDigitSignalingSystem	Char - -1, 0 or 1 Default value is -1, DTMF - 0, MF - 1.	Defines which Signaling System to use - MF or DTMF on both directions (detection\generation).

The default values for all parameters are set to -1, which are the state machine values.

The replacement towards the CAS state machine takes place at the CAS application initialization only for none default value (-1).

Note: You can change the default\replace state machine initialization parameters only when the state machine is not in use, reset or when it is not related to any trunk. If it is related, you must delete the trunk.

11 RTP/RTCP Payload Types

Latest RTP Payload Types are defined in RFC 3551. For coders that should have dynamic Payload types, proprietary default values have been defined. These defaults are appropriate when working with AudioCodes devices 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 relevant product's Release Notes for the product's supported list of coders.

11.1 Payload Types Defined in RFC 3551

Table 11-1: Payload Types Defined in RFC 3551

Payload Type	Description	Basic Packet Rate [msec]
0	G.711 µ-law	20
3	MS-GSM	20
3	GSM & GSM-EFR	20
4	G.723 (6.3/5.3 kbps)	30
8	G.711 A-law	20
15	G.728	20
18	G.729	20
36	G.726-24	20
38	G.726-40	20
62	QCELP (13.3 kbps)	20
63	G.729E	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	

Note: QCELP-13 default value (63) is not equal to the RFC 3551 value (12) due to backward compatible problem.

11.2 Payload Types Not Defined in RFC 3551

Table 11-2: Payload Types Not Defined in RFC 3551

Payload Type	Description	Basic Packet Rate [msec]
2	G.726 32 kbps	20
35	G.726 16 kbps	20
36	G.726 24 kbps	20
38	G.726 40 kbps	20
39	G.727 16 kbps	20
40	G.727 24-16 kbps	20
41	G.727 24 kbps	20
42	G.727 32-16 kbps	20
43	G.727 32-24 kbps	20
44	G.727-32 kbps	20
45	G.727 40-16 kbps	20
46	G.727 40-24 kbps	20
47	G.727 40-32 kbps	20
56	Transparent PCM	20
60	EVRC	20
61	QCELP_8	20
62	QCELP_13	20
64	AMR & AMR WB	20
65	iLBC	20/30
66	G.722 - 48	20
67	G.722 - 56	20
68	EVRC B (4GV)	20
69	G.729EV	20
70	EG.711	10, 20, 30 *
78	BV16	20

Table 11-2: Payload Types Not Defined in RFC 3551

Payload Type	Description	Basic Packet Rate [msec]
90	Linear PCM	20

* The 30 msec duration for EG.711 is not supported in 6310, 8410, Mediant 1000 Analog, Mediant 3000, IPmedia 3000 and MediaPack 1xx blades.

11.3 Default Dynamic Payload Types which are Not Voice Coders

Table 11-3: Dynamic Payload Types Not Defined in RFC 3551

Payload Type	Description
96	RFC 2833
102	Fax Bypass
103	Modem Bypass
104	RFC 2198
105	NSE
120	No Operation

11.4 Default RTP/RTCP/T.38 Port Allocation

The local default UDP ports for Audio, Video & Fax media streams are set according to the following formula:

Channel's local UDP port = BaseUDPPort + ChannelID*10 + PORT_OFFSET

Table 11-4: Local UDP Port Offsets Table

PORT TYPE	PORT_OFFSET
Audio RTP	0
Audio RTCP	1
T.38	2
Video RTP	4
Video RTCP	5

The *BaseUDPPort* is a configurable parameter which by default is set to 4000.

Example:- The T.38 local UDP port of channel No. 30 would be: 4000 + 30*10 + 2 = 4302.

Reader's Notes

12 DTMF, Fax & Modem Transport Modes

12.1 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.

12.2 Fax/Modem Settings

Users may choose from one of the following transport methods for Fax, V34Fax and for each modem type (V.21/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.

12.3 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. The ECM Fax Mode can be enabled/disabled using the FaxRelayECMEnable parameter settings.

There is a (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 FaxRelayRedundancyDepth configuration parameter.

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

12.4 Configuring Fax/Modem ByPass Mode

When VxxTransportType= 2 (FaxModemBypass, Vxx can be one of the following: V32 / V22 / V21/ Bell/ V.34/ Fax/ V34Fax), 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 Fax relay is enabled, the Answer Tone mode packets are relayed as Fax relay packets.

When the EnableFaxModemInbandNetworkDetection parameter is enabled under the conditions discussed above, a detection of an Answer Tone from the network triggers a switch to bypass mode in addition to the local Fax/Modem tone detections. However, only a high bit-rate coder voice session effectively detects the Answer Tone sent by a remote Endpoint

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 using 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.

It is possible to fine tune fax and modem bypass output line signal levels by appropriately setting the FaxBypassOutputGain and/or ModemBypassOutputGain configuration parameters.

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

12.5 Configuring Fax/Modem Bypass NSE mode

Setting the NSEMode to **1** configures the answering Fax/Modem channel to send NSE packets to the calling Fax/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).

12.6 Supporting V.34 Faxes

Note: The v34faxtransporttype parameter is only supported on **TP-6310, TP-8410, IPM-6310, IPM-8410, Mediant 3000 and IPmedia 3000.**

AudioCodes provides special configuration of the V.34 (Super G3) fax transport method for the channel through the *v34faxtransporttype* parameter.

Note that for using fallback to T.38 mode at v34 fax, both faxtransporttype and v34faxtransporttype should be configured to work in relay mode.

The expected upcoming events will be the same as for any G3 Fax transfer.

Note: For all the setups described below, the CNG tone detector is disabled.

12.6.1 Using Bypass Mechanism for V.34 Fax Transmission

Configuration:

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

Expected events for V.34 Fax to V.34 Fax - Bypass Mode are shown in the table below.

Table 12-1: V.34 Fax to V.34 Fax - 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 device changes its status to bypass mode upon receiving fax bypass packet from the remote side.

Note that AudioCodes recommends this setup since it reaches the full rate of modem/fax transfer. 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.

12.6.2 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.34 Fax **to** V.34 Fax - Events Only Mode are shown in the table below.

Table 12-2: V.34 Fax to V.34 Fax - Events Only Mode

Calling	Answering

Table 12-2: V.34 Fax to V.34 Fax - Events Only Mode

Calling	Answering
	EV_DETECT_ANSWER_TONE
	EV_DETECT_FAX

12.6.3 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 & V.34 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.34 Fax to V.34 Fax - Relay Mode are shown in the table below.

Table 12-3: V.34 Fax to V.34 Fax - Relay Mode

Calling	Answering
	EV_DETECT_ANSWER_TONE
	EV_DETECT_FAX
EV_DETECT_FAX	
EV_END_FAX	EV_END_FAX

13 Utilities

This section describes the functionality and operation of a list of utilities supplied with the TrunkPack software package.

13.1 API Demonstration Utility

⚠ **Note:** This sub-section on API Demonstration Utility is not applicable to **MediaPack**.

LOCATION:

```
.\VoP_API_Library\VoPLib_Tcl_Extension\<OS>\<CPU>\apirunce
```

DESCRIPTION:

This utility is designed to serve both as a reference for using the VoPLib and as demo applications, which the user can run immediately after installing the device/module.

OPERATION:

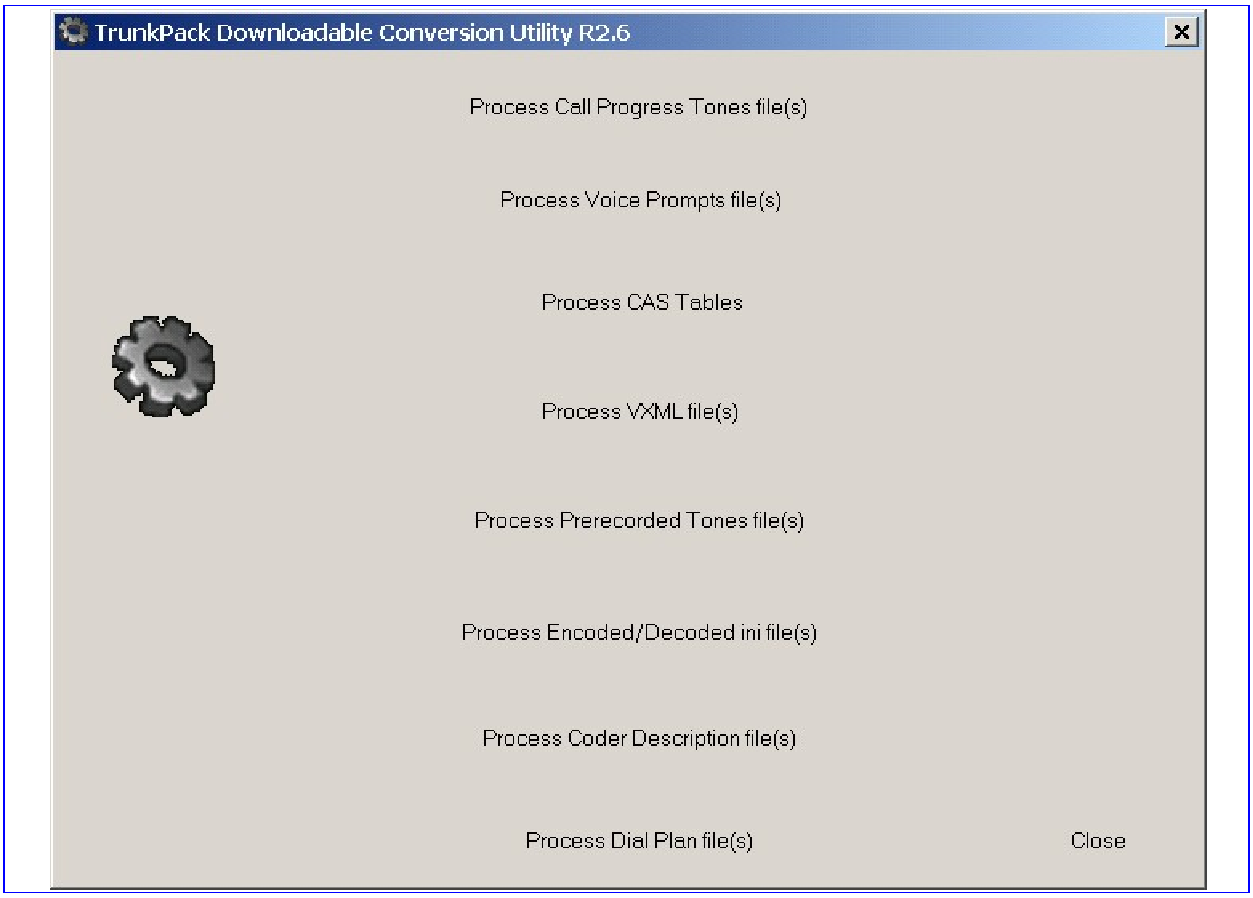
The Apirunce Application - TCL-based demo program available for Linux™, Solaris™ and Windows™ OSs. With this application the user can build scripts online or offline and execute them. All new APIs in this version are supported by Apirunce.

13.2 TrunkPack Downloadable Conversion Utility

LOCATION:

```
.\Utilities\DConvert\DConvert.exe
```

Figure 13-1: TrunkPack Downloadable Conversion Utility R2.6.2



TrunkPack Downloadable Conversion Utility R2.6
Process Call Progress Tones file(s)
Process Voice Prompts file(s)
Process CAS Tables
Process VXML file(s)
Process Prerecorded Tones file(s)
Process Encoded/Decoded ini file(s)
Process Coder Description file(s)
Process Dial Plan file(s)
Close

This utility is used to generate the following:

- Process Call Progress Tones file(s)
- Process Voice Prompts file(s)
- Process CAS Tables
- Process Prerecorded Tones file(s)
- Process Encoded/Decoded ***ini*** file(s)
- Process Coder Description file(s)
- Process Dial Plan file(s)

Using the MediaPack, the above files can be used when:

- Using an *ini* file during BootP/DHCP session
- Using the Web Interface

Some files may have usage restrictions as described under their usage information.

Using all devices (except for the MediaPack and 260/UNI), the files constructed using these utilities can be used when:

- Configuring the device 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.

The above files can be used when configuring the device using the VoPLib function acOpenBoard().

Some files may have usage restrictions as described under their usage information.

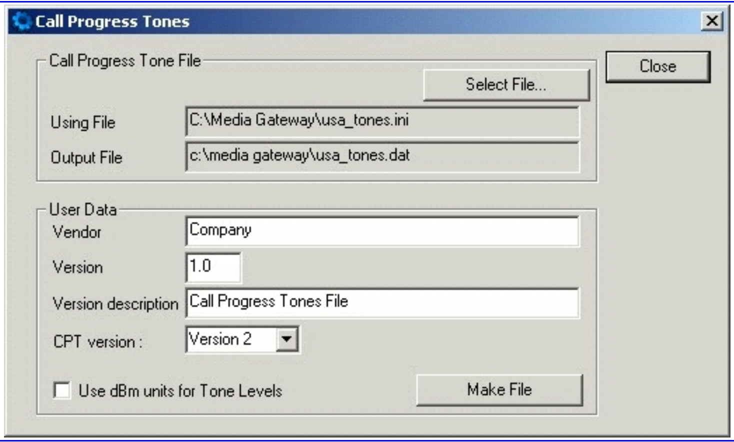
13.2.1 Process Call Progress Tones File(s)

➢ To convert a CPT ini file to a binary dat file, take these 8 steps:

1. For **MediaPack**, create a CPT *ini* file using the direction in Modifying the Call Progress Tones File & Distinctive Ringing File on page 584, or by editing a CPT *ini* file provided by AudioCodes.

   For **devices other than MediaPack**, create a CPT *ini* file using the direction in 'Modifying the Call Progress Tones File' on page 583, or by editing a CPT *ini* file provided by AudioCodes.
2. Execute *DConvert.exe* and click the **Process Call Progress Tones file(s)** button. The Call Progress Tones dialog appears.

Figure 13-2: Call Progress Tones Screen



Using File: C:\Media Gateway\usa_tones.ini

Output File: c:\media gateway\usa_tones.dat

Vendor: Company

Version: 1.0

Version description: Call Progress Tones File

CPT version : Version 2

☐ Use dBm units for Tone Levels

3. Click the **Select File . . .** button and navigate to the location of the CPT *ini* file that you want to convert.
4. Select the required file and click **Open**. The name and path of both the CPT *ini* file and the *dat* file appear in the **Using File** field and **Output File** field respectively. (The file names and paths are identical except for the file extension.)
5. Fill in the **Vendor, Version** and **Version Description** fields.
   - **Vendor** field - 256 characters maximum
   - **Version** field - must be made up an integer, followed by a period '**.**', then followed by another integer (e.g., 1.2, 23.4, 5.22)
   - **Description** field - 256 characters maximum
6. The default value of the CPT version drop-down list is **Version 3**. Do one of the following:
   - If the software version release you are using is 4.4, in the **CPT Version** drop-down list, select **Version 2**.
   - If the software device version release is prior to version 4.4, in the **CPT Version** drop-down list, select **Version 1** (to maintain backward compatibility).
7. The **Use dBm units for tone levels** checkbox unchecked by default. To use -dBm units for setting the Call Progress Tone and User Defined Tone Levels, click

a checkmark into the **Use dBm units for tone levels** checkbox. This checkbox should be checked to maintain backward compatibility.

Note: The default value of the **dBm units for tone levels** checkbox is left unchecked for backward compatibility with versions prior to version 4.4.

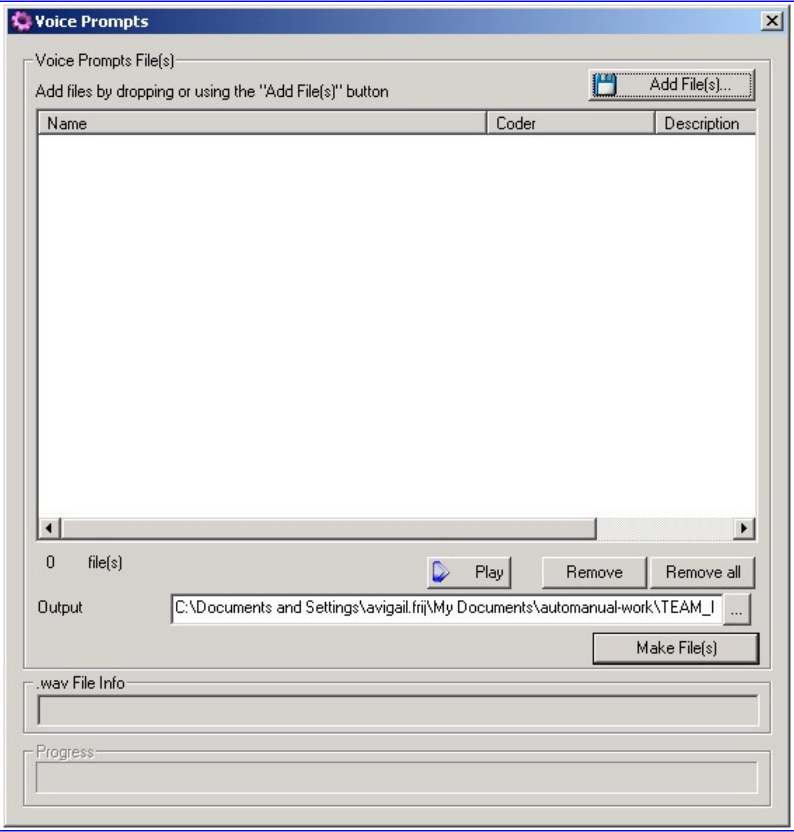
8. Click the **Make File** button. The *dat* file is generated and placed in the same directory as shown in the **Output File** field. A message box informing you that the operation was successful indicates that the process is completed.

13.2.2 Process Voice Prompts File(s)

➢ **To generate a Voice Prompts file, take these 12 steps:**

1. Create raw Voice Prompt files according to the instructions in the section on “Relaying DTMF/MF Digits” in the “VoPLib User’s Manual”, (Document #: LTRT-844xx). Note that starting from version 1.2 (device version 4.2), **DConvert** accepts *wav* files as well.
2. Execute *DConvert.exe* and click the **Process Voice Prompts file(s)** button. The Voice Prompts window appears.

Figure 13-3: Voice Prompts Screen



Voice Prompts File(s)

Add files by dropping or using the "Add File(s)" button

Name	Coder	Description

0 file(s)

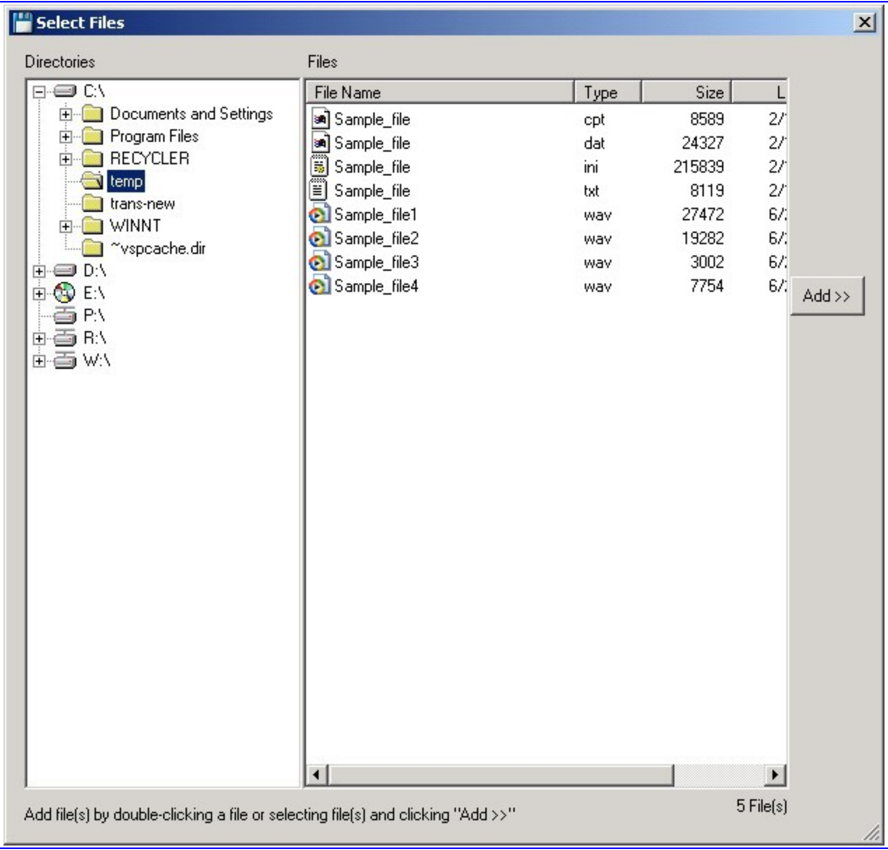
Output C:\Documents and Settings\avigail.frij\My Documents\automanual-work\TEAM_I

3. Select the raw **Voice Prompt** files (created in Step 1) step either by one of these actions:
   a. Click the **Add Files** button in the upper right corner. The Add Files window appears. (Refer to the figure, "Select Files Window" below.)

b. Navigate to the appropriate file.

c. Select it and click the **Add>>** button. To close the **Add Files** window, click the ☒ Exit button. (Press the **Esc** key to cancel changes.)

Figure 13-4: Select Files Window

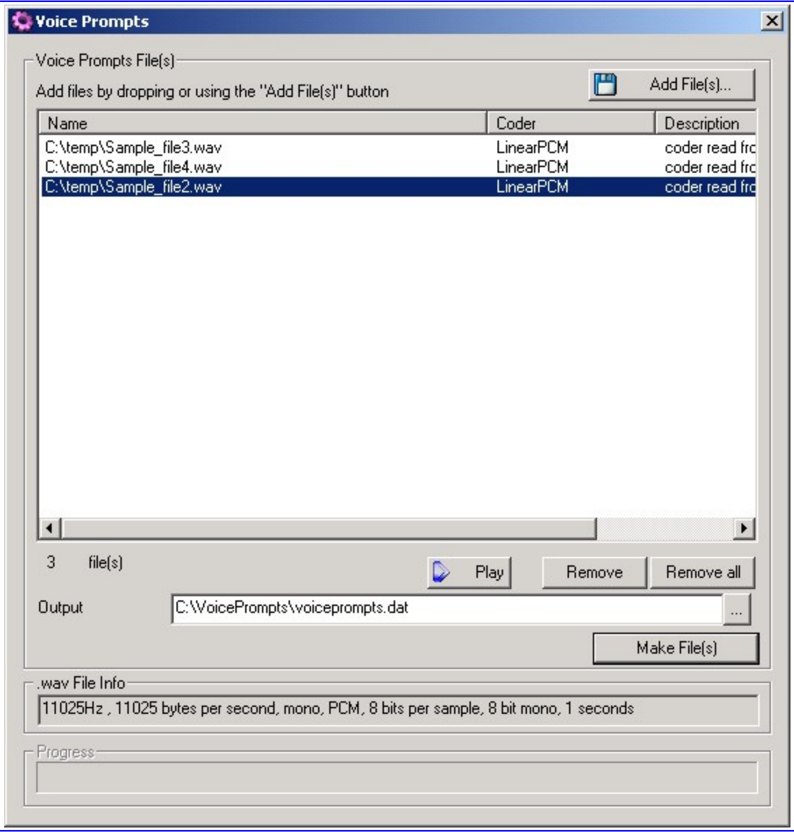


File Name	Type	Size	L
Sample_file	cpt	8589	2/
Sample_file	dat	24327	2/
Sample_file	ini	215839	2/
Sample_file	txt	8119	2/
Sample_file1	wav	27472	6/
Sample_file2	wav	19282	6/
Sample_file3	wav	3002	6/
Sample_file4	wav	7754	6/

Add file(s) by double-clicking a file or selecting file(s) and clicking "Add >>"

5 File(s)

Figure 13-5: 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

11025Hz , 11025 bytes per second, mono, PCM, 8 bits per sample, 8 bit mono, 1 seconds

d. Drag and drop files onto the **Voice Prompts** window.

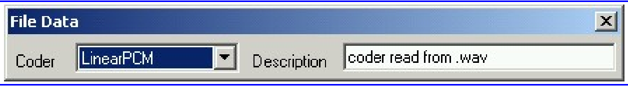
4. Arrange the files as required by dragging and dropping them from one location in the list to another location.

Note: The sequence of files in the “Add Files...” window defines the Voice Prompt ID.

5. 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.

6. 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 13-6: File Data Window



Coder LinearPCM Description coder read from .wav

7. From the **Coder** drop-down list, select a coder type (to be used by the acPlayVoicePrompt() function).

8. In the **Description** field, enter a description (optional).

> **Note:** For *wav* files, a coder is automatically selected from the *wav* file header.

9. Close the File Data dialog by clicking on the Exit button. (Press the **Esc** key to cancel changes.). You are returned to the Voice Prompts window.
10. The default **Output** file name is *voiceprompts.dat*.You can modify it. Or,

    Use the ... Browse button to select a different Output file. Navigate to the required file and select it. The selected file name and its path appear in the **Output** field.
11. Click the **Make File(s)** button to generate the Voice Prompts file. The Progress bar at the bottom of the window is activated. The *dat* file is generated and placed in the same directory as shown in the **Output File** field. A message box informing you that the operation was successful indicates that the process is completed.
12. 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.

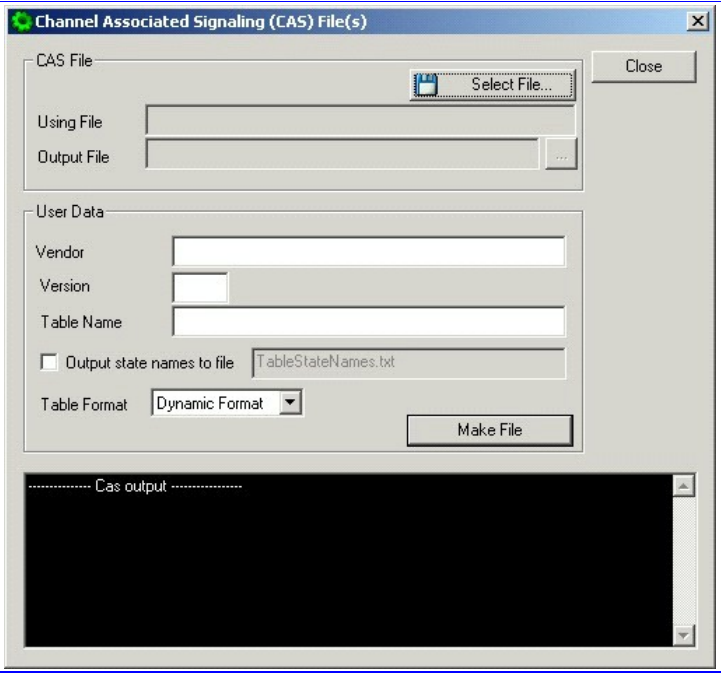
13.2.3 Process CAS Tables

➢ To produce a CAS table, take these 10 steps:

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 the “VoPLib User’s Manual”, Document #: LTRT-844xx.
2. Copy the files generated in the previous step (or at least the *xxx.h* file) to the same directory in which *DConvert.exe* is located and make sure that the two following files, *CASSetup.h* and *CPP.exe,* are also located in this same directory.

3. Execute *DConvert.exe* and click the **Process CAS Tables** button. The Call Associated Signaling (CAS) Window appears.

Figure 13-7: Call Associated Signaling (CAS) Screen


Channel Associated Signaling (CAS) File(s)
CAS File
Select File...
Close
Using File
Output File
...
User Data
Vendor
Version
Table Name
Output state names to file
Table Format
Make File
------------- Cas output ---------------

□ Output state names to file: TableStateNames.txt

Table Format: Dynamic Format

4. Click the **Select File** button. A Browse window appears.
5. Navigate to the required 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. The Table Name is also automatically designated.)
6. Fill in the **Vendor** and **Version** fields.
   - **Vendor** Field - 32 characters maximum
   - **Version** Field - must be made up an integer, followed by a period "**.**", then followed by another integer (e.g., 1.2, 23.4, 5.22)
7. Modify the **Table Name** if required.
8. For troubleshooting purposes, you can click a check into the **Output state names to file** checkbox. This activates the file name field in which the default file name, **TableState Names.txt** appears. You can modify the file name if required. The file is located in the same directory as the **Using file** and **Output file** designated above.
9. In the **Table Format** select box, choose the format you want to use:
   - Old Format - This format is supported in all versions. Many CAS features are not supported in this format.
   - New Format - supported from Ver. 4.2 and on. From 5.2 and on - there will be new features that this format will not support.
   - Dynamic Format - supported from Ver. 5.2 and on. There may be 5.2 features that will supported only in this format.
     The size of the file with dynamic format is significantly lower.
10. Click the **Make File** button. The *dat* file is generated and placed in the same directory as shown in the **Output File** field. A message box informing you that the operation was successful indicates that the process is completed.

On the bottom of the Call Assisted Signaling (CAS) Files(s) window, the CAS output log box displays the log generated by the process. It can be copied as needed. The information in it is **NOT** retained after the window is closed.

> **Note:** The process verifies the input file for validity. Invalid data causes an error and the process is aborted. For more details, refer to the log box.

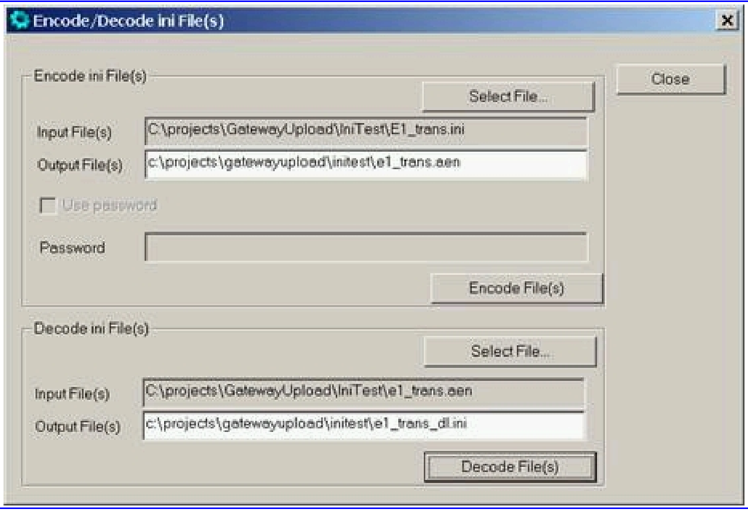
The *ini* file can be both encoded and decoded using **DConvert**. Encoding usually takes place before downloading an *ini* file to the device while decoding usually takes place after uploading an ***ini*** file from the device.

➢ To Encode an *ini* file, take these 5 steps:

1. Prior to the encoding process, the user should prepare the appropriate *ini* file either by uploading from the device or by constructing one.

   Execute *DConvert.exe* and click the **Process Encoded *ini* file(s)** button. The Encoded *ini* file(s) window appears.

Figure 13-8: Encoded ini File(s) Screen



Input File(s) C:\projects\GatewayUpload\IniTest\E1_trans.ini

Output File(s) c:\projects\gatewayupload\initest\e1_trans.aen

□ Use password

Password

Input File(s) C:\projects\GatewayUpload\IniTest\e1_trans.aen

Output File(s) c:\projects\gatewayupload\initest\e1_trans_dl.ini

2. In the **Encode *Ini* File(s)** area, click the **Select File…** Button. A Browse window appears.
3. Navigate to the required location and select the *ini* file to be encoded. (This automatically designates the output file as the same name and path, but with the *aen* extension.

> **Note:** The Password field is to be implemented in a future version.

4. Click the **Encode File(s)** button. The encoded file is generated and placed in the same directory as shown in the **Output File** field. A message box informing you that the operation was successful indicates that the process is completed.

The encoded *ini* file can be loaded using the regular *ini* file procedure. To upload a file from a device, use the Web Interface.

- **To Decode an *ini* file, follow these 4 steps:**

1. Prior to the decoding process, the user should prepare the appropriate encoded *ini* file either by uploading from the device or by using the encoding process on an existing *ini* file.
2. Execute *DConvert.exe* and click the **Process Encoded** *ini* **file(s)** button.
3. In the **Decode *Ini* File(s)** area, click **Select File(s)** and select the file to be decoded. (This automatically designates the output file as the same name and path, but with the extension, *_dl.ini*.
4. Click the **Decode File(s)** button. The decoded file is generated and placed in the same directory as shown in the **Output File** field. A message box informing you that the operation was successful indicates that the process is completed.

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

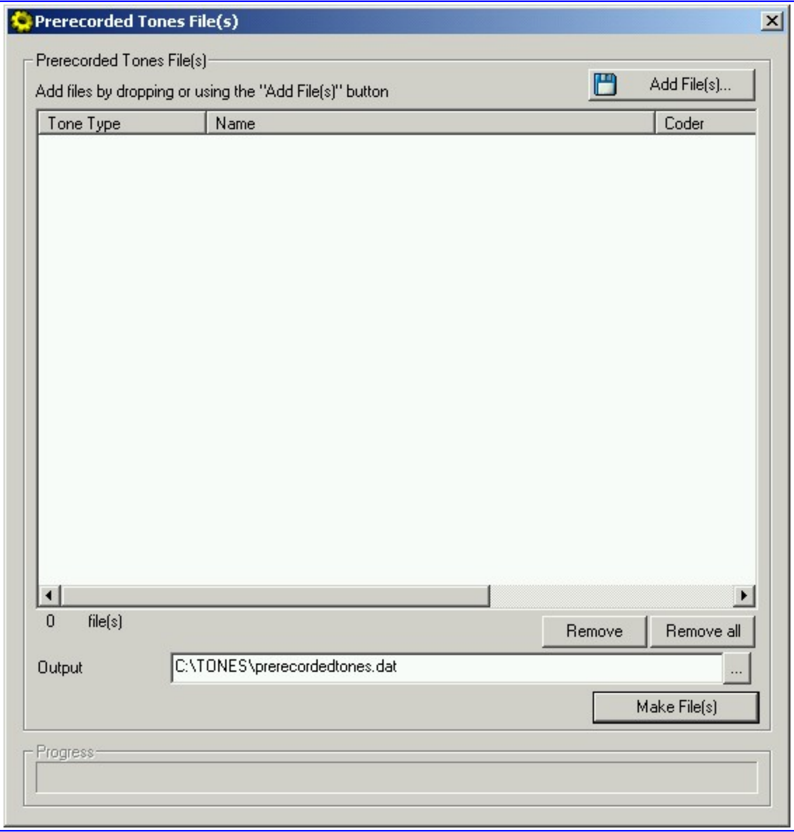
13.2.4 Process Prerecorded Tones File(s)

- **To generate a Prerecorded Tones file, take these 11 steps:**

1. Prior to the conversion process, the user should prepare the appropriate prerecorded tones file(s).

2. Execute *DConvert.exe* and press the **Process Prerecorded Tones file(s)** button. The Prerecorded Tones file(s) window appears.

Figure 13-9: Prerecorded Tones File(s) Screen



Prerecorded Tones File(s)
Prerecorded Tones File(s)
Add files by dropping or using the "Add File(s)" button
Add File(s)...
Tone Type
Name
Coder
0 file(s)
Remove
Remove all
Output
...
Make File(s)
Progress

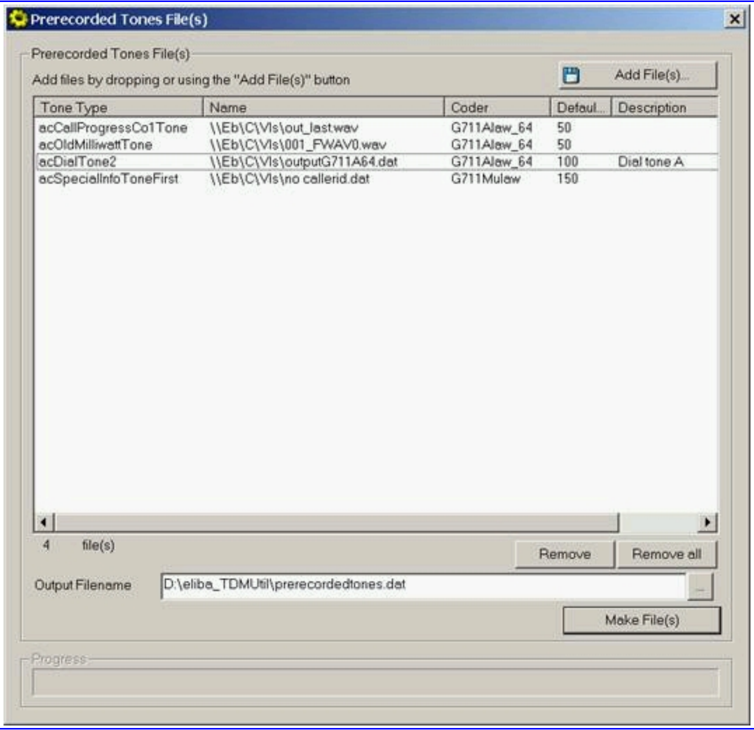
C:\TONES\prerecordedtones.dat

3. Select the raw Prerecorded Tones files (created in Step 1) utilizing one of these actions:
   a. Click the **Add Files** button in the upper right corner. The Add Files window appears. (Refer to the figure, Select Files Window.)

      Navigate to the appropriate file.

Select it and click the **Add>>** button. (To close the Add Files window, click the ☒ Exit button. Press the **Esc** key to cancel changes.) You are returned to the Prerecorded Tones file(s) window.

Figure 13-10: Prerecorded Tones File(s) Screen with wav Files



Prerecorded Tones File(s)

Add files by dropping or using the "Add File(s)" button

Tone Type	Name	Coder	Defaul...	Description
acCallProgressCo1Tone	\\Eb\C\VIs\out_last.wav	G711Alaw_64	50	
acOldMilliwattTone	\\Eb\C\VIs\001_FWAV0.wav	G711Alaw_64	50	
acDialTone2	\\Eb\C\VIs\outputG711A64.dat	G711Alaw_64	100	Dial tone A
acSpecialInfoToneFirst	\\Eb\C\VIs\no callerid.dat	G711Mulaw	150	

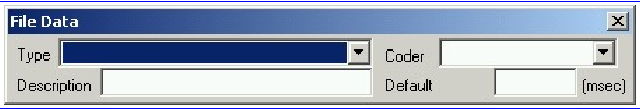
4 file(s)

Output Filename: D:\eliba_TDMUtil\prerecordedtones.dat

b. Drag and drop files onto the Prerecorded Tones File(s) Screen.

4. To define a tone type, coder and default duration for each file, select the file (or group of files to be set the same) and double click or right click on it. The File Date window appears.

Figure 13-11: File Data Dialog Box



5. From the **Type** drop-down list, select a Ring parameter type.

6. From the **Coder** drop-down list, select a coder type (G.711 A-law_64, G.711 µ-law, or Linear PCM).

7. In the **Description** field, enter a description (optional).

8. In the **Default** field, enter the duration in msec.

9. Click the ☒ Exit button. (Press the **Esc** key to cancel changes.) You are returned to the Prerecorded Tones file(s) window.

10. The default **Output** file name is *prerecordedtones.dat*. You can modify it. Or,

Use the ▯ Browse button to select a different Output file. Navigate to the required file and select it. The selected file name and its path appear in the **Output** field.

11. Click **Make File(s)** button. The Progress bar at the bottom of the window is activated. The *dat* file is generated and placed in the same directory as shown in the **Output File** field. A message box informing you that the operation was successful indicates that the process is completed.

13.2.5 Process Encoded/Decoded ini File(s)

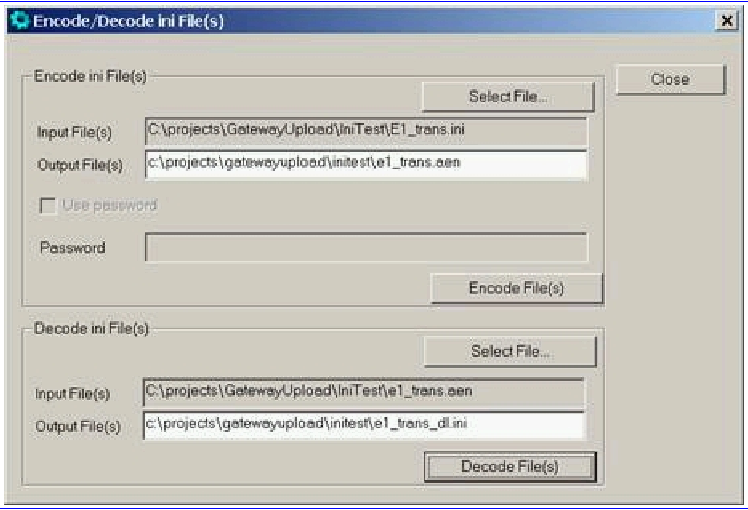
The ***ini*** file can be both encoded and decoded using **DConvert**. Encoding usually takes place before downloading an ***ini*** file to the device while decoding usually takes place after uploading an ***ini*** file from the device.

➢ To Encode an *ini* file, take these 4 steps:

1. Prior to the encoding process, the user should prepare the appropriate ***ini*** file either by uploading from the device or by constructing one (refer to 'Initialization (ini) File' on page 29).

   Execute *DConvert.exe* and click the **Process Encoded *ini* file(s)** button. The Encoded ***ini*** file(s) window appears.

Figure 13-12: Encoded ini File(s) Screen



Input File(s) C:\projects\GatewayUpload\IniTest\E1_trans.ini

Output File(s) c:\projects\gatewayupload\initest\e1_trans.aen

☐ Use password

Password

Input File(s) C:\projects\GatewayUpload\IniTest\e1_trans.aen

Output File(s) c:\projects\gatewayupload\initest\e1_trans_dl.ini

2. In the **Encode *ini* File(s)** area, click the **Select File…** Button. A Browse window appears.
3. Navigate to the required location and select the ***ini*** file to be encoded. (This automatically designates the output file as the same name and path, but with the *aen* extension.

Note: The Password field is to be implemented in a future version.

4. Click the **Encode File(s)** button. The encoded file is generated and placed in the same directory as shown in the **Output File** field. A message box informing you that the operation was successful indicates that the process is completed.

The encoded ***ini*** file can be loaded using the regular ***ini*** file procedure. To upload a file from a device, use the Web Interface (refer to 'Software Update').

➢ To Decode an *ini* file, follow these 4 steps:

1. Prior to the decoding process, the user should prepare the appropriate encoded ***ini*** file either by uploading from the device or by using the encoding process on an existing ***ini*** file.
2. Execute *DConvert.exe* and click the **Process Encoded *ini* file(s)** button.
3. In the **Decode *ini* File(s)** area, click **Select File(s)** and select the *aen* file to be decoded. (This automatically designates the output file as the same name and path, but with the extension, *_dl.ini*.
4. Click the **Decode File(s)** button. The decoded file is generated and placed in the same directory as shown in the **Output File** field. A message box informing you that the operation was successful indicates that the process is completed.

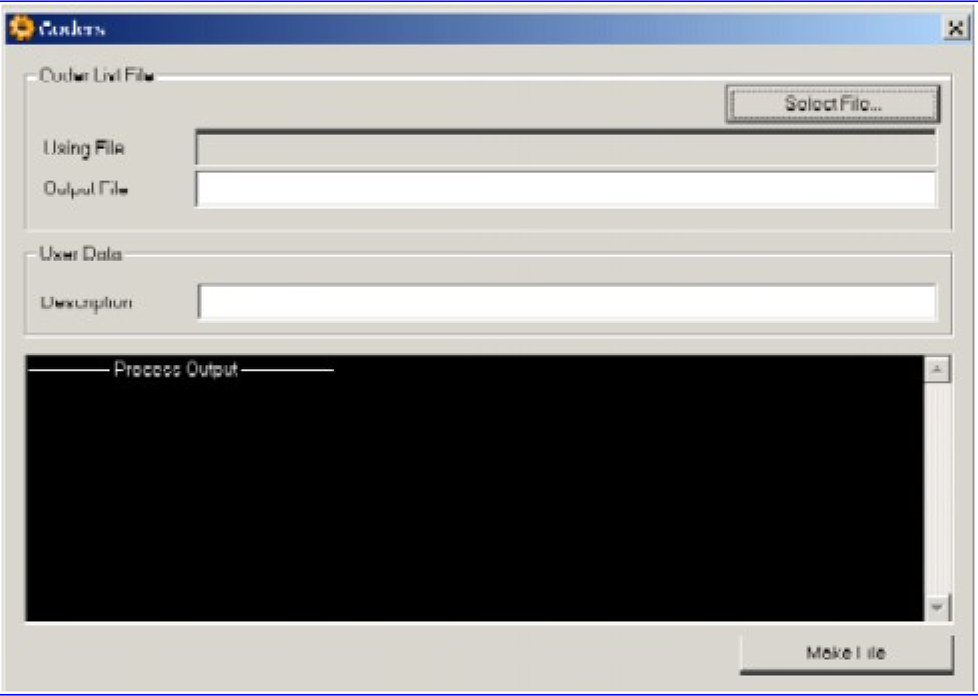
Note: The decoding process verifies the input file for validity. Any change made to the encoded file causes an error and the decoding process is aborted.

13.2.6 Process Coder Description File(s)

➢ To produce a Coder Description file, take these 7 steps:

1. Construct a Coder Description text file according to the instructions in Coder Table File on page 587.
2. Execute DConvert.exe and click the Process Coder Description button. The Coders Window appears.

Figure 13-13: Coders Screen



3. Click the **Select File** button. A Browse window appears.

4. Navigate to the required 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). The output file name may be altered.
5. Fill in the **Description** field. This step is optional. The maximum description length is 64 chars.
6. Click the **Make File** button. The .dat file is generated and placed in the same directory as shown in the **Output File** field. A message box informing you that the operation was successful indicates that the process has been completed.
7. On the bottom of the Coders window, the Coders output log box displays the log generated by the process. It may be copied as needed. This information is **NOT** retained after the window has been closed.

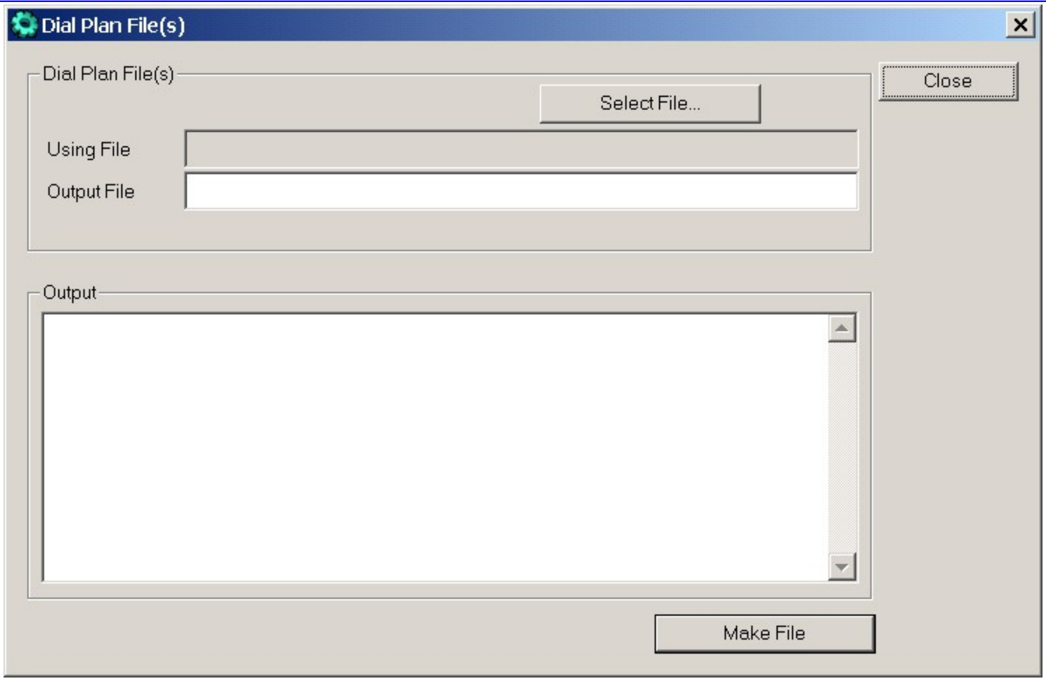
Note: The process verifies the input file for validity. Invalid data will cause an error and abort the process. In this case the log box will contain further information.

13.2.7 Process Dial Plan File(s)

➢ To produce a Dial Plan file, take these 6 steps:

1. Construct a Dial Plan text file according to the instructions in Dial Plan File on page 592.
2. Execute DConvert.exe and click the Process Coder Description button. The Dial Plan window appears.

Figure 13-14: Dial Plan Screen



3. Click the **Select File** button. A Browse window appears.

4. Navigate to the required 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). The output file name may be altered.
5. Click the **Make File** button. The .dat file is generated and placed in the same directory as shown in the **Output File** field. A message box informing you that the operation was successful indicates that the process has been completed.
6. On the bottom of the Coders window, the "Output" log box displays the log generated by the process. It may be copied as needed. This information is NOT retained after the window has been closed.

Note: The process verifies the input file for validity. Invalid data will cause an error and abort the process. In this case the log box will contain further information.

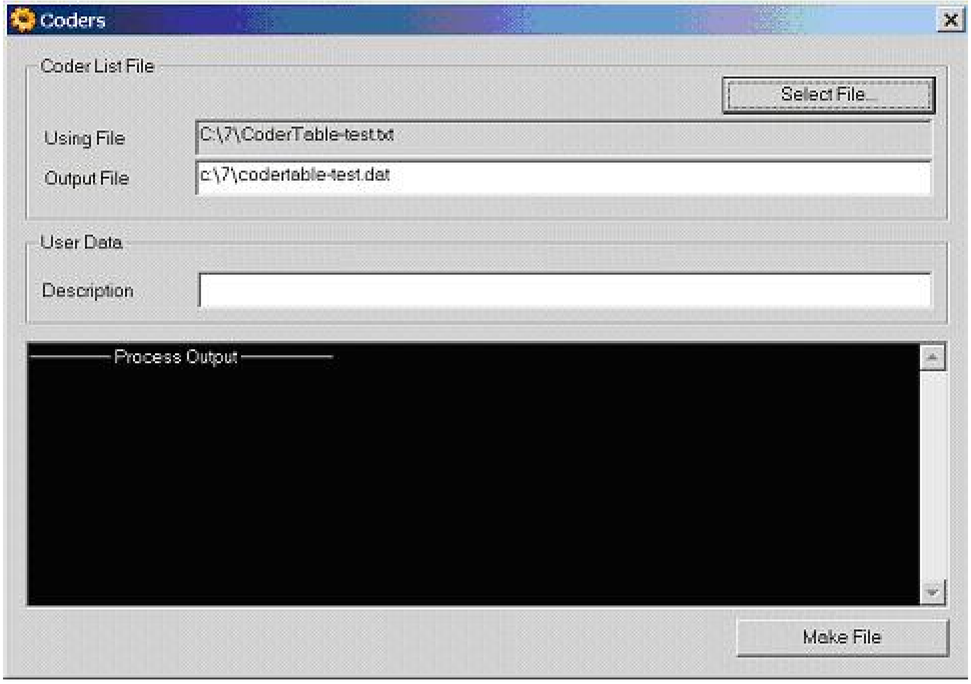
13.2.8 Process Coder Table File(s)

Coder description files are used to define coders properties for the control protocol in use.

➢ To Encode an ini file, take these 6 steps:

1. Create a coder table file. An example is shown in Coder Table File on page 587.
2. Run **DConvert**.
3. Select the 'Process Coder Table file(s)' menu option from the main menu.. The Coders screen appears as shown in the figure below.

Figure 13-15: Process Coder Table Screen



Using File: C:\7\CoderTable-test.txt

Output File: c:\7\codertable-test.dat

4. Click on **Select File** under Coder List File, and select the filename to be decoded. (This automatically designates the output file as the same name and path, but with the dat extension.)
5. An optional description may be added at the “Description” field.

Click the Make File button. The decoded file is generated and placed in the same directory as shown in the Output File field. A message box appears, informing you that the operation was successful and the process has been completed.

> **Note:** This process checks the file for validity and ignores any illegal lines. The output pane displays the error messages.

13.3 PSTN Trace Utilities

> **Note:** This sub-section is NOT applicable to **MediaPack**.

LOCATION:

```
.\Utilities\PSTN Trace Utility
```

DESCRIPTION:

These utilities are designed to convert Wireshark log files containing the PSTN trace to text format. The user does not have to filter the Wireshark log files. The files can contain a variety of network messages. The following converter can extract only the PSTN trace related messages.

OPERATION:

Generating a Trace/audit Text File for ISDN/SS7 Protocols

➢ To generate a readable text file out of the Wireshark log file when using ISDN/SS7 protocols, take these 2 steps:

1. Copy the Wireshark log file to the same directory in which the translation utility Convert_pCap.bat is located. The following files should reside in the same directory:
   - PcapToNBBin.exe
   - CONVERT_TRACE.BAT
   - Dumpview.exe
   - Dumpview.cfg
   - ReadMe.txt.

   Carefully read the ReadMe.txt in order to understand the usage of the translation utility.
2. Run the Convert_pCap.bat. The text file is created.

13.4 Collect and Read the PSTN Trace via Wireshark

Note: This sub-section is NOT applicable to **MediaPack**.

Enabling the PSTN trace is done via the Debug Recording tool (refer to Debug Recording on page 59). The PSTN messages sent by the device can be collected and read using the Wireshark (Network Protocol Analyzer: www.wireshark.org) application. A special plug-in has to be used to facilitate this.

The Wireshark plug-in can be found in .\Utilities\WiresharkPlugins.

13.5 WinDriver Utilities

Note: This sub-section is NOT applicable to **MediaPack**, **Mediant 2000** and **6310/3000** devices.

LOCATION:

```
PCI DRIVER\linux\intel\32BIT
PCI DRIVER\windows\intel\32BIT\util
PCI DRIVER\solaris\intel\32BIT\util
PCI DRIVER\solaris\sparc\32bit\util
PCI DRIVER\solaris\sparc\64bit\util
```

DESCRIPTION:

WinDriver™ is a device driver created by Jungo™. The utilities supplied in these directories are distributed by Jungo™ and are useful in debugging PCI-related problems (such as cases in which the device is not recognized by acInitLib()).

OPERATION:

Contact AudioCodes Support for any debugging problems. When appropriate, one or more of the utilities should be run to enable AudioCodes and/or Jungo to debug the problem.

13.6 Call Progress Tones Wizard (MediaPack Only)

Note : This section is applicable to **MediaPack** only.

This section describes the Call Progress Tones Wizard (CPTWizard), an application designed to help the provisioning of a MediaPack FXO gateway, by recording and analyzing Call Progress Tones generated by any PBX or telephone network.

13.6.1 About this Software

- This wizard helps detect the call progress tones generated by your PBX (or telephone exchange), and creates basic call progress tone *ini* and *dat* files, providing a good starting point when configuring a MediaPack FXO gateway. (The ***ini*** file contains definitions for all relevant call progress tones; the ***dat*** file is suitable for downloading to the gateway. The ***dat*** file is the same file the DConvert would produce when processing the ***ini*** file.)
- To use this wizard, you need a device FXO gateway connected to your PBX with 2 physical phone lines. This gateway should be configured with the factory-default settings, and should not be used for phone calls during operation of the wizard.
- Firmware version 4.2 and above is required on the gateway.

13.6.2 Installation

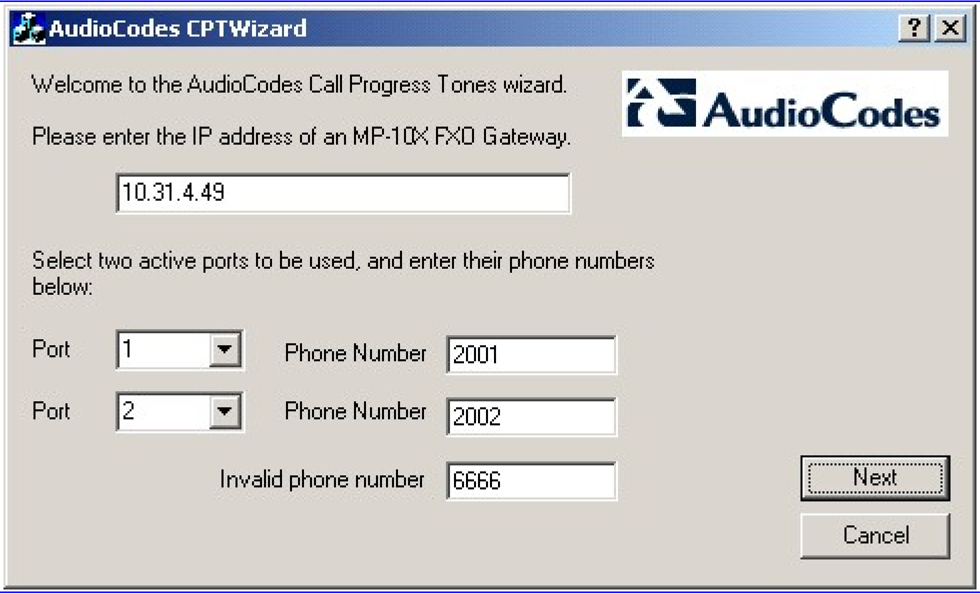
- CPTWizard can be installed on any Windows 2000 or Windows XP based PC. Windows-compliant networking and audio peripherals are required for full functionality.
- To install CPTWizard, copy the files from the installation media to any folder on the PC's hard disk. No further setup is required.
- Approximately 5 MB of hard disk space are required.

13.6.3 Initial Settings

➢ To start CPTWizard take these 4 steps:

1. Double-click on your copy of the CPTWizard.exe program file. The initial settings dialog is displayed:

Figure 13-16: Initial Settings Dialog



AudioCodes CPTWizard

Welcome to the AudioCodes Call Progress Tones wizard.

Please enter the IP address of an MP-10X FXO Gateway.

10.31.4.49

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

Port 1 Phone Number 2001

Port 2 Phone Number 2002

Invalid phone number 6666

2. In the appropriate fields, fill in the gateway's IP address, select which of the gateway's ports are connected to your PBX, and specify the phone number for each extension.

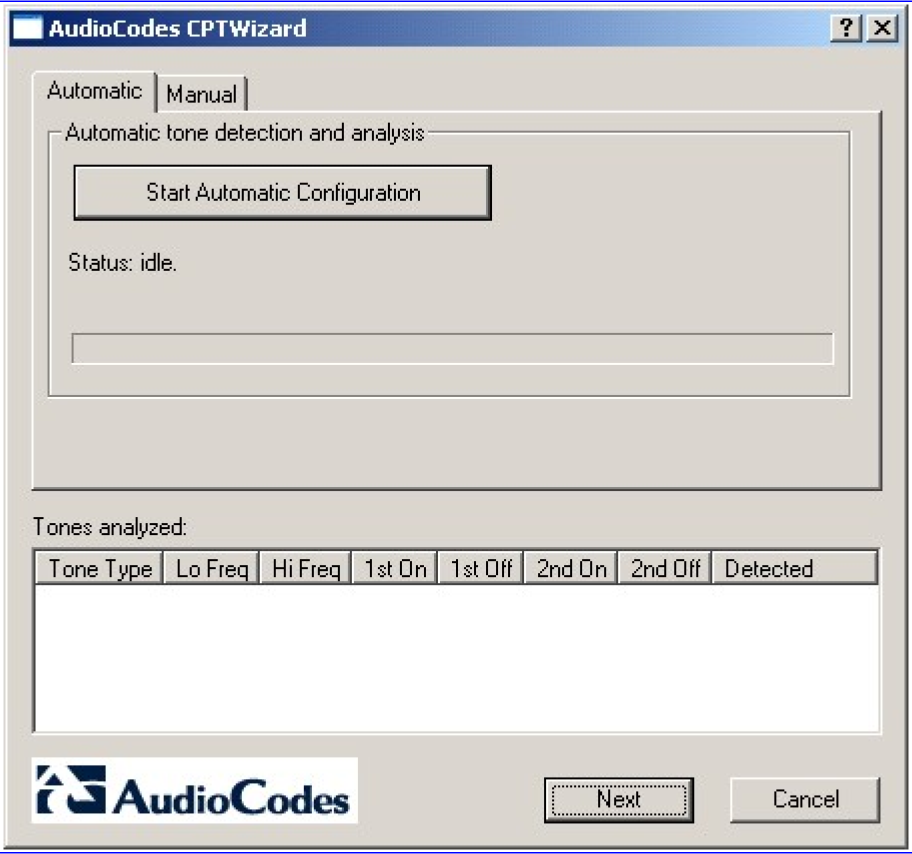
3. In the “Invalid phone number” box, enter a number which generates a “fast busy” tone when dialed. Usually, any incorrect phone number should cause a “fast busy” tone.
4. When the parameters are entered correctly, press NEXT.

Note: CPTWizard connects to the gateway using the TPNCP protocol. If this protocol has been disabled in the gateway configuration, CPTWizard does not display the next dialog and an error is reported.

13.6.4 Recording Dialog – Automatic Mode

Once the connection to the device FXO gateway is established, the recording dialog is displayed:

Figure 13-17: Recording Dialog



Status: idle.

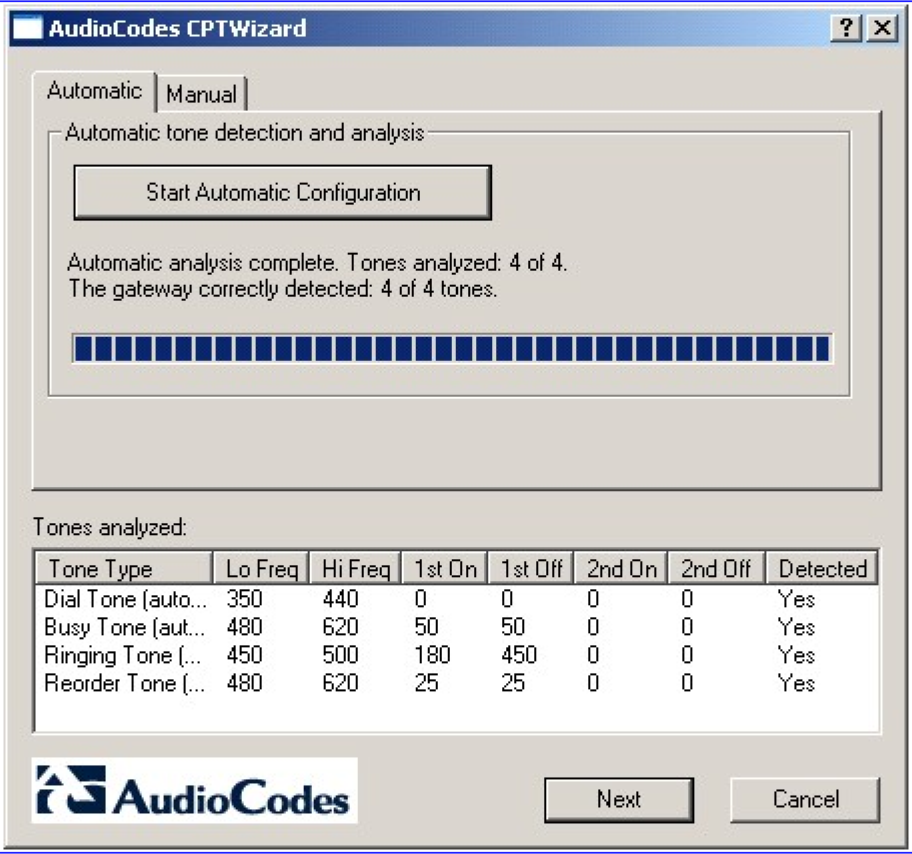
Tone Type	Lo Freq	Hi Freq	1st On	1st Off	2nd On	2nd Off	Detected

➢ **To start Recording Dialog in Automatic Mode take these 5 steps:**

1. To start the detection process, press the “Start Automatic Configuration” button. The wizard will start a call progress tone detection sequence (the operation should be about 60 seconds long), as follows:
   - Set port 1 off-hook, listen to the dial tone
   - Set port 1 and port 2 off-hook, dial port 2’s number, listen to the busy tone
   - Set port 1 off-hook, dial port 2’s number, listen to the ringback tone
   - Set port 1 off-hook, dial an invalid number, listen to the reorder tone

2. The wizard will then analyze the recorded call progress tones, and display a message specifying which tones were detected (by the gateway) and analyzed (by the wizard) correctly. At the end of a successful detection operation, the dialog displays the results shown in the figure below:

Figure 13-18: Recording Dialog after Automatic Detection



AudioCodes CPTWizard
?
Automatic
Manual
Automatic tone detection and analysis
Start Automatic Configuration
Next
Cancel
AudioCodes

Automatic analysis complete. Tones analyzed: 4 of 4.
The gateway correctly detected: 4 of 4 tones.

Tones analyzed:

Tone Type	Lo Freq	Hi Freq	1st On	1st Off	2nd On	2nd Off	Detected
Dial Tone (auto...	350	440	0	0	0	0	Yes
Busy Tone (aut...	480	620	50	50	0	0	Yes
Ringing Tone (...	450	500	180	450	0	0	Yes
Reorder Tone (...	480	620	25	25	0	0	Yes

3. All four call progress tones are saved in the same directory as the CPTWizard.exe file, with the following names:
   - cpt_recorded_dialtone.pcm
   - cpt_recorded_busytone.pcm
   - cpt_recorded_ringtone.pcm
   - cpt_recorded_invalidtone.pcm
4. All files are saved as standard A-law PCM at 8000 bits per sample.

Note 1: If the gateway is configured correctly (with a call progress tones *dat* file downloaded to the gateway), all four call progress tones shall be **detected** by the gateway. By noting whether the gateway detects the tones or not, you can determine how well the call progress tones *dat* file matches your PBX. During the first run of CPTWizard, it is probable that the gateway might not detect any tones.

Note 2: Some tones cannot be detected by the device gateway hardware (such as 3-frequency tones and complex cadences). CPTWizard is therefore limited to detecting only those tones which can be detected on the device gateway.

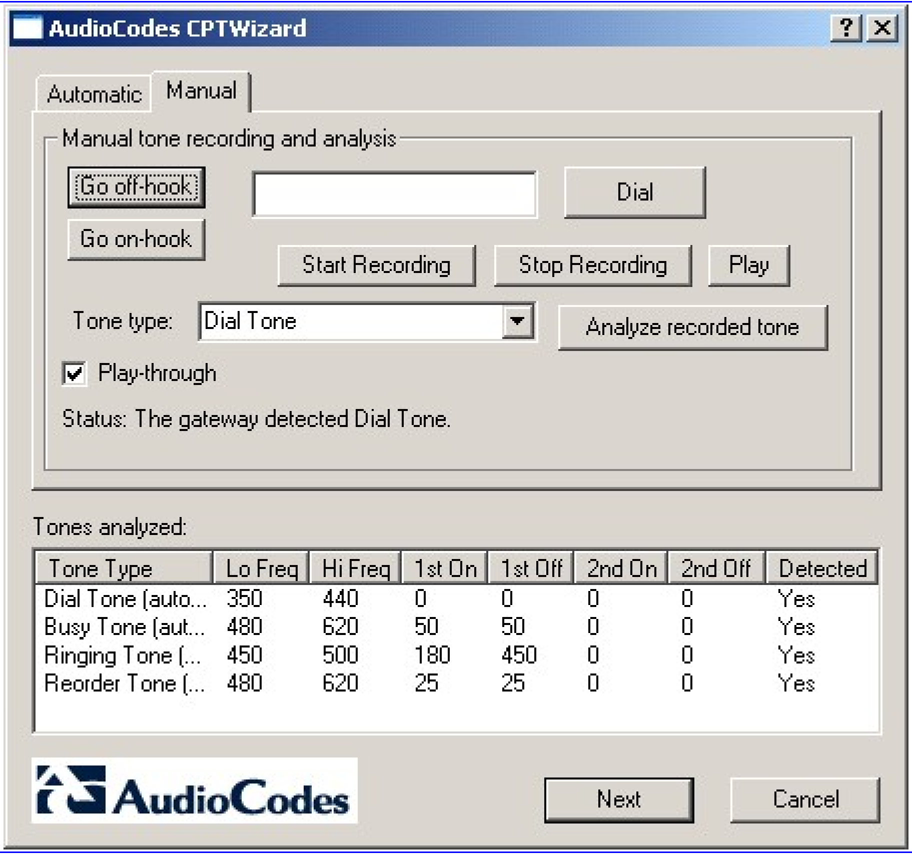
5. At this stage, you can either press NEXT to generate call progress tone *ini* file *dat* files and end the wizard, or continue to manual recording mode.

13.6.5 Recording Dialog – Manual Mode

➢ To start Recording Dialog in Manual Mode take these 6 steps:

1. Choose the “Manual” tab at the top of the recording dialog, it is then possible to record and analyze more tones, which are included in the call progress tone *ini* and *dat* files.

Figure 13-19: Recording Dialog in Manual Mode



Tone type: Dial Tone

☑ Play-through

Status: The gateway detected Dial Tone.

Tones analyzed:

Tone Type	Lo Freq	Hi Freq	1st On	1st Off	2nd On	2nd Off	Detected
Dial Tone (auto...	350	440	0	0	0	0	Yes
Busy Tone (aut...	480	620	50	50	0	0	Yes
Ringing Tone (...	450	500	180	450	0	0	Yes
Reorder Tone (...	480	620	25	25	0	0	Yes

2. For easy operation, use the play-through check box to hear the tones through your PC speakers.
3. Press the “Set off-hook” button, enter a number to dial in the Dial box, and press the Dial button. When you’re ready to record, press the “Start Recording” button; when the required tone is complete press “Stop Recording”. (The recorded tone will be saved as “cpt_manual_tone.pcm”.)

Note: Due to some PC audio hardware limitations, you may hear “clicks” in play-through mode. It is safe to ignore these clicks.

4. Select the tone type from the drop-down list, and press “Analyze”. The analyzed tone is added to the list at the bottom of the dialog. It is possible to record and analyze several different tones for the same tone type (e.g., different types of “busy” signal).

5. Repeat the process for more tones, as necessary.
6. When you're done adding tones to the list, click **Next** to generate a call progress tone *ini* file and end the wizard.

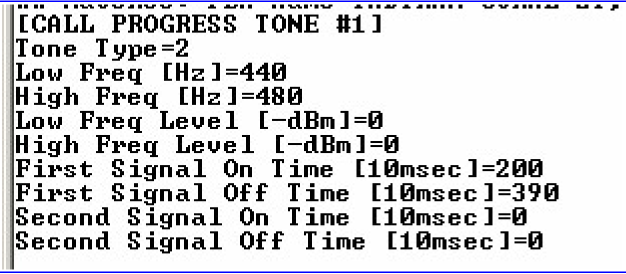
13.6.6 The Call Progress Tone ini and dat Files

Once the wizard completes the call progress tone detection, the call_progress_tones.ini text file and the call_progress_tones.dat binary file are created in the same directory as CPTWizard.exe. The call_progress_tones.dat binary file is now ready for download to the media gateway, and it contains the same output which the DConvert utility would produce when processing the ini file.

The ini file contains:

- Information about each tone recorded and analyzed by the wizard. This includes frequencies and cadence (on/off) times, and is required when converting the *ini* file to *dat.*

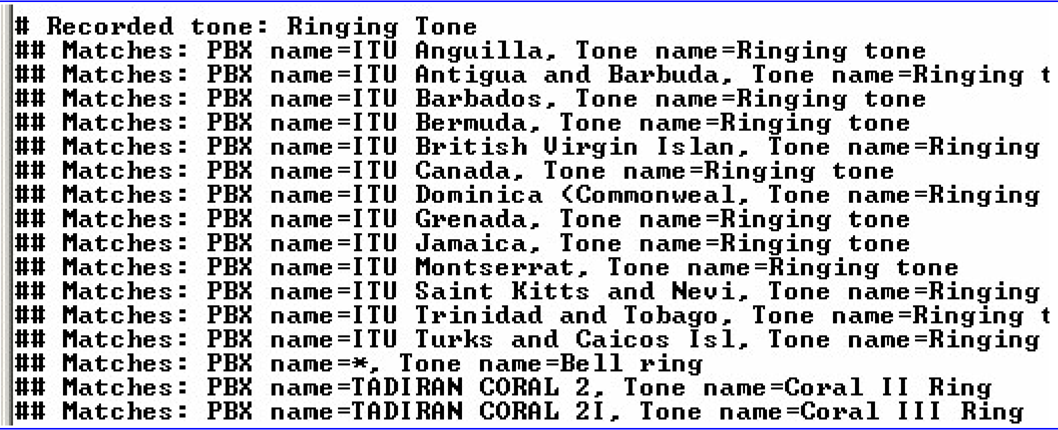
Figure 13-20: Call Progress Tone Properties



```
[CALL PROGRESS TONE #1]
Tone Type=2
Low Freq [Hz]=440
High Freq [Hz]=480
Low Freq Level [-dBm]=0
High Freq Level [-dBm]=0
First Signal On Time [10msec]=200
First Signal Off Time [10msec]=390
Second Signal On Time [10msec]=0
Second Signal Off Time [10msec]=0
```

- Information related to possible matches of each tone with the CPTWizard internal database of well-known tones. This information is specified as comments in the file, and is ignored when converting the *ini* file to *dat.*

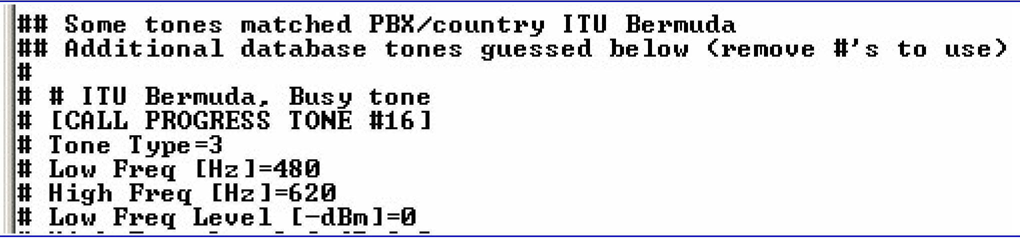
Figure 13-21: Call Progress Tone Database Matches



```
# Recorded tone: Ringing Tone
## Matches: PBX name=ITU Anguilla, Tone name=Ringing tone
## Matches: PBX name=ITU Antigua and Barbuda, Tone name=Ringing t
## Matches: PBX name=ITU Barbados, Tone name=Ringing tone
## Matches: PBX name=ITU Bermuda, Tone name=Ringing tone
## Matches: PBX name=ITU British Virgin Islan, Tone name=Ringing
## Matches: PBX name=ITU Canada, Tone name=Ringing tone
## Matches: PBX name=ITU Dominica (Commonweal, Tone name=Ringing
## Matches: PBX name=ITU Grenada, Tone name=Ringing tone
## Matches: PBX name=ITU Jamaica, Tone name=Ringing tone
## Matches: PBX name=ITU Montserrat, Tone name=Ringing tone
## Matches: PBX name=ITU Saint Kitts and Nevi, Tone name=Ringing
## Matches: PBX name=ITU Trinidad and Tobago, Tone name=Ringing t
## Matches: PBX name=ITU Turks and Caicos Isl, Tone name=Ringing
## Matches: PBX name=*, Tone name=Bell ring
## Matches: PBX name=TADIRAN CORAL 2, Tone name=Coral II Ring
## Matches: PBX name=TADIRAN CORAL 2I, Tone name=Coral III Ring
```

- Information related to matches of all tones recorded with the CPTWizard internal database. The database is scanned to find one or more PBX definitions which match all recorded tones (i.e. both dial tone, busy tone, ringing tone, reorder tone and any other manually-recorded tone – all match the definitions of the PBX). If a match is found, the entire PBX definition is reported in the *ini* file using the same format.

Figure 13-22: Full PBX/Country Database Match



```
## Some tones matched PBX/country ITU Bermuda
## Additional database tones guessed below (remove #'s to use)
#
# # ITU Bermuda, Busy tone
# [CALL PROGRESS TONE #16]
# Tone Type=3
# Low Freq [Hz]=480
# High Freq [Hz]=620
# Low Freq Level [-dBm]=0
```

- If a match is found with the database, consider using the database definitions instead of the recorded definitions, as they might be more accurate.
- For full operability of the MP-11X FXO gateway, it may be necessary to edit this file and add more call progress tone definitions. Sample call progress tone *ini* files are available in the release package.
- When the call progress tones *ini* is complete, the corresponding *dat* file is ready for downloading. After loading this file to the gateway, repeat the automatic detection phase discussed above, and verify that the gateway detects all four call progress tones correctly.
- Manually changing the *ini* file causes the *dat* file to be outdated. It needs to be re-generated according to the new *ini file*. A *dat* file may be re-generated by pressing the "Regenerate" button at the final dialog or by using the DConvert utility.

14 List of Abbreviations

Table 14-1: List of Abbreviations

Abbreviation	Meaning
AAL1	ATM Adaptation Layer 1 – Used in North America for voice traffic. It provides support for constant bit rate (voice) traffic
AAL2	ATM Adaptation Layer 2 – Used to transmit standard and compressed voice transmissions including silence suppression. It can support both constant and variable bit rates.
ADPCM	Adaptive Differential PCM - voice compression
AIS	Alarm Indication Signal
ASN.1	Abstract Syntax Notation
ATM	Asynchronous Transmission Mode – A connection based transport mechanism that is based on 53 byte cells
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)
CLIP	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

Table 14-1: List of Abbreviations

Abbreviation	Meaning
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
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 blades
IPM-260/UNI	AudioCodes IPmedia PCI VoIP Media Processing blade, to 240 ports
IPM-1610	AudioCodes IPmedia cPCI VoIP Media Processing blade, to 240 ports
IPM-6310	AudioCodes IPmedia VoIP Media Processing blade, to 2016 voice/fax/data independent multiple LBR channels
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)

Table 14-1: List of Abbreviations

Abbreviation	Meaning
MGC	Media Gateway Controller
MGCP	Media Gateway Control Protocol
MIB	Management Information Base
MP-112	AudioCodes 2-port Analog MediaPack Media Gateway
MP-114	AudioCodes 4-port Analog MediaPack Media Gateway
MP-118	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
NIC	Network Interface Card
OSI	Open Systems Interconnection (Industry Standard)
PCI	Personal Computer Interface (Industry Standard)
PCM	Pulse Code Modulation
PDU	Protocol Data Unit
POTS	Plain Old Telephone System or Service
PRI	Primary Rate Interface in ISDN
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RAI	Remote Alarm Indication
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 blade, 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

Table 14-1: List of Abbreviations

Abbreviation	Meaning
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.
TGCP	Trunking Gateway Control Protocol
TPNCP	AudioCodes TrunkPack Network Control Protocol.
TP-260/UNI	AudioCodes TrunkPack VoIP/260 Voice over IP PCI media streaming blade, up to 240 ports
TP-1610	AudioCodes TrunkPack VoIP cPCI media streaming blade, to 480 ports
TP-6310	AudioCodes TrunkPack VoIP Media Processing blade, to 2016 voice/fax/data independent multiple LBR channels
TPM-1100	AudioCodes TrunkPack Module
TrunkPack	AudioCodes series of voice compression blades
T1	1.544 Mbps USA Digital Transmission System (see E1 and DS1)
T3	44.736 Mbps USA Digital Transmission System, Encapsulates 28 T1 streams, also called DS3
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

15 Index

Product Reference Manual

AudioCodes

www.audiocodes.com