Mediant™ Media Gateways

MediaPack™ Media Gateways

IPmedia™ Media Servers

Product Reference Manual

MGCP & MEGACO Protocols

HD VoIP Sounds Better

AudioCodes

Version 6.6

December 2015

Document # LTRT-77009

Table of Contents

This page is intentionally left blank.

Notice

This Product Reference Manual provides an extremely comprehensive description of MGCP and MEGACO Network Control Protocols and their compliance.

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

Date Published: December 6, 2015

Trademarks

AudioCodes, AC, HD VoIP, HD VoIP Sounds Better, IPmedia, Mediant, MediaPack, What's Inside Matters, OSN, SmartTAP, VMAS, VoIPerfect, VoIPerfectHD, Your Gateway To VoIP, 3GX, VocaNOM and CloudBond 365 are trademarks or registered trademarks of AudioCodes Limited All other products or trademarks are property of their respective owners. Product specifications are subject to change without notice.

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 services are provided by AudioCodes or by an authorized AudioCodes Service Partner. For more information on how to buy technical support for AudioCodes products and for contact information, please visit our Web site at www.audiocodes.com/support.

Related Documentation

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

- **MGCP MEGACO Product Reference Manual** (this manual) - provides an extremely comprehensive description of MGCP and MEGACO Network Control Protocols and their compliance.
- **User's Manual** contains the Product overview; software package, startup and initialization; Web GUI-based management; Diagnostics and Product Specification.
- **MEGACO Release Notes** - 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.

Document Revision Record

LTRT	Description
77008	Initial document release for Version 6.6.
77009	Updated Note in Section 5.5 – SS7 is not applicable to Mediant 3000.

Documentation Feedback

AudioCodes continually strives to produce high quality documentation. If you have any comments (suggestions or errors) regarding this document, please fill out the mentation Feedback form on our Web site at http://www.audiocodes.com/downloads

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

Note: For information on which devices are supported in this software version and how to fully configure any device, please refer to the device's User's Manual and Release Notes.

This page is intentionally left blank.

2 System Initialization Process

This section describes the Initialization Procedures and Configuration Options for the Mediant 3000 System. It includes:

- Startup Process (see below)
- Configuration Parameters and Files (refer to 'Configuration Parameters and Files' on page 21)
- BootP/DHCP (refer to 'Using BootP/DHCP' on page 31)
- Software Upgrade
- High Availability Aspects

2.1 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 31).
- Web Interface (refer to Configuration Using the Web Interface).
- Automatic Update facility (refer to 'Automatic Update Facility' on page 26).

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.

Notes:

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

2.1.1 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'). The lists of parameters are provided in the *ini* File Parameters chapter of the Product Reference Manual.
- **Table** (refer to 'Table Structure' on page 24). The lists of parameters are provided in Table Parameters.

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]
```

2.1.1.1 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.
- A carriage-return/line-feed 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'
```

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 'ini File Parameters' on page 164.

2.1.1.2 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_INT_ID_TABLE 7 =  50, BELFAST12, 7, 2, 0;
SS7_SIG_INT_ID_TABLE 8 =  51, AMSTERDAM, 7, 2, 1;
```

```
[ \SS7_SIG_INT_ID_TABLE ]
```

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

Table Structure Example

IF ID Index	IF ID Value	SS7_SIG_IF_ID _NAME	SS7_SIG_IF_ID_ OWNER_GROUP	SS7_SIG_IF_ID _LAYER	SS7_SIG_IF_I D_NAI
7	50	BELFAST12	7	2	0
8	51	AMSTERDAM	7	2	1

2.1.1.2.1 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 the 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-page, 127979, 9;
Items_Table 2 = Computer-pad, 111111, $$;
[\Items_Table]
```

2.1.1.2.2 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.

Note: When obtaining the ini file for the Mediant 1000, Mediant 600 and Mediant 800, the current running hardware entities are added to the ini file header. The user may use this information in order to help analyze potential faulty situations.

2.1.2 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 34), 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.

Note:

For the following parameters, the URLs are reset to their default value on successful Autoupdate. Subsequent Autoupdates without re-initializing the parameters are not supported.

- CptFileUrl
- PrtFileUrl
- FXSCoeffFileUrl
- FXOCoeffFileUrl
- CasFileUrl
- DialPlanFileUrl
- TLSPkeyFileUrl
- TLSCertFileUrl
- TLSRootFileUrl
- WebLogoFileUrl
- V5PortConfigurationFileURL

➢ **To utilize Automatic Updates for deploying the device with minimum manual configuration:**

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'
```

If you implement the Automatic Update mechanism, the device must not be configured using the Web interface. If you configure parameters in the Web interface and save (burn) the new settings to the device's flash memory, the *IniFileURL* parameter (defining the URL to the *ini* file for Automatic Updates) is automatically set to 0 (i.e., Automatic Updates is disabled).

The Web interface provides a safeguard for the Automatic Update mechanism. If the IniFileURL parameter is defined with a URL value (i.e., Automatic Updates is enabled), then by default, the 'Burn To FLASH' field under the Reset Configuration group in the Web interface's 'Maintenance Actions' page is automatically set to "No". Therefore, this prevents an unintended burn-to-flash when resetting the device.

However, if configuration settings in the Web Interface were burnt to flash, you can re-instate the Automatic Update mechanism, by loading to the device, the *ini* file that includes the correct IniFileURL parameter setting, using the Web interface or BootP.

2.2 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.

  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.

2.3 Using BootP/DHCP

Notes:

- **This sub-section is not applicable to Mediant 3000 HA.**
- 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.
- The default time duration between BootP/DHCP requests is set to 1 second. This can be changed by the BootPDelay *ini* file parameter. Also, the default number of requests is 3 and can be changed by the *BootPRetries ini* file parameter. Both parameters can also be set using the Command Line Switches in the BootP reply packet.
- 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.

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.

Both DHCP and BootP are network protocols that enable a device to discover its assigned IP address; DHCP differs from BootP in that it provides a time-limited "lease" to the assigned 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 private (local) IP address of the device.

2.3.1 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. Note that in High Availability (HA) mode, this IP address is only private (local) and is not the HA System (global) IP address that must be configured separately through the Interface Table.
- **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.

2.3.1.1 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:**

1. In the **Boot File** field, leave the filename defined in the field as it is (e.g., *ramxxx.cmp*).
2. After "*cmp",* leave a space and 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.

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.
-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.

Command Line Switch Descriptions

-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 34.
-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.

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 21. 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.

2.3.2 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. On networks which support this setting, 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).

2.3.3 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**.

2.3.4 Secure Startup

The TFTP protocol is not considered secure; some network operators block it using firewalls. If loading configuration from flash memory is not desired, the device may be configured to retrieve the configuration upon each start-up from a remote server, using a secure protocol such as HTTPS.

➢ **To work with HTTPS instead of TFTP:**

- 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 37 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 26.

2.3.5 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:

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	XXXXXXXXXXX	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.

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

This page is intentionally left blank.

3 Management Functions

Two types of Management are detailed in this section:

- Command-line Interface - refer to 'Command-line Interface' on page 37
- SNMP - refer to 'Using SNMP-based Management' on page 63

3.1 CLI-Based Management

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:**

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
```

Notes:

- The username and password are case-sensitive.
- The CLI username and password can be altered by the media gateway administrator. Multiple users can be defined.
- 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.

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.
	exit	Terminates the CLI session

CLI Commands and their Options

Purpose	Commands	Description
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.

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.
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

General Commands

Command	Short Format	Arguments	Description
			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:

Sub-commands of command 'SHow MGCP' / 'SHow MEGACO'

Sub-command	Description
conf	Displays the overall configuration of MGCP/MEGACO, including: ▪ MGCP/MEGACO version string ▪ Configured call-agent data ▪ Endpoint-name / Termination-name pattern used by the call agent ▪ Various feature flags ▪ List of supported packages
perf	Displays performance statistics, including: ▪ Current/Total number of voice calls ▪ Average call length (calculated in 15-minute intervals) ▪ Number of messages sent/received from the call agent ▪ Number of successful/failed commands, per command type ▪ 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.

Sub-commands of command 'SHow MGCP' / 'SHow MEGACO'

calls	Displays a list of all active calls, with the following information per each call: ▪ Endpoint/Termination name (Trunk/B-Channel) ▪ Mode (recvonly / sendonly / sendrecv) ▪ DSP device used for the call ▪ Call duration in seconds For MGCP: ▪ Call ID (value of "C:" parameter specified by the call agent) ▪ Connection ID (value of "I:" parameter selected by the device) ▪ RTP port numbers For MEGACO: ▪ Context ID ▪ Call ID (Internal Channel Handler used for the call) ▪ Local Address ▪ Remote Address ▪ CallType - The call type of the call (T2I – TDM to IP, I2I – IP to IP, T2T – TDM to TDM, Conf - Conference). ▪ 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. **Note for MEGACO:** For non-TDM to IP calls, (context with PSTN and IP terminations) the CallID, DSP device, Mode, Local Address, Remote Address and Media Type fields are not valid. It contains information of the first selected RTP termination only.
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: ▪ Trunk and B-channel and internal channel ID used for call ▪ Call duration in seconds and call start time ▪ RTP information – on/off, vocoder used, Source and Destination IP addresses and ports. ▪ DSP information – which device is used, echo cancelation length, etc. ▪ RTCP information – number of bytes Tx/Rx, quality (jitter/delay), etc. This command also has an optional <TimeSet> parameter. The value of <TimeSet> should be positive value. When this parameter is used, the output will be displayed after <timeset> seconds and will include RTCP information relevant to <timeset> interval only. **Notes and Constraints**: ▪ Two "show mgcp/megaco detail" commands on the same endpoint/termination are not supported. Error on second command is generated. ▪ In case the command is canceled, no statistics will be reported. Only a "Canceled" string will be printed. ▪ The deletion of the endpoint/termination in the middle of the "show

Sub-commands of command 'SHow MGCP' / 'SHow MEGACO'

	mgcp/megaco detail" command will report an error to the CLI. ▪ The "show megaco detail" with the TimeSet option is not supported on context with physical termination only.
dur	Displays call duration averages for the past 48 hours. For every hour, the following items are displayed: ▪ Number of calls completed during that hour. ▪ Average call duration for completed calls (seconds).
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: ▪ A counter per each type of failed response to a CRCX command (e.g., a 501 response counter, a 502 response counter …). ▪ A counter per each type of failed response to a MDCX command (e.g., a 501 response counter, a 502 response counter …). ▪ 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: ▪ Number of call attempts made in the last second. (This value is updated every second). ▪ 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.

MGCP 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 (Internal ChannelId - 85)
```

```
DSP device:              48
RTP:                     On sendonly
Coder:                   PCMU, packetization 20 ms
Echo Canceller:           On, length 128 ms
Silence Compression:     Off
High Pass Filter:        On
DTMF Detection:          On
Voice Volume:            0 dB
Input Gain:              0 dB
Jitter buffer length:    10 ms
DTMF Transport Type:     3-acRFC2833RalayDTMF
Call Duration:           53 seconds
Local RTP address:       10.4.4.34 port4840
Remote RTP address:      10.4.64.13 port 4840
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
```

MEGACO Example

```
/>sh megaco


Usage:
   SHow MEGACO CONF             Displays MEGACO configuration
   SHow MEGACO PERF             Displays MEGACO performance
statistics
   SHow MEGACO NER              Displays MEGACO network efficiency
rate
   SHow MEGACO CALLS  [Trunk#]  Displays currently active Contexts
   SHow MEGACO DETAIL <CID=callID|TerminationName >
               Displays detailed data for the specified call
   SHow MEGACO DETAIL <CID=callID|TerminationName> <TimeSet>
              Displays detailed data for the specified call after
<TimeSet> time and with statistics relevant to this period of time
only
   SHow MEGACO DUR              Displays history of call duration
averages
   SHow MEGACO ERR              Displays Megaco statistics for
Failure responses
   SHow MEGACO CS               Displays Megaco statistics for
Calls per second


/>sh megaco nerf
Network Efficiency Rate:
      Add    success/total = 6/6 = 100%
      Modify success/total = 0/0 = 100%
```

```
/>sh megaco calls


Termination Context(CON) CallID(CID) Time(T) LocalAddress(L)
RemoteAddress(R)
            Mode(M) DSP(D) CallType(CT) MediaType(MT)
gws0c1 CON=1 CID=0 T=92 L=10.4.4.35 port 4000 R=10.4.4.35 port
4000 M=sendrecv D
=0 CT=T2I MT=Voice
gws0c2 CON=2 CID=1 T=80 L=10.4.4.35 port 4010 R=10.4.4.35 port
4010 M=sendrecv D
=1 CT=T2I MT=Voice
gws0c3 CON=3 CID=2 T=70 L=10.4.4.35 port 4020 R=10.4.4.35 port
4020 M=sendrecv D
=2 CT=T2I MT=Voice


/>sh megaco detail gws0c1


Context:                1 (associated terminations:  gws0c1
gwrtp/0)
Trunk/BChannel:         0/1 (Internal ChannelId - 0)
DSP device:             0
RTP:                    On sendrecv
Coder:                  PCMU, packetization 20 ms
Echo Canceller:         On, length 128 ms
Silence Compression:    Off
High Pass Filter:       On
DTMF Detection:         On
Voice Volume:           0 dB
```

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:

Subcommands of Call Detail Reports (CDR) Command

Subcommand	Description
start [syslog | file | both]	Starts generating CDR records. If 'syslog' is specified, the records are sent to the Syslog. If 'file' is specified, the records are collected in a file 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

Subcommands of Call Detail Reports (CDR) Command

Subcommand	Description
	(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:

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:

Management Commands

Command	Short Format	Arguments	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.

PSTN Commands

Command	Short Format	Arguments	Description
DeleteCasFile	PS/CAS/DCF	<TableIndex>	When setting <TableIndex> to the value of "-1", delete all CAS files on the blade. (Other options are not supported.)
PstnLoopCommands	PS/PH/PLC	<TrunkId> <LoopCode> <BChannel>	Activates a loopback on a specific trunk and BChannel. For loop on all trunks, set BChannel = (-1). LoopCode: 0 - NO_LOOPS 1 - REMOTE_LOOP (whole trunks only) Not available for BRI trunks. 2 - LINE_PAYLOAD_LOOP (whole trunks only). Not available for BRI, DS3 or Sonet/SDH trunks 3 - LOCAL_ALL_CHANNELS_LOOP (whole trunks only). Not available for DS3 or Sonet/SDH trunks.
PstnSendAlarm	PS/PH/PSA	<TrunkId> <AlarmSendCode>	Sends an alarm signal at the Tx interface or specific TrunkId. AlarmSendCode: 0 - NO_ALARMS (means stop sending AIS) 1 - AIS_ALARM 2 - STOP_RAI_ALARM 3 - SEND_RAI_ALARM

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
Trunk 22 is sending AIS alarm.


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:**

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:**

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):**

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.

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).

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 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.

Trace Rules

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. 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. RemoteIPAddr - enables to capture the number (according to the 'NumOfCalls' parameter) of next call, but with the special condition that these next calls should use ONLY the specific Remote IP Address. For example: 'AddNextCallTrace All 10 Dynamic 10.31.2.85' In this example, we will record the next 10 dynamic RTP calls that will activate the RTP to specific remote IP address (in this case 10.31.2.85 address).
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.

Trace Rules

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 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.

DR Activation

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

3.1.5 Changing the Network Parameters via CLI

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

3.1.5.1 Accessing the CLI

➢ **To access the CLI via the RS-232 port:**

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., HyperTerminal) 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.

3.1.5.1.1 Assigning an IP Address

➢ **To assign an IP address via the CLI:**

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 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 (acGateway, AcAlarm, acMedia, acControl, acAnalog 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 three 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).

3.2.1.2.1 Host Resource MIB (RFC 2790)

Host Resource MIB (RFC 2790) is used for managing host systems. The term host is any computer that communicates with other similar computers connected to the Internet and that is directly used by one or more human beings.

The following Host Resources MIB objects have been added:

- hrSystem group
- hrStorage group (basic only)
- hrDevice group (CPU, RAM, Flash - basic only)
- hrSWRunPerf (basic only)
- hrSWInstalled (OS only)

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.

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

The 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 File Management

The SNMP Interface offers downloading, uploading and removing of files.

3.2.4.1 Downloading a File to the Device

The URL file is set in the appropriate MIB object under the *acSysHTTPClient* subtree. Refer to the subtree object description for the URL form. The download can be scheduled using the *acSysHTTPClientAutoUpdatePredefinedTime* and *acSysHTTPClientAutoUpdateFrequency* objects.

It can also be a manual process using *acSysActionSetAutoUpdate*. In this case only and as long as one URL is SET at a time, the result of the action can be viewed in *acSysActionSetAutoUpdateActionResult*.

For both cases a *acHTTPDownloadResult* trap will also be sent indicating the success or failure of the process.

acSysActionSetActionId can be SET to any value chosen and can be used to indicate that a certain manager has performed the action.

A successful process also ends with the file name in the appropriate object under the *acSysFile* subtree, in the *acCASFileTable* or the *acAuxiliaryFiles* subtree, along with the URL being erased from the object under the *acSysHTTPClient* subtree.

A new SNMP object, acSysActionSetApplyINImethodthat, has been added to enable the EMS to perform VoIP ini file downloads without parsing (similar to the wizard mode). This object is located under the system MIB.

Notes:

- The action result (both in the *acSysActionSetAutoUpdateActionResult* object and *acHTTPDownloadResult* trap) for the Voice Prompt and XML indicates only that the file reached the device and has no indication on the application's ability to parse the file.
- The action result in *acSysActionSetAutoUpdateActionResult* is reliable as long as only one file is downloaded at a time.

3.2.4.2 Uploading and Removing a File

File upload is the procedure of sending a file from the device to the manager. Removing a file, is erasing it from the blade, an offline action that requires a reset for it to be applied. The *acSysUpload* subtree holds all relevant objects.

acSysUploadFileURI indicates the file name and location along with the file transfer protocol (HTTP, NFS). For example – "http:\\server\filename.txt".

acSysUploadFileType and *acSysUploadFileNumber* are used to determine which file is going to be uploaded along with its instance when relevant (for CAS or Video Font).

acSysUploadActionID is at the disposal of the manager and can be used to indicate that a certain manager has performed the action.

acSysUploadActionType determines the action that will take place and triggers it off simultaneously.

Note: File removal is supported for all files, other than iniFile(1), v5PortFile(16) and CAS files. v5PortFile removal is done via acV5PortActionType.

3.2.5 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:

- **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.
- **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.

Performance Monitoring MIBs have a fixed format:

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.

3.2.5.1 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.

3.2.5.2 Reporting Congestion in Performance Monitoring

Note: This sub-section on Reporting Congestion in Performance Monitoring is only applicable to **6310** devices.

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
- Extension Resources (used for Conference 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*
- *acPMCongestionConferenceResourcesTable*

3.2.5.3 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.
- System MIB
- Entity MIB
- IF MIB
- **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.
- **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.

Note: RTCP-XR is NOT supported on 3000/6310/8410 devices.

- **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
    - sonetPathCurrentTable
    - sonetPathIntervalTable

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

- Traps (all defined in the AcBoard MIB):
  - acSonetSectionLOFAlarm
  - acSonetSectionLOSAlarm
  - acSonetLineAISAlarm
  - acSonetLineRDIAlarm
  - acSonetIfHwFailureAlarm
  - acSonetPathSTSLOPAlarm
  - acSonetPathSTSAISAlarm
  - acSonetPathSTSRDIAlarm
  - acSonetPathUnequippedAlarm
  - acSonetPathSignalLabelMismatchAlarm

(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 **8410** devices.

- **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)

There is also support for channelized DS3 to OC3 mapping, so that actual interface will be OC3, but it will bring 3 DS3 interfaces into the blade.

In contrast to the DS3 coaxial interfaces, DS3 to OC3 mapping is SONET-based and hence is APS-protected (for more information please refer to 'PSTN' on page 301.

- **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.

The AcBoard MIB includes **acTrap**

Note: The AcBoard MIB is being **phased out**.

- **AcAnalog MIB** (Applicable to **MediaPack and Mediant 1000**)
- **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.
  - acMediaProcessOverloadAlarm - OID:1.3.6.1.4.1.5003.9.10.1.21.2.0.81 that is sent upon overload of the device's media processing and interfaces. (Applicable to Mediant 3000)
- **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.
  - acSysInterfaceStatusTable- supports the Networking multiple interfaces feature status. This table reflects all the active system's interfaces. The line indices consist of both the Entry Index and the Type Index.

    The table contains the following columns:
    - Entry Index - Related Interface index in the interface configuration table (in case the table is empty - meaning there is only single IP the index will appear with 0).
    - Type Index - 1 for IP Address and 2 for IPv6 Link-Local Address.
    - Application Types - The type assigned to the interface.
    - Status Mode - Interface configuration mode.
    - IP Address - IP Address (can be either IPv4 or IPv6) for this interface.
    - Prefix Length - The number of '1' bits in this interface's net mask.
    - Gateway - Default Gateway.
    - Vlan ID - VLAN ID of this interface.
    - Name - Interface's name.
  - 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.
- **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.

Notes: 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.
- 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).

Full proprietary trap definitions and trap Varbinds are found in AcBoard MIB and AcAlarm MIB. For a detailed inventory of traps, refer to 'SNMP Traps' on page 108.

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.
- **acNTPServerStatusAlarm** - This is raised when the connection to the NTP server is lost. It is cleared when the connection has been re-established. Unset time (as a result of no connection to NTP server) may result with functionality degradation and failure in the device.

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 HA systems only - the HA system is faulty and therefore there is no HA.
- **acHASystemConfigMismatchAlarm** – for HA systems only - configuration to the modules in the HA system us uneven causing instability.

- **acHASystemSwitchOverAlarm** – for HA systems only - a switch over from the active to the redundant module has occurred.
- **acSWUpgradeAlarm** - Raised for SW upgrade process errors.

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 IN-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 IN-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.
- **acSS7RedundancyAlarm** – This alarm is used only in case of shared-point-codes configuration. It is raised if the x-link between boards becomes OUT-OF-SERVICE. The alarm is cleared when the x-link between boards becomes IN-SERVICE.
- **acSS7AliasPcStateChangeAlarm** - This alarm is raised if the operational state of the SS7 alias point code becomes BUSY. The alarm is cleared when the operational state of the node becomes IN-SERVICE or OFFLINE.
- **acSS7UalGroupStateChangeAlarm** - This alarm is raised if the ASP group state is not ACTIVE. The alarm is cleared when the state of the ASP group becomes ACTIVE.

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 NOT applicable 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

Note: The following DS3 traps are only applicable to **6310/3000 devices**.

- acDS3RAIAlarm - DS3 RAI alarm
- acDS3AISAlarm - DS3 AIS alarm
- acDS3LOFAlarm - DS3 LOF alarm
- acDS3LOSAlarm - DS3 LOS 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**)

3.2.6 Performance Monitoring Parameters

The device's SNMP MIB PM parameters are listed in the subsequent subsections.

3.2.6.1 DS3 Performance Monitoring

Note: These PM parameters are applicable only to Mediant 3000 with the TP-6310 blade.

DS3 Performance Monitoring

MIB Name	Gauge/ Counter	Description
dsx3IntervalPESs	Gauge	The counter associated with the number of P-bit Errored Seconds. EMS Parameter Name: DS3 PESs
dsx3IntervalPSESs	Gauge	The counter associated with the number of P-bit Severely Errored Seconds. EMS Parameter Name: DS3 PSESs
dsx3IntervalUASs	Gauge	The counter associated with the number of Unavailable Seconds. This object may decrease if the occurrence of unavailable seconds occurs across an interval boundary. EMS Parameter Name: DS3 UASs
dsx3IntervalLCVs	Gauge	The counter associated with the number of Line Coding Violations. EMS Parameter Name: DS3 LCVs
DS3 PCVs	Gauge	The counter associated with the number of P-bit Coding Violations. EMS Parameter Name: dsx3IntervalPCVs
dsx3IntervalLESs	Gauge	The number of Line Errored Seconds (BPVs or illegal zero sequences). EMS Parameter Name: DS3 LESs
dsx3IntervalCCVs	Gauge	The number of C-bit Coding Violations. EMS Parameter Name: DS3 CCVs
dsx3IntervalCESs	Gauge	The number of C-bit Errored Seconds. EMS Parameter Name: DS3 CESs
dsx3IntervalCSESs	Gauge	The number of C-bit Severely Errored Seconds. EMS Parameter Name: DS3 CSESs
dsx3CurrentPESs	-	The counter associated with the number of P-bit Errored Seconds.
dsx3CurrentPSESs	-	The counter associated with the number of P-bit Severely Errored Seconds.
dsx3CurrentUASs	-	The counter associated with the number of Unavailable Seconds.

DS3 Performance Monitoring

MIB Name	Gauge/ Counter	Description
dsx3CurrentLCVs	-	The counter associated with the number of Line Coding Violations.
dsx3CurrentPCVs	-	The counter associated with the number of P-bit Coding Violations.
dsx3CurrentLESs	-	The number of Line Errored Seconds.
dsx3CurrentCCVs	-	The number of C-bit Coding Violations.
dsx3CurrentCESs	-	The number of C-bit Errored Seconds.
dsx3CurrentCSESs	-	The number of C-bit Severely Errored Seconds.

3.2.6.2 Fiber Group Performance Monitoring

Note: These PM parameters are applicable only to Mediant 3000 with the TP-6310 blade.

Fiber Group Performance Monitoring

MIB Name	Gauge/ Counter	Description
sonetSectionCurrentESs	Gauge	The counter associated with the number of Errored Seconds encountered by a SONET/SDH Section in the current 15 minute interval. EMS Parameter Name: Section ESs
sonetSectionCurrentSESs	Gauge	The counter associated with the number of Severely Errored Seconds encountered by a SONET/SDH Section in the current 15 minute interval. EMS Parameter Name: Section SESs
sonetSectionCurrentCVs	Gauge	The counter associated with the number of Coding Violations encountered by a SONET/SDH Section in the current 15 minute interval. EMS Parameter Name: Section CVs
sonetLineCurrentESs	Gauge	The counter associated with the number of Errored Seconds encountered by a SONET/SDH Line in the current 15 minute interval. EMS Parameter Name: Line ESs
sonetLineCurrentSESs	Gauge	The counter associated with the number of Severely Errored Seconds encountered by a SONET/SDH Line in the current 15 minute interval. EMS Parameter Name: Line SESs

Fiber Group Performance Monitoring

MIB Name	Gauge/ Counter	Description
sonetLineCurrentCVs	Gauge	The counter associated with the number of Coding Violations encountered by a SONET/SDH Line in the current 15 minute interval. EMS Parameter Name: Line CVs
sonetLineCurrentUASs	Gauge	The counter associated with the number of Unavailable Seconds encountered by a SONET/SDH Line in the current 15 minute interval. EMS Parameter Name: Line UASs
sonetPathCurrentESs	Gauge	The counter associated with the number of Errored Seconds encountered by a SONET/SDH Path in the current 15 minute interval. EMS Parameter Name: Path ESs
sonetPathCurrentSESs	Gauge	The counter associated with the number of Severely Errored Seconds encountered by a SONET/SDH Path in the current 15 minute interval. EMS Parameter Name: Path SESs
sonetPathCurrentCVs	Gauge	The counter associated with the number of Coding Violations encountered by a SONET/SDH Path in the current 15 minute interval. EMS Parameter Name: Path CVs
sonetPathCurrentUASs	Gauge	The counter associated with the number of Unavailable Seconds encountered by a Path in the current 15 minute interval. EMS Parameter Name: Path UASs

3.2.6.2.1 System IP Performance Monitoring

System IP Performance Monitoring

MIB Name	Gauge/ Counter	Description
acPMNetUtilKBytesVolumeTx	Counter	Counts the total number of outgoing Kbytes (1000 bytes) from the interface during the last interval. EMS Parameter Name: Number of Outgoing KBytes
acPMNetUtilKBytesVolumeRx	Counter	Counts the total number of Kbytes (1000 bytes) received on the interface, including those received in error, during the last interval. EMS Parameter Name: Number of Incoming KBytes
acPMNetUtilPacketsVolumeTx	Counter	Counts the total number of outgoing Packets from the interface during the last interval. EMS Parameter Name: Number of Outgoing Pkts
acPMNetUtilPacketsVolumeRx	Counter	Counts the total number of Packets received on the interface, including those received in error, during the last interval. EMS Parameter Name: Number of Incoming Pkts

System IP Performance Monitoring

MIB Name	Gauge/ Counter	Description
AcPMNetUtilDiscardedPackets Val	Counter	Counts the total number of malformed IP Packets received on the interface during the last interval. These are packets which are corrupted or discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, etc. EMS Parameter Name: Number of Incoming Discarded Pkts
acPMNetUtilKBytesTotalTx	Gauge	This attribute counts the Current total number of outgoing Kbytes (1000 bytes) from the interface, so far from the beginning of the current collection interval as indicated by time Interval. EMS Parameter Name: Number of Outgoing KBytes
acPMNetUtilKBytesTotalRx	Gauge	This attribute counts the total number of Kbytes (1000 bytes) received on the interface, including those received in error, so far from the beginning of the current collection interval as indicated by time Interval. EMS Parameter Name: Number of Incoming KBytes
acPMNetUtilPacketsTotalTx	Gauge	This attribute counts the Current total number of outgoing Packets from the interface, so far from the beginning of the current collection interval as indicated by time Interval. EMS Parameter Name: Number of Outgoing Pkts
acPMNetUtilPacketsTotalRx	Gauge	This attribute counts the Current total number of Packets received on the interface, including those received in error, so far from the beginning of the current collection interval as indicated by time Interval. EMS Parameter Name: Number of Incoming Pkts
AcPMNetUtilDiscardedPackets Total	Gauge	This attribute counts the Current total number of malformed IP Packets received on the interface from the beginning of the current collection interval. These are packets which are corrupted or discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, etc. EMS Parameter Name: Number of Incoming Discarded Pkts

3.2.6.2.2 VoP Call Statistics Performance Monitoring

Note: These PM parameters are not applicable to the MediaPack series.

VoP Call Statistics Performance Monitoring

MIB Name	Gauge / Counter	Description
acPMActiveContextCountAverage	Gauge	Indicates the average number of voice calls connected on the gateway since the last clear. EMS Parameter Name: Num of Active Contexts Avg
acPMActiveContextCountMin	Gauge	Indicates the minimum number of voice calls connected on the gateway since the last clear. EMS Parameter Name: Num of Active Contexts Min
acPMActiveContextCountMax	Gauge	Indicates the maximum number of voice calls connected on the gateway since the last clear. EMS Parameter Name: Num of Active Contexts Max
acPMChannelsPerCoderAverageG711	Gauge	Indicates the average number of G.711 calls present on the TPM. EMS Parameter Name: G711 Active Calls Avg
acPMChannelsPerCoderAverageG723	Gauge	Indicates the average number of G.723 calls present on the TPM. This attribute is only displayed if the G.723 Codec is provisioned on the DSP template. EMS Parameter Name: G723 Active Calls Avg
acPMChannelsPerCoderAverageG728	Gauge	Indicates the average number of G.728 calls present on the TPM. This attribute is only displayed if the G.728 Codec is provisioned on the DSP template. EMS Parameter Name: G728 Active Calls Avg
acPMChannelsPerCoderAverageG729a	Gauge	Indicates the average number of G.729a calls present on the TPM. This attribute is only displayed if the G.729a Codec is provisioned on the DSP. EMS Parameter Name: G729a Active Calls Avg
acPMChannelsPerCoderAverageG729e	Gauge	Indicates the average number of G.729e calls present on the TPM. This attribute is only displayed if the G.729e Codec is provisioned on the DSP template. EMS Parameter Name: G729e Active Calls Avg

VoP Call Statistics Performance Monitoring

MIB Name	Gauge / Counter	Description
acPMChannelsPerCoderAverageAMR	Gauge	Indicates the average number of AMR calls present on the TPM. This attribute is only displayed if the AMR Codec is provisioned on the DSP template. EMS Parameter Name: AMR Active Calls Avg
acPMMegacoTransactionRequestsPerSecTable	Gauge	Indicates how many Megaco transactions are received at the board per second. This Performance Monitor helps administrators determine the cause of performance issues on the network, if they occur. CPU overload, for example, may result if the Megaco call agent sends too many Megaco transactions per second to the board.
acPMModuleRTPPacketLossRxMax	Gauge	Indicates the Max Rx RTP Packet loss (reported by RTCP) per TPM, up to this point in time during the collection interval, as indicated by the time Interval. EMS Parameter Name: Rx RTP Packet Loss Max
acPMModuleRTPPacketLossTxMax	Gauge	Indicates the Max Tx RTP Packet loss (reported by RTCP) per TPM, up to this point in time during the collection interval, as indicated by the time Interval. EMS Parameter Name: Tx RTP Packet Loss Max
acPMModulePacketDelayAverage	Gauge	Indicates the average RTP packets delay per TPM, up to this point in time during the collection interval, as indicated by the time Interval. EMS Parameter Name: RTP delay Average
acPMModulePacketDelayMax	Gauge	Indicates the maximum RTP packets delay per TPM, up to this point in time during the collection interval, as indicated by the time Interval. EMS Parameter Name: RTP delay Max
acPMModulePacketDelayMin	Gauge	Indicates the minimum RTP packets delay per TPM, up to this point in time during the collection interval, as indicated by the time Interval. EMS Parameter Name: RTP delay Min
acPMModulePacketJitterAverage	Gauge	Indicates the average RTP packets jitter per TPM, up to this point in time during the collection interval, as indicated by the time Interval. EMS Parameter Name: RTP jitter Average
acPMModulePacketJitterMin	Gauge	Indicates the minimum RTP packets jitter per TPM, up to this point in time during the collection interval, as indicated by the time Interval. EMS Parameter Name: RTP jitter Min
acPMModulePacketJitterMax	Gauge	Indicates the maximum RTP packets jitter per TPM, up to this point in time during the collection interval, as indicated by the time Interval. EMS Parameter Name: RTP jitter Max

VoP Call Statistics Performance Monitoring

MIB Name	Gauge / Counter	Description
acPMModuleRTPBytesRxMax	Gauge	Indicates the Max Tx RTP Bytes per TPM, up to this point in time during the collection interval, as indicated by the time Interval. EMS Parameter Name: Rx RTP Bytes Max
acPMModuleRTPBytesTxMax	Gauge	Indicates the Max Rx RTP Bytes per TPM, up to this point in time during the collection interval, as indicated by the time Interval. EMS Parameter Name: Tx RTP Bytes Max
acPMModuleRTPPacketsRxMax	Gauge	Indicates the Max Rx RTP Packets per TPM, up to this point in time during the collection interval, as indicated by the time Interval. EMS Parameter Name: Rx RTP Packets Max
acPMModuleRTPPacketsTxMax	Gauge	Indicates the Max Tx RTP Packets per TPM, up to this point in time during the collection interval, as indicated by the time Interval. EMS Parameter Name: Tx RTP Packets Max
acPMActiveContextCountVal	Gauge	Indicates the current number of voice calls connected on the box since last clear. EMS Parameter Name: Num of Active Contexts
acPMChannelsPerCoderValG711	Gauge	This attribute indicates the current number of G711 calls present on the TPM. EMS Parameter Name: G711 Active Calls
acPMChannelsPerCoderValG723	Gauge	This attribute indicates the current number of G723 calls present on the TPM.This attribute is only displayed if the G723 Codec is provisioned on the DSP template. EMS Parameter Name: G723 Active Calls
acPMChannelsPerCoderValG728	Gauge	This attribute indicates the current number of G728 calls present on the TPM.This attribute is only displayed if the G728 Codec is provisioned on the DSP template. EMS Parameter Name: G728 Active Calls
acPMChannelsPerCoderValG729a	Gauge	This attribute indicates the current number of G729a calls present on the TPM.This attribute is only displayed if the G729a Codec is provisioned on the DSP. EMS Parameter Name: G729a Active Calls
acPMChannelsPerCoderValG729e	Gauge	This attribute indicates the current number of G729e calls present on the TPM.This attribute is only displayed if the G729e Codec is provisioned on the DSP template. EMS Parameter Name: G729e Active Calls
acPMChannelsPerCoderValAMR	Gauge	This attribute indicates the current number of AMR calls present on the TPM.This attribute is only displayed if the AMR Codec is provisioned on the DSP template. EMS Parameter Name: AMR Active Calls

VoP Call Statistics Performance Monitoring

MIB Name	Gauge / Counter	Description
acPMModuleRTPPacketLossRxTotal	Gauge	The total number of RTP packet loss reported by RTCP since last reset. EMS Parameter Name: Rx Packet Loss current
acPMModuleRTPPacketLossTxTotal	Gauge	The total number of RTP packet loss reported by RTCP since last reset. EMS Parameter Name: Tx Packets Loss current
acPMModuleRTPPacketsRxTotal	Gauge	The total number of packets received since last reset. EMS Parameter Name: Rx Packets Current
acPMModuleRTPPacketsTxTotal	Gauge	The total number of RTP packets transmitted since last reset. EMS Parameter Name: Rx Packets Current

3.2.6.2.3 Common Control Performance Monitoring

Note: These PM parameters are not applicable to the MediaPack series.

Common Control Performance Monitoring

MIB Name	Gauge/ Counter	Description
acPMCPConnectionLifetimeAverage	Counter	Indicates the Connection lifetime, in seconds. EMS Parameter Name: Lifetime in seconds Avg
acPMCPConnectionLifetimeMin	Counter	Indicates the Connection lifetime, in seconds. EMS Parameter Name: Lifetime in seconds Min
acPMCPConnectionLifetimeMax	Counter	Indicates the Connection lifetime, in seconds. EMS Parameter Name: Lifetime in seconds Max
acPMCPCommandCounterValRx	Counter	Indicates the MGC response counters. EMS Parameter Name: MGC response counters
acPMCPCommandCounterValTx	Counter	Indicates the MGC command counters. EMS Parameter Name: MGC command counters
acPMCPRetransmissionCountValRx	Counter	Counts the number of incoming retransmissions. EMS Parameter Name: MGC Rx retransmissions
acPMCPRetransmissionCountValTx	Counter	Counts the number of transactions retransmissions sent from the board. EMS Parameter Name: MGC Tx retransmissions
acPMCPCallAttemptsPerSecAverage	Counter	Average of call attempts (successful and unsuccessful) per second, during last interval. EMS Parameter Name: Call Attempts Per Sec Average

Common Control Performance Monitoring

MIB Name	Gauge/ Counter	Description
acPMCPCallAttemptsPerSecMax	Counter	Maximum of call attempts (successful and unsuccessful) per second, during last interval. EMS Parameter Name: Call Attempts Per Sec Max
acPMCPCallAttemptsPerSecMin	Counter	Minimum of call attempts (successful and unsuccessful) per second, during last interval. EMS Parameter Name: Call Attempts Per Sec Min
acPMCPConnectionLifetimeVolume	Counter	The Connection lifetime in seconds. EMS Parameter Name: Lifetime in seconds
acPMCPCommandCounterTotalTx	Gauge	MGC command counters. EMS Parameter Name: MGC Tx command counters
acPMCPCommandCounterTotalRx	Gauge	MGC response counters. EMS Parameter Name: MGC Rx command counters
acPMCPRetransmissionCountTotalTx	Gauge	Number of transactions retransmissions sent from the board. EMS Parameter Name: MGC Tx retransmissions
acPMCPRetransmissionCountTotalRx	Gauge	Number of incoming retransmissions. EMS Parameter Name: MGC Rx retransmissions
acPMCPCallAttemptsPerSecVal	Gauge	Number of Call attempts (successful and unsuccessful) per second, during current interval. EMS Parameter Name: Call Attempts Per Sec

3.2.6.2.4 Trunk Statistics Performance Monitoring

Note: These PM parameters are applicable only to Digital PSTN devices.

Trunk Statistics Performance Monitoring

MIB Name	Gauge/ Counter	Description
acPMTrunkUtilizationAverage	Gauge	Indicates the Average of simultaneously busy DS0 channels on this Trunk up to this point in time during the collection interval, as indicated by the Time Interval. A busy channel is when the Physical DS0 Termination isn't in Null context or OOS. A Trunk is either E1 or T1. EMS Parameter Name: Trunk utilization Avg

Trunk Statistics Performance Monitoring

MIB Name	Gauge/ Counter	Description
acPMTrunkUtilizationMin	Gauge	Indicates the Minimum of simultaneously busy DS0 channels on this Trunk up to this point in time during the collection interval, as indicated by the Time Interval. A busy channel is when the Physical DS0 Termination isn't in Null context or OOS. A Trunk is either E1 or T1. EMS Parameter Name: Trunk utilization Min
acPMTrunkUtilizationMax	Gauge	Indicates the Maximum of simultaneously busy DS0 channels on this Trunk up to this point in time during the collection interval, as indicated by the Time Interval. A busy channel is when the Physical DS0 Termination isn't in Null context or OOS. A Trunk is either E1 or T1. EMS Parameter Name: Trunk utilization Max
dsx1IntervalESs	Gauge	Indicates the number of Errored Seconds. EMS Parameter Name: Trunk Errored Seconds
dsx1IntervalCSSs	Gauge	Indicates the number of Controlled Slip Seconds. EMS Parameter Name: Trunk Controlled Slip Seconds
dsx1IntervalPCVs	Gauge	Indicates the number of Path Coding Violations. EMS Parameter Name: Trunk Path Coding Violations
dsx1IntervalBESs	Gauge	Indicates the number of Bursty Errored Seconds. EMS Parameter Name: Trunk Bursty Errored Seconds
acPMTrunkUtilizationVal	Gauge	This attribute indicates the Current simultaneous busy DS0 channels on this Trunk. A busy channel is when the Physical DS0 Termination isn't in Null context or OOS. A Trunk is either E1 or T1. EMS Parameter Name: Trunk utilization
acPMPSTNTrunkActivitySecondsTotal	Gauge	This attribute amount of call duration per timeslot and E1 since last clear. EMS Parameter Name: Trunk Calls Duration
dsx1TotalESs	Gauge	This attribute indicates amount of Errored Seconds encountered by a DS1 interface in the previous 24 hour interval. Invalid 15 minute intervals count as 0. EMS Parameter Name: Trunk Errored Seconds
dsx1TotalCSSs	Gauge	This attribute indicates amount of Controlled Slip Seconds encountered by a DS1 interface in the previous 24 hour interval. Invalid 15 minute intervals count as 0. EMS Parameter Name: Trunk Controlled Slip Seconds

Trunk Statistics Performance Monitoring

MIB Name	Gauge/ Counter	Description
dsx1TotalPCVs	Gauge	This attribute indicates amount of Path Coding Violations encountered by a DS1 interface in the previous 24 hour interval. Invalid 15 minute intervals count as 0. EMS Parameter Name: Trunk Path Coding Violations
dsx1TotalBESs	Gauge	This attribute indicates amount of Bursty Errored Seconds encountered by a DS1 interface in the previous 24 hour interval. Invalid 15 minute intervals count as 0. EMS Parameter Name: Trunk Bursty Errored Seconds

3.2.6.2.5 SS7 Performance Monitoring

Note: These PM parameters are applicable only to Mediant 2000 and Mediant 3000.

SS7 Linkset Performance Monitoring

MIB Name	Gauge/ Counter	Description
acPMSS7SN0LSOutOfServiceTimeAboveHighThreshold	Counter	Percent of interval time for which gauge is above what was determined as the high threshold. In this case - 'Out Of Service'. EMS Parameter Name: SS7 LinkSet SN0 Out of Service %
acPMSS7SN1LSOutOfServiceTimeAboveHighThreshold	Counter	Percent of interval time for which gauge is above what was determined as the high threshold. In this case - 'Out Of Service'. EMS Parameter Name: SS7 LinkSet SN1 Out of Service %

SS7 Link Performance Monitoring

MIB Name	Gauge/ Counter	Description
acPMSS7TxLSSUVal	Counter	Value of gauge or counter. EMS Parameter Name: SS7 Link Transmitted LSSU's

MIB Name	Gauge/ Counter	Description
acPMSS7RxLSSUVal	Counter	Value of gauge or counter. EMS Parameter Name: SS7 Link Received LSSU's
acPMSS7TxFISUVal	Counter	Value of gauge or counter. EMS Parameter Name: SS7 Link Transmitted FISU's
acPMSS7RxFISUVal	Counter	Value of gauge or counter. EMS Parameter Name: SS7 Link Received FISU's
acPMSS7DiscMSUVal	Counter	Value of gauge or counter. EMS Parameter Name: SS7 Link Discarded MSU's
acPMSS7InServiceTimeAboveHighThreshold	Counter	Percent of interval time for which gauge is above what was determined as the high threshold. In this case - 'In Service'. EMS Parameter Name: SS7 Link In Service %
acPMSS7OutOfServiceTimeAboveHighThreshold	Counter	Percent of interval time for which gauge is above what was determined as the high threshold. In this case - 'Out Of Service'. EMS Parameter Name: SS7 Link Out Of Service %

3.2.7 Toplogy MIB - Objects

3.2.7.1 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.

3.2.7.2 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:

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

DS1 Digital Interfaces

ifTable	Values
ifLinkUpDownTrapEnable	Set to enabled(1)
ifHighSpeed	Speed of line in Mega-bits per second: 2
ifConnectorPresent	Set to true(1) normally
ifCounterDiscontinuityTime	Always zero.

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.

Gigabit Ethernet Interface

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.
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.
ifCounterDiscontinuityTime	As defined in IfMIB.

SONET / SDH Interface (Applicable to 6310 Devices)

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 (Applicable to 6310 Devices)

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

DS3 Interfaces (Applicable to 6310 Devices)

ifTable	Values
	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.

3.2.8 SNMP Interface Details

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

There are several alternatives for SNMP security:

- Use SNMPv2c community strings.
- Use SNMPv3 User-based Security Model (USM) users.
- Use SNMP encoded over IPSec. For more details, refer to 'Security' on page 707.
- Use some combinations of the above.

For *ini* file encoding, refer to 'Utilities' on page 787.

3.2.8.1 SNMP Community Names

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

Up to 5 read-only community strings, 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.

SNMP Predefined Groups

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

3.2.8.1.1 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.

3.2.8.1.2 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.

➢ **To add a read-only community string, v2user:**

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:**

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:**

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,:**

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:**

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:**

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.

3.2.8.2 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:

SNMPv3 Security Levels

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

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

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)

3.2.8.3 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.

SNMPv3 Table Columns Description

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

SNMPUsers_PrivProtocol	Privacy protocol to be used for this user. Possible values are 0 (none), 1 (DES), 2 (3DES), 3 (AES128), 4 (AES192), 5 (AES256)	0
SNMPUsers_AuthKey	Authentication key.	""
SNMPUsers_PrivKey	Privacy key.	""
SNMPUsers_Group	The group that this user is associated with. Possible values are 0 (read-only group), 1 (read-write group), and 2 (trap group). The actual group will be ReadGroup<sl>, ReadWriteGroup<sl> or TrapGroup<sl> where <sl> is the SecurityLevel (1=noAuthNoPriv, 2=authNoPriv, 3=authPriv)	0

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

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

The following sample configuration creates 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.

3.2.8.4 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:**

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:**

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:**

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:**

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

3.2.8.5 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 work 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').

3.2.8.5.1 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.

3.2.8.5.2 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:**

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:**

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):**

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 last Trusted Manager:**

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.

3.2.8.6 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.

3.2.8.7 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.

3.2.8.7.1 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.

3.2.8.7.2 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.

3.2.8.7.3 Configuring SNMP Engine ID

SNMP Engine ID Configuration:
SnmpEngineIDString *ini* file parameter – is a string of 36 characters. This parameter will work offline. The parameter default value will be 00:00:00:00:00:00:00:00:00:00:00:00 12 Hex characters.
The provided key must be set with 12 HEX values delimited by ':'.
If the supplied key will not pass validation of the 12 HEX values input or will be set with the default value, the engine ID will be generated the same way it was done according to the RFC.
Before setting this parameter, all SNMPv3 users must be deleted, otherwise the configuration will be ignored.

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 are defined in the snmpManagersTable:

   - snmpTrapManagerSending
   - snmpManagerIsUsed
   - snmpManagerTrapPort
   - 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).

Notes:

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

- 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:**

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:**

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:**

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:**

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

➢ **To enable a trap destination:**

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

3.2.9 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.

3.2.10 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.

3.2.11 Gateway Severity

The *acSysStateGWSeverity* parameter reflects the highest active alarm severity on the device. The possible options are:

- noAlarm (0)
- indeterminate (1)
- warning (2)
- minor (3)
- major (4)
- critical (5)

3.2.12 SNMP for AMS

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

3.2.12.1 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

3.2.12.2 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 108):

- **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.

3.2.13 High Availability Systems

Note: High Availability Systems are 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.

3.2.14 Administrative State Control

3.2.14.1 Node Maintenance

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

- acSysActionAdminState - Determines the gateway's desired operational state.
  - locked: Shutdown the GW In the time frame set by acgwAdminStateLockControl.
  - shuttingDown: (read only) Graceful Shutdown is being carried out.
  - unlocked: GW is in service.
- acSysActionAdminStateLockTimeout - 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.

3.2.14.2 Graceful Shutdown

acSysActionAdminState 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 acSysActionAdminState to perform a lock, acSysActionAdminStateLockTimeout 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 575.

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

Note: 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.

3.2.15 SNMP Traps

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.

3.2.15.1 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'.

3.2.15.2 Alarms Applicable to all Devices

Note: For 3000 devices, the source format is : **System#<n>**.

For devices other than 3000, the source format is: **Board#<n>**.

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.

acBoardFatalError Alarm Trap

Alarm:	acBoardFatalError
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.

acBoardTemperatureAlarm Alarm Trap

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

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

acBoardEvResettingBoard Alarm Trap

Alarm:	acBoardEvResettingBoard
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.

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**

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

acgwAdminStateChange Alarm Trap

Alarm:	acgwAdminStateChange
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.

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
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.

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
Corrective Action:	Ensure Ethernet IF is well connected.

acNTPServerStatusAlarm

Alarm:	acNTPServerStatusAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.71
Default Severity	Major
Event Type:	communicationsAlarm
Probable Cause:	communicationsSubsystemFailure
Alarm Text:	NTP server alarm. No connection to NTP server.
Status Changes:	
Condition:	No communication to NTP server from the start.
Alarm status:	Major
Condition:	No communication to NTP server after the time was already set once.
Alarm status:	Minor

acNTPServerStatusAlarm

Alarm:	acNTPServerStatusAlarm
Corrective Action:	Fix NTP communication. (The NTP server is down or the NTP Server IP configured in the blade is incorrect).

3.2.16 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.)**

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.

acActiveAlarmTableOverflow Alarm Trap

Alarm:	acActiveAlarmTableOverflow
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.

Note: 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).

For 3000 devices, the source format is: Module#<n>/EthernetLink#0.

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
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

3.2.16.1 Alarms Applicable to Mediant 3000 Only

Note: The Timing Module Alarms component is: Chassis#0/TimingManager#0.

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.

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 escalated 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.

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:	

Notes:

- The source varbind text for the alarm under the component below is Chassis#0/FanTray#0.
- For **Mediant 1000**, only the following are applicable: acFanTray Alarm and acPowerSupply Alarm.

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.

acFanTrayAlarm Alarm Trap

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.

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.

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.

acUserInputAlarm Alarm Trap

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

3.2.16.1.1Component: Chassis#0

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

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.

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.

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.
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.

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.

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

Note: Only acFanTray Alarm and acPowerSupply Alarm are applicable to the Mediant 1000.

.

3.2.16.1.2Component: System#0/Module#<m>

Notes:

- The following is only applicable to **3000** devices.
- 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.

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

acHASystemFaultAlarm Alarm Trap

Alarm:	acHASystemFaultAlarm
	HW fault. invalid TER Type HW fault. invalid TER active/redundant state HW fault. Error reading GbE state Redundant module is missing Unable to sync SW versions Redundant is not connecting Redundant is not reconnecting after deliberate restart No Ethernet Link in redundant module SA module faulty or missing
Condition:	HA feature is active and the redundant module is in start up mode and hasn't connected yet.
Alarm status:	Minor
<text> value:	Waiting for redundant to connect
Condition:	HA system is active.
Alarm status:	Cleared

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

acHASystemConfigMismatchAlarm Alarm Trap

Alarm:	acHASystemConfigMismatchAlarm

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
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.

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

acBoardEthernetLinkAlarm Alarm Trap

Alarm:	acBoardEthernetLinkAlarm
Condition:	Fault on both interfaces
Alarm status:	critical
<text> value:	No Ethernet link
Condition:	Fault on single interface of the Redundant module.
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.

3.2.16.2 Audio Provisioning Alarm (Applicable to IPmedia 2000 / IPmedia 3000)

Note: The source varbind text for all the alarms under the component below is System#0/AudioStaging#0.

acAudioProvisioningAlarm Alarm Trap

Alarm:	acAudioProvisioningAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.14
Default Severity	critical
Event Type:	processingErrorAlarm
Probable Cause:	configurationOrCustomizationError (7)

acAudioProvisioningAlarm Alarm Trap

Alarm:	acAudioProvisioningAlarm
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.

3.2.16.2.1SS7 Alarms

Notes:

- **For 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.
- **For devices other than 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.

acSS7LinkStateChangeAlarm Trap

Alarm:	acSS7LinkStateChangeAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.19
Default Severity	Major
Event Type:	communicationsAlarm
Probable Cause:	other
Alarm Text:	*** SS7 *** Link %i is %s $s
Status Changes:	
Condition:	Operational state of the SS7 link becomes ‘BUSY’.
Alarm status:	Major

acSS7LinkStateChangeAlarm Trap

Alarm:	acSS7LinkStateChangeAlarm
<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.

acSS7LinkInhibitStateChangeAlarm Trap

Alarm:	acSS7LinkInhibitStateChangeAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.20
Default Severity	Major
Event Type:	communicationsAlarm
Probable Cause:	other
Alarm Text:	*** SS7 *** Link %i (SP %i linkset %i slc %i) is %s
Status Changes:	
Condition:	SS7 link becomes inhibited (local or remote).
Alarm status:	Major
<text> value:	%i - <Link number> %i - <SP number> %i - <Link-Set number> %i - <SLC number> %s - <congestion state>: { "UNINHIBITED", "INHIBITED" }
Additional Info1 varbind	INHIBITED
Condition:	Link becomes uninhibited - local AND remote

acSS7LinkInhibitStateChangeAlarm Trap

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.**

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**

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: %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

acSS7LinkCongestionStateChangeAlarmTrap

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.**

acSS7LinkSetStateChangeAlarm Trap

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

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 SN (signaling node) number. **(Applicable to 3000 devices.)**

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

acSS7SNSetStateChangeAlarmTrap

Alarm:	acSS7SNSetStateChangeAlarm
<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.

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
Note:	**Support pending**

The source varbind text for all the alarms under the component below is Board#<n>/SS7#0/SS7SN#<i>/ss7aliaspc#<m> where n is the slot number ,i is the SN number and m is the alias pc number. (Not Applicable to 3000 devices.)

The source varbind text for all the alarms under the component below is System#0/SS7#0/SS7SN#<i>/ss7aliaspc#<m> where m ,i is the SN number and is the alias pc number. (Applicable to 3000 devices.)

acSS7AliasPcStateChangeAlarm Trap

Alarm:	acSS7AliasPcStateChangeAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.73
Default Severity	Major
Event Type:	communicationsAlarm
Probable Cause:	other
Alarm Text:	*** SS7 *** Alias PC %d [sn %d, apc_idx %d] is %s
Status Changes:	
Condition:	Alias PC state changes

acSS7AliasPcStateChangeAlarm Trap

Alarm:	acSS7AliasPcStateChangeAlarm
Alarm status:	Major
<text> value:	%d – The Alias PC value %d - sn index %d - Alias pc index %s – new state ("OFFLINE", "BUSY", "INSERVICE")
Additional Info1 varbind	
Condition:	Alias PC state is not BUSY
Alarm status:	Clear
Corrective Action:	

The source varbind text for all the alarms under the component below is Board#<n>/SS7#0/ss7ualgroup#<m> where n is the slot number and m is the alias pc number. (Not Applicable to 3000 devices.)

The source varbind text for all the alarms under the component below is System#0/SS7#0/ss7ualgroup#<m> where m is the ual group number. (Applicable to 3000 devices.)

acSS7UalGroupStateChangeAlarm Trap

Alarm:	acSS7UalGroupStateChangeAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.74
Default Severity	Major
Event Type:	communicationsAlarm
Probable Cause:	other
Alarm Text:	*** SS7 *** Group Id %j Asp status is %s
Status Changes:	
Condition:	Group ASP status changes.
Alarm status:	Major
<text> value:	%i - Group number %s - New state ("NO_SCTP", "SCTP_ASSOCIATE", SCTP_FAILURE", "ASP_DOWN", "ASP_INACTIVE", "ASP_ACTIVE")
Additional Info1 varbind	
Condition:	When group ASP status changes to "ASP_ACTIVE"
Alarm status:	cleared

acSS7UalGroupStateChangeAlarm Trap

Alarm:	acSS7UalGroupStateChangeAlarm
Corrective Action:	

3.2.16.3 Alarms Applicable to MediaPack and Mediant 1000

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

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

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.

acAnalogPortHighTemperature Trap

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

3.2.16.4 Alarms Applicable to Mediant 1000 Only

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

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

acHwFailureAlarm Alarm Trap

<text> value:	IF-Module Mismatch

acBoardFatalError

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.

3.2.16.4.1SONET Alarms - Applicable to 6310 Devices Only

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

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

AcSonetSectionLOFAlarm Alarm Trap

<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

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
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

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

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

3.2.16.4.2Path Alarms Applicable to 6310 Devices Only

Note: The source format for the alarm below is Interfaces#0/Path#<m>.

acSonetPathSTSLOPAlarm

Alarm:	acSonetPathSTSLOPAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.61
Default Severity	critical
Event Type:	communicationsAlarm
Probable Cause:	receiveFailure
Alarm Text:	Sonet Path STS AIS alarm.
Status Changes:	
Condition:	LOP condition is present on Path #n.
Alarm status:	critical
<text> value:	LOP
Note:	The sonetPathCurrentStatus in the sonetPathCurrentTable have a value sonetPathSTSLOP (2).
Condition:	LOP condition is not present.
Alarm status:	cleared

acSonetPathSTSAISAlarm

Alarm:	acSonetPathSTSAISAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.62
Default Severity	critical
Event Type:	communicationsAlarm
Probable Cause:	receiveFailure
Alarm Text:	Sonet Path STS AIS alarm..
Status Changes:	
Condition:	AIS condition is present on Path #n.
Alarm status:	critical
<text> value:	AIS
Note:	The sonetPathCurrentStatus in the sonetPathCurrentTable have a value

	sonetPathSTSAIS(4).
Condition:	AIS condition is not present.
Alarm status:	cleared

acSonetPathSTSRDIAlarm

Alarm:	acSonetPathSTSRDIAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.63
Default Severity	critical
Event Type:	communicationsAlarm
Probable Cause:	transmitFailure
Alarm Text:	Sonet Path STS RDI alarm.
Status Changes:	
Condition:	RDI condition is present on Path #n.
Alarm status:	critical
<text> value:	RDI
Note:	The sonetPathCurrentStatus in the sonetPathCurrentTable have a value sonetPathSTSRDI(8).
Condition:	RDI condition is not present.
Alarm status:	cleared

acSonetPathUnequippedAlarm

Alarm:	acSonetPathUnequippedAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.64
Default Severity	critical
Event Type:	communicationsAlarm
Probable Cause:	receiveFailure
Alarm Text:	Sonet Path Unequipped alarm.
Status Changes:	
Condition:	Unequipped condition is present on Path #n.
Alarm status:	critical
<text> value:	Unequipped
Note:	The sonetPathCurrentStatus in the sonetPathCurrentTable have a value sonetPathUnequipped(16).
Condition:	Unequipped condition is not present.
Alarm status:	cleared

acSonetPathSignalLabelMismatchAlarm

Alarm:	acSonetPathSignalLabelMismatchAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.65
Default Severity	critical
Event Type:	communicationsAlarm
Probable Cause:	receiveFailure
Alarm Text:	Sonet Path Signal Label Mismatch alarm.
Status Changes:	
Condition:	Signal Label Mismatch condition is present on Path #n.
Alarm status:	critical
<text> value:	SignalLabelMismatch
Note:	The sonetPathCurrentStatus in the sonetPathCurrentTable have a value sonetPathSignalLabelMismatch(32).
Condition:	Signal Label Mismatch condition is not present.
Alarm status:	cleared

3.2.16.4.3DS3 Alarms Applicable to 6310 Only

Note: The source format for the alarm below is Interfaces#0/DS3#<m>.

acDS3RAIAlarm

Alarm:	acDS3RAIAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.66
Default Severity	critical
Event Type:	communicationsAlarm
Probable Cause:	transmitFailure
Alarm Text:	DS3 RAI alarm.
Status Changes:	
Condition:	RAI condition is present on DS3-Line #n.
Alarm status:	critical
<text> value:	RAI
Note:	The dsx3LineStatusfield in the dsx3ConfigTablewill have a value dsx3RcvRAIFailure(2).

Condition:	RIA condition is not present.
Alarm status:	cleared

acDS3AISAlarm

Alarm:	acDS3AISAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.67
Default Severity	critical
Event Type:	communicationsAlarm
Probable Cause:	receiveFailure
Alarm Text:	DS3 AIS alarm.
Status Changes:	
Condition:	AIS condition is present on DS3-Line #n.
Alarm status:	critical
<text> value:	AIS
Note:	The dsx3LineStatusfield in the dsx3ConfigTablewill have a value dsx3RcvAIS(8).
Condition:	AIS condition is not present.
Alarm status:	cleared

acDS3LOFAlarm

Alarm:	acDS3LOFAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.68
Default Severity	critical
Event Type:	communicationsAlarm
Probable Cause:	lossOfFrame
Alarm Text:	DS3 LOF alarm.
Status Changes:	
Condition:	LOF condition is present on DS3-Line #n.
Alarm status:	critical
<text> value:	LOF
Note:	The dsx3LineStatusfield in the dsx3ConfigTablewill have a value dsx3LOF (32).
Condition:	LOF condition is not present.
Alarm status:	cleared

acDS3LOSAlarm

Alarm:	acDS3LOSAlarm
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.69
Default Severity	critical
Event Type:	communicationsAlarm
Probable Cause:	lossOfSignal
Alarm Text:	DS3 LOS alarm.
Status Changes:	
Condition:	LOS condition is present on DS3-Line #n.
Alarm status:	critical
<text> value:	LOS
Note:	The dsx3LineStatusfield in the dsx3ConfigTablewill have a value dsx3LOS (64).
Condition:	LOS condition is not present.
Alarm status:	cleared

3.2.16.4.4Trunk Alarms Applicable to Digital Gateways Only

Note: 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.

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.

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.
Status Changes:	
Condition:	Near end LOS

acTrunksAlarmNearEndLOS Alarm Trap

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

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.

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
Alarm status:	cleared
Corrective action:	None.

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.

3.2.16.5 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.

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’
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**

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

acHTTPDownloadResult Log Trap

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

acSSHConnectionStatus

Alarm:	acSSHConnectionStatus
OID:	1.3.6.1.4.1.5003.9.10.1.21.2.0.77
Default Severity	indeterminate
Event Type:	environmentalAlarm

acSSHConnectionStatus

Alarm:	acSSHConnectionStatus
Probable Cause:	other
Alarm Text:	"SSH successful login from IP address %s, user %s" "SSH unsuccessful login attempt from IP address %s, user %s"
Status Changes:	
Condition:	SSH connection attempt.
Alarm status:	
<text> value:	%s – remote IP %s – user name
Additional Info1 varbind	
Condition:	SSH connection attempt – success of failure.
Alarm status:	
Corrective Action:	

acBoardConfigurationError Alarm Trap

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

3.2.16.6 Other Traps

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

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.**

authenticationFailure Trap

Trap Name:	authenticationFailure
OID:	1.3.6.1.6.3.1.1.5.5
MIB	SNMPv2-MIB
Note:	**This is a trap from the standard SNMP MIB.**

LinkUp Trap

Trap Name:	LinkUp
OID:	1.3.6.1.6.3.1.1.5.4
MIB	IF-MIB
Note:	**This is a trap from the standard SNMP MIB.**

LinkDown Trap

Trap Name:	LinkDown
OID:	1.3.6.1.6.3.1.1.5.3
MIB	IF-MIB
Note:	**This is a trap from the standard SNMP MIB.**

entConfigChange Trap

Trap Name:	entConfigChange
OID:	1.3.6.1.2.1.4.7.2.0.1
MIB	ENTITY-MIB

entConfigChange Trap

Trap Name:	entConfigChange
Note:	**This is a trap from the standard SNMP MIB.**

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**.

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.

3.2.16.7 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.

3.2.17 SNMP Alarms in Syslog

All SNMP alarms are sent to the Syslog server using the following formats:

- **Raise Alarm**

RAISE-ALARM: <Alarm Name>; Textual Description: <Textual Description>; Severity <Alarm Severity>; Source <Alarm Source>; Unique ID: <Alarm Unique ID >;

If exists Additional Info1:/ Additional Info2:/ Additional Info3:

Message Severity is determined as follows:

Determining Message Severity

SNMP Alarm Severity	AudioCodes' Syslog Severity
Critical	RecoverableMsg
Major	RecoverableMsg
Minor	RecoverableMsg
Warning	Notice
Indeterminate	Notice
Cleared	Notice

- **Clear Alarm**

CLEAR-ALARM: <Alarm Name>; Textual Description: <Textual Description>; Severity <Alarm Severity>; Source <Alarm Source>; Unique ID: <Alarm Unique ID >;

If exists Additional Info1:/ Additional Info2:/ Additional Info3:

3.2.18 Getting Started with SNMP

This section provides directions for getting started and quickly setting up the device for management functionality using AudioCodes SNMP MIBs.

3.2.18.1 Basic SNMP Configuration Setup

This subsection provides a description of the required SNMP configuration when first accessing the SNMP agent running on the device.

To access the device's SNMP agent, there are a few parameters that can be configured if you wish not to use default settings. The SNMP agent default settings include the following:

- SNMP agent is enabled.
- Port 161 in the agent is used for SNMP GET/SET commands.
- No default trap managers are defined and therefore, the device does not send traps.
- The Trap destination port is 162.
- The SNMP agent is accessible to all SNMP managers (i.e., no trusted managers).
- SNMP Protocol version - SNMPv2c with 'public' and 'private' as the read-only and read-write community strings respectively.

Configuring these SNMP attributes is described in the following subsections:

3.2.18.1.1 Disabling SNMP

To disable SNMP, set the following in the *ini* file:

```
DisableSNMP = 1
```

3.2.18.1.2 Configuring SNMP Port

To configure the agent's SNMP port in the *ini* file, set the following:

```
SNMPPort = <x>
; where 'x' is the port number.
```

3.2.18.1.3 Configuring Trap Mangers (Trap Destination)

Configuring Trap Managers (i.e., trap destinations) includes defining IP address and port. This configuration corresponds to the snmpTargetAddrTable. The agent supports up to five separate trap destinations. For each manager, you need to set the manager IP address and trap-receiving port along with enabling the sending to that manager. Trap managers can be configured using *ini* file, SNMP, or Web interface.

In addition, you can associate 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.

- **Using *ini* File:** Two options that can be used separately or together:
  - Explicit IP address:

```
SNMPMANAGERTABLEIP_x=<IP address>
SNMPMANAGERISUSED_x=1
SNMPMANAGERTRAPSENDINGENABLE_x=1
SNMPMANAGERTRAPPORT_x=162 ;(optional)
```

Where *x* is the entry index from 0 to 4.

- Manager host name:

```
SNMPTrapManagerHostName = <'host name on network'>
```

For example: 'myMananger.corp.MyCompany.com'

The host name is translated into the IP address using DNS resolution and is then defined as the fifth (last) trap manager. Until the address is resolved, some traps are expected to be lost.

Notes:

- This option also requires you to configure the DNS server IP address using the *ini* file: DNSPriServerIP =<IP address>.
- This option results in the fifth manager being overrun by the resolved IP address. Online changes to the Manager table will also be overrun.

■ **Using SNMP:** The trap managers are SET using the SNMPTargetMIB MIB object.

- To add an SNMPv2 trap destination:
  - ♦ Add a row to the snmpTargetAddrTable with these values:
  - ♦ Name=trapN, where *N* is an unused number between 0 and 4.
  - ♦ TagList=AC_TRAP
  - ♦ Params=v2cparamsm

  All changes to the trap destination configuration take effect immediately.

➢ **To add an SNMPv3 trap destination:**

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

   - To delete a trap destination:
3. Remove the appropriate row from the snmpTargetAddrTable.
4. If this is the last trap destination associated with this user and security level, you can also delete the appropriate row from the snmpTargetParamsTable.

   - To modify a trap destination, change the IP address and or port number for the appropriate row in the snmpTargetAddrTable for an existing trap destination. The same effect can be achieved by removing a row and adding a new row.
   - To disable a trap destination, change TagList on the appropriate row in the snmpTargetAddrTable to the empty string.
   - To enable a trap destination, change TagList on the appropriate row in the snmpTargetAddrTable to "AC_TRAP".

- **Using Web Interface:** The Trap Destination table appears in the 'SNMP Trap Destinations' page, accessed from the 'Management Setting' page. The check box on the left indicates if the row is used. The three columns are used to set IP address, port and enable trap sending. The SNMPv3 Settings table, also accessed from the 'Management Setting' page is used for setting trap users.
  - To add a trap user: In the field near the **Add Index** button, enter the index of the row you want to add (0 to 9), and then click the button. The row is now available for configuration. The five columns include name, authentication protocol, privacy protocol, authentication key and privacy key. After configuring the columns, click **Apply**.
  - To delete a row: Select the corresponding index field, and then click **Delete**.

3.2.18.1.4 Configuring Trap Destination Port

For configuring the trap destination port, refer to trap managers, above.

3.2.18.1.5 Configuring Trusted Managers

The configuration of trusted managers determines which managers can access the device. You can define up to five trusted managers.

Notes:

- 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.
- Trusted managers are therefore, not supported in SNMPv3 – thus they apply only when the device is set to use SNMPv2c.
- If trusted managers are defined, then all community strings work from all trusted managers. That is, there is no way to associate a community string with particular trusted managers.

The configuration can be done via *ini* file, SNMP and Web.

- **Using *ini* file:** SNMPTRUSTEDMGR_x = <IP address>, where *x* is the entry index 0 to 4.
- **Using SNMP:** To configure Trusted Managers, the EM must use the SNMP-COMMUNITY-MIB, snmpCommunityMIB, and snmpTargetMIB.

➢ **To add the first Trusted Manager:**

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: 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.
4. Add a row to the snmpTargetAddrTable with these values:
   - Name=mgrN, where N is an unused number between 0 and 4.
   - TagList=MGR
   - Params=v2cparams
5. 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): 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: 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.

6. Set the value of the TransportTag field on each row in the snmpCommunityTable to the empty string.
7. 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.

- **Using Web** – Under the 'Management' tab select 'Management Settings' in the 'Management Settings' menu. In the main display press the 'SNMP Trusted Managers' arrow. The Web now displays the table. Use the 'Submit' button for applying your configuration. Use the check boxes for deleting.

3.2.18.2 Familiarizing Yourself with AudioCodes MIBs

AudioCodes' proprietary MIBs are located in the AudioCodes subtree (OID 1.3.6.1.4.1.5003). A classification within the subtree separates the MIBs according to the following:

- Configuration and status MIBs – in the acBoardMibs subtree
- Performance monitoring MIBs – in the acPerformance subtree
- Proprietary Carrier Grade Alarm MIB – in the acFault subtree

In the acBoardsMibs and acPerformance subtrees, the different MIB modules are grouped according to different virtual modules of AudioCodes' devices. In general, the division is as follows (a more detailed breakdown of the MIBs is discussed below):

- **acBoardMibs subtrees:**
  - **acBoard MIB:** proprietary traps.
  - **acMedia MIB:** DSP and media related objects. This MIB includes the configuration and status of DSP, voice, modem, fax, RTP/RTCP related objects. This MIB is relevant to all devices.
  - **acControl MIB:** mostly MEGACO and MGCP CP related objects. A number of objects are also related to SIP. The MIB is divided into subtrees that are common to both MEGACO and MGCP and subtrees that are specific to the different CPs. This MIB is relevant to all devices.
  - **acAnalog MIB:** all objects in this MIB are related only to the configuration, status and line testing or resetting of analog interfaces. This MIB is relevant to devices with analog interfaces only.
  - **acPSTN MIB:** configuration and status of trunk related objects only. Most of the MIB objects are trunk specific. This MIB is relevant to devices with digital PSTN interfaces only.
  - **acSystem MIB:** configuration and status of a wide range of general objects along with chassis related objects and a variety of actions that can be instigated. The MIB is relevant to all devices.
  - **acSS7 MIB:** configuration and status of SS7 related objects only. This MIB is relevant to Mediant 2000 and Mediant 3000.
  - **acV5 MIB:** configuration and status of v5.2 related objects only. This MIB is relevant to Mediant 3000/TP-6310.
- **acPerformance subtrees:**
  - acPMMedia, acPMControl, acPMAnalog, acPMPSTN, acPMSystem, and acPMSS7: module specific parameters performance monitoring MIBs
  - acPMMediaServer MIB: performance monitoring specifically for MediaServer related parameters (IVR, BCT, Conference and Trunk-Testing)
- **acFault subtree:** only one MIB exists – the acAlarm which is a proprietary simplification of the standard notificationLogMIB and alarmMIB (both are also supported).

The structure of the different MIBs is similar, depending on the subtree in which they reside. The MIBs in the acBoardMibs subtree have a very similar structure (except the acBoard and acGateway MIBs). Each MIB can be made up of four major subtrees:

- **Configuration subtree:** mostly read-write objects, tables and scalars. The relevant module's configuration is done via these objects.
- **Status subtree:** read-only objects, tables and scalars. Module status is collected by these objects.
- **Action subtree:** read-write objects that are used to instigate actions on the device (such as reset, save configuration, and so on) and read-only objects used to receive the actions' results.
- **Chassis subtree (in acSystem MIB only):** read-write and read-only objects related to chassis control and management (this includes, fan trays, power supply modules, PSTN IF modules, etc').

The acBoard MIB contains some deprecated objects and current proprietary trap definitions.

The acGateway MIB contains only the configuration subtree which in return is divided into common, H323 subtrees. The H323 subtree is mostly deprecated or obsolete.

3.2.18.3 Performance Monitoring Overview

Performance monitoring (PM) is available for a Third-Party Performance Monitoring System through an SNMP interface and can be polled at any interval by an external poller or utility in the management server or other off device system.

This section describes AudioCodes proprietary performance measurements (PM) MIB.

The device's performance measurements are provided by several proprietary MIBs (located under the "acPerformance" subtree (see below for more detail on each of the MIBs):

- **acPMMedia:** for media (voice) related monitoring such as RTP and DSP.
- **acPMControl:** for Control Protocol related monitoring such as connections, commands.
- **acPMAnalog:** Analog channels off-hook state (applicable to devices with analog interfaces only)
- **acPMPSTN:** for PSTN related monitoring such as channel use, trunk utilization.
- **cPMSystem:** for general (system related) monitoring.
- **acPMSS7:** for SS7 specific monitoring.
- **acPMMediaServer:** for Media Server specific monitoring. (Applicable to the 3000/6310/8410 devices)

Performance Monitoring MIBs have a fixed format. They all have an identical structure consisting of two major subtrees:

- **Configuration subtree:** allows configuration of general attributes of the MIB and specific attributes of the monitored objects.
- **Data subtree:** this is where the monitored information is found.

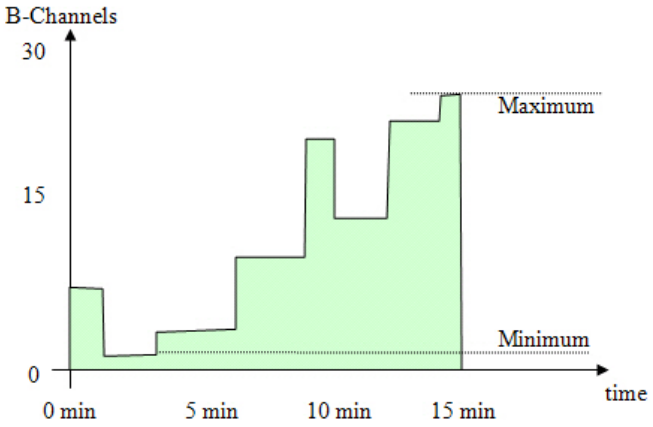
The information supplied by the device is divided into time intervals (default is 15 minutes). These intervals are used as a key in the tables. Thus, the monitoring results are presented in tables. There are one or two indices in each table. If there are two, the first is a sub-set in the table (e.g., 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).

Some of the PM parameters support a history with more than two intervals. These include the MEGACO parameters, IVR requests, IVR-play-collect, IVR-play-record, BCT contexts, conference calls, trunk-test calls and digit-collect requests.

Note: The interval's start time is synchronized with the device's clock so that they begin on the hour. If you are using NTP, then it is likely that the last interval within the first hour after device start up will be cut short to accommodate for this synchronization.

Following is a graphic example of one monitored parameter, in this case the number of utilized B-channels in a single trunk:

Figure 1: One Monitored Parameter Example



The x-axis is the time within the interval. The y-axis is the number of used channels. The parameter's value is a gauge. While the interval index is 0 (thus it is the current interval, any GET on the parameter value will return y-axis value for the graph at that moment in time. When the interval is over (index 1 or 2) the value is no longer relevant but there are other attributes such as the average – in this case the area in green divided by the interval length in seconds.

The configuration subtree includes:

- **Reset Total Counters:** resets the 'total' (see below) objects in all the MIB's tables if they are defined.
- **Attributes subtrees:** a number of subtrees in which scalars are used to configure the high and low thresholds for relevant tables.

The Data subtree consists of monitored data and statistics:

- **Time From Start Of Interval object:** GETs the time in seconds from the beginning of the current interval.
- **Data tables:** all have similar structure. Not all possible columns appear in all of them. The specific structure of a table (i.e. what columns are defined) is parameter specific. The only column that always appears is the interval column. The information in each column is a statistical attribute of the parameter being looked at.

Note: When an attribute value is -1, it means that the attribute isn't relevant at that point of time.

The columns are:

- Table specific index – table key.
- Interval – index, 0,1,2 – table key.
- Val – value of gauge or counter. This is the snapshot view of current device activity.
  - Counter – cumulative, only increases in value.
  - Gauge – fluctuates in value, value increases and decreases.
- Average – within the period length.
- Max – gauge high water mark.
- Min - gauge low water mark.

- Volume – number of times gauge or counter was updated, indicating the volume of change. For example:
  - For a trunk utilization element, the volume indicates how many calls were made and released.
  - For the Ethernet connection status element, the volume indicates how many network connections and disconnections occurred.
- TimeBelowLowThreshod – Percent of interval time for which the gauge is below the determined low threshold.
- TimeAboveHighThreshod – Percent of interval time for which the gauge is above the determined high threshold.
- TimeBetweenThresholds – Percent of interval time for which the gauge is between thresholds.
- FullDayAverage – 24 hour average.
- Total – relevant when using counters. Sums all counter values so far. It resets only once every 24 hours.
- StateChanges – the number of times a state (mostly active/non-active) was toggled.

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 returns to under the threshold. The 'source' varbind in the trap indicates the object for which the threshold is being crossed.

Expansions for the different MIBs.

- **acPMMedia:** Consists of data related to voice, DSPs coders etc. This MIB includes the following parameters:
  - Number of active DSP channels
  - Channels used for each coder
  - Discarded packets in robust RTP filter
  - Media Networking subtree - an array of packet behavior parameters such as delay, jitter, transmitted/received and lost RTP bytes and packets.
  - Media Networking Aggregated subtree - displays similar data only for the entire device and includes TDM-IP and IP-IP calls.
  - Channel Utilization subtree - parameters regarding channel use by fax, modem, TDM-IP calls, RTP, SRTP, multicast source and modem relay.
  - Streaming Cache subtree - hit count, miss count and server request count.
- **acPMControl:** Control Protocol related monitoring is divided into three groups – MEGACO, MGCP and SIP. The MIB includes the following parameters:
  - CP Connection subtree – general for all three control protocols. Its parameters include connection lifetime/state, counters for commands, retransmissions, active contexts, command success/failure and process time, transaction processing time and call attempts.
  - The remaining three subtrees are self explanatory and are CP specific.
- **acPMAnalog:** Analog channels statistics - one table only (offhook state).
- **acPMPSTN:** All statistics in this MIB are per trunk:
  - Number of active channels.
  - Trunk activity.
  - Number of channels that are in/out of service and in maintenance.
- **acPMSystem:** This detailed MIB is for general (system related) monitoring:
  - IP connection.
  - Discarded UDP packets due to unknown port.

- System Net Utils subtree – transmitted/received bytes/packets, discarded packets.
- System Network subtree – DHCP response time/request count. STUN related statistics.
- System Sctp subtree – SCTP sent/received/retransmitted bytes, retransmission attempts.
- IPsec security associations.
- System Multicast subtree – multicast IP packets received, multicast IP packets conveying UDP payload packets received/rejected, IGMP packets/general-queries/specific-queries received, IGMP membership-report/leave-group sent messages.
- System Congestion subtree – congestion state for general resources, DSP resources, IP resources, conference resources. (ATM resources table is obsolete).
- System NFS subtree – NFS related parameters.
- System MSBG subtree – includes received good/bad octets, received undersized/oversized/discarded packets, received MAC errors, received FSC error packets, transmitted octets/packets/collisions/late-packets.

- **acPMSS7:** The SS7 related data is divided into four subtrees:
  - SS7 Links Subtree – transmitted/received MSU/LSSU/FISU/Signal-units/octets, MTP2 no acknowledge received, Discarded MS, in/out of service.
  - SS7 Link Sets subtree – out of service state.
  - SS7 Signaling Nodes subtree – transmitted/received MTP3 octets/MSU/UPU, MTP3 MSU discarded/discarded-due-to-routing-data-error, MTP3 TFC messages.
  - SS7 MTP2 Timers subtree - expiry of timers 1 to 7.
- **acPMMediaServer:** (Applicable to the 3000/6310/8410 devices) The Media Server related data is divided into four subtrees:
  - IVR subtree – play requests, play progress/duration/collect/collect-in-progress/collect-duration/record/record-in-progress/record-duration, digit-collect requests, digit-collect in-progress/duration.
  - BCT subtree – BCT contexts, BCT in-progress/duration.
  - Conference subtree – conference calls, conference in-progress/duration.
  - Trunk Test subtree – trunk test requested, trunk tests in-progress/duration.

3.2.18.4 Traps and Alarms

AudioCodes supports standard traps and proprietary traps. Most of the proprietary traps are alarm traps, that is, they can be raised and cleared. Thus, they are referred to as *alarm traps*. All the standard traps are non-alarm traps, referred to as *log traps*. The complete list of all supported traps is mentioned in previous subsections.

The proprietary traps are defined under the acBoardTrapDefinitions subtree.

The standard MIB traps supported include the following:

- coldStart
- authenticationFailure
- linkDown
- linkup
- dsx1LineStatusChange
- rtcpXrVoipThresholdViolation
- dsx3LineStatusChange

- entConfigChange

This subsection describes the device's configuration so that traps are sent out to user-defined managers under SNMPv2c or SNMPv3. It continues with an explanation on the 'carrier grade alarm' abilities and usage.

3.2.18.4.1 Device Configuration

For a device to send out traps to specified managers the most basic configuration are the trap targets. More advanced configuration includes the Trap Community String or traps over SNMPv3.

- Destination IP address and port (refer to Basic SNMP Configuration Setup)
- Trap Community String: The default Trap Community String is 'trapuser'. There is only 1 for the entire device. It can be configured via *ini* file, SNMP or Web:
  - INI file: SNMPTRAPCOMMUNITYSTRING = <your community string here>.
  - SNMP: add a new community string to the snmpCommunityTable. To associate the traps to the new Community String change the snmpTargetParamsSecurityName in the snmpTargetParamsTable so it coincides with the snmpCommunitySecurityName object. If you wish, you can remove the older Trap Community String from snmpCommunityTable (however, it is not mandatory).
  - Web: under the 'Management' tab, choose 'Management Settings' in the 'Management Settings' menu. On the page, click the **SNMP Community String** arrow to display the table. Use the **Submit** button to apply your configuration. You can't delete the Trap Community String, only modify its value.
- SNMPv3 Settings: When using SNMPv3 settings it is important to note that by default the trap configuration remains such that the traps are sent out in SNMPv2c mode. To have traps sent out in SNMPv3, you can use either *ini* file or SNMP:
  - INI file: amongst the SNMPv3 users ensure that you also define a trap user (the value of 2 in the SNMPUsersGroup indicates the trap user). For example: you can have the SNMP users table defined with a read-write user, 'rwmd5des' with MD5 authentication and DES privacy, along with a trap user, 'tmd5no' with SHA authentication and DES privacy:

```
[ SNMPUsers ]
FORMAT SNMPUsers_Index = SNMPUsers_Username,
SNMPUsers_AuthProtocol, SNMPUsers_PrivProtocol, SNMPUsers_AuthKey,
SNMPUsers_PrivKey, SNMPUsers_Group;
SNMPUsers 1 = rwmd5des, 1, 1, myauthkey, myprivkey, 1;
SNMPUsers 2 = tshades, 2, 1, myauthkey, myprivkey, 2
[ \SNMPUsers ]
```

Notes:

- If you define a trap user only, the device runs in SNMPv3 mode but will not be accessible as there are no defined read-write or even read-only users.
- If you define non-default community strings (SNMPv2c), you need to access the device via SNMPv2c.

Along with this configuration, you also need to associate the trap targets (managers) with the user:

```
SNMPMANAGERTRAPUSER_x=tshades
```

where *x* is the target index and can be between 0 and 4.

Any targets that are defined in the *ini* file where this last parameter isn't defined, receives SNMPv2c traps.

- SNMP: change snmpTargetAddrParams object to the user of your choice adding the letters 'usm' as prefix (ensure it's a trap user). For example, the 'tshades' user should be added as 'usmtshades'.

3.2.18.4.2 Carrier Grade Alarm (CGA)

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 a manager to determine which alarms are currently active in the device. That is, the device maintains an active alarm table.
- The device allows a manager to detect lost alarms and clear notifications (sequence number in trap, current sequence number MIB object).
- The device allows a manager 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 manager 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 history and current active alarm information.

As part of CGA, the device supports the following:

- **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 proprietary AcAlarm MIB (this is a simple, one-row per alarm table that is easy to view with a MIB browser)
  - alarmActiveTable and alarmActiveVariableTable in the IETF standard AcAlarm MIB (rooted in the MIB tree)
- **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 raised or cleared traps. Two views of the alarm history table are supported by the agent:
  - acAlarmHistoryTable in the proprietary AcAlarm MIB (this is a simple, one-row per alarm table that is easy to view with a MIB browser)
  - nlmLogTable and nlmLogVariableTable in the standard NOTIFICATION-LOG-MIB

3.3 Voice Menu

Note: The 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.

Note: The Voice Menu will be automatically disabled as soon as the administrator password is set to a value other than the default "Admin".

➢ **To configure networking parameters for the device:**

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 26. The Voice Menu can be used to specify the configuration URL.

➢ **To set a configuration URL:**

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.
6. 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:

Configuration Parameters

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

3.4 INI File-Based Management

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

- System Parameters (refer to 'System Parameters' on page 165)
- Daylight Saving Parameters (refer to
- PSTN Parameters (refer to "PSTN Parameters" on page 210)
- Infrastructure Parameters (refer to "Infrastructure Parameters" on page 178)
- Media Processing Parameters (refer to 'Media Processing Parameters' on page 192)
- Control Protocols Parameters (refer to 'Control Protocol Parameters' on page 232)
- MGCP Specific Parameters (refer to "MGCP Specific Parameters" on page 248)
- MEGACO Specific Parameters (refer to "MEGACO Specific Parameters" on page 253)
- SNMP Parameters (refer to "SNMP Parameters" on page 260)
- Web Interface Parameters (refer to "Web Interface Parameters" on page 256)
- Voice Streaming Parameters (refer to "Voice Streaming Parameters" on page 263)
- SCTP Parameters (refer to "SCTP Parameters" on page 266)
- SS7 Parameters (refer to 'ini File Table Parameters' on page 361)
- Advanced Audio Server Parameters (refer to "Advanced Audio Server Parameters" on page 268)
- Routing Parameters (refer to
- IPsec Parameters (refer to 'IPsec Parameters' on page 242)
- SRTP Parameters (refer to SRTP Parameters)
- NFS Parameters (refer to 'NFS Parameters' on page 243)
- Analog Parameters (refer to 'Analog Parameters' on page 227) (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.

3.4.1.1 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.

System Parameters - ALL

Parameter Name	Description	Default	Range
[ActivityListToLog] Web: Activity Types to Report via Activity Log Messages	Defines the Activity Log mechanism of the device, which sends log messages to a Syslog server for reporting certain types of Web operations according to the below user-defined filters. ▪ **[pvc]** Parameters Value Change = Changes made on-the-fly to parameters. Note that the ini file parameter, EnableParametersMonitoring can also be used to set this option, using values [0] (disable) or [1] (enable). ▪ **[afl]** Auxiliary Files Loading = Loading of auxiliary files. ▪ **[dr]** Device Reset = Reset of device via the 'Maintenance Actions page. **Note:** For this option to take effect, a device reset is required. ▪ **[fb]** Flash Memory Burning = Burning of files or parameters to flash (in 'Maintenance Actions page). ▪ **[swu]** Device Software Update = cmp file loading via the Software Upgrade Wizard. ▪ **[ard]** Access to Restricted Domains = Access to restricted domains, which include the following Web pages: ✓ (1) ini parameters (AdminPage) ✓ (2) General Security Settings ✓ (3) Configuration File ✓ (4) IP Security Proposal / IP Security Associations Tables	Empty string	Refer to: Supported activity codes in the Description column.

System Parameters - ALL

Parameter Name	Description	Default	Range
	✓ (5) Software Upgrade Key Status ✓ (6) Firewall Settings ✓ (7) Web & Telnet Access List ✓ (8) WEB User Accounts ▪ **[naa]** Non-Authorized Access = Attempt to access the Web interface with a false or empty user name or password. ▪ **[spc]** Sensitive Parameters Value Change = Changes made to sensitive parameters: ✓ (1) IP Address ✓ (2) Subnet Mask ✓ (3) Default Gateway IP Address ✓ (4) ActivityListToLog ▪ **[ll]** Login and Logout = Every login and logout attempt. For example: ActivityListToLog = 'pvc', 'afl', 'dr', 'fb', 'swu', 'ard', 'naa', 'spc' **Note:** For the ini file, values must be enclosed in single quotation marks.		
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 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.	0	0 to 2

System Parameters - ALL

Parameter Name	Description	Default	Range
[AUPDVerifyCertificates] EMS: AUPD Verify Certificates	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
[AutoUpdatePredefinedTime]	Schedules an automatic update to a predefined time of the day.	NULL	'HH:MM' (24-hour format)
[BehaviorUponRadiusTimeout] Web: Device Behavior Upon RADIUS Timeout	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
[DayLightSavingTimeEnable]	Determines whether to enable the time adjustment to daylight saving time while updating the time from the NTP server. 0 - Disable 1 - Enable	0	0 or 1
[DayLightSavingTimeEnd]	This parameter defines the date and time of ending day light time in current year. Format mm:dd:hh:mm	NULL	12 String Max Length

System Parameters - ALL

Parameter Name	Description	Default	Range
[DayLightSavingTimeOffset]	Refers to Daylight saving time offset in minutes. When the DayLightSavingTimeEnable parameter is enabled (set to 1) , the DayLightSavingTimeOffset parameter determines the offset size in minutes.	60	0 - 120
[DayLightSavingTimeStart]	This parameter defines the date and time of starting daylight time in the current year. Format mm:dd:hh:mm	NULL	12 String Max Length
[DefaultAccessLevel] Web: Default Access Level	This parameter defines the default access level for the device. Default value is 'Security Administrator' (= 200).	200	0 to 255
[DialPlanFileName] Web: Dial Plan File EMS: Dial Plan File Name	This parameter is used to indicate name of the file containing the Dial Plan.	NULL	See Descr.
[DialPlanFileUrl]	URL for downloading a Dial Plan file using the Automatic Update facility.	NULL	See Descr.
[DisableWebConfig] EMS: Disable WEB Config	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] Web: DNS Primary Server IP EMS: DNS Primary Server	This parameter defines the DNS primary server's IP address.	0.0.0.0	Legal IP address
[DNSSecServerIP] Web: DNS Secondary Server IP EMS: DNS Secondary Server	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

System Parameters - ALL

Parameter Name	Description	Default	Range
[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] Web: Enable STUN EMS: STUN Enable	This parameter is used to enable the STUN module, used for NAT traversal of UDP packets.	0	0 or 1
[EnableSyslog] Web: Enable Syslog EMS: Syslog Enable	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 = Disable 1 = Enable	0	0 or 1
[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
[IKEcertificateExtValidate] EMS: IKE Certificate Ext Validate	Enables or disables certificate extension checking for IKE. 0 = Disable 1 = Enable	0	0 or 1
[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
[IniFileURL]	This parameter provides a link to an **.ini* file to be downloaded from a remote server.	NULL	See Descr.

System Parameters - ALL

Parameter Name	Description	Default	Range
[InitialShellCommand]	A Command Shell command to be executed during initialization. Several commands can be entered (each separated by a semicolon).	NULL	-
[M3KHAEnabled]	Enables/disables the HA subsystem module on the TP-6310/TP-8410 running Mediant 3000. 0 = Disable 1 = Enable	1	0 or 1
[NATBindingDefaultTimeout] EMS: Binding Life Time	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] Web: NTP Server IP Address EMS: Server IP Address	Defines the IP address (in dotted-decimal notation or as an FQDN) of the NTP server. The advantage of using an FQDN is that multiple IP addresses can be resolved from the DNS server, providing NTP server redundancy.	0.0.0.0 (i.e., internal NTP client is disabled).	Legal IP address
Web: NTP Secondary Server IP **[NTPSecondaryServerIP]**	Defines a second NTP server's address as an FQDN or an IP address (in dotted-decimal notation). This NTP is used for redundancy; if the primary NTP server fails, then this NTP server is used.	0.0.0.0	
[NTPServerUTCOffset] Web: NTP UTC Offset EMS: UTC Offset	This parameter is used to define the NTP time to offset, in seconds.	0	-43200 to +43200 seconds
[NTPUpdateInterval] Web: NTP Update Interval EMS: Update Interval	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] EMS: OCSP Default Response	Determines default OCSP behavior when the server cannot be contacted. 0 = Reject peer certificate 1 = Allow peer certificate	0	0 or 1
[OcspEnable] EMS: OCSP Enable	Enables or disables certificate checking via OCSP. 0 – Disable 1 – Enable	0	0 or 1

System Parameters - ALL

Parameter Name	Description	Default	Range
[OcspSecondaryServerIP] EMS: OCSP Secondary Server IP	This parameter defines the OCSP secondary server's IP address.	NULL	Legal IP address
[OcspServerIP] EMS: OCSP Server IP	This parameter defines the OCSP server's IP address.	0.0.0.0	Legal IP address
[OcspServerPort] EMS: OCSP Server Port	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] Web: Local RADIUS Password Cache Mode	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] Web: Local RADIUS Password Cache Timeout	Expiry time [sec] of locally stored RADIUS password cache. -1 = No Expiry; 0 = No Cache	300 seconds	-1 or 0
[RGCONFFILEURL]	This parameter provides a link to a rg_conf file to be downloaded from a remote server.	NULL	255 String Max Length
[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.

System Parameters - ALL

Parameter Name	Description	Default	Range
[SSHAdminKey] EMS: SSH Admin Key	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.
SSHEnableLastLoginMessage	Enables or disables the Last-Login message in SSH sessions. 0 - Disable 1 - Enable	1	0 or 1
SSHMaxBinaryPacketSize	Configures the maximum packet size for SSH packets, in bytes.	35000	582 to 35000
SSHMaxLoginAttempts	Configure the number of allowed SSH incorrect login attempts.	3	1 to 3
SSHMaxPayloadSize	Configures the maximum uncompressed payload size for SSH packets, in bytes.	32768	550 to 32768
SSHMaxSessions	Configures the maximum allowed number of SSH sessions.	2	1 or 2
[SSHRequirePublicKey] EMS: SSH Require Public Key	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] Web: SSH Server Enable EMS: SSH Server Enable	Enables or disables the embedded SSH server. 0 = Disable; 1= Enable	0	0 or 1
[SSHServerPort] Web: SSH Server Port EMS: SSH Server Port	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. Maximum of 64 bytes.	0.0.0.0	String[64]
[STUNServerPrimaryIP] Web: STUN Server Primary IP EMS: Primary Server IP	Defines the primary STUN Server IP address.	0.0.0.0	Legal IP address
[STUNServerSecondaryIP] Web: STUN Server Secondary IP EMS: Secondary Server IP	Defines the secondary STUN server IP address.	0.0.0.0	Legal IP address

System Parameters - ALL

Parameter Name	Description	Default	Range
SyslogFacility	Determines the facility number at syslog messages. Possible values: 16 = local use 0 (local0) 17 = local use 1 (local1) .. 23 = local use 0 (local7)	16	16 to 23
[SyslogServerIP] Web: Syslog Server IP Address EMS: Syslog Server IP Address	This parameter defines the IP address in dotted format notation. e.g., 192.10.1.255	0.0.0.0	Legal IP address
[SyslogServerPort] Web: Syslog Server Port EMS: Syslog Server Port Number	Defines Port number of Syslog Server.	514	Legal Port Number
[TelnetServerEnable] Web: Embedded Telnet Server EMS: Server Enable	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] Web: Telnet Server Idle Timeout EMS: Server Idle Disconnect	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] Web: Telnet Server TCP Port EMS: Server Port	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	NULL	See Descr.

System Parameters - ALL

Parameter Name	Description	Default	Range
	Automatic Update facility.		
[TLSClientCipherString]	Cipher-suite selection string for TLS clients. This parameter complements HTTPSCipherString (which affects TLS servers). For possible values and additional details, refer to: http://www.openssl.org/docs/apps/ciphers.html	"ALL:!ADH"	255 String Max Length
[TLSPkeyFileUrl]	URL for downloading a TLS private key file using the Automatic Update facility.	NULL	See Descr.
TLSPkeySize	Defines the RSA key size (in bits) for newly-generated keys.	1024	512, 768, 1024, or 2048
[TLSRootFileUrl]	URL for downloading a TLS trusted root certificate file using the Automatic Update facility.	NULL	See Descr.
[TLSVersion] EMS: TLS Version Web: TLS Version	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) 1 = TLS 1.0 will always be used	0	0 or 1
[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.

System Parameters - 6310

Parameter Name	Description	Default	Range
[SerialBaudRate] EMS: Baud Rate	Determines the value of the RS-232 baud rate. The valid values are: 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200. **(Applicable to TP-6310, TP-8410, Mediant 3000, IPM-6310, IPM-8410 and IPmedia 3000.)**	115200	See Descr.

System Parameters - IPM

Parameter Name	Description	Default	Range
[APSSegmentsFileUrl]	Provides a link to an XML segments file, to be downloaded from a remote server. See the chapter 'Automatic Update Facility' for supported URL options.	Not applicable	Not applicable

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] EMS: Enable Performance Threshold Alarms	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

[SystemOperationStateChange Profile]	This parameter defines the System Operation State Change Profile. 0 = Disable 1 = Nortel AMS ATM Refer to the enumerator acSystemOperationStateChange Profile enum for the possible values.	0	Integer >0
[TrunkingToAnalogFunctionality Profile]	This parameter defines the Trunking to Analog Functionality Profile. 0 = Disable; 1 = Enable MelCAS/LoopStart/GroundStart to Analog Functionality Refer to the enumerator acTrunkingToAnalogFunctionality Profile enum for the possible values.	0	Integer >0

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.

3.4.1.2 Daylight Saving Parameters

The daylight saving time parameters are described in the table below.

Daylight Saving Parameters

Parameter Name	Description	Default	Range
Web: Day Light Saving Time **EMS: Mode** **[DayLightSavingTimeEnable]**	Enables daylight saving time. ▪ **[0]** Disable (default) ▪ **[1]** Enable	0	0 - 1
Web: Start Time or Day of Month Start EMS: Start [**DayLightSavingTimeStart**]	Defines the date and time when daylight saving begins. This value can be configured using any of the following formats: ▪ Day of year - mm:dd:hh:mm, where: ✓ ***mm*** denotes month ✓ ***dd*** denotes date of the month ✓ ***hh*** denotes hour ✓ ***mm*** denotes minutes For example, "05:01:08:00" denotes daylight saving starting from May 1 at 8 A.M. ▪ Day of month - mm:day/wk:hh:mm, where: ✓ ***mm*** denotes month (e.g., 04) ✓ ***day*** denotes day of week (e.g., FRI) ✓ ***wk*** denotes week of the month (e.g., 03) ✓ ***hh*** denotes hour (e.g., 23) ✓ ***mm*** denotes minutes (e.g., 10) For example, "04:FRI/03:23:00" denotes Friday, the third week of April, at 11 P.M. The 'wk' field can be 1-5, where 5 denotes the last occurrence of the specified day in the specified month. For example, "04:FRI/05:23:00" denotes the last Friday of April, at 11 P.M.		
Web: End Time or Day of Month End EMS: End **[DayLightSavingTimeEnd]**	Defines the date and time when daylight saving ends. For a description of the format of this value, see the DayLightSavingTimeStart parameter.		
Web/EMS: Offset **[DayLightSavingTimeOffset**]	Defines the daylight saving time offset (in minutes).	60	0 -120

3.4.1.3 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.

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 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	3	1 to 7 & 15
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
bspTimingModuleCfgTimingMode	Synchronizes the Gateway with one of the PSTN interfaces. Possible values:	0	0 to 2

Infrastructure Parameters

Parameter Name	Description	Default	Range
	0 - TM_Standalone_MODE Non-synchronized mode - each board or TPM is synchronized internally from one of the PSTN interfaces without using the SAT timing module. **1 - TM_External_MODE** External Timing mode - use the SAT trunks Centralized Line Timing to synchronize the Gateway with one of the PSTN interfaces. **2 - TM_LineSync_MODE** Distributed Line Timing mode - without using the SAT timing module.		
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

Infrastructure Parameters

Parameter Name	Description	Default	Range
EnableDetectRemoteMACChange	GARP)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) **Note:** If the Gateway is situated in a network subnet, which is connected to other gateways via a router that uses protocol VRRP for redundancy, this parameter must be set to 0 or 2.	2	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

Infrastructure Parameters

Parameter Name	Description	Default	Range
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 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	1	0 to 3
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

Infrastructure Parameters

Parameter Name	Description	Default	Range
EnableNetworkPhysicalSeparation	Enables Network Physical Separation in Supported Hardware. Allows the user to have separate port for each Network. Requires suitable hardware. 0 = Disabled; 1 = Enabled Note: In order for this parameter to take effect, a device reset is required.	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 **Note:** In order for this parameter to take effect, a device reset is required.	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, etc. This is not available in DHCP.	0	0 or 1

Infrastructure Parameters

Parameter Name	Description	Default	Range
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
ICMPUnreachableReport Interval	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
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
StaticRouteTable_Prefix Length	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
StaticRouteTable_Destination	Comprises the Destination column of the static routing rules that users can add to.	NULL	Legal IP address

Infrastructure Parameters

Parameter Name	Description	Default	Range
StaticRouteTable_Gateway	Comprises the gateways column of the static routing rules that users can add.	NULL	Legal IP address
StaticRouteTable_Interface Name	Comprises the interface name of the static routing rules that users can add.	Unknown	String [17]
StaticRouteTable_Description	Comprises a description of the static route	“”	String [30]
SctpIPAddress	Defines the source IP address for the SCTP traffic. This parameter may only be used when working in a single interface scenario. When working with Multiple Interfaces, the OAMP IP Address or a CONTROL IP address is used (according to the EnableSCTPasControl parameter). The valid value is an IP address in dotted-decimal notation (xxx.xxx.xxx.xxx). The default is 0.0.0.0 (i.e., the main source IP is used).	0.0.0.0	xxx.xxx.xxx.xxx
StreamingCacheDecision Interval	Defines the streaming cache decision interval in minutes. If -1 is set, decision will be made upon each cache request.	4	-1 to 0xFFFF
StreamingCacheNumOf Descriptors	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
TMLoopBackExternalRef1	Enables loopback state on Reference Interface 1, by connecting Rx path towards Tx path. 0 - Disable 1 - Enable	0	0 or 1
TMLoopBackExternalRef2	Enables loopback state on Reference Interface 2, by connecting Rx path towards Tx path. 0 - Disable 1 - Enable	0	0 or 1

Infrastructure Parameters

Parameter Name	Description	Default	Range
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 TPNCP host.	0	0 or 1
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
WANIPAddress	Sets the WAN address to be used by VoIP signaling applications.	0.0.0.0	IPv4 address in dotted decimal notation xxx.xxx.xxx.xxx

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

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

Infrastructure Parameters – MediaPack & Mediant 1000

Parameter Name	Description	Default	Range
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 = Normal Serial ▪ 1 = Serial SMDI ▪ 2 = Ericsson flavor of SMDI ▪ 3 = NEC Ics flavor of SMDI	0	0 to 3
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

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. 0 - Feature is disabled. Do not Disable any H.100 family clocks on trunk failure. 1 - Disable NETREF 2 - Disable H.100 Clocks, Master A/B 3 - Disable both NETREF and H.100	0	0 to 3
EnableBitTask	Enables the bit task. 0 = Disabled; 1 = Enabled	1	0 or 1
EnableDNSasOAM	Sets location of the DNS. If the parameter is set and the device is functioning in multiple IPs mode, DNS is on the 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, 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.**	0	0 or 1
EnableNTPasOAM	Sets the location of the Network Time Protocol (NTP). If this parameter is set and the device is functioning in multiple IPs mode, the NTP is located on the OAM network. If not, the NTP is located on the control network. 0 = Disable 1 = Enable	1	0 or 1

Infrastructure Parameters – TP

Parameter Name	Description	Default	Range
EnableSCTPasControl	Defines the location of the SCTP (Stream Control Transmission Protocol). If this parameter is set and the device is operating in multiple IPs mode, the SCTP is located on Control network. If not,the SCTP is located on OAMP network. 0 = Disable; 1 = Enable Note: In order for this parameter to take effect, a device reset is required.	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
EnableVoicePathBIT Test	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
LocalOAMDefaultGW	Sets the Default gateway for Operation, Administration, Management and Provisioning (OAMP) interface when operating in a single interface scenario without a Networking Interface Table.	0.0.0.0	Legal IP address in subnet
LocalOAMIPAddress	Sets the IP address of the OAMP interface when operating in a single interface scenario without a Networking Interface Table.	0.0.0.0	Legal IP address
LocalOAMSubnetMask	Sets Subnet Mask for the OAMP interface, when operating in a single interface scenario without a Networking Interface Table.	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

Infrastructure Parameters – TP

Parameter Name	Description	Default	Range
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
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 For 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

Infrastructure Parameters – TP

Parameter Name	Description	Default	Range
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
TDMBusLocal Reference	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)
TDMBusmasterSlave Selection	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
TDMBusNetref OUTPUTMODE	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 • 2 = 2.048 MHz	0	0 to 2
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
TDMBusOutputStarting Channel	Defines the outgoing TDM Timeslot for device's channel #0. The remaining channels are organized sequentially.	0	0 to 127

Infrastructure Parameters – TP

Parameter Name	Description	Default	Range
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	Mediant 1000, TP-6310, TP-8410 use 2	0, 1, 2, & 4 to 8
VLANBRONZESERVICECLASSPRIORITY	Sets the priority for the Bronze service class content.	2	0 to 7
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
VLANMODE	Sets the VLAN functionality. 0 = Disable 1 = Enable **Notes:** ▪ In order for this parameter to take effect, a device reset is required. ▪ In order to work with more than a single network interface, VLANs must be activated.	0	0 or 1
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

Infrastructure Parameters – TP

Parameter Name	Description	Default	Range
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
WanInterfaceName	Sets the WAN interface name to be used by VOIP signalling applications. Refer to CLI documentation for interface name formats.	""	See Descr.

3.4.1.4 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.

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
AMDSensitivityLevel	Determines the AMD (Answer Machine Detector) level of detection sensitivity. It has 16 levels of sensitivity : 0 = Best detection of an answering machine 15 = Best detection of a live call	8	0 to 15

Media Processing Parameters

AMDSensitivityParameterSuit	Determines the serial number of the AMD parameter suit: 0 = USA Parameter Suit with normal detection sensitivity resolution (8 sensitivity levels) 1 = USA Parameter Suit with high detection sensitivity resolution (16 sensitivity levels) 2-7 : other countries parameter suits with up to 16 sensitivity levels.	0	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 packet 2 = Redundancy depth of 2 packets 3 = Redundancy depth of 3 packets	0	0 to 3
AriaProtocolSupport	Enables or disables the Aria encryption protocol. Enabling this parameter might reduce the board channel capacity. 0 = Disable 1 = Enable	0	0 or 1
Web: AMR Payload Format [AmrOctetAlignedEnable]	Defines the AMR payload format type. ▪ **[0]** Bandwidth Efficient ▪ **[1]** Octet Aligned (default) If octet-align is absent both from the local and remote SDP, the response takes the value from the configuration. The octet-align MUST be symmetric. Therefore, if the remote SDP exists, the local side is set according to it.	1	0 or 1

Media Processing Parameters

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
BrokenConnectionEventActivation Mode	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 standard. • 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. 0 = Disable 1 = Enable	1	0 or 1
CallProgressTonesFilename	Defines Call Progress Tone filenames (downloaded by TFTP).	Null	

Media Processing Parameters

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 different channel capacity and different features supported.	0	0 to 255
DTMFDetectorEnable	Enables or disables detection of DTMF signaling. 0 = Disable	1	0 or 1

Media Processing Parameters

	1 = Enable		
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
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
EnableFaxModemInbandNetwork	Enables or disables in-band network	0	0 to 1

Media Processing Parameters

Detection	detection related to fax/modem. ▪ [0] = Disable (default) ▪ [1] = Enable When this parameter is enabled on Bypass and transparent with events mode (VxxTransportType = 2 or 3), 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. This can be useful when, for example, the payload of voice and bypass is the same, allowing the originator to switch to bypass mode as well.		
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
EnableSidWithNonePayloadType 13	For coders that don't support the SID in their main standard (such as G.711), it defines whether or not RTP packets without Payload Type 13 as SID are supported. ▪ 0 = Not supported ▪ 1 = Supported	1	0 or 1
EnableSilenceCompression	Enables or disables Silence Suppression Mode. ▪ 0 = Disable = SILENCE_COMPRESION_DISABLE ▪ 1 = Enable =	0	0 to 2

Media Processing Parameters

	SILENCE_COMPRESION_ENABLE ▪ 2 = Enable without adaptation = SILENCE_COMPRESION_ENABLE_NOISE_ADAPTATION_DISABLE		
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
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	0 to 3

Media Processing Parameters

	• 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)		
FaxModemBypassCoderType	Users can use this parameter to set the fax/modem bypass coder (according to acTCoders). 0 = G.711 A-Law 1 = G.711 Mu-Law	0	0 or 1
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 dBm to -3.5 dBm in steps of 1 dBm.	-12	-18 to -3
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 • 6 = 16800 bps • 7 = 19200 bps • 8 = 21600 bps • 9 = 24000 bps • 10 = 26400 bps • 11 = 28800 bps • 12 = 31200 bps • 13 = 33600 bps	13	0 to 13
FaxRelayRedundancyDepth	Determines the depth of redundancy for fax packets. This parameter is applicable only to non-V.21 packets.	0	0 to 2

Media Processing Parameters

	▪ 0 = No redundancy ▪ 1 = 1 packet redundancy ▪ 2 = 2 packet redundancy		
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 • 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

Media Processing Parameters

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 • 15 = G729EV_RATE_UNDEFINED	0	0 to 11, 15
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	-	See Descr.

Media Processing Parameters

	relevant when using the PSTN interface with CAS protocols. Range = 0x0 to 0xF		
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.
MFSS5DetectorEnable	Enables or disables detection of MF SS5 line signaling. ▪ 0 = Disable ▪ 1 = Enable	0	0 or 1

Media Processing Parameters

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
NoiseReductionActivation Direction	Defines Noise Reduction activation direction: • 0 = from TDM side • 1 = from Network side	0	0 to 1
NoiseReductionIntensity	Defines Noise Reduction ntensity: • 0 - Weakest • 8 – Normal • 15 - Strongest	8	See Descr.
NoOpEnable	Enables / disables the No-op packets sending mode. • 0 = Disable • 1 = Enable	0	0 to 1
NoOpInterval	Sets the No-op packets sending interval. Parameter value in milliseconds default value - 10 sec (10000 msec) Range = 20 to 600000 (20 msec - 10 min - 10 min = 600000)	10000	20 to 600000

Media Processing Parameters

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 Payload type of sent packets.	96	96 to 127

Media Processing Parameters

RTPNOOPPayloadType	User can modify the Noop packets RTP Payload type by setting this parameter.	120	96 to 127
RTPRedundancyDepth	Enables the device to generate RFC 2198 redundant packets. This can be used for packet loss where the missing information (audio) can be reconstructed at the receiver's end from the redundant data that arrives in subsequent packets. This is required, for example, in wireless networks where a high percentage (up to 50%) of packet loss can be experienced. ▪ **[0]** = (Default) Disable. ▪ **[1]** = Enable - previous voice payload packet is added to current packet. ▪ **[2]** = Previous two voice payload packets are added to the current packet. ▪ **[3]** = Previous three voice payload packets are added to the current packet. ▪ **[4]** = Previous four voice payload packets are added to the current packet. ▪ **[5]** = Previous five voice payload packets are added to the current packet. To use this RTP redundancy level, you need to set the DSPVersionTemplateNumber parameter to 4 or 7. The coders that support this redundancy level include: ✓ DSPVersionTemplateNumber = 4 (all coders) ✓ DSPVersionTemplateNumber = 7 (only G.729 and iLBC coders) **Notes:** When enabled, you can configure the payload type, using the RFC2198PayloadType parameter. The RTP redundancy dynamic payload type can be included in the SDP, by using the EnableRTPRedundancyNegotiation parameter. This parameter can also be configured in an IP Profile.	0	
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

Media Processing Parameters

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 device's transferring method of TTY signals during a call. • 0 = Disable (default) • 1 = Bypass - entering HBR on TTY tone detection. The HBR coder is selected by the FaxModemBypassCoderType parameter. The network packets that are generated and received during the bypass period are regular voice RTP packets (per the selected bypass coder), but with a different RTP payload type (103). You can change this payload type by using the ModemBypassPayloadType parameter. When TTY signal transmission ends, reverse switching from bypass coder to regular voice coder is performed. • 2 = Relay (signals sent over the EVRC/B or AMR codec) - TTY phone device transfer using In-Band Relay mode for TTY signal transport. The following TTY relay standards are supported: ✓ - 3GPP2 C.S0014-0-3 and 3GPP2C.s0028-0 v2.0 over EVRC and EVRC B. ✓ - Text over Cellular Text Modem (CTM) over an existing speech path when using AMR or GSM EFR	0	0 or 2

Media Processing Parameters

	coders. **Note:** To support TTY Relay (2), you must configure the device to use the EVRC/B, AMR, or GSM EFR coder.		
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 • 1 = Enable	0	0 or 1
V1501AllocationProfile	Selects the V.150.1 profile, determining how many DSP channels have V.150.1 support.	0	0 to 3
V1501SSERedundancyDepth	SSE is a part of V150.1 modem relay protocol and SSE messages are sent over RTP. SSE redundancy refers to the sending of SSE messages several times to increase reliability. This parameter determines the number of times each SSE message is to be resent.	3	1 - 6
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 • 4 = AnsMute	2	0 to 4
V34ModemTransportType	Sets the V.34 modem transport method. • 0 = Transparent • 2 = Bypass • 3 = Transparent with Events • 4 = AnsMute	2	0 to 4

Media Processing Parameters

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	Sets the voice quality monitoring (RTCP-XR) mode. • 0 = Disable • 1 = Enable All	0	0 or 1

3.4.1.4.1 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.

DSP Template Table

ini File Field Name	Description	Default Value	Valid 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%.

3.4.1.5 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.

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

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 Layer 2. Applicable in BRI trunks only. Point-to-point = 0; Point-to-Multipoint = 1	0	0 or 1
CASAddressingDelimiters	Determines if delimiters are added to the received address or received ANI digits string. • 0 = Disable • 1 = Enable When this parameter is enabled, delimiters such as '*', '#', and 'ST' are added to the received address or received ANI digits string. When this parameter is disabled, the address and ANI strings remain without delimiters.	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 the CAS protocol configuration files.	NULL	0 to 7

PSTN Parameters - TP

Parameter Name	Description	Default	Range
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 (Use the value from state machine)	0, 1
CasStateMachineDigitSignalingSystem	Defines which Signaling System to use MF or DTMF for detection & generation. 0 = DTMF; 1 = MF	-1 (Use the value from state machine)	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 (Use the value from state machine)	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 (Use the value from state machine)	See Descr.
CasStateMachineGenerateDigitOnTime	Overrides the CAS state machine global parameter 'Generate digit on-time' msec	-1 (Use the value from state machine)	Not applicable
CasStateMachineGenerateInterDigitTime	Overrides the CAS state machine global parameter 'Generate digit off-time' msec	-1 (Use the value from state machine)	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 (Use the value from state machine)	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 (Use the value from state machine)	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
CASChannelIndex	This parameter determines which CAS protocol file to use on each BChannel at the trunk. The index values	-	String with syntax as described in CAS Trunk

PSTN Parameters - TP

Parameter Name	Description	Default	Range
	corresponds to the number configured for the parameter CASFileName_X. Range = not greater than the parameter defining the PSTN CAS Table Num.		Parameters in 'CAS' on page 347.
CASTablesNum	This parameter defines the quantity of CAS 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	0	0 to 8
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

PSTN Parameters - TP

Parameter Name	Description	Default	Range
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
	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).		
DPNSSNumReal Channels	This parameter is relevant only to protocol ISDN DPNSS. Defines the number of real channels.	30	1 to 30
DPNSSNumVirtual Channels	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	See Descr.	0, 1, a, b, c, A, B, C, D, E, F

PSTN Parameters - TP

Parameter Name	Description	Default	Range
	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)		
ISDNDuplicateQ931BuffMode	Activates / de-activates delivery of raw Q.931 messages. For a detailed description, refer to 'ISDN Flexible Behavior' on page 317.	0	0 to 255
ISDNGeneralCCBehavior	This is the bit-field used to determine several general ISDN behavior options. For a detailed description, refer to 'ISDN Flexible Behavior' on page 317.	0	Not applicable

PSTN Parameters - TP

Parameter Name	Description	Default	Range
ISDNIBehavior	Bit-field used to determine several behavior options that influence the behaviour of the Q.931 protocol. ▪ **[0]** = Disable (default) ▪ **[1]** NO STATUS ON UNKNOWN IE = Q.931 Status message isn't sent if Q.931 received message contains an unknown/unrecognized IE. By default, the Status message is sent. Note: This value is applicable only to ISDN variants in which sending of Status message is optional. ▪ **[2]** NO STATUS ON INV OP IE = Q.931 Status message isn't sent if an optional IE with invalid content is received. By default, the Status message is sent. **Note:** This option is applicable only to ISDN variants in which sending of Status message is optional. ▪ **[4]** ACCEPT UNKNOWN FAC IE = Accepts unknown/unrecognized Facility IE. Otherwise, the Q.931 message that contains the unknown Facility IE is rejected (default). **Note:** This option is applicable only to ISDN variants where a complete ASN1 decoding is performed on Facility IE. ▪ **[128]** SEND USER CONNECT ACK = The Connect ACK message is sent in response to received Q.931 Connect; otherwise, the Connect ACK is not sent. **Note:** This option is applicable only to Euro ISDN User side outgoing calls. ▪ **[512]** EXPLICIT INTERFACE ID = Enables to configure T1 NFAS Interface ID (refer to the parameter ISDNNFASInterfaceID_x). **Note:** This value is applicable only to 4/5ESS, DMS, NI-2 and HKT variants. ▪ **[2048]** ALWAYS EXPLICIT = Always set the Channel Identification IE to explicit Interface ID, even if the B-channel is on the same trunk as the D-channel. **Note:** This value is applicable only to 4/5ESS, DMS and NI-2 variants.	0	Not applicable

PSTN Parameters - TP

Parameter Name	Description	Default	Range
	▪ **[32768]** ACCEPT MU LAW =Mu-Law is also accepted in ETSI. ▪ **[65536]** EXPLICIT PRES SCREENING = The calling party number (octet 3a) is always present even when presentation and screening are at their default. **Note:** This option is applicable only to ETSI, NI-2, and 5ESS. ▪ **[131072]** STATUS INCOMPATIBLE STATE = Clears the call on receipt of Q.931 Status with incompatible state. Otherwise, no action is taken (default). ▪ **[262144]** STATUS ERROR CAUSE = Clear call on receipt of Status according to cause value. ▪ **[524288]** ACCEPT A LAW =A-Law is also accepted in 5ESS. ▪ **[2097152]** RESTART INDICATION = Upon receipt of a Restart message, acEV_PSTN_RESTART_CONFIRM is generated. ▪ **[4194304]** FORCED RESTART = On data link (re)initialization, send RESTART if there is no call. ▪ **[67108864]** NS ACCEPT ANY CAUSE = Accept any Q.850 Cause IE from ISDN. **Note:** This option is applicable only to Euro ISDN. ▪ **[536870912]** Alcatel coding for redirect number and display name is accepted by the device. Note: This option is applicable only to QSIG (and relevant for specific Alcatel PBXs such as OXE). ▪ **[1073741824]** QSI ENCODE INTEGER = If this bit is set, INTEGER ASN.1 type is used in operator coding (compliant to new ECMA standards); otherwise, OBJECT IDENTIFIER ASN.1 type is used. **Note:** This option is applicable only to QSIG. ▪ **[2147483648]** 5ESS National Mode For Bch Maintenance = Use the National mode of AT&T 5ESS for B-channel maintenance. **Notes:** ▪ To configure the device to support		

PSTN Parameters - TP

Parameter Name	Description	Default	Range
	several ISDNIBehavior features, enter a summation of the individual feature values. For example, to support both [512] and [2048] features, set the parameter ISDNIBehavior is set to 2560 (i.e., 512 + 2048). ▪ When configuring in the Web interface, to select the options click the arrow button and then for each required option select 1 to enable.		
ISDNInCallsBehavior	This is the bit-field used to determine several behavior options that influence how the ISDN Stack INCOMING calls behave. For a detailed description, refer to 'ISDN Flexible Behavior' on page 317.	0	Not applicable
ISDNNFASInterfaceID	Defines the Interface ID. Works with NS_EXPLICIT_INTERFACE_ID. For a detailed description, refer to 'ISDN Flexible Behavior' on page 317.	(unsigned char)-1	0 to 255
ISDNNSBehaviour2	This is the Bit-field used to determine several behavior options, which influence how the Q.931 protocol behaves. For a detailed description, refer to 'ISDN Flexible Behavior' on page 317.	0	Not applicable
ISDNOutCallsBehavior	This is the bit-field used to determine several behavior options that influence how the ISDN Stack OUTGOING calls behave. For a detailed description, refer to 'ISDN Flexible Behavior' on page 317.	0	Not applicable
ISDNTimerT310	Defines the T310 override timer. Started / stopped on receipt of Q.931 messages. Valid values: ▪ 0 = 10 sec ▪ Any positive value (up to 600) = overrides default	0	0 to 600
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

PSTN Parameters - TP

Parameter Name	Description	Default	Range
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) ▪ 2 = Use HDB3 line code (for E1 trunks only)	0	0 to 2
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	0	1-23, 26, 28-31, & 34-40, 50-57

PSTN Parameters - TP

Parameter Name	Description	Default	Range
	▪ 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 ▪ BRI_EURO_ISDN = 50 ▪ BRI_QSIG = 54 • BRI_FRENCH_VN6_ISDN = 55 • BRI_NTT_ISDN = 56		
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

PSTN Parameters - TP

Parameter Name	Description	Default	Range
TDMBusPSTNAutoClock Enable	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
TDMBusPSTNAutoClock RevertingEnable	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
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

PSTN Parameters - TP

Parameter Name	Description	Default	Range
TraceLevel	▪ Defines the Trace level: NO_TRACE = 0 ▪ FULL_ISDN_TRACE = 1 ▪ LAYER3_ISDN_TRACE = 2 ▪ ONLY_ISDN_Q931_MSGS_ TRACE = 3 ▪ LAYER3_ISDN_TRACE_NO_ DUPLICATION = 4 ▪ FULL_ISDN_TRACE_WITH_ DUPLICATION = 5 ▪ ISDN_Q931_RAW_DATA_TRACE = 6 ▪ ISDN_Q921_RAW_DATA_TRACE = 7 ▪ ISDN_Q931_Q921_RAW_DATA_T RACE = 8 ▪ SS7_MTP2 = 10 ▪ SS7_MTP2_AND_APPLI = 11 ▪ SS7_MTP2_SL_L3_NO_MSU = 12 ▪ SS7_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
V5Indication	Indicates that V5 configuration is enabled. ▪ 0 = Disabled ▪ 1 = Enabled	0	0 to 1

PSTN Parameters - TP

Parameter Name	Description	Default	Range
V5NumberOfCChannels	Indicates the number of timeslots used as communication channels in a V5 trunk. ▪ 0 - No C-channel, the trunk is voice only. ▪ 1 - Timeslot 16 is used for the V5 signaling.	0	0 to 1
V5ProtocolSide	Indicates the V5 side of the V5 protocol served by the trunk. ▪ 0 - AN side ▪ 1 - LE side	1	0 to 1

3.4.1.5.1 PSTN SDH/SONET Parameters

PSTN Parameters

Parameter Name	Description	Default	Range
SDHFbrGrp_Mapping_Type	Determines the SDH/SONET mapping type (signal label and payload mapping type) for the PSTN interface. This is typically selected per Fiber Group. ▪ **[0]** VT1.5 Asynchronous = Asynchronous VT1.5 and DS1 (for OC-3). ▪ **[1]** TU-12 Asynchronous = Asynchronous TU12 and E1 (for STM-1). ▪ **[2]** TU-11 Byte Synchronous = VT1.5 Byte Synchronous mapping (for OC-3). ▪ **[3]** Asynchronous DS3 = Asynchronous mapping of DS3 in STS1, DS3 channelized to DS1's - asynchronous mapping of channelized DS3 to OC-3, so that the actual interface is OC-3 but mapped to three DS3 trunk interfaces (DS1 > DS3 > STS-1 > OC-3). ▪ **[15]** UNDEFINED = (Default) Not defined. **Notes:** For this parameter to take effect, a device reset is required. The setting of this parameter should be in coordination with the parameters SDHFbrGrp_SDHSONETMode and ProtocolType.	15	0, 1, 2, 3, 15

PSTN Parameters

Parameter Name	Description	Default	Range
	This parameter is applicable only when TDMBusType = acFRAMERS (2) and PSTNTransmissionType = Optical SONET or SDH Transmission type(1). When option [3] is selected, the clock source is automatically set to 'Local Board' (synchronization supplied by device) and cannot be changed. For more details regarding this parameter, see the SDHFbrGrp_SdhSonetMode parameter.		
SDHFbrGrp_SDHSONET Mode	Determines the SDH/SONET mode for the PSTN interface, typically per Fiber Group. ▪ 0 = Unknown ▪ 1 = STM1 ▪ 2 = OC3 Should be in coordination with other parameters as follows: - PSTNTransmissionType - ProtocolType When STM-1 mode is selected, set the SDHFbrGrp_Mapping_Type parameter to 1 (i.e., Asynchronous TU12 and E1) and ProtocolType for E1. When OC-3 mode is selected, set the SDHFbrGrp_Mapping_Type parameter to: ▪ 0 (Asynchronous VT1.5), ▪ 2 (Byte-synchronous VT1.5), ▪ or 3 (Asynchronous mapping of DS3 in STS1), ▪ and ProtocolType for DS1. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is relevant only when the TDMBusType parameter is set to acFRAMERS (2), PSTNTransmissionType is set to SONET/SDH (1), and SdhFbrGrp_SdhSonetMode is not set to UNKNOWN (0).	0	0 to 2

PSTN Parameters

Parameter Name	Description	Default	Range
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. Applicable only to the TP-6310, Mediant 3000 /6310, IPM-6310, IPmedia 3000/6310. • 0 = Not protected, APS not activated • 1 = Protected, APS activated	1	0, 1
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/TP-6310, IPM-6310, IPmedia 3000/IPM-6310. Applicable only when the SDHFbrGrp_Protected parameter is set to 1 (Protected).	0	0 to 1
SDHFbrGrp_APS_Revert Mode	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/TP-6310, IPM-6310, IPmedia 3000/IPM-6310. Applicable only when the SDHFbrGrp_Protected parameter is set to 1 (Protected).	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/TP-6310, IPM-6310, IPmedia 3000/IPM-6310 Applicable only when the SDHFbrGrp_APS_Revert Mode parameter is set to 1 (Revertive).	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
SonetSdhMediumCircuitIdentifier	SDH / SONET circuit name.	“”	NA

PSTN Parameters

Parameter Name	Description	Default	Range
SDHFbrGrp_KLM_Numbering_Scheme	The scheme for VC/VT numbering on STM-1/OC3. Once a scheme is selected, there is a mapping of 3 variables (K, L, M) that identify the VC/VT inside the STM-1/OC3 to a trunk number. The trunk number changes from 0 to 62 or from 0 to 83). ▪ K is the TUG3 number (SDH) or STS-1 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 PSTNTransmissionType=Optical SONET or SDH Transmission type(1).	0	0 to 2

3.4.1.5.2 Clock Timing Parameters

The Clock Timing parameters are described in the table below.

Clock Timing Parameters

Parameter	Description
Web: Timing Module Mode EMS: Mode **[TMMode]**	Determines the device's Timing Synchronization mode. ▪ [0] StandAlone = Standalone Synchronization mode - without using the SAT timing module (default). ▪ [1] External = External, where each blade is synchronized from one of the PSTN interfaces (without using the SAT timing module). This must be selected for the BITS Synchronization mode. ▪ [2] LineSync = Line Synchronization mode - synchronizes the device with one of the PSTN interfaces.
Web: External Interface Type EMS: External IF Type **[TMExternalIFType]**	Defines the external BITS reference transmission type for both primary and secondary interfaces. ▪ [0] E1 CRC4 (default) ▪ [1] E1 CAS ▪ [2] E1 FAS ▪ [3] T1 D4 ▪ [4] T1 ESF ▪ [5] T12
Web: T1 External Reference Transmit Line Build Out EMS: T1 Line Build Out **[TMT1LineBuildOut]**	Defines the transmission power between the timing module on the SAT blade and the T1 external reference clock (External Reference Transmit Line Build Out). ▪ [0] = DSX 1.0 to 133 feet 0 dB CSU ▪ [1] = DSX 1.133 to 266 feet (default) ▪ [2] = DSX 1.266 to 399_feet ▪ [3] = DSX 1.399 to 533 feet ▪ [4] = DSX 1.533 to 655 feet ▪ [7] = DSX 1.0 to 133 feet 0 dB CSU plus enable transmit and receive gapped clock Note: This parameter is applicable only when using BITS Synchronization (i.e., the parameter TMMode is set to 1).
Web: E1 External Reference Transmit Line Build Out EMS: E1 Line Build Out **[TME1LineBuildOut]**	Defines the transmission power (in ohm) between the timing module on the SAT blade and the E1 external reference clock. • [0] E 75 ohms normal • [1] E 120 ohms normal (default) • [4] E 75 ohms with high return loss • [5] E 120 ohms with high return loss • [6] E 75 ohms normal plus enable transmit and receive gapped clock • [7] E 120 ohms normal plus enable transmit and receive gapped clock • Note: This parameter is applicable only when using BITS Synchronization (i.e., the parameter TMMode is set to 1).

Clock Timing Parameters

Web: Reference Validation Time EMS: Validation Time **[TMReferenceValidationTime]**	Configures the Reference validation time. The valid range is 0 to 15 minutes. The default is 1 minute. The resolution is 1 minute. Note: This parameter is applicable only for External timing (i.e., BITS) and Line timing references (refer to the parameter TMMode).

3.4.1.6 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.

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	Defines 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 VMWI standard. ▪ 0 = Between_Rings ▪ 1 = Not_Ring_Related	0	0 to 1
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

Analog Parameters

Parameter Name	Description	Default	Range
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 Direct Access Arrangements (DAA). Possible values: • 0 = DC remover is disabled • 1 = DC remover is enabled	0	0 to 1
EnableOnHookActiveMode	Enables the Duslic active mode in onhook state. Possible values: 0 = Active mode is disabled (device in power down mode) 1 = Active mode 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
ExternalLifeLinePorts	Sets the number of FXS ports which will be connected to an external lifeline and will be disabled (maximum value is half of the number of FXS ports). **Note:** In order for this parameter to take effect, a device reset is required.	0	0 to 24

Analog Parameters

Parameter Name	Description	Default	Range
ETSIVMWITypeOneStandard	Selects the number denoting the ETSI VMWI Type 1 Standard. ▪ 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	0	0 to 6
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	15	0 to 15

Analog Parameters

Parameter Name	Description	Default	Range
	▪ 5 = CPT & CurrentDisconnect ▪ 7 = CPT & PolarityReversal & CurrentDisconnect ▪ 8 = Silence		
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 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.	700 msec	25 to 1500
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

Analog Parameters

Parameter Name	Description	Default	Range
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 200 msec, then enter 150 msec, i.e. 200 minus 50).	300 msec	25 to 300 msec
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
OffhookDebounceTiming	Sets the off hook detection debounce timing. ▪ 10 - 22 ms ▪ 11 - 24 ms ▪ 12 - 26 ms ▪ 13 - 28 ms ▪ 14 - 30 ms ▪ 15 - 32 ms	15	0 to 15
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
TimeToSampleAnalogLineVoltage	Determines the time to sample the analog line voltage after offhook, for the current disconnect threshold.	1000	100 to 2500

Analog Parameters

Parameter Name	Description	Default	Range
Winktime	Defines the time elapsed between two consecutive polarity reversals.	200	All

3.4.1.7 Control Protocol Parameters

The table below lists and describes the parameters, contained in the *ini* file, that are common to all
Call Control (CC) protocols. Use this table as a reference when modifying *ini* file parameter values.

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
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. **NOTE: Deprecated in MEGACO.**	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.	0	0 or 2

Control Protocol Parameters - ALL

Parameter Name	Description	Default	Range
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]
cpMediaResourceOptimization	Determines the DSP allocation method in an IP-to-IP call. If this parameter is set to (0) DISABLE_SINGLE_DSP_ALLOCATION, then two IP terminations with the same configuration will cause an allocation of 0 DSPs and two terminations with different configuration will cause allocation of two DSPs. • If this parameter is set to (1) ENABLE_SINGLE_DSP_ALLOCATION, then two IP terminations with the same configuration will cause an allocation of 0 DSPs transcoding with an allocation of one DSP will be used if possible, and two terminations with different configuration will cause allocation of two DSPs. • If this parameter is set to (2) ENFORCE_FULL_DSP_ALLOCATION, then two IP terminations with any configuration will cause an allocation of two DSPs.	2	0,1,2
cpCipherSuiteType	Defines the default cipher type for the control protocol: 0 = none; 1 = TGCP; 2 = SRTP	0	0, 1, 2
cpDigitMapLongTimer	Defines the inter-digit long timer (L Symbol) value in milliseconds, in a digit map. This timer is typically activated between collected digits when the end of the pattern has not yet been reached. When the value is -1, the hardcoded value (16000) is set.	-1	-1 and greater
CPMediaIPVersionPreference	Configures which address types the blade will offer and in what order of preference. • 0 - Prefer IPV6 • 1 - Prefer IPV4 • 2 - IPv6 Only • 3 - IPv4 Only	0	0 to 3
cpDigitMapShortTimer	Defines the short timer (S Symbol)	-1	-1 to 65535

Control Protocol Parameters - ALL

Parameter Name	Description	Default	Range
	value in milliseconds, in a digit map. This timer is typically activated when the repetition symbol "." exists. When the value is -1, the hardcoded value (3000) is set.		
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.	PCMU for 6300 device group PCMA for all other devices	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. The exact meaning of each bit is described in the MEGACO section of this manual. Every new RFC support should be turned on or off with this parameter.	unsigned Integer > 0	See Descr.
CPSDPSESSIONOWNER	Defines the owner/creator of the session	-	String[31].
CPTransportType	Defines the transport type for the control protocol: ▪ 0 = UDP ▪ 1 = TCP ▪ 2 = SCTP	0	0, 1, 2
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 the digit map, in increments of 10. For MEGACO, it	-1	-1 or 1 to 65535

Control Protocol Parameters - ALL

Parameter Name	Description	Default	Range
	represents the start timer. For all other protocols it represents the end timer. When the value is -1, it is set to 16000 milliseconds.		
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 (if playing is requested from Call Agent). 0 = Off; 1 = On	1	0 or 1
GatewayName	For MGCP - Defines the media gateway's identification name towards the MGCP Call Agent. If undefined, the gateway name holds the IP address of the device.	MGCP: AudioCodes.com	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	1	0 to 2, 4, 8

Control Protocol Parameters - ALL

Parameter Name	Description	Default	Range
	▪ 8 = SIP		
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 disconnection (in seconds).	30	>0
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	PCMU	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
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 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. **Note:** In order for this parameter to take effect, a device reset is required. **Deprecated in MEGACO.**	NULL	Legal IP Address
ProvisionedCallAgentsPorts	Use this parameter to define a list of up to 10 Call Agent UDP ports separated by a comma ',' or a semi-	0	0 to 65535

Control Protocol Parameters - ALL

Parameter Name	Description	Default	Range
	colon ';' for each Call Agent defined by parameter used to specify Allowed Call Agent Address. Using the default value (0), means that port 2944 will be used. Note: In order for this parameter to take effect, a device reset is required. **Deprecated in MEGACO.**		
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 Base.	1	0 or 1
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. **Note:** A default value of "0" indicates an interval value of 5000 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

Control Protocol Parameters - ALL

Parameter Name	Description	Default	Range
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

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	0	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

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

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

3.4.1.8 Routing Parameters

The IP network routing parameters are described in the table below.

IP Network Routing Parameters

Parameter	Description
Web: Disable ICMP Redirects **[DisableICMPRedirects]**	Determines whether the device accepts or ignores ICMP Redirect messages. **[0]** Disable = (Default) ICMP Redirect messages are handled by the device. **[1]** Enable = ICMP Redirect messages are ignored.
Static IP Routing Table	
Web/EMS: IP Routing Table **[StaticRouteTable]**	Defines up to 30 static IP routing rules for the device. These rules can be associated with IP interfaces defined in the Multiple Interface table (InterfaceTable parameter). The routing decision for sending the outgoing IP packet is based on the source subnet/VLAN. If not associated with an IP interface, the static IP rule is based on destination IP address. When the destination of an outgoing IP packet does not match one of the subnets defined in the Multiple Interface table, the device searches this table for an entry that matches the requested destination host/network. If such an entry is found, the device sends the packet to the indicated router (i.e., next hop). If no explicit entry is found, the packet is sent to the default gateway according to the source interface of the packet (if defined). The format of this parameter is as follows: **[StaticRouteTable]** FORMAT StaticRouteTable_Index = StaticRouteTable_InterfaceName, StaticRouteTable_Destination, StaticRouteTable_PrefixLength, StaticRouteTable_Gateway, StaticRouteTable_Description; [\StaticRouteTable]
Destination IP Address **[StaticRouteTable_Destination]**	Defines the IP address of the destination host/network. The destination can be a single host or a whole subnet, depending on the Prefix Length configured for this routing rule.
Prefix Length **[StaticRouteTable_PrefixLength]**	Defines the Classless Inter-Domain Routing (CIDR)-style representation of a dotted-decimal subnet notation, of the destination host/network. The CIDR-style representation uses a suffix indicating the number of bits that are set in the dotted-decimal format. For example, 16 is synonymous with subnet 255.255.0.0.
The address of the host/network you want to reach is determined by an AND operation that is applied to the fields 'Destination IP Address' and 'Prefix Length'. For example, to reach the network 10.8.x.x, enter 10.8.0.0 in the 'Destination IP Address' field and 16 in the 'Prefix Length'. As a result of the AND operation, the value of the last two octets in the 'Destination IP Address' field is ignored. To reach a specific host, enter its IP address in the 'Destination IP Address' field and 32 in the 'Prefix Length' field.	

IP Network Routing Parameters

Parameter	Description
Gateway IP Address **[StaticRouteTable_Gateway]**	Defines the IP address of the router (next hop) used for traffic destined to the subnet/host as defined in the 'Destination IP Address' / 'Prefix Length' field. **Note:** The Gateway address must be in the same subnet as the IP address of the interface over which you configure this static routing rule.
Metric	Defines the number of hops needed to reach the specified destination. **Note:** The recommended value for this parameter is 1. This parameter must be set to a number greater than 0 for the routing rule to be valid. Routing entries with Hop Count equals 0 are local routes set automatically by the device.
Interface Name **[StaticRouteTable_InterfaceName]**	Assigns a network interface through which the 'Gateway IP Address' is reached. This is the string value as configured for the network interface in the 'Interface Name' field of the Multiple Interface table. **Note**: The IP address of the 'Gateway IP Address' field must be in the same subnet as this interface's IP address.
Status	Read-only field displaying the status of the static IP route: ▪ **"Active"** - routing rule is used ny the device. ▪ **"Inactive"** - routing rule is not applied. When the destination IP address is not on the same segment with the next hop or the interface does not exist, the route state changes to "Inactive".

3.4.1.9 IPsec Parameters

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
IPsecPolicyRemote SubnetMask	Secure Internet Protocol (IPsec) Policy - Subnet Mask of the Remote IPSec Address.	255.255.255.255	Legal Subnet
IPsecPolicyRemote TunnelIPAddress	Secure Internet Protocol (IPsec) Policy - IP Address of the Remote IPSec Tunnel Endpoint.	0.0.0.0	Legal IP address

3.4.1.10 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 for details.

NFS Parameters

Parameter	Description
Index	The row index of the remote file system. The valid range is 1 to 16.
Host Or IP **[NFSServers_HostOrIP]**	The domain name or IP address of the NFS server. If a domain name is provided, a DNS server must be configured.
Root Path **[NFSServers_RootPath]**	Path to the root of the remote file system in the format: /[path]. For example, '/audio'.
NFS Version **[NFSServers_NfsVersion]**	NFS version used to access the remote file system. ▪ **[2]** NFS Version 2 ▪ **[3]** NFS Version 3 (default)
Authentication Type **[NFSServers_AuthType]**	Authentication method used for accessing the remote file system. ▪ **[0]** Null ▪ **[1]** Unix (default)
User ID **[NFSServers_UID]**	User ID used in authentication when using Unix. The valid range is 0 to 65537. The default is 0.
Group ID **[NFSServers_GID]**	Group ID used in authentication when using Unix. The valid range is 0 to 65537. The default is 1.
VLAN Type **[NFSServers_VlanType]**	The VLAN type for accessing the remote file system. ▪ **[0]** OAM ▪ **[1]** MEDIA (default) **Note:** This parameter applies only if VLANs are enabled or if Multiple IPs is configured.
NFSBasePort	Start of the range of numbers used for local UDP ports used by the NFS client. Up to maxChannels+maxNumNfsServers local ports are used. Default value is 47000. Range is 0 to 65535.

3.4.1.11 SRTP Parameters

The Secure Real-Time Transport Protocol (SRTP) parameters are described in the table below.

SRTP Parameters

Parameter	Description
Web: Media Security EMS: Enable Media Security **[EnableMediaSecurity]**	Enables Secure Real-Time Transport Protocol (SRTP). ▪ **[0]** Disable = (Default) SRTP is disabled. ▪ **[1]** Enable = SRTP is enabled. **Notes:** For this parameter to take effect, a device reset is required. SRTP causes a reduction of 12.5% in available channels (i.e., 84 channels instead of 96 per DSP) when using DSP Template 0. In other words, instead of 2,016 (21 * 96) channels, 1,764 (21 * 84) are available. For a depopulated system, 504 channels are available (6 * 84). The DSP-depopulated blade is used for configurations with 21 spans and less. Therefore, no capacity reduction occurs in these sizes for T1 configurations (504/24 = 21 T1s). However, only 15 E1s instead of 16 (504/32 = 15.75) may be available.
Web/EMS: Media Security Behavior **[MediaSecurity Behaviour]**	Determines the device's mode of operation when SRTP is used (i.e., when the parameter EnableMediaSecurity is set to 1). ▪ **[0]** Preferable = (Default) The device initiates encrypted calls. However, if negotiation of the cipher suite fails, an unencrypted call is established. Incoming calls that don't include encryption information are accepted. ▪ **[1]** Mandatory = The device initiates encrypted calls, but if negotiation of the cipher suite fails, the call is terminated. Incoming calls that don't include encryption information are rejected. ▪ **[2]** Disable = The IP Profile for which this parameter is set does not support encrypted calls (i.e., SRTP). ▪ **[3]** Preferable - Single Media = The device sends SDP with a single media ('m=') line only (e.g., m=audio 6000 RTP/AVP 4 0 70 96) with RTP/AVP and crypto keys. The remote UA can respond with SRTP or RTP parameters: ✓ If the remote UA does not support SRTP, it uses RTP and ignores the crypto lines. ✓ In the opposite direction, if the device receives an SDP offer with a single media (as shown above), it responds with SRTP (RTP/SAVP) if the EnableMediaSecurity parameter is set to 1. If SRTP is not supported (i.e., EnableMediaSecurity is set to 0), it responds with RTP. **Notes:** ▪ Before configuring this parameter, set the EnableMediaSecurity parameter to 1. ▪ If this parameter is set to Preferable [3] and two 'm=' lines are received in the SDP offer, the device prefers the SAVP (secure audio video profile) regardless of the order in the SDP. ▪ Option [2] Disable is applicable only to IP Profiles.

SRTP Parameters

Parameter	Description
	▪ This parameter can also be configured per IP Profile, using the IPProfile parameter (see 'Configuring IP Profiles' on page).
Web: Master Key Identifier (MKI) Size EMS: Packet MKI Size **[SRTPTxPacketMKISize]**	Defines the size (in bytes) of the Master Key Identifier (MKI) in SRTP Tx packets. The range is 0 to 4. The default is 0 (i.e., new keys are generated without MKI). **Notes:** For the GW/IP-to-IP application, the device only initiates the MKI size. You can also configure MKI size in an IP Profile. For the SBC application, the device can forward MKI size as is for SRTP-to-SRTP flows or override the MKI size during negotiation, using IP Profiles. This can be done on the inbound or outbound leg.
Web:Symmetric MKI Negotiation EMS: Enable Symmetric MKI **[EnableSymmetricMKI]**	Enables symmetric MKI negotiation. • **[0]** Disable = (Default) The device includes the MKI in its 200 OK response according to the SRTPTxPacketMKISize parameter (if set to 0, then it is not included; if set to any other value, it is included with this value). • **[1]** Enable = The answer crypto line contains (or excludes) an MKI value according to the selected crypto line in the offer. For example, assume that the device receives an INVITE containing the following two crypto lines in SDP: a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:TAaxNnQt8/qLQMnDuG4vxYfWl6K7eBK/ufk04pR4|2^31|1:1 a=crypto:3 AES_CM_128_HMAC_SHA1_80 inline:bnuYZnMxSfUiGitviWJZmzr7OF3AiRO0l5Vnh0kH|2^31 The first crypto line includes the MKI parameter "1:1". In the 200 OK response, the device selects one of the crypto lines (i.e., '2' or '3'). Typically, it selects the first line that supports the crypto suite. However, for SRTP-to-SRTP in SBC sessions, it can be determined by the remote side on the outgoing leg. If the device selects crypto line '2', it includes the MKI parameter in its answer SDP, for example: a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:R1VyA1xV/qwBjkEklu4kSJyl3wCtYeZLq1/QFuxw|2^31|1:1 If the device selects a crypto line that does not contain the MKI parameter, then the MKI parameter is not included in the crypto line in the SDP answer (even if the SRTPTxPacketMKISize parameter is set to any value other than 0). **Notes:** • To enable symmetric MKI, the SRTPTxPacketMKISize parameter must be set to any value other than 0. • You can also enable MKI negotiation per IP Profile.

SRTP Parameters

Parameter	Description
Web/EMS: SRTP offered Suites **[SRTPofferedSuites]**	Defines the offered crypto suites (cipher encryption algorithms) for SRTP. ▪ **[0]** = (Default) All available crypto suites. ▪ **[1]** CIPHER SUITES AES CM 128 HMAC SHA1 80 = device uses AES-CM encryption with a 128-bit key and HMAC-SHA1 message authentication with a 80-bit tag. ▪ **[2]** CIPHER SUITES AES CM 128 HMAC SHA1 32 = device uses AES-CM encryption with a 128-bit key and HMAC-SHA1 message authentication with a 32-bit tag. ▪ **[4]** CIPHER SUITES ARIA CM 128 HMAC SHA1 80 = device uses ARIA encryption algorithm with a 128-bit key and HMAC-SHA1 message authentication with a 32-bit tag. ▪ **[8]** CIPHER SUITES ARIA CM 192 HMAC SHA1 80 = device uses ARIA encryption algorithm with a 192-bit key and HMAC-SHA1 message authentication with a 32-bit tag. **Notes:** ▪ For enabling ARIA encryption, use the AriaProtocolSupport parameter. ▪ This parameter also affects the selection of the crypto in the device's answer. For example, if the device receives an offer with two crypto lines containing HMAC_SHA1_80 and HMAC_SHA_32, it uses the HMAC_SHA_32 key in its SIP 200 OK response if the parameter is set to 2.
Web: Aria Protocol Support **[AriaProtocolSupport]**	Enables ARIA algorithm cipher encryption for SRTP. This is an alternative option to the existing support for the AES algorithm. ARIA is a symmetric key block cipher algorithm standard developed by the Korean National Security Research Institute. **[0]** Disable (default) **[1]** Enable **Notes:** To configure the ARIA bit-key encryption size (128 or 192 bit) with HMAC SHA-1 cryptographic hash function, use the SRTPofferedSuites parameter. For ARIA encryption of SRTP, the device must be installed with the relevant Software License Key.
Web: Disable Authentication On Transmitted RTP Packets EMS: RTP AuthenticationDisable Tx **[RTPAuthenticationDisableTx]**	Enables authentication on transmitted RTP packets in a secured RTP session. ▪ **[0]** Enable (default) ▪ **[1]** Disable
Web: Disable Encryption On Transmitted RTP Packets EMS: RTP EncryptionDisable Tx **[RTPEncryptionDisableTx]**	Enables encryption on transmitted RTP packets in a secured RTP session. ▪ **[0]** Enable (default) ▪ **[1]** Disable
Web: Disable Encryption On Transmitted RTCP Packets EMS: RTCP EncryptionDisable Tx	Enables encryption on transmitted RTCP packets in a secured RTP session. ▪ **[0]** Enable (default)

SRTP Parameters

Parameter	Description
[RTCPEncryptionDisableTx]	▪ **[1]** Disable
[ResetSRTPStateUponRekey]	Enables synchronization of the SRTP state between the device and a server when a new SRTP key is generated upon a session expire. This feature ensures that the roll-over counter (ROC), one of the parameters used in the SRTP encryption/decryption process of the SRTP packets, is synchronized on both sides for transmit and receive packets. ▪ **[0]** = (Default) Disabled. ROC is not reset on the device side. ▪ **[1]** = Enabled. If the session expires causing a session refresh through a re-INVITE, the device or server generates a new key and the device resets the ROC index (and other SRTP fields) as done by the server, resulting in a synchronized SRTP. **Notes:** This feature can also be configured for an IP Profile. If this feature is disabled and the server resets the ROC upon a re-key generation, one-way voice may occur.

3.4.1.12 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.

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
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).	0	≥0

MGCP Parameters - ALL

Parameter Name	Description	Default	Range
MGCPBufferingTimeout	Sets the timer for buffering digit after a digit map match was found and until new RQNT arrives. The timer is in seconds. The number of buffered digits is limited to 64.	0	≥0
MGCPDIGESTPASSWORD	Defines the MGCP Digest security password.	See Descr.	Up to 39 characters
MGCPDIGESTUSERNAME	Defines the MGCP Digest security user name.	See Descr.	Up to 39 characters
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 characters
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
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
MGCPTrunkNaming	Defines the trunk and B-channel naming pattern used by the gateway.	'ds*/tr*'	String [63]

MGCP Parameters - ALL

Parameter Name	Description	Default	Range
Pattern	The '*' signs will be replaced with a trunk or B-channel number or with a wild-card sign.		
MGCPUseAudioPort ForT38	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
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.
MGHistoryBuffer TimeLim	Defines the time (in seconds) that a transaction is kept in the history buffer.	30	≥ 0
QuarantineMode State	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 _x (0-2)	Defines for each of the redundant Call Agents, the IP addresses to be used for the initial Restart in Progress message (RSIP). Each of these parameters are optional.	NULL	xxx.xxx.xxx.xxx
RedundantAgentPort _x(0-2)	Defines the Port Number for each redundant Call Agent.	Port 2427	0 to 65534
RSIPOnNetworkDisc onnection	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

MGCP Parameters - ALL

Parameter Name	Description	Default	Range
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 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.	1	0 or 1
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

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]

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 signals/events operations prior to CRCX. 0 = Deactivate; 1 = Activate	0	0 or 1

3.4.1.13 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.

MEGACO Parameters - ALL

Parameter Name	Description	Default	Range
DigitMapName	Name of the provisioned digit map.	NULL	String [30]
DIGITMAPPING	The digit map patterns separated by a vertical bar (|), as defined in H.248.1 Section 7.1.14.	NULL	String [151]
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]
MEGACOCheckLegalityOfMGC	This parameter is specified if MEGACO rejects commands from a MGC not in the provisioned list. • 1 = Reject • 0 = Don't Check	1	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 4294967295 (0 to FFFFFFFF)
MEGACOCOTTESTTYPE	The continuity test type (Set per trunk). One of the following values: 0 = THRH, 1 = THRL, 2 = TLRH • T = Transmit • R = Receive • H = Continuity high signal • L = Continuity low signal	0	0, 1, 2
MEGACOEncoding	Sets the MEGACO coding method. 0 = Text mode	0	0
MEGACOHangTermTimeout	Default timeout (in seconds) for sending Hanging Termination event, when a request for Hanging Termination is sent without parameters.	0	0 -65535

MEGACO Parameters - ALL

Parameter Name	Description	Default	Range
MEGACOTdmHairPinningMode	Determines which Hair-pinning mode is to be used: Mode 0 - will create a TDM to TDM connection through IP software Loopback Mode 1 - will create “pure” Hair-pinning i.e. TDM to TDM connection through PSTN.	0	0, 1
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)
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_Num	Defines the starting number for each name's RTP termination level.	0	Positive number

MEGACO Parameters - IPM

Parameter Name	Description	Default	Range
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.**	50	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]

MEGACO Parameters - TP

Parameter Name	Description TP	Default	Range
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

3.4.1.14 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.

Web Parameters - ALL

Parameter Name	Description	Default	Range
DenyAuthentication Timer	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
HTTPSCertFile Name	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
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

Web Parameters - ALL

Parameter Name	Description	Default	Range
HTTPSRootFile Name	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.	145 pixels	0 to 200 pixels
RadiusAccLocalPort	RADIUS access local port. Predefined UDP port used for access to the RADIUS ACC server. Note: In order for this parameter to take effect, a device reset is required.	801	Any integer
RadiusAuthLocal Port	RADIUS authentication local port. Predefined UDP port used for authentication with the RADIUS server. Note: In order for this parameter to take effect, a device reset is required.	800	Any integer
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
RADIUSDouble DecodeURL	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

Web Parameters - ALL

Parameter Name	Description	Default	Range
RADIUSAuthPort	RADIUS authentication port. Predefined UDP port used for authentication with the RADIUS Server.	1645	Any integer
RADIUS Retransmission	RADIUS packets retransmission. Number of retransmissions for the same request.	3	1 to 10
RADIUSTo	RADIUS Response Time Out. Time to wait for the response before a retransmission is needed.	10 seconds	1 to 30 seconds.
RadiusVSAAccess Attribute	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
RadiusVSAVendor ID	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]

Web Parameters - ALL

Parameter Name	Description	Default	Range
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
WanMgmtHttpPort	Determines the WAN HTTP port of the device. If set to 0, WAN HTTP access will not be possible.	0	0 to 65535 (other restrictions may apply in this range)
WanMgmtHttpsPort	Determines the WAN HTTPS port of the device. If set to 0, WAN HTTPS access will not be possible.	0	0 to 65535 (other restrictions may apply in this range)
WanMgmtSSHPort	Determines the WAN SSH port of the device. If set to 0, WAN SSH access will not be possible.	0	0 to 65535 (other restrictions may apply in this range)
WanMgmtTelnet Port	Determines the WAN Telnet port of the device. If set to 0, WAN Telnet access will not be possible.	0	0 to 65535 (other restrictions may apply in this range)
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) 2, digest authentication (MD5) used for HTTP, while basic authentication used for HTTPS. Note that turning on RADIUS login forces basic authentication.	0	0 to 2
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	0	0, 1

Web Parameters - ALL

Parameter Name	Description	Default	Range
	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		

3.4.1.15 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.

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).	500, 100 for MediaPack	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 a network manager, and provides a non-volatile 'handle' for the physical entity.	NULL	String [255]
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
IFLinkUpDownTrapEnable	Enables and disables the LinkUp and LinkDown traps. 1 = Enable 2 = Disable	2	1 or 2
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	0	0 or 1

SNMP Parameters - ALL

Parameter Name	Description	Default	Range
	defined in the parameter defining NAT Binding Default Timeout. 0 = Disable; 1 = Enable		
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]
SnmpEngineIDString	Sets the SNMP EngineID. 12 HEX Octets in the following format xx:xx:...:xx For example: 00:11:22:33:44:55:66:77:88:99:aa:bb	DEFAULT_ENGINE_ID_STRING	String[35]
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 trap format, authentication level and encryption level. By default, the trap is associated with the SNMP trap community string.	An empty string	0 to 33 characters.
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
SNMPREADONLYCOMMUNITYSTRING	Used to define a read-only community string.	“public”	String[19]
SNMPREADWRITECOMMUNITYSTRING	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	String[100]

SNMP Parameters - ALL

Parameter Name	Description	Default	Range
		1.3.6.1.4.1.5003.8.1.1	
SNMPTRAPCOMMUNITYSTRING	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]
Web: Trap User [SNMPManagerTrapUser_x]	Defines the SNMPv3 USM user or SNMPv2 user to associate with the trap destination. This determines the trap format, authentication level, and encryption level. By default, it is associated with the SNMPv2 user (SNMP trap community string). The valid value is a string.		
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]
WanMgmtSnmpPort	This parameter specifies the WAN port number for SNMP requests and responses. If set to 0, WAN access to SNMP is disabled. Note: In order for this parameter to take effect, a device reset is required.	0	0 to 65534

SNMP Parameters - MediaPack & Mediant 1000

Parameter Name	Description	Default	Range
ChassisPhysicalAssetID	This object is a user-assigned asset tracking identifier for the Mediant 1000 chassis as specified by an EMS, and provides non-volatile storage of this information. For Mediant 1000 only.	NULL	String[255] Mediant 1000 only.

3.4.1.16 Voice Streaming Parameters

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.

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, retransmission 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 ServerRespond Timeout).	0 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 90000
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 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.**	0	See Descr.
StreamingCacheRefreshTime	Determines the dynamic caching refresh rate (in minutes). This refresh timeout avoids the scenario in which files played from the cache have	-1	-1 to 0xFFFF

Voice Streaming Parameters

Parameter Name	Description	Default	Range
	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.**		
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 Timeout	Defines the maximum time in 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	5000	100 to 10000

Voice Streaming Parameters

Parameter Name	Description	Default	Range
	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.		
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.	-	-

3.4.1.17 SCTP Parameters

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

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 are available: 0 = adler32 checksum mechanism 1 = crc32c checksum mechanism (improved mechanism)	0	0 or 1
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

SCTP Parameters - All Digital Devices

Parameter Name	Description	Default	Range
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

3.4.1.18 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.

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
AMSForceRepositoryUpdateEnabled	Indicates that a new audio 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	Enables/Disables AMS Advanced IVR play functionality. 0 = Disable 1 = Enable	0	0 or 1
AMSSecondary Language	Defines the Secondary Language for the AMS.	NULL	String[3] - language ISO string

Advanced Audio Server Parameters

Parameter Name	Description	Default	Range
APSEnabled	Indicates if the system should expect to use APS bundles (vp.dat and segments.xml files), or if the system should expect the vp.dat file only. 1 - APS bundle should be used. 0 - The system should use vp.dat only.	1	0 or 1
AudioStagingAutoSwitchoverEnabled	Indicates if audio repository should automatically be activated after downloading of the APS bundle. 0 - Audio is not to be automatically activated. 1 - Audio is to be automatically activated.	1	0 or 1

3.4.2 ini File Table Parameters

3.4.2.1 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.

NFS Servers Table Parameters

Parameter Name	Description	Default	Range
NFSServers_Index	Table row index.	N/A	0 to 4
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_NfsVersion	The NFS version to use in accessing this remote NFS file system.	3	2 or 3
NFSServers_AuthType	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_VLANType	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

3.4.2.2 DS3 Configuration Table Parameters

Note: 'T3' and 'DS3' are terms used interchangeably. This section is only applicable to the TP-6310 and Mediant 3000.

DS3 Configuration Table Parameters

ini File Field Name	Description	Default Value	Valid Range
DS3CONFIG_ FramingMethod	Used to select the physical DS3 framing method for the interface. **Applicable only to TP-6310/T3 and Mediant 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 and Mediant 3000/T3.** 0 = DS3Clock is recovered from the line 1 = DS3 trunk clock source is provided by the device's internal clock	1	0 to 1
DS3CONFIG_Line BuildOut	Used to select the DS3 line build out. **Applicable only to the TP-6310/DS3 and Mediant 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_Circuit Identifier	Defines the Interface SNMP-name represented by a string of up to 15 characters.	Empty string	string
DS3CONFIG_TrapE nable	Enables or disables SNMP traps for DS3 1 = Enable 0 = Disable	1	0 or 1
DS3CONFIG_PmOn Off	Enables or disables DS3 performance monitoring 1 = Enable 0 = Disable	1	0 or 1
DS3CONFIG_Tappi ngEnable	Enables or disables special DS3 Tapping mode. In this mode the interface is capable of receiving a signal with additional 14dB attenuation to the	0	0 or 1

	standard maximum length. 1 = Enable 0 = Disable		
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 2 = Administrative status Down (currently not supported)	1	1 or 2

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, 4 ;
```

This means that the interface should be configured with framing method M23, using External clock source and line built-out LEVEL 3.

3.4.2.3 MEGACO Gateway Configuration Table Parameters

The VGW can be configured via Management using the MegacoGtwConfigurationTable table (see reference in *ini* file tables). Configuration is offline and requires reset to the device.

VGW Configuration Table

Parameter Name	Description	Parameter Type	Default Value
MegacoGtwConfiguration Table_Index	Virtual GW table Index. The unique Virtual Media Gateway Id. Mandatory and Unique (not used in the Web interface)	0-2	""
MegacoGtwConfiguration Table_VirtualGWName	Virtual GW Name. The descriptive name of the Virtual GW. Mandatory and Unique.	string – 17 chars	_
MegacoGtwConfigurationTable_IPv4InterfaceName	IPv4 Interface Name. The pointer (Interface ID) to Network Interface Table for IP version 4 address interface ID . Should contain valid value from NW Interface Table. DEFAULT will cause retrieving default interface name for Control from NW.	String – 17 chars according to Interface Table	"DEFAULT"
MegacoGtwConfiguration Table_LocalPort	Local Port Defines the port to be used by the Media Gateway Controller to communicate with this Virtual GW. Port 0 is illegal.	Short	2944
MegacoGtwConfiguration Table_MID	MID Defines the Virtual GW Message Identifier . Optional,	string – 69 chars	""
MegacoGtwConfiguration Table_LoadWeight	Load Weight Defines the weight of the VGW out of the total load. VGW with higher values process more messages. Select the next VGW from the list based on load weight. Server with a higher weight process more messages. By default, the weight of the each VGW is "1". This means the load between each VGW is equal, and each VGW processes one message on the Round-Robin loop. Generally, the VGW load percentage (%) out of the total load can be calculated by VGW load weight / Total weight.	1-5	1

VGW Configuration Table

MegacoGtwConfigurationTable_MediaRealmName	Media Realm Name defines the default Media Realm name (pointer to media realm table). Optional.	String – 40 chars according to the Media Realm table	""
MegacoGtwConfigurationTable_MegacoVersion	Megaco Version define the Megaco Version. Optional. **Note:** In case Megaco Version is set to a value greater than "3" – the default value will be used.	1-3	2
MegacoGtwConfigurationTable_ServiceChangeProfile	Service Change Profile defines the service change profile. Optional.	String – 64 chars	"TGW"
MegacoGtwConfigurationTable_AssociatedMembersList	Associated Members List defines the list of the trunks/interfaces. Comma and range can be used. Each trunk should be associated with a single VGW only. "Default Line" refers to the unused trunks. The Trunks range begins with "1".	String – 64 chars	"DEFAULT LINE"

3.4.2.4 MEGACO Gateway Controller Link Table Parameters

MGCs for each VGW are configured via Management using the *MegacoGtwControllerLinkTable* table (see reference in ini file tables). MGC configuration is offline and requires reset to the device.

MEGACO Gateway Controller Link Table

Parameter Name	Description	Parameter Type	Default Value
MegacoGtwControllerLinkTable_Index	Controller Link Index defines the unique Controller Link Index. Mandatory parameter (not used in the Web interface).	0-9	-
MegacoGtwControllerLinkTable_VirtualGWIndex	Virtual GW index defined a pointer - the Virtual GW Name. Mandatory parameter. According to MegacoGatewayConfigutationTable table. (not used in the Web interface)	String – 17 chars	-
MegacoGtwControllerLinkTable_ActivityOrder	Activity Order defines the order in which MGC becomes active. Mandatory parameter. MGC with Activity order that was set to “1” will be the first MGC active. Upon disconnect MGC that its activity order set to “2” will become the next active and so on.	1-10	-
MegacoGtwControllerLinkTable_TransportType	Transport Type defines the signaling transport type to be used o communication with this MGC .	UDP=0,TCP=1,SCTP=2	UDP
MegacoGtwControllerLinkTable_MGCAddressFormat	MG Controller Address Format defines the format of the MGC address.	IPv4=0,DNS=2IPv6 (=1) Not supported	IPv4
MegacoGtwControllerLinkTable_MGControllerAddress	MG Controller Address defines the domain name or IP address of the MG Controller. If a domain name is provided, then a DNS server must be configured. Mandatory parameter.	String – 39 chars	“”
MegacoGtwControllerLinkTable_MGControllerPort	MG Controller Port defines the port number MGC is listening for. Port 0 is illegal.	Short	2944

Note: The same local port and same network interface for different virtual gateways can’t be used for SCTP transport. The VGW on SCTP should be on the first control interface only (SCTP can run only on first control interface).

The MGC Address format should match the IP Interface Name at VGW configuration table. For example: if IPv4 is set at “MG Controller Address Format “ IPv4Interface Name must be set at VGW configuration table.

4 Network Configuration

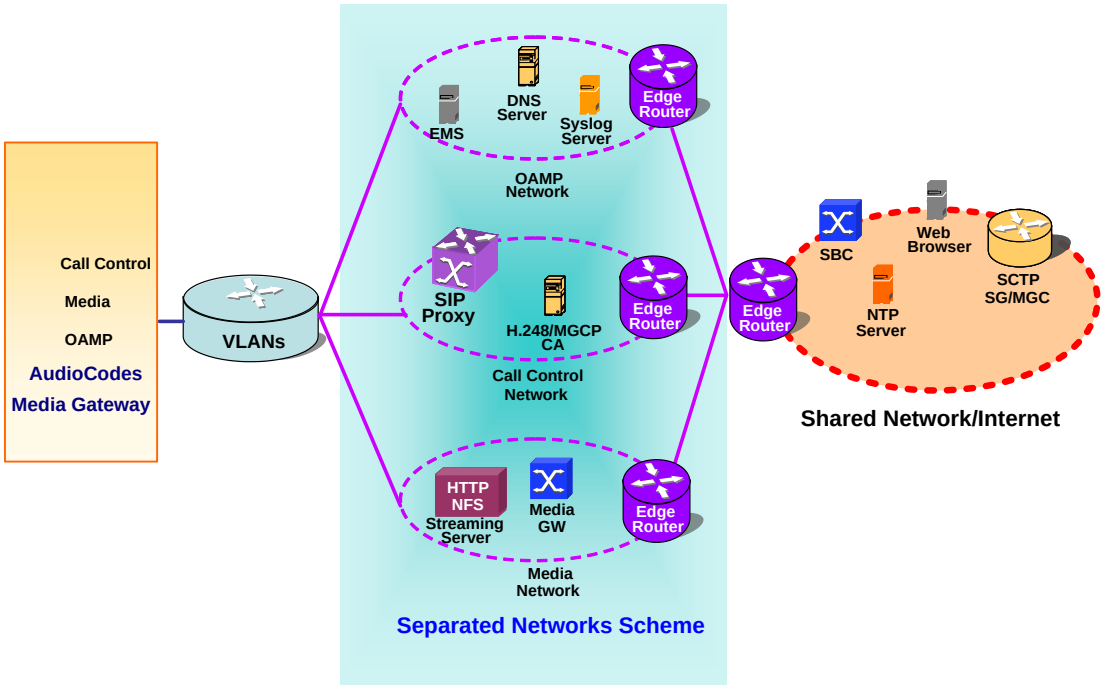
The device allows the user to configure up to 16 / 32 different IP addresses with associated VLANs, via an Interface Table. Complementing the Interface Table is a configurable Routing Table, allowing the user to define routing rules for non-local hosts/subnets.

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 2: 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 Network Interface Table, which is configurable via ini file, Web & SNMP interfaces.

Note: The following section contains references to IPv6. These references are applicable only to TP-6310, TP-8410 and Mediant 3000 systems.

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.

Multiple Interface Table

Index	Allowed Application Types	Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	OAMP	IPv4 Manual	10.31.174.50	16	0.0.0.0	4	ManagementI F
1	Control	IPv4 Manual	10.32.174.50	16	0.0.0.0	5	ControlIF
2	Media	IPv4 Manual	10.33.174.50	16	10.33.0.1	6	Media1IF
3	Media	IPv4 Manual	10.34.174.50	16	0.0.0.0	7	Media2IF
4	Media	IPv4 Manual	10.35.174.50	16	10.35.0.1	8	Media3IF
5	Media	IPv4 Manual	10.36.174.50	16	0.0.0.0	9	Media4IF
6	Media	IPv4 Manual	10.37.174.50	16	0.0.0.0	10	Media5IF
7	Media	IPv4 Manual	10.38.174.50	16	0.0.0.0	11	Media6IF
8	Media	IPv4 Manual	10.39.174.50	16	10.39.0.1	12	Media7IF
9	Media	IPv4 Manual	10.40.174.50	16	0.0.0.0	13	Media8IF
10	Media + Control	IPv4 Manual	10.41.174.50	16	0.0.0.0	14	Media9IF
11	Media	IPv4 Manual	10.42.174.50	16	0.0.0.0	15	Media10IF
12	Media	IPv4 Manual	10.43.174.50	16	10.43.0.1	16	Media11IF
13	Media	IPv4 Manual	10.44.174.50	16	0.0.0.0	17	Media12IF
14	Media	IPv4 Manual	10.45.174.50	16	10.45.0.1	18	Media13IF
15	Media + Control	IPv4 Manual	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).

4.1.2 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. Several 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.

4.1.2.1 Index Column

This column holds the index of each interface. Possible values are 0 to 15 (or 31 in the Mediant 3000 systems). Each interface index must be unique.

4.1.2.2 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 applications, such as Web, Telnet, SSH, SNMP.
- CONTROL – Call Control Protocols. Examples of Control applications include: SIP, MGCP, MEGACO.
- MEDIA – RTP streams of Voice/Video.
- Various combinations of the above mentioned types.

The following table shows the possible values of this column and their descriptions:

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	OAMP, MEDIA & CONTROL: All of the application types will be allowed on this interface.

For valid configuration guidelines, refer to 'Interface Table Configuration Summary & Guidelines' on page 285 for more information.

4.1.2.3 Interface Mode Column

The Interface Mode column determines the method that this interface uses to acquire its IP Address. For IPv4 Manual IP Address assignment, use “IPv4 Manual” (10).

IPv6 addresses may be assigned in two ways:

- “IPv6 Manual” (4)
- “IPv6 Manual Prefix” (3)

Refer to 'Configuring IPv6' on page 285 for more information regarding IPv6 interface modes.

4.1.2.4 IP Address and Prefix Length Columns

These columns allow the user to configure an IPv4 or IPv6 IP Address and its related subnet mask.

The “Prefix Length” column holds the Classless Inter-Domain Routing (CIDR)-style representation of a dotted decimal subnet notation. The CIDR-style representation uses a suffix indicating the number of bits which are set in the dotted decimal format. i.e. 192.168.0.0/16 is synonymous with 192.168.0.0 and a subnet of 255.255.0.0 (Refer to 'http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing' for more information).

This CIDR notation lists the number of ‘1’ bits in the subnet mask. So, a subnet mask of 255.0.0.0 (when broken down to its binary format) 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.

For IPv4 Interfaces, the prefix length values range from 0 to 31. For IPv6 interfaces, the prefix length must be set to 64.

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 for OAMP applications in the Interface Table will be available when booting from Flash again.

This allows it to work with a temporary address for initial management and configuration purposes while retaining the address to be used for deployment.

4.1.2.5 Gateway Column

This column defines a default gateway for an interface. A default gateway may be configured for each interface. The Gateway Address provided in the “Gateway” column MUST be on the same subnet as the interface address.

A separate routing table allows configuring additional routing rules. Refer to the Routing Table sub-section for more details.

Note: The default gateway configured in the example below (200.200.85.1) will only be available for the applications allowed on that interface (Media & Control). Outgoing management traffic (originated in interface 0) will never be directed to this default gateway.

Configured Default Gateway Example

Index	Allowed Application Types	Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	OAMP	IPv4 Manual	192.168.85.14	16	0.0.0.0	100	Mgmt
1	Media & Control	IPv4 Manual	200.200.85.14	24	200.200.85.1	200	CntrlMedia

A separate routing table allows configuring routing rules. Configuring the following routing rule, enables OAMP applications to access peers on subnet 17.17.0.0 via gateway 192.168.0.1.

Separate Routing Table Example

Destination	Prefix Length	Gateway	Interface	Metric	Status
17.17.0.0	16	192.168.0.1	0	1	Active

Refer to the Routing Table sub-section for more details.

4.1.2.6 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 of the same address family.

One IPv4 interface and one IPv6 interface may share the same VLAN ID, allowing hybrid networks on a single broadcast domain.

4.1.2.7 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.

4.1.3 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.

4.1.3.1 Booting using DHCP

The *DHCPEnable* parameter enables the system to boot while acquiring an IP address from a DHCP server. Note that when using this method, Interface Table/VLANs and other advanced configuration options will be disabled.

4.1.3.2 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 291. The default value for this parameter is 0 (disabled).

4.1.3.3 '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.

Notes:

- If *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, make sure that the value of the *VlanNativeVlanId* parameter is different than any VLAN ID in the table.

4.1.3.4 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:

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 (media traffic)
VLANPremiumServiceClassControlPriority	6	Sets the priority for the Premium service class content (control traffic)
VLANGoldServiceClassPriority	4	Sets the priority for the Gold service class content (streaming traffic)
VLANBronzeServiceClassPriority	2	Sets the priority for the Bronze service class content (OAMP traffic)
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 (media traffic)
PremiumServiceClassControlDiffServ	40	Sets the DiffServ for the Premium service class content (control traffic)
GoldServiceClassDiffServ	26	Sets the DiffServ for the Gold service class content (streaming traffic)
BronzeServiceClassDiffServ	10	Sets the DiffServ for the Bronze service class content (OAMP traffic)

4.1.3.5 Applications with Assignable 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

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 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

4.1.4 Configuring IPv6

Note: TP-6310, TP-8410 and Mediant 3000 support IPv6 configuration.

IPv6 interfaces may be assigned in two ways:

- IPv6 Manual - where the IPv6 address (128 bits) will be set manually.
- IPv6 Manual Prefix - where the IPv6 prefix (higher 64 bits) will be set manually, while the interface ID (the lower 64 bits) will be derived from the modules MAC Address.

For both methods, the only prefix length supported is 64.

Two configurations are available with IPv6:

- with VLANs
- without VLANs

Without VLANs, only one IPv4 interface may be defined in addition to the IPv6 interface. The IPv4 interface "Allowed Application Types" must be set to OAMP, MEDIA & CONTROL, while the IPv6 interface "Allowed Application Types" may be MEDIA, CONTROL or both.

With VLANs, there must be at least one IPv4 interface allowing OAMP applications, one allowing MEDIA applications, and one allowing CONTROL applications (it may very well be the same IPv4 interface). In addition to this mandatory IPv4 interface (or interfaces), more IPv4 and IPv6 interfaces may be defined.

While no two IPv4 interfaces or two IPv6 interfaces may have the same VLAN ID, a combination of one of each address family is permitted.

IPv6 interfaces support MEDIA, CONTROL, or a combination of MEDIA & CONTROL applications. IPv6 interfaces do not support OAMP applications.

4.1.5 Interface Table Configuration Summary & Guidelines

Interface Table configurations must adhere to the following rules:

- Up to 16 different interfaces may be defined (32 in Mediant 3000 systems).
- The indices used must be in the range between 0 to 15 (31 in Mediant 3000 systems).
- 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 a value of 0-31 for IPv4 interfaces and a value of 64 for IPv6 interfaces.

- Only one IPv4 interface with OAMP "Allowed Application Types" **must** be configured. At least one IPv4 interface with CONTROL "Allowed Application Types" **must** be configured. At least one IPv4 interface with MEDIA "Allowed Application Types" **must** be configured. One or more IPv6 interface with CONTROL "Allowed Application Types" **may** be configured. One or more IPv6 interface with MEDIA "Allowed Application Types" may be configured. These application types **may** be mixed (i.e. OAMP and CONTROL). Here are some examples for interface configuration:
  - One IPv4 interface with "Allowed Application Types" – OAMP, MEDIA & CONTROL (without VLANs).
  - One IPv4 interface with "Allowed Application Types" – OAMP, MEDIA & CONTROL plus one IPv6 interface with "Allowed Application Types" – MEDIA & CONTROL (without VLANs).
  - One IPv4 interface with "Allowed Application Types" – OAMP, one other or more IPv4 interfaces with "Allowed Application Types" – CONTROL, and one or more IPv4 interfaces with "Allowed Application Types" – MEDIA (with VLANs).
  - One IPv4 interface with "Allowed Application Types" – OAMP & MEDIA, one other or more IPv4 interfaces with "Allowed Application Types" – MEDIA & CONTROL, zero or more IPv6 interfaces with "Allowed Application Types" MEDIA & CONTROL (with VLANs).
  - Other configurations are also possible while keeping to the above-mentioned rule.
- The default gateway can be configured for each interface. The Gateway address MUST be in the same subnet as the interface; other Routing Rules may be specified in the Routing Table. Refer to the Routing Table section below, 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" for IPv4, and to "::" for IPv6.
- 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 IPv4 Interfaces, the "Interface Mode" column must be set to "IPv4 Manual" (numeric value 10). IPv6 Interfaces "Interface Mode" can be set to either "IPv6 Manual" (numeric value 4), or "IPv6 Manual Prefix" (numeric value 3).
- When defining more than one interface of the same address family, VLANs must 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. In this scenario, multiple network interface capabilities (including IPv6) will not be available.
- 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).
- Qualities 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 configured 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.

Notes:

- When configuring the module via the Web Interface, it is possible to perform a quick validation of the configured Interface Table and VLAN definitions.
  On the Interface Table web page, click on the **Done** button.
- It is highly recommended to use this button when configuring Multiple Interfaces and VLANs via the Web Interface, to ensure the configuration is complete and valid.

4.1.6 Troubleshooting - Interface Table

If any of the above guidelines are violated, the system will fall back to a "safe mode" configuration, consisting of a single IPv4 interface and 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 in the IPv4 interfaces.
- An IPv6 interface was defined with a prefix length different than 64.
- There are too many interfaces with "Allowed Application Types" of OAMP. Only one interface defined but the "Allowed Application Types" column is not set to "O+M+C" (numeric value 6).
- An IPv4 interface was defined with "Interface Type" different than "IPv4 Manual" (10).
- An IPv6 interface was defined with "Interface Type" different than "IPv6 Manual" (4) or "IPv6 Manual Prefix" (3).
- An IPv6 interface was defined with an IPv4 address in the "Gateway" column (including "0.0.0.0").
- Two interfaces of the same address family 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 Routing Table Overview

The Routing Table consists of the following:

Routing Table Layout

Destination	Prefix Length	Gateway	Interface Name	Metric	Status
201.201.0.0	16	192.168.0.1	O+M+C	1	Active
202.202.0.0	16	192.168.0.2	O+M+C	1	Active
203.203.0.0	16	192.168.0.3	O+M+C	1	Active
225.225.0.0	16	192.168.0.25	O+M+C	1	Inactive

4.2.2 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 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 Prefix Length Column

This column holds the Classless Inter-Domain Routing (CIDR)- style representation of a dotted decimal subnet notation. The CIDR-style representation uses a suffix indicating the number of bits which are set in the dotted decimal format. i.e., 16 is synonymous with subnet of 255.255.0.0.

4.2.2.3 Gateway Column

This 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 Interface Column

This column holds the Interface index (in the Interface Table) from which the gateway address is reached.

Figure 3: Interface Column

The Interface Table:

Index	Allowed Application Types	Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	0	10	10.31.174.50	16	0.0.0.0	4	ManagementIF
1	2	10	10.32.174.50	16	0.0.0.0	5	ControlIF
2	1	10	10.33.174.50	16	10.33.0.1	6	Media1IF
3	1	10	10.34.174.50	16	0.0.0.0	7	Media2IF
4	5	4	2000::1:10:33:174:50	64	::	8	V6MedCtrl

The Routing Table:

Destination	Prefix Length	Subnet Mask	Gateway	Interface Name	Hop Count
201.201.0.0	16		10.31.174.1	ManagementIF	1

Left Blank

The Gateway address resides on the subnet configured in interface Index 0 at the Interface Table. The Next Hop will be accessible via Interface 0.

Note: The Interface Address family must be coherent with the Routing Rule Address family. IPv4 interfaces cannot be selected in an IPv6 routing rule, and vice versa.

4.2.2.5 Metric Column

This column MUST be set to 1 for each routing rule.

4.2.2.6 Status Column

This column holds the status of each static route. Possible values are either '*Active*' or '*Inactive*'. When the destination IP Address is not on the same segment as the next hop or the interface doesn't exist, the Route state is changed to '*Inactive'.*

4.2.3 Routing Table Configuration Summary & Guidelines

Note: Routing Table configurations must adhere to the following rules:

- Up to 30 different routing rules may be defined.
- The Prefix Length replaces the dotted decimal Subnet Mask presentation. This column must have a value of 0-31 for IPv4 interfaces and a value of 64 for IPv6 interfaces.
- The Gateway IP Address must be available on one of the local subnets.
- The Interface selected in the routing rule, must be of the same address family as the rule defined.
- The Metric column must be set to 1.
- Network Configuration changes are offline. The new configuration should be saved and will be available at the next startup.

4.2.4 Troubleshooting - Routing Table

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 rejected 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.
- More than 30 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 the device's User Manual for more information on the Web Interface.

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_InterfaceMode, InterfaceTable_IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable_VlanID, InterfaceTable_InterfaceName;

InterfaceTable 0  = 6, 10, 192.168.85.14,  16, 192.168.0.1, 1,
myAll;
[\InterfaceTable]


; VLAN related parameters:
VlanMode = 0
VlanNativeVlanId = 1


 StaticRouteTable ]
FORMAT StaticRouteTable_Index = StaticRouteTable_InterfaceName,
StaticRouteTable_Destination, StaticRouteTable_PrefixLength,
StaticRouteTable_Gateway, StaticRouteTable_Description;
StaticRouteTable 0 = O+M+C, 201.201.0.0, 16, 192.168.0.2, desc1;
StaticRouteTable 1 = Media, 202.202.0.0, 16, 192.168.0.3, desc2;
 [ \StaticRouteTable ]



; 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, OAMP, Media and Control applications are allowed) and a default gateway (192.168.0.1)
- 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; and the NTP application is configured to act as an OAMP application.

Notes:

- Lines that begin with a semicolon are considered a remark and are ignored.
- The Interface Table configuration via *ini* file, **MUST** have the 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 OAMP, Media and Control applications:

Interface Table - Example 1

Index	Allowed Application Types	Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	OAMP, Media & Control	IPv4 Manual	192.168.85.14	16	192.168.0.1	1	myInterface

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 Name	Metric
201.201.0.0	16		192.168.0.2	O+M+C	1
202.202.0.0	16		192.168.0.3	Media	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_InterfaceMode, InterfaceTable_IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable_VlanID, InterfaceTable_InterfaceName;

InterfaceTable 0  = 6, 10, 192.168.85.14,  16, 192.168.0.1, 1,
myAll;
[\InterfaceTable]


[ StaticRouteTable ]
FORMAT StaticRouteTable_Index = StaticRouteTable_InterfaceName,
StaticRouteTable_Destination, StaticRouteTable_PrefixLength,
StaticRouteTable_Gateway, StaticRouteTable_Description;
StaticRouteTable 0 = O+M+C, 201.201.0.0, 16, 192.168.0.2, desc1;
StaticRouteTable 1 = Media, 202.202.0.0, 16, 192.168.0.3, desc2;
 [ \StaticRouteTable ]
```

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:

Interface Table - Example 2

Index	Allowed Application Types	Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	OAMP	IPv4 Manual	192.168.85.14	16	0.0.0.0	1	ManagementIF
1	Control	IPv4 Manual	200.200.85.14	24	0.0.0.0	200	myControlIF
2	Media	IPv4 Manual	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 from a host in 176.85.49.0 / 24:

Destination	Prefix Length	Subnet Mask	Gateway	Interface Name	Metric
176.85.49.0	24		192.168.0.1	ManagementIF	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_InterfaceMode, InterfaceTable_IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable_VlanID, InterfaceTable_InterfaceName;

InterfaceTable 0  = 0, 10, 192.168.85.14,  16, 0.0.0.0, 1,
ManagementIF;
InterfaceTable 1  = 2, 10, 200.200.85.14,  24, 0.0.0.0, 200,
myControlIF;
InterfaceTable 2  = 1, 10, 211.211.85.14,  24, 211.211.85.1, 211,
myMediaIF;
[\InterfaceTable]


; VLAN related parameters:
VlanMode = 1
VlanNativeVlanId = 1


[ StaticRouteTable ]
FORMAT StaticRouteTable_Index = StaticRouteTable_InterfaceName,
StaticRouteTable_Destination, StaticRouteTable_PrefixLength,
StaticRouteTable_Gateway, StaticRouteTable_Description;
StaticRouteTable 0 = ManagementIF, 176.85.49.0, 24, 192.168.0.1,
desc1;
 [ \StaticRouteTable ]
```

Example 3 - One interface exclusively for management (OAMP applications) and three others for Call Control and RTP (CONTROL and MEDIA applications):

The Interface Table will be configured with four interfaces. One is exclusively for Management purposes, and the three are for Call Control and RTP Media applications. Two of them are IPv4 interfaces and the third is an IPv6 interface:

Interface Table - Example 3

Index	Allowed Application Types	Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	OAMP	IPv4 Manual	192.168.85.14	16	0.0.0.0	1	Mgmt
1	Media & Control	IPv4 Manual	200.200.85.14	24	200.200.85.1	201	CntrlMedia1
2	Media & Control	IPv4 Manual	200.200.86.14	24	0.0.0.0	202	CntrlMedia2
3	Media & Control	IPv6 Manual	2000::1:200:200:86:14	64	::	202	V6CntrlMedia2

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 from a host in 176.85.49.0 / 24.

Destination	Prefix Length	Subnet Mask	Gateway	Interface Name	Metric
176.85.49.0	24		192.168.0.1	Mgmt	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_InterfaceMode, InterfaceTable_IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable_VlanID, InterfaceTable_InterfaceName;

InterfaceTable 0  = 0, 10, 192.168.85.14,  16, 0.0.0.0, 1, Mgmt;
InterfaceTable 1  = 5, 10, 200.200.85.14,  24, 200.200.85.1, 201,
CntrlMedia1;
InterfaceTable 2  = 5, 10, 200.200.86.14,  24, 0.0.0.0, 202,
CntrlMedia2;
InterfaceTable 3  = 5, 4, 2000::1:200:200:86:14,  64, ::, 202,
V6CntrlMedia2;


 [\InterfaceTable]


; VLAN related parameters:
VlanMode = 1
VlanNativeVlanId = 1


[ StaticRouteTable ]
```

```
FORMAT StaticRouteTable_Index = StaticRouteTable_InterfaceName,
StaticRouteTable_Destination, StaticRouteTable_PrefixLength,
StaticRouteTable_Gateway, StaticRouteTable_Description;
StaticRouteTable 0 = Mgmt, 176.85.49.0, 24, 192.168.0.1, desc1;
 [ \StaticRouteTable ]
```

4.3.3 Getting Started with the Mediant 3000 System in High Availability Mode

Note: This sub-section is only applicable to **Mediant 3000**.

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 blade in the Mediant 3000 system will boot as 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 for maintenance only (prior to entering HA mode).

The active blade will use the address configured by the networking IF table. The redundant blade will disconnect from the network once the system has gone into HA mode.

When the system is loading from BootP/TFTP (first configuration setup), HA is disabled. After resetting the Active blade from the Web Interface/EMS, it will load from flash and the HA will be activated.

4.3.3.1 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.

4.3.3.2 Configuring the Mediant 3000 for Multiple Interfaces via *ini* File

The following example shows a sample configuration of five interfaces, including IPv6 and VLANs.

First, allocate three IP Addresses to the system:

IP Address	Details
192.168.85.14 200.200.85.14 211.211.85.14	The Mediant 3000 Global IP Addresses. These addresses will be used as the Management, Control and Media IP Address of the system.
192.168.85.15	A private IP Address for the blade in slot #1 (received from BootP)
192.168.85.16	A private IP Address for the blade in slot #2 (received from BootP)

The Interface Table will be configured with three interfaces - one exclusively for each application type:

- OAMP Applications
- Call Control Applications
- 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 from a host in 176.85.49.0 / 24:

Destination	Prefix Length	Subnet Mask	Next Hop	Interface Name	Metric
176.85.49.0	24		192.168.0.1	ManagementIF	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
```

Ensure that the private addresses are not configured in the Interface Table.

The same *ini* file should be loaded to both blades.

4.3.3.3 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.

Notes:

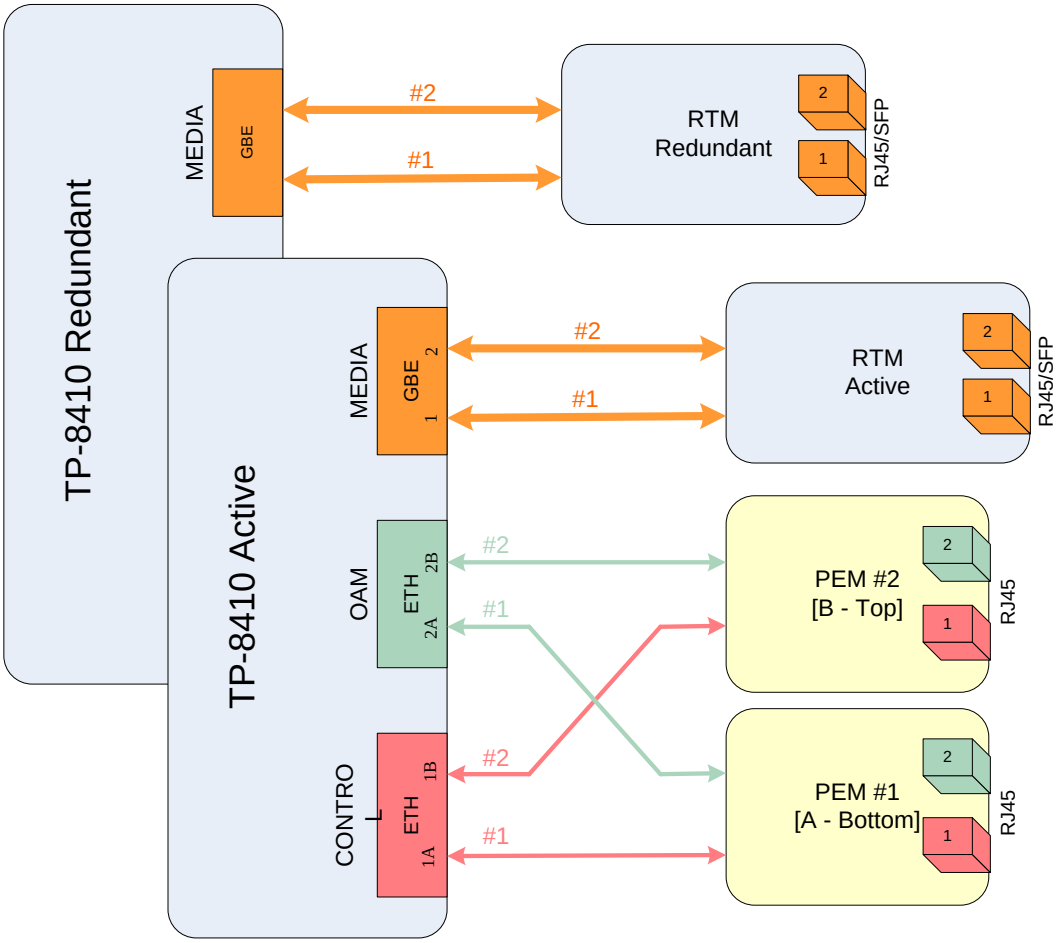
- Physical Network Separation currently supports three interfaces. All three interfaces must be configured.
- When working in this mode, OAMP traffic is sent and received from/to the designated physical port (on the PEM); this port is not the same port used when working with a single physical interface.

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: 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.

4.3.3.3.1 Configuring your System to the Separate Physical Interfaces Scheme using *.ini Files

➢ **To prepare the Mediant 3000 to work with Multiple Interfaces and network physical interfaces separation:**

1. Prepare an ini file with parameters as shown in 'Setting Up Your System via ini File' on page 291. Ensure that the *EnableNetworkPhysicalSeparation* ini file parameter has been added and is set to "1".
2. Insert only a single blade into the system. Each blade should be configured separately. It does not matter which blade is configured first.
3. 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.
4. Verify that the following message is sent to the Syslog: *"Updating Flash to work in Network Separation Mode in the next Boot".*
5. Repeat steps 3 to 5 for the second blade.
6. 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.
7. 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:**

1. Prepare an *ini* file with parameters t as shown in 'Setting Up Your System via ini File' on page 291.
   Ensure that the *"EnableNetworkPhysicalSeparation" ini* file parameter has been added and is set to "0".
2. Insert a single blade into the system. Each blade should be configured separately, while the other blade is not inserted into the Mediant 3000. It does not matter which blade is configured first.
3. Make sure your Ethernet cable is connected to the PEM.
4. 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.
5. Verify that the following message is sent to the Syslog: *"Updating Flash to work in Non Network Separation Mode in the next Boot".*
6. Repeat steps 5 to 7 with the second blade.
7. 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.

4.3.3.3.2 Configuring your System to the Separate Physical Interfaces Scheme using the Web Interface

Refer to the **Using Network Physical Separation** chapter of the Mediant 3000 and TP-8410 MGCP-MEGACO User's Manual, for more information.

5 PSTN

5.1 PSTN Description

This chapter describes a set of the PSTN protocols supported by the product, as well as the basic physical layer interfaces. Not all of the protocols or physical layer interfaces are available on all devices.

PSTN (Public Switched Telephone Network) is the network of the world's public circuit-switched telephone networks.

5.1.1 PSTN Protocols

PSTN protocols are different signaling techniques used for call control. They provide the user with a User-network signaling mechanism or signaling capabilities between various components of the telephone network.

- SS7 - Signaling System #7 is a set of telephony signaling protocols which are used to set up most of the world's public switched telephone network telephone calls. Please refer to 'SS7 Functionality & Configuration' on page 352 for more information.
- ISDN - Integrated Services Digital Network is an all digital communications line that allows voice, data and video to be transmitted at very high speeds over standard communication lines. Please refer to 'ISDN' on page 308 for more information.
- CAS / Robbed Bit Signaling - CAS (Channel Associated Signaling) is a method of carrying signaling information for many timeslots of digital trunk. Specific bits in the frame are allocated to specific timeslots. Please refer to 'CAS' on page 347 for more information.
- V5 - This is a standardized protocol suite for connection of Access Networks (AN) to the Local Exchange (LE), usually implemented in a traditional Class-5 switch. The Access Network itself typically has PSTN and/or ISDN user ports available for the customer. The V5.2 interfaces are based on G.703/G.704 interfaces at 2,048 kbit/s (E1). V5.1 is a single 2,048 kbit/s interface, whereas V5.2 may consist of up to 16 2,048 kbit/s links, configurable by the operator. Please refer to 'V5 Protocol' on page 449 for more information.

5.1.2 PSTN Physical Interfaces

Physical interfaces are different types of digital carriers for voice and signaling used to connect different pieces of PSTN equipment.

- E1 - Provides a 2.048 MHz electrical interface. This carrier is used in most of world, including Europe, Mexico, and South America.
- T1 / DS1 - Provides a 1.544 MHz electrical interface. This carrier is used in North America and Korea.
- J1 - Provides a 1.544 MHz electrical interface. This carrier is used in Japan.
- ISDN BRI - Provides a 192 KHz electrical interface, with actual bandwidth of 144 KHz. This carrier is used mostly in Europe.
- T3 / DS3 - Provides a 44.736 MHz electrical interface. This carrier is used in North America. The T3 electrical interface consists of two coaxial cables - one for transmission and one for receiving - at a line impedance of 75 Ohm. The product supports channelized DS3 transport and can bear up to 28 T1 virtual carriers inside single DS3 interface.

- STM-1 - Synchronous Transport Module level-1 is a basic unit of framing in the Synchronous Digital Hierarchy (SDH). Synchronous optical networking is used in Europe and operates at 155.52 Mbit/s. The device has a protected STM1 interface that bears 63 E1 virtual carriers inside it.
- OC3 - Optical Carrier level 3 is a network line with transmission speeds of 155.52 Mbit/s using fiber optics. OC-3 is defined in the Synchronous Optical NETwork (SONET). The device has a protected OC3 interface that bears 84 DS1 virtual carriers.

5.1.3 PSTN Low-Layer Applications

- Configuration
- Alarms
- Performance Monitoring
- SDH/SONET Automatic Protection Switch (APS)
- Clocking

Refer to 'PSTN Low-Layer Applications' on page 413 for more information.

5.2 Configuring the PSTN Interface and Protocols

5.2.1 Common E1 / T1 Trunk Parameters

Most of the PSTN interface and protocol configuration can be done using the following board configuration parameters. These parameters can be configured using the *ini* file, Web Interface or EMS GUI.

5.2.1.1 ProtocolType

The protocol type parameter defines the **interface** and the specific telephony **protocol**:

- E1, T1 or BRI physical *line type*
- Layer 3 specific telephony protocol or transparent operation mode

Note: All E1/T1 trunks must be of *the same line type* (E1 or T1). BRI trunks can work with both E1 or T1.

Protocols Available from AudioCodes

Protocol Type Values	Description
acPROTOCOL_TYPE_E1_EURO_ISDN	ISDN PRI Pan-European (CTR4) protocol with many ISDN information elements and call control messages accessible to users.
acPROTOCOL_TYPE_T1_CAS	Common T1 robbed bits protocols including E&M wink start, E&M immediate start, E&M delay dial/start and loop-start and ground start. Users can change the protocol parameters and even the entire state machine by using the CAS downloadable file.
acPROTOCOL_TYPE_T1_RAW_CAS	Signaling information such as ABCD robbed bits, in-band Call Progress Tones, and DTMF digit detection and generation, are accessible to users via the API events and functions. Users can implement call control or relay various events with their own host software. Users can control the default PCM and ABCD bits idle pattern via acOpenBoard().
acPROTOCOL_TYPE_T1_TRANSPARENT	Transparent protocol, where no signaling is to be provided by the TrunkPack series. Time slots 1 to 24 of all trunks are mapped to DSP channels (e.g., SS7 application).
acPROTOCOL_TYPE_E1_TRANSPARENT_31	Transparent protocol, where no signaling is to be provided by the TrunkPack series boards. Time slots 1 to 31 of each trunk are mapped to the DSP channels (e.g., SS7 application).

Protocols Available from AudioCodes

Protocol Type Values	Description
acPROTOCOL_TYPE_E1_TRANSPARENT_30	Transparent protocol, where no signaling is to be provided by the TrunkPack series boards. Time slots 1 to 31, excluding time slot 16 of all trunks, are mapped to DSP channels (e.g., SS7 application).
acPROTOCOL_TYPE_E1_MFCR2	Common E1 MFC/R2 CAS protocols (including line signaling and compelled register signaling). Script files library includes national variants such as China, Israel, Mexico, Philippines and more. Users can change the protocol parameters and even the entire state machine by editing the protocol textual table that is downloaded to the board on initiating acOpenBoard().
acPROTOCOL_TYPE_E1_CAS	Common E1 CAS protocols (including line signaling and MF/DTMF address transfer) Script files library (related to E1) including R2D and R2D modified variants. Users can change the protocol parameters and even the entire state machine by using the CAS downloadable file.
acPROTOCOL_TYPE_E1_RAW_CAS	Signaling information such as ABCD line signaling (time-slot 16), In-Band Call Progress Tones, and DTMF digits detection and generation, are accessible to Users via the API events and functions. Users can implement call control or relay various events with their own host software. Users can control the default PCM and ABCD bits idle pattern via *ini* file parameters IdlePcmPattern and IdleAbcdPattern.
acPROTOCOL_TYPE_T1_NI2_ISDN	NATIONAL ISDN 2 PRI protocol with many ISDN information elements and call control messages accessible to users via the PSTN library API.
acPROTOCOL_TYPE_T1_4ESS_ISDN	ISDN PRI protocol for the Lucent™/AT&T™ 4ESS switch with many ISDN information elements and call control messages accessible to Users via the PSTN library API.
acPROTOCOL_TYPE_T1_5ESS_9_ISDN	ISDN PRI protocol for the Lucent™/AT&T™ 5ESS-9 switch with many ISDN information elements and call control messages accessible to users via the PSTN library API. (refer to the Note below)
acPROTOCOL_TYPE_T1_5ESS_10_ISDN	ISDN PRI protocol for the Lucent™/AT&T™ 5ESS-10 switch with many ISDN information elements and call control messages accessible to Users via the PSTN library API. (refer to the Note below)

Protocols Available from AudioCodes

Protocol Type Values	Description
acPROTOCOL_TYPE_T1_DMS100_ISDN	ISDN PRI protocol for the Nortel™ DMS switch with many ISDN information elements and call control messages accessible to users via the PSTN library API.
acPROTOCOL_TYPE_J1_TRANSPARENT	-
acPROTOCOL_TYPE_T1_NTT_ISDN	ISDN PRI protocol for the Japan - Nippon Telegraph Telephone (known also as INS 1500)
acPROTOCOL_TYPE_E1_AUSTEL_ISDN	ISDN PRI protocol for the Australian Telecom
acPROTOCOL_TYPE_E1_HKT_ISDN	ISDN PRI protocol for the Hong Kong - HKT
acPROTOCOL_TYPE_E1_KOR_ISDN	ISDN PRI protocol for Korean Operator (similar to ETSI)
acPROTOCOL_TYPE_T1_HKT_ISDN	ISDN PRI protocol for the Hong Kong - HKT
acPROTOCOL_TYPE_E1_QSIG	ECMA 143 QSIG over E1
acPROTOCOL_TYPE_E1_TNZ_ISDN	ISDN PRI protocol for Telecom New Zealand (similar to ETSI)
acPROTOCOL_TYPE_T1_QSIG	ECMA 143 QSIG over T1
acPROTOCOL_TYPE_T1_IUA or acPROTOCOL_TYPE_E1_IUA	Relay of the ISDN signaling messages using SIGTRAN IUA and SCTP protocols (refer to 'IUA/DUA' on page 393 for details on IUA protocol).
acPROTOCOL_TYPE_E1_FRENCH_VN6_ISDN	France Telecom VN6
acPROTOCOL_TYPE_E1_FRENCH_VN3_ISDN	France Telecom VN3
acPROTOCOL_TYPE_T1_EURO_ISDN	ISDN PRI protocol for EURO over T1
acPROTOCOL_TYPE_T1_DMS100_MERIDIAN_ISDN	ISDN PRI protocol for the Nortel™ DMS Meridian switch. This protocol includes numerous ISDN information elements and call control messages accessible to users via the PSTN API.
acPROTOCOL_TYPE_T1_NI1_ISDN	National ISDN 1 PRI protocol with numerous ISDN information elements and call control messages accessible to users via the PSTN library API.
acPROTOCOL_TYPE_E1_DUA	Relay of DPNSS layer 3 messages using DUA extension for IUA layer (refer to 'IUA/DUA' on page 393 for details on IUA and DUA protocols).
acPROTOCOL_TYPE_E1_NI2_ISDN	NATIONAL ISDN 2 PRI protocol over E1
acPROTOCOL_TYPE_V5	V5 protocol according to ETS 300 324-1 and ETS 300 347-1
acPROTOCOL_TYPE_BRI_EURO_ISDN	EURO_ISDN over BRI

Protocols Available from AudioCodes

Protocol Type Values	Description
acPROTOCOL_TYPE_BRI_QSIG	QSIG over BRI
acPROTOCOL_TYPE_BRI_FRENCH_VN6_ISDN	VN6 over BRI
acPROTOCOL_TYPE_BRI_NTT_ISDN	NTT over BRI

Notes:

- E1 and T1 protocol families cannot be configured together on the same Digital Media Gateway.
- The *acPROTOCOL_TYPE_T1_5ESS_9_ISDN* variant and the *acPROTOCOL_TYPE_T1_5ESS_10_ISDN* variant are identical.
- Setting *ProtocolType* to *acPROTOCOL_TYPE_NONE* causes Loss of Signal on the remote side.

5.2.1.2 FramingMethod

The *FramingMethod* parameter selects the physical framing method for each trunk. For T1, there are two primary Framing Methods:

- Super Frame (SF 12-frame multi-frame)
- Extended Super Frame (ESF 24-frame multi-frame)

E1 framing is assumed to be 16-frame multi-frame with CRC4. The default value is acEXTENDED_SUPER_FRAME for E1,T1 and J1.

Notes:

- In T1, the default value (acEXTENDED_SUPER_FRAME) is converted to acT1_FRAMING_ESF_CRC6.
- In E1, the default value (acEXTENDED_SUPER_FRAME) is converted to acE1_FRAMING_MFF_CRC4_EXT (automatic mode, only if CRC is identified in the Rx, it sends CRC in the Tx, otherwise no CRC).
- In J1, the FramingMethod is always set to acT1_FRMAING_ESF_CRC6_JT.
- For other values, consult with AudioCodes FAE personnel.

5.2.1.3 LineCode

Line codes for T1 and E1 trunks (according to acTLineCode enumerator). Used to select B8ZS or AMI for T1 spans and HDB3 or AMI for E1 spans.

5.2.1.4 LineBuildOut.LOSS

This parameter is used to control the loss for different lengths of the line (according to acTLineBuildOutLOSS enumerator): 0 dB, -7.5 dB, -15 dB, or -22.5 dB. This parameter is applicable only in T1 trunks.

5.2.1.5 LineBuildOut.OverWrite

The overwrite parameter, which is zero by default (no overwrite), enables users to write to the 3 Pulse Mask (XPM) registers, thus controlling the Trunk analog pulse shape (applicable to E1 and T1 trunks).

5.2.1.6 TraceLevel

The protocol trace level per-trunk when ACL debug record is active. After activating the debug record, the TraceLevel filters the type of messages which it sends to the debug-record host. For more information on TraceLevel, refer to 'PSTN Parameters' on page 210.

For Clock Management parameters please refer to 'Infrastructure Parameters' on page 178.

For ISDN Specific Trunk's Parameters please refer to 'ISDN Specific Trunk's Parameters' on page 311.

5.3 ISDN

The following describes ISDN.

5.3.1 ISDN Overview

ISDN is a major advance in the provision of an international standard for the integration of audio, data, fax, signaling, video, voice, and their digital transmission over high speed lines.

ISDN currently exists in two forms: Basic Rate Interface (BRI) and Primary Rate Interface (PRI), both based on a 64 kbps (kilobits per second) module called the B-channel.

PRI, also called Primary Rate Access (PRA), combines multiple B-channels within T1 or E1 PCM (Pulse Code Modulated) trunks to provide a high speed service. Both E1 and T1 structures are based on multiples of the 64 kbps B-channels, but with a different framing structure.

With T1 trunks operating in North America and Japan at 1.544 Mbps, 23 B-channels and one 64 kbps D-channel are provided (23B+D). The T1 structure consists of a further 8 kbps for framing, providing a total of 1.544 Mbps.

With E1 trunks operating in Europe and other parts of the world at 2.048 Mbps, 30 B-channels and one 64 kbps D-channel (time-slot 16) are provided (30B+D). The E1 structure consists of a further 64 kbps in time-slot 0 for framing, providing a total of 2.048 Mbps.

ISDN BRI consists of two 64 kbps B-channels for the transmission of voice, data and other end-user information, and a 16 kbps D-channel for data and control, multiplexed together in a digital bidirectional 144 kbps bit stream.

5.3.1.1 OSI Seven Layer Protocol Stack

The ITU Telecommunication Standardization Sector (ITU-T), one of the three sectors of the International Telecommunication Union (ITU), has promoted the Open Systems Interconnection (OSI) reference model (OSI-RM) for several years.

The OSI-RM is a 7-layer protocol stack based on the structure originally designed by the International Standards Organization (ISO), providing standardized software functionality of the layers and of the interfaces between them.

Having been developed for data communications, adopted for telephony and now extended for Internet Protocol (IP) communication, the stack can have several standards applying to an individual layer. So it is important to ensure functional compatibility within the software (refer to the OSI Seven Layer Stack figure below).

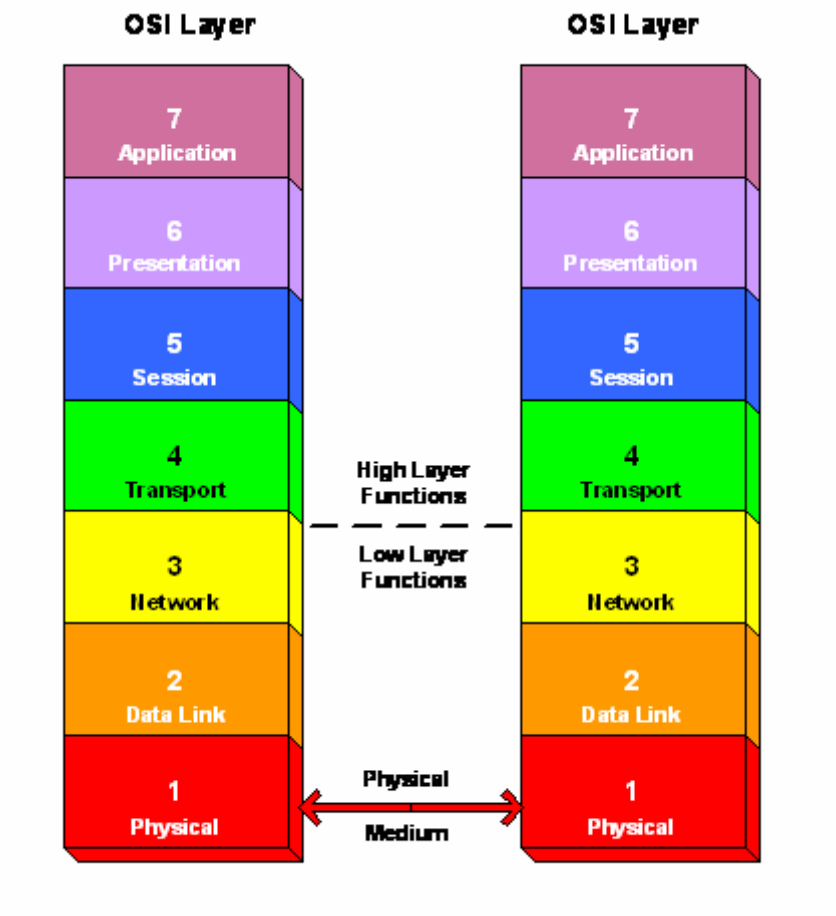
- The Physical Layer converts between the signal of the physical medium and the data stream of the Data Link Layer. For the ISDN PRI and BRI protocols, ITU-T Recommendations I.430, I.431 apply.
- The Data Link Layer presents the data to the higher layers in an organized, error-free manner. The Data Link Layer protocol is called the Line Access Protocol for the D-channel (LAPD) and ITU-T Recommendations I.440, I.441 and Q.920, Q.921 apply.
- The Network Layer provides transparency between the Data Link Layer below and the Transport layer above. The Network Layer protocol is called the Call Control and also provides control and routing functionality. For this layer, ITU-T Recommendations I.450, I.451 and Q.930, Q.931 apply.

The above three layers of the OSI model are referred to as the 'low layer functions'. The following four layers are referred to as the 'high layer functions'.

- The Transport Layer provides connection parameters for the 'transparent network' according to the requirement of the session layer above.

- The Session Layer provides interaction between the Applications in layer 7 presented to it from the Presentation layer above it.
- The Presentation Layer presents the Applications in layer 7 and the network Session in layer 5 to each other, without interference from protocol or syntax.
- The Application Layer at the top of the stack contains the user application.

Figure 5: OSI Seven Layer Stack



5.3.1.2 ISDN Variants

ISDN has been implemented in many different switches by many different vendors. Some of these are well defined and have a specification. Below are the ISDN variants supported by AudioCodes' devices.

5.3.1.2.1 Supported Variants

Supported Variants	
acPROTOCOL_TYPE_E1_EURO_ISDN	= 1
acPROTOCOL_TYPE_T1_NI2_ISDN	= 10
acPROTOCOL_TYPE_T1_4ESS_ISDN	= 11
acPROTOCOL_TYPE_T1_5ESS_9_ISDN	= 12 (refer to the Note below)
acPROTOCOL_TYPE_T1_5ESS_10_ISDN	= 13 (refer to the Note below)
acPROTOCOL_TYPE_T1_DMS100_ISDN	= 14

Supported Variants	
acPROTOCOL_TYPE_T1_NTT_ISDN	= 16 (Japan – Nippon TT)
acPROTOCOL_TYPE_E1_AUSTEL_ISDN	= 17 (Australian Telecom)
acPROTOCOL_TYPE_E1_HKT_ISDN	= 18 (Hong Kong – HKT)
acPROTOCOL_TYPE_E1_KOR_ISDN	= 19 (Korean Operator)
acPROTOCOL_TYPE_T1_HKT_ISDN	= 20 (European Hong Kong – HKT over T1)
acPROTOCOL_TYPE_E1_QSIG	= 21 (QSIG)
acPROTOCOL_TYPE_E1_TNZ_ISDN	= 22 (Telecom New Zealand)
acPROTOCOL_TYPE_T1_QSIG	= 23 (QSIG over T1)
acPROTOCOL_TYPE_E1_IUA	= 28 (IUA for E1)
acPROTOCOL_TYPE_T1_IUA	= 29 (IUA for T1)
acPROTOCOL_TYPE_E1_FRENCH_VN6_ISDN	= 30 (French VN6)
acPROTOCOL_TYPE_E1_FRENCH_VN3_ISDN	= 31 (French VN3)
acPROTOCOL_TYPE_T1_EURO_ISDN	= 34 (ETSI over T1)
acPROTOCOL_TYPE_T1_DMS100_MERIDIAN_ISDN	= 35
acPROTOCOL_TYPE_T1_NI1_ISDN	= 36 (NI1)
acPROTOCOL_TYPE_E1_DUA	= 37 (E1 DUA)
acPROTOCOL_TYPE_E1_NI2_ISDN	= 40 (NI2 over E1)
acPROTOCOL_TYPE_BRI_EURO_ISDN	= 50 (EURO_ISDN over BRI)

Note: The acPROTOCOL_TYPE_T1_5ESS_9_ISDN variant and the acPROTOCOL_TYPE_T1_5ESS_10_ISDN variant are identical.

AudioCodes supports ISDN Hot Redundancy.

5.3.2 ISDN PRI

The ISDN protocols support the following functions:

- ISDN call control.
- En-bloc and overlap dialing modes.
- Dealing with incoming and outgoing call collisions.
- Dealing with clearing call collisions.
- Different E1/T1 variants on the same blade.

5.3.3 ISDN Specific Trunk's Parameters

- Additional ISDN Behavior configuration parameters (Refer to 'ISDN Flexible Behavior' on page 317 for more details.)
- **DCHConfig:** D-channel configuration setting is applicable only to ISDN PRI protocols that support the NFAS and/or D-channel backup procedures. Only US variants - NI2, DMS, 4ESS and 5ESS-10 are supported for, and affected by, this configuration. The following combinations, in the table below are allowed:

PSTN Interface and Protocol Configuration Example for PRI Trunks

TrunkId 0	TrunkId 1	TrunkId 2	TrunkId 3	Description
Primary	Primary	Primary	Primary	Normal and default configuration for PRI protocols for 23 bearer channels in T1 trunks / 30 bearer channels in E1 trunks, and one CCS channel for each span.
Primary	NFAS	NFAS	NFAS	NFAS mode using the first span (Id=0) carrying the CCS channel and used to set-up calls for all four spans. Thus, the other three spans (ID=1,2,3) each supports 24 bearer channels in T1 trunks / 31 bearer channels in E1 trunks.
Primary	BACKUP	NFAS	NFAS	DCH_BACKUP mode using the first span's CCS has a primary signaling channel setting up the calls for both spans. The 2nd span supports 23 bearer channels in T1 trunks / 30 bearer channels in E1 trunks and its CCS channel is used as a back-up or stand-by channel. In the case of malfunction, the 2nd span changes roles with the 1st span and becomes the primary span. The 3rd and 4th spans each support 24 bearer channels in T1 trunks / 31 bearer channels in E1 trunks.

- **NFAS Group** – Several trunks can be controlled by a single D-channel (Primary). Each NFAS trunk must belong to a group. This parameter assigns each NFAS trunk to a specific group. Valid values: 0 = does not belong to an NFAS group, 1-9 = belongs to the group number 1-9.

  For detailed information and for additional ISDN NFAS configuration parameters, refer to 'ISDN NFAS (PRI only)' on page 313.

Note: This feature is supported only for NFAS.

5.3.4 ISDN BRI

Note: For supported American variants over BRI, please contact your nearest AudioCodes representative.

5.3.4.1 BRI Characteristics

Layer2 Mode:

A BRI port can be configured to work either in Point-to-Point or in Point–to–Multipoint mode. The Point-to-Point configuration is selected if the user intends to connect a single device/application to the ISDN port. Otherwise, a Point–to–Multipoint mode is selected.

A BRI port configured as a user always works in Point-to-Point mode.

A BRI port configured as a network can work either in Point-to-Point or in Point–to–Multipoint mode.

TEI Assignment:

TEI (Terminal endpoint identifier) is a number to uniquely identify each device directly connected to the BRI port.

TEI Assignment is the procedure applied in order to assign a TEI value to an ISDN phone.

The TEI is 0 to 63 for fixed assignment and 64 to 126 for automatic assignment.

In AudioCodes implementation, an interface in Network mode always works in Point-to-Multipoint Automatic TEI assignment. This option is usually used in cases where the interface (BRI port) has one or more devices connected to it.

When configuring a Point-to-Point mode, the static (fixed) TEI assignment applies.

5.3.4.2 SPID (ITU-T Recommendation Q.932)

The SPID (Service Profile Identifier) is required in ISDN BRI American variants, (DMS100, NI2 and 5ESS) for call establishment. This feature is required in Point to Multipoint mode, on the network side only.

The Mediant 1000 blade supports the SPID feature on ISDN-BRI trunks when configured appropriately.

Only ISDN terminals that have their TEI dynamically assigned (64-126) require terminal identification procedures. Such terminals must perform the SPID initialization procedure by sending the SPID value to the Mediant 1000 blade – which functions as the network. It then replies by sending a specific Endpoint Identifier (EID) back to the terminal.

Once an ISDN terminal has completed its initialization, the network can directly address it by including "any EID" in SETUP messages.

Notes:

- Each protocol variant has a different SPID format. (e.g., see the NI2 SPID format in the *BellCore GR-2941-CORE* specification.)
- The Mediant 1000 does not consider these formats as they are transmitted transparently.

5.3.5 ISDN NFAS (PRI only)

With ISDN Non-Facility Associated Signaling (NFAS) you can use a single D channel to control multiple PRI interfaces. A backup D channel can also be configured for use when the primary NFAS D channel fails.

5.3.5.1 Benefits

On a ISDN trunk, one channel carries signaling for all B channels. This is called the D channel. The D channel is typically carried in channel 24 on T1 trunks and in channel 16 on E1 trunks. Use of a single D channel to control multiple PRI interfaces can free one B channel on each interface to carry other traffic (voice, for example).

If a D-channel backup is configured, then any failure of the date link connection in the primary
D-signaling channel results in an immediate switchover to the backup D channel without disconnecting currently connected users.

Note: If a backup D-channel is configured, its D-channel is unavailable to carry traffic like B-channel; therefore, only those trunks that are neither primary nor backup can have all channels available for traffic.

5.3.5.2 NFAS Implementation Limitations

- Maximum NFAS groups: 12 (Valid NFAS group numbers are only 1 to 12)
- Maximum trunks in NFAS group: 10

5.3.5.3 List of Terms

NFAS trunk	PRI channel group configured to have no NFAS D channel; all its channels are B channels. This particular trunk uses the D channel configured in the primary trunk of its NFAS group for signaling.
NFAS group	PRI channel group (a group of interfaces) under control of a single D channel. The channel group can include all the ISDN channels on multiple trunks.
NFAS member	PRI interface in an NFAS group.

5.3.5.4 ISDN Variants that Support NFAS

Following are the ISDN variants that support NFAS:

- acPROTOCOL_TYPE_T1_NI2_ISDN = 10
- acPROTOCOL_TYPE_E1_NI2_ISDN = 40 NI2 over E1
- acPROTOCOL_TYPE_T1_4ESS_ISDN = 11
- acPROTOCOL_TYPE_T1_5ESS_9_ISDN = 12
- acPROTOCOL_TYPE_T1_5ESS_10_ISDN = 13
- acPROTOCOL_TYPE_T1_DMS100_ISDN = 14
- acPROTOCOL_TYPE_T1_NTT_ISDN = 16 Japan – Nippon TT
- acPROTOCOL_TYPE_T1_HKT_ISDN = 20 European Hong Kong – HKT over T1

- acPROTOCOL_TYPE_T1_DMS100_MERIDIAN_ISDN = 35
- acPROTOCOL_TYPE_T1_NI1_ISDN = 36 NI1

5.3.5.5 Using ISDN NFAS Blade Parameters

The following are the prerequisites for configuring NFAS:

The trunk is configured as NFAS in: BladeParam.TrunkConfig[i].NfasGroupNumber. The range is 0 to 9 (default = 0 = not NFAS trunk). Valid NFAS group numbers are only 1 to 9. Zero (0) indicates that this trunk is not NFAS (in this case, parameters ISDNNFASInterfaceID and DchConfig are ignored).

- The D-channel configuration is defined in the data structure BladeParam.TrunkConfig[i].DchConfig.
- Possible values are:
  - acDCH_CONFIG_PRIMARY = 0, (Default)
  - acDCH_CONFIG_BACKUP = 1
  - acDCH_CONFIG_NFAS = 2
- Primary Trunk refers to the D-channel is on that trunk.
- Backup Trunk refers to the backup D-channel is on that trunk.
- NFAS Trunk refers to a trunk whose channels are all B-channels.
- The NFAS group members can be defined without any order restriction (for on-the-fly NFAS configuration, refer to Section , '', on page . One of them must be configured as the PRIMARY Trunk and has the D-channel in it. As an option, another trunk may be configured as BACKUP Trunk. The rest of the group are defined as NFAS Trunks.
- Example configuration of an NFAS group of 3 trunks (trunks 0,2,3):

```
BladeParams.TrunkConfig[0].NfasGroupNumber = 1
BladeParams.TrunkConfig[2].NfasGroupNumber = 1
BladeParams.TrunkConfig[3].NfasGroupNumber = 1
BladeParams.TrunkConfig[0].DchConfig = acDCH_CONFIG_PRIMARY
BladeParams.TrunkConfig[2].DchConfig = acDCH_CONFIG_NFAS
BladeParams.TrunkConfig[3].DchConfig = acDCH_CONFIG_NFAS
```

- Example configuration of 2 NFAS groups of 3 trunks each:

```
BladeParams.TrunkConfig[1].NfasGroupNumber = 2
BladeParams.TrunkConfig[2].NfasGroupNumber = 1
BladeParams.TrunkConfig[3].NfasGroupNumber = 2
BladeParams.TrunkConfig[4].NfasGroupNumber = 2
BladeParams.TrunkConfig[5].NfasGroupNumber = 1
BladeParams.TrunkConfig[6].NfasGroupNumber = 1
BladeParams.TrunkConfig[1].DchConfig = acDCH_CONFIG_NFAS
BladeParams.TrunkConfig[2].DchConfig = acDCH_CONFIG_PRIMARY
BladeParams.TrunkConfig[3].DchConfig = acDCH_CONFIG_NFAS
BladeParams.TrunkConfig[4].DchConfig = acDCH_CONFIG_PRIMARY
BladeParams.TrunkConfig[5].DchConfig = acDCH_CONFIG_NFAS
BladeParams.TrunkConfig[6].DchConfig = acDCH_CONFIG_NFAS
```

5.3.5.6 ISDN NFAS Members

In the PRI messages, there is a Channel ID that identifies the trunk and B-channel.

The NFAS Member is related through an 'Interface ID'.

On AudioCodes' blades, the default for the interface ID is the number of the TrunkId.

The Interface ID must be provisioned and agreed on both sides (by the switch technician and the TrunkPack technician) before operating the PRI.

5.3.5.7 Changing the Interface ID in the Blade Parameters

To provision the Interface ID, the following must be performed:

A 'Pseudo Code' Example

```
blade_params.TrunkConfig[i].ProtocolSpecific.ISDNTrunk.ISDN_NFAS_I
nterfaceID = V_Int_Id;
where V_Int_Id is the interface Id you wish to provision for Trunk
i
blade_params.TrunkConfig[i].ProtocolSpecific.ISDNTrunk.ISDNQ931Lay
erResponseBehavior =
(ValueBitFild  |  NS_EXPLICIT_INTERFACE_ID)

where: ValueBitFild is your bit field of
"ISDNQ931LayerResponseBehavior" field
and:   NS_EXPLICIT_INTERFACE_ID is the FLAG for enabling this
feature

and:        #define NS_EXPLICIT_INTERFACE_ID  0x0200
```

An *ini* File Example (NFAS group of 4 Trunks)

```
NFASGroupNumber_0 = 1
NFASGroupNumber_1 = 1
NFASGroupNumber_2 = 1
NFASGroupNumber_3 = 1

DchConfig_0 = 0
DchConfig_1 = 2
DchConfig_2 = 2
DchConfig_3 = 2

ISDNNFASInterfaceID_0 = 0
ISDNNFASInterfaceID_1 = 1
ISDNNFASInterfaceID_2 = 2
ISDNNFASInterfaceID_3 = 3

ISDNIBehavior_0 = 512
ISDNIBehavior_1 = 512
ISDNIBehavior_2 = 512
ISDNIBehavior_3 = 51
```

5.3.5.8 Developing an Application with the DMS100 Switch

To develop an application with the DMS100 switch, get acquainted with the following DMS100 Interface Identifier (IID) conventions:

- IID 0 on Trunk with primary D-channel
- IID 1 on Trunk with backup D-channel
- IID 2 on Trunk with 24 channels in consecutive order
- IID 3 on Trunk with 24 channels in consecutive order

If the backup D-channel is not used, do not use IID 1.

Example Configuration of 2 NFAS Groups of 4 Trunks Each

```
BladeParams.TrunkConfig[0].NfasGroupNumber = 1
```

```
BladeParams.TrunkConfig[1].NfasGroupNumber = 1
BladeParams.TrunkConfig[2].NfasGroupNumber = 1
BladeParams.TrunkConfig[3].NfasGroupNumber = 1
BladeParams.TrunkConfig[4].NfasGroupNumber = 2
BladeParams.TrunkConfig[5].NfasGroupNumber = 2
BladeParams.TrunkConfig[6].NfasGroupNumber = 2
BladeParams.TrunkConfig[7].NfasGroupNumber = 2

BladeParams.TrunkConfig[0].DchConfig = acDCH_CONFIG_PRIMARY
BladeParams.TrunkConfig[1].DchConfig = acDCH_CONFIG_NFAS
BladeParams.TrunkConfig[2].DchConfig = acDCH_CONFIG_NFAS
BladeParams.TrunkConfig[3].DchConfig = acDCH_CONFIG_NFAS
BladeParams.TrunkConfig[4].DchConfig = acDCH_CONFIG_PRIMARY
BladeParams.TrunkConfig[5].DchConfig = acDCH_CONFIG_BACKUP
BladeParams.TrunkConfig[6].DchConfig = acDCH_CONFIG_NFAS
BladeParams.TrunkConfig[7].DchConfig = acDCH_CONFIG_NFAS

/* set the bit-flag  to force the interface ID , and not use the
default value */
/* where: ValueBitFild is your bit field of
"ISDNQ931LayerResponseBehavior" field */
for (int I=0;I<MAX_TRUNK;I++)
{

BladeParams.TrunkConfig[I].ProtocolSpecific.ISDNTrunk.ISDNQ931Laye
rResponseBehavior =
                                                    (ValueBitFild  |
NS_EXPLICIT_INTERFACE_ID);
}

/* set the IID values for the TrunkPack working with DMS100 */

BladeParams.TrunkConfig[0].ProtocolSpecific.ISDNTrunk.ISDN_NFAS_In
terfaceID = 0;
BladeParams.TrunkConfig[1].ProtocolSpecific.ISDNTrunk.ISDN_NFAS_In
terfaceID = 2;
BladeParams.TrunkConfig[2].ProtocolSpecific.ISDNTrunk.ISDN_NFAS_In
terfaceID = 3;
BladeParams.TrunkConfig[3].ProtocolSpecific.ISDNTrunk.ISDN_NFAS_In
terfaceID = 4;
BladeParams.TrunkConfig[4].ProtocolSpecific.ISDNTrunk.ISDN_NFAS_In
terfaceID = 0;
BladeParams.TrunkConfig[5].ProtocolSpecific.ISDNTrunk.ISDN_NFAS_In
terfaceID = 2;
BladeParams.TrunkConfig[6].ProtocolSpecific.ISDNTrunk.ISDN_NFAS_In
terfaceID = 3;
BladeParams.TrunkConfig[7].ProtocolSpecific.ISDNTrunk.ISDN_NFAS_In
terfaceID = 4;
```

5.3.5.9 Related ini File Parameters

The following are the *ini* file parameters associated with this feature

- NFASGROUPNUMBER
- ISDNNFASINTERFACEID
- ISDNIBEHAVIOR
- DCHCONFIG

5.3.6 ISDN Flexible Behavior

ISDN has been implemented in many different switches and many different PBXs by different vendors. Although there is a well-defined specification for most variants, some implementations might vary from the specification. To provide the flexibility to also support those that do not always conform to the specification, some fields were added to AudioCodes' blade parameters.

The following fields can be configured for every trunk:

- BladeParams.TrunkConfig[i].ProtocolSpecific.ISDNTrunk.ISDNQ931LayerResponse Behavior
- BladeParams.TrunkConfig[i].ProtocolSpecific.ISDNTrunk.ISDNOutgoingCallsBehavior
- BladeParams.TrunkConfig[i].ProtocolSpecific.ISDNTrunk.ISDNIncomingCallsBehavior
- BladeParams.TrunkConfig[i].ProtocolSpecific.ISDNTrunk.ISDNGeneralCCBehavior
- BladeParams.TrunkConfig[i].ProtocolSpecific.ISDNTrunk.ISDNNSBehaviour2

Field	Description
ISDNOutgoingCallsBehavior	Bit-field used to determine several behavior options which influence how the ISDN Stack OUTGOING calls behave.
ISDNIncomingCallsBehavior	Bit-field used to determine several behavior options, which influence how the ISDN Stack INCOMING calls behave.
ISDNQ931LayerResponseBehavior	Bit-field used to determine several behavior options, which influence how the Q.931 protocol behaves.
ISDNGeneralCCBehavior	Bit-field used to determine several general CC behavior options.
ISDNNSBehaviour2	Bit-field used to determine several behavior options, which influence how the Q.931 protocol behaves (Network Specific Layer).

5.3.6.1 Using *ini* File Parameters

The following are the parameters in the *ini* file:

- ISDNOUTCALLSBEHAVIOR (= ISDNOutgoingCallsBehavior parameter)
- ISDNINCALLSBEHAVIOR (= ISDNIncomingCallsBehavior parameter)
- ISDNIBEHAVIOR (= ISDNQ931LayerResponseBehavior parameter)
- ISDNGENERALCCBEHAVIOR
- ISDNNSBehaviour2
- BriLayer2Mode (Indicates "Point-to-Point" or "Point-to-Multipoint" mode for BRI ports.

5.3.6.2 ISDNOutgoingCallsBehavior

- CC_USER_SENDING_COMPLETE bit:
  Automatic generation of the information element 'Sending-complete' in SETUP (as result of PlaceCall). When this bit is set, the blade does not automatically generate the information element 'Sending-complete' in the SETUP. The user application must request it in the function PlaceCall. If this bit is not set, the blade generates it automatically in the SETUP message only.
- CC_USE_MU_LAW bit:
  - bit set: CC sends G.711- μ-Law in outgoing voice calls
  - bit cleared: CC sends G.711-A-Law in outgoing voice calls.
  - Applicable only to the Korean variant.
- CC_DIAL_WITH_KEYPAD bit:
  When this bit is set, CC uses the KEYPAD IE to store the called number digits instead of the CALLED_NB IE. Only applicable to the KOR variant (Korean network). Useful for Korean switches that do not accept the Called_nb IE.
- CC_STORE_CHAN_ID_IN_SETUP bit:
  When this bit is set, CC forces the sending of a Channel-id IE in an outgoing SETUP message even if it is not required by the standard (i.e., it is optional only), and no Channel-id has been specified in the establishment request. This is useful to increase compatibility with switches that require it. On BRI lines, the Channel-id IE indicates 'any channel'. On PRI lines, it indicates a not yet used channel ID, preferred only.
- CC_USE_A_LAW bit:
  - bit set: CC sends G.711 A-Law in outgoing voice calls
  - bit cleared: CC sends default G.711 –Law in outgoing voice calls.
  - Applicable to E10 variant.
- CC_USER_CALLING_NB_TYPE_PLAN bit:
  When this bit is set, CC sends the type/plan fields of the calling_nb IE that are provided by the user, instead of forcing them according to the number of digits (local/isdn for 7 digits, national/isdn for 10 digits, and unknown/unknown for other), which is the default behavior. Applies to NI2, DMS and E10 only.
- CC_ACCEPT_IA5_NB bit:
  When this bit is set, the CC accepts any IA5 character in the called_nb and calling_nb strings, and isn't restricted to extended-digits only (0-9,*,#).

ISDNOutgoingCallsBehavior

ini File Field Name	Value	Description
CC_USER_SENDING_COMPLETE	0x0002	Automatic generation of the information element 'Sending-complete' in SETUP
CC_USE_MU_LAW	0x0010	μ-Law / A-Law in outgoing voice
CC_DIAL_WITH_KEYPAD	0x0080	Using the KEYPAD IE to store the called number digits
CC_STORE_CHAN_ID_IN_SETUP	0x0100	Force the sending of a Channel-id IE in an outgoing SETUP message
CC_USE_A_LAW	0x0200	μ-Law / A-Law in outgoing voice (applicable to the E10 variant)
CC_USER_CALLING_NB_TYPE_P	0x0400	The blade sends the type/plan fields of the

ISDNOutgoingCallsBehavior

ini File Field Name	Value	Description
LAN		calling_nb IE as provided by the user
CC_ACCEPT_IA5_NB	0x0800	Accepts any IA5 character in the source and destination number.

Examples:

BladeParams.TrunkConfig[i].ProtocolSpecific.ISDNTrunk.ISDNOutgoingCallsBehavior = 0x0002

Only CC_USER_SENDING_COMPLETE

BladeParams.TrunkConfig[i].ProtocolSpecific.ISDNTrunk.ISDNOutgoingCallsBehavior = 0x0012

CC_USER_SENDING_COMPLETE and CC_USE_MU_LAW

5.3.6.3 ISDNIncomingCallsBehavior

- CC_DATA_CONN_RS bit:
  Automatic answering on NOT TELEPHONY incoming calls:
  When this bit is set, the blade sends a CONNECT (answer) message on NOT TELEPHONY incoming calls.
- CC_VOICE_CONN_RS bit:
  Automatic answering on TELEPHONY incoming calls:
  When this bit is set, the blade sends a CONNECT (answer) message on TELEPHONY incoming calls.
- CC_CHAN_ID_IN_FIRST_RS bit:
  When this bit is set, the ISDN Stack forces the sending of the Channel-id Information Element in the first response to an incoming call indication. When this bit is not set, the ISDN stack forces it only if it is mandatory for the running operator, or if it is on the NT-side.
- CC_USER_SETUP_ACK bit:
  When this bit is set, the ISDN stack does NOT automatically generate a Setup ack. When this bit is not set, the ISDN Stack sends a Setup Ack when it detects overlap receiving in progress.
- CC_CHAN_ID_IN_CALL_PROCEED bit:
  When this bit is set, the ISDN stack forces the sending of the Channel-id Information Element (IE) in the Call-Proceeding message. When this bit is not set, the ISDN stack forces it only if it is mandatory for the running operator or if it is on NT-side.
- CC_PROGR_IND_IN_SETUP_ACK bit:
  When this bit is set, the blade automatically sends a Progress-Indicator #8 ('in-band tones/announcements available') in the Setup-Ack message that is sent back automatically by the blade when it receives and accepts a Setup message in overlap-receiving mode (e.g., no digit in the Setup), for Voice calls only (BC encoded as Speech or 3.1-Audio).This allows tone generation to the Terminals after the handset is off-hooked.
- CC_USER_SCREEN_INDICATOR bit:
  This bit is used when a SETUP message includes 2 calling numbers. When this bit is not set (default), the blade will choose the calling number which has a network screen indicator. If the bit is set, the calling number with user screen indicator is chosen. If both calling numbers have the same screen indicator, the first one is selected.

ISDNIncomingCallsBehavior

ini File Field Name	Value	Description
CC_DATA_CONN_RS	0x00000020	Sends a CONNECT (answer) message on NOT TELEPHONY incoming calls.
CC_VOICE_CONN_RS	0x00000040	Blade sends a CONNECT (answer) message on TELEPHONY incoming calls.
CC_CHAN_ID_IN_FIRST_RS	0x00000800	Send Channel-id Information Element in the first response to an incoming call indication.
CC_USER_SETUP_ACK	0x00001000	Do NOT automatically generate a Setup ack.
CC_CHAN_ID_IN_CALL_PROCEED	0x00002000	Forces Channel-id IE in the Call-Proceeding message.
CC_PROGR_IND_IN_SETUP_ACK	0x00010000	Sends Progress-Indicator #8 ('in-band tones / announcements available') in the Setup-Ack message.
CC_USER_SCREEN_INDICATOR	0x80000000	Handles a SETUP message that includes two calling numbers.

Examples:

- BladeParams.TrunkConfig[i].ProtocolSpecific.ISDNTrunk.ISDNIncomingCallsBehavior = 0x3800

CC_CHAN_ID_IN_FIRST_RS, CC_USER_SETUP_ACK and C_CHAN_ID_IN_CALL_PROCEED

- BladeParams.TrunkConfig[i].ProtocolSpecific.ISDNTrunk. ISDNIncomingCallsBehavior = 0x0830

CC_CHAN_ID_IN_FIRST_RS and CC_VOICE_CONN_RS and CC_DATA_CONN_RS

5.3.6.4 ISDNQ931LayerResponseBehavior

- NS_NO_STATUS_ON_UNKNOWN_IE bit:
  No sending of a STATUS message on receipt of an unknown IE. If this bit is set, the ISDN Stack does not generate a STATUS message after having received a message containing 1 or more unknown/unrecognized IE(s). This bit applies only for network variants for which the sending of STATUS, in this case, is optional (where recommendation says 'a STATUS MAY be returned').
- NS_NO_STATUS_ON_INV_OP_IE bit:
  No sending of a STATUS message on receipt of an optional IE with invalid content. If this bit is set, the ISDN Stack does not generate a STATUS message after having received a message containing 1 or more optional IE(s) with invalid content. This bit applies only for network variants for which the sending of STATUS, in this case, is optional (where recommendation says 'a STATUS MAY be returned').
- NS_ACCEPT_UNKNOWN_FAC_IE bit:
  Unknown/unrecognized Facility IE accepted: If this bit is set, the NS does not reject a received message containing a Facility IE that it does not recognize and acts as if the IE was correct. It applies in network variants where a complete ASN1 decoding is made on Facility IE.
- NS_SEND_USER_CONNECT_ACK bit:
  Only applicable when the configuration is EuroISDN, user-side outgoing call. If this bit is set, the blade sends a CONNECT_ACK message when a CONNECT is received.
- NS_EXPLICIT_INTERFACE_ID bit:
  Only applicable when the configuration is AT4, DMS, E10, NI2 or HKT, TE and NT2 sides or NFAS, both incoming and outgoing calls. If this bit is set, ISDN calls on NFAS have an interface ID different to the default one (refer to the NFAS Appendix). The Interface ID value is taken from the Bladeparam of the NFAS Member Trunk config. (i.e., blade_params.TrunkConfig[i].
  ProtocolSpecific.ISDNTrunk.ISDN_NFAS_InterfaceID).
- NS_ALWAYS_EXPLICIT bit:
  Only applicable when the configuration is AT4, DMS, E10 or NI2. If this bit is set, the channel ID is always set to EXPLICIT interface ID even if the B-channel is on the link bearing the
  D-channel.
- NS_ACCEPT_MU_LAW bit:
  Only applicable when the configuration is ETSI. If this bit is set, Mu-law (μ-Law) is also accepted. Otherwise only A-Law is accepted.
- NS_EXPLICIT_PRES_SCREENING bit:
  Only applicable when the configuration is ETSI. If this bit is set, the calling party number octet 3a is always present, even if presentation and screening have default values. Otherwise the usual rule is used.
- NS_STATUS_INCOMPATIBLE_STATE bit:
  Only applicable when the configuration is NI2, QSIG or ETSI. If this bit is set, the call is cleared on receipt of STATUS with incompatible state. Otherwise no action is taken.
- NS_STATUS_ERROR_CAUSE bit:
  Only applicable when the configuration is QSIG or ETSI. If this bit is set, the call can be cleared on receipt of STATUS according to cause value. Otherwise no action is taken.

- NS_ACCEPT_A_LAW bit:
  Only applicable when the configuration is E10. If this bit is set, A-Law is also accepted. Otherwise only μ-Law is accepted.
- NS_RESTART_INDICATION bit:
  If this bit is set, the event acEV_PSTN_RESTART_CONFIRM is generated on receipt of a RESTART message. Note that the default value of this bit is set.
- NS_FORCED_RESTART bit:
  If this bit is set, [on data link (re)initialization] NS forces RESTART sending if no call (for PRI only).
- NS_BRI_DL_ALWAYS_UP bit:
  Only applicable when the PH layer is BRI. If this bit is set, NS always tries to activate the DL when it is set to DeActive.
- NS_QSI_ENCODE_INTEGER bit:
  Only applicable when the configuration is QSIG.
  If this bit is set, INTEGER ASN.1 type is used in operator coding (compliant to new ECMA standards).
  Otherwise (default behavior), OBJECT IDENTIFIER ASN.1 type is used.
- NS_5ESS_NATIONAL bit:
  Only applicable when the configuration is E10.
  If this bit is set, NS runs the National mode of AT&T 5ESS for B channel maintenance.
  Otherwise, NS runs the Custom mode.

ISDNQ931LayerResponseBehavior

ini File Field Name	Value	Description
NS_NO_STATUS_ON_UNKNOWN_IE	0x00000001	Do not generate a STATUS message after receiving unknown Ies.
NS_NO_STATUS_ON_INV_OP_IE	0x00000002	Do not generate a STATUS message after receiving a bad optional Ies.
NS_ACCEPT_UNKNOWN_FAC_IE	0x00000004	Unknown/unrecognized Facility IE accepted
NS_SEND_USER_CONNECT_ACK	0x00000080	Send a CONNECT_ACK message when receiving a CONNECT
NS_EXPLICIT_INTERFACE_ID	0x00000200	ISDN calls on NFAS have an interface ID different to the default.
NS_ALWAYS_EXPLICIT	0x00000800	Always set the channel ID to EXPLICIT interface ID even if the B-channel is on the link bearing the D-channel.
NS_ACCEPT_MU_LAW	0x00008000	Mu-Law is also accepted in ETSI.
NS_EXPLICIT_PRES_SCREENING	0x00010000	Calling party number octet is always presented.
NS_STATUS_INCOMPATIBLE_STATE	0x00020000	Clear call on receipt of STATUS with incompatible state.
NS_STATUS_ERROR_CAUSE	0x00040000	Clear call on receipt of STATUS according to cause value.
NS_ACCEPT_A_LAW	0x00080000	A-Law is also accepted in 5ESS.
NS_RESTART_INDICATION	0x00200000	acEV_PSTN_RESTART_CONFIRM is generated on receipt of a RESTART message.

ini File Field Name	Value	Description
NS_FORCED_RESTART	0x00400000	On data link (re)initialization, send RESTART if there is no call.
NS_BRI_DL_ALWAYS_UP	0x08000000	BRI with data link always up.
NS_QSI_ENCODE_INTEGER	0x40000000	Applicable only in QSIG. Determines the operator coding: INTEGER ASN.1 type or OBJECT IDENTIFIER ASN.1 type.
NS_5ESS_NATIONAL	0x80000000	Use the National mode of AT&T 5ESS for B channel maintenance.

Examples:

- BladeParams.TrunkConfig[i].ProtocolSpecific.ISDNTrunk.ISDNQ931LayerResponseBehavior=0x2001

NS_ EXPLICIT_INTERFACE_ID and NS_NO_STATUS_ON_UNKNOWN_IE

- BladeParams.TrunkConfig[i].ProtocolSpecific.ISDNTrunk.ISDNQ931LayerResponseBehavior=0x0080

Only NS_SEND_USER_CONNECT_ACK

5.3.6.5 ISDNNSBehaviour2

- NS_BEHAVIOUR2_ANY_UUI bit:
  When this bit is set, any UUI user information is accepted for any protocol discriminator. This feature has been inserted in order to comply with non-standard switches (e.g. AVAYA).

ISDNNSBehaviour2

ini File Field Name	Value	Description
NS_BEHAVIOUR2_ANY_UUI	0x0008	When this bit is set, any UUI user information is accepted for any protocol discriminator.

Example:

BladeParams.TrunkConfig[i].ProtocolSpecific.ISDNTrunk.ISDNNSBehaviour2 = 0x0008

5.3.6.6 ISDNGeneralCCBehaviour

- CC_REVERSE_CHAN_ALLOC_ALGO bit:
  The default channel-id allocation algorithm is as follows:
  TE-side: find the highest channel-id first
  NT-side: find the lowest channel-id first.
  When this bit is reset, CC uses this default behavior. When this bit is set, CC reverses the algorithm.
- CC_CHAN_ID_16_ALLOWED bit:
  Applies to PRI/E1 lines (30B+D) only. Enables handling the differences between the newer QSIG standard (ETS 300-172) and other ETSI-based standards (ETS 300-102 and ETS 300-403) in the conversion of B-channel ID values into timeslot values:
  - In 'regular ETSI' standards, the timeslot is identical to the B-channel ID value, and the range for both is 1 to 15 and 17 to 31. The D-channel is identified as channel-id #16 and carried into timeslot #16. In newer QSIG standards, the channel-id range is 1 to 30 but the timeslot range is still 1 to 15 and 17 to 31. The D-channel is not identified as channel-id #16, but is still carried into the timeslot #16.

  When this bit is set, the channel ID #16 is considered as a valid B-channel ID, but timeslot values are converted to reflect the range 1 to 15 and 17 to 31. This is the new QSIG mode of operation. When this bit is not set (default), the channel_id #16 is not allowed, as for all ETSI-like standards.
- CC_USE_T1_PRI bit:
  When this bit is set, the PRI interface type is forced to T1 (= 23B+D) rather than E1 (= 30B+D) for the ETSI variant. This is usable in Taiwan, where E1 and T1 lines can be used depending on the switches. When this bit is not set, then the default PRI interface type applies (E1 for ETSI).
- CC_USE_E1_PRI bit:
  When this bit is set, the PRI interface type is forced to E1 (=30B+D) rather than T1 (=23B+D) for USA variants. This is usable in Taiwan, where E1 and T1 lines can be used depending on the switches on E1 lines. When this bit is not set, then the default PRI interface type applies (E1 for ETSI).
- CC_START_WITH_B_CHAN_OOS bit:
  When this bit is set, the B-channels are started in the Out-Of-Service state (OOS), not in the In-Service state (IS). They are not available for call processing until brought back In-Service through a Restart message. This is useful on US PRI lines that start in OOS state by default and can also be useful for Fractional PRI lines.
- CC_CHAN_ALLOC_LOWEST bit:
  When this bit is set, CC allocates B-channels starting from the lowest available B-channel ID. When the CC_CHAN_ALLOC_LOWEST bit is set, the CC_REVERSE_CHAN_ALLOC_ALGO and CC_CHAN_ALLOC_HIGHEST bits are ignored.
- CC_CHAN_ALLOC_HIGHEST bit:
  When this bit is set, CC allocates B-channels starting from the highest available B-channel ID. When the CC_CHAN_ALLOC_HIGHEST bit is set, the CC_REVERSE_CHAN_ALLOC_ALGO bit is ignored.
- CC_TRANSPARENT_UUI bit:
  When this bit is set, the UUI-protocol implementation of CC is disabled allowing the application to freely send UUI elements in any primitive regardless of the UUI-protocol requirements (UUI Implicit Service 1). This allows more flexible application control on the UUI. When this bit is not set (default behavior), CC implements the UUI-protocol as specified in the ETS 300-403 standards for Implicit Service 1.
- CC_GTD5_TBCT bit:
  Only applicable when the configuration is NI2. When this bit is set, CC sends implements the GTD-5 Switch variant of the TBCT Supplementary Service, as specified in the FSD 01-02-40AG Feature Specification Document. Otherwise (default

behavior), TBCT is implemented as specified in the GR-2865-CORE specification.

ISDNGeneralCCBehaviour

ini File Field Name	Value	Description
CC_REVERSE_CHAN_ALLOC_ALGO	0x00000008	Channel ID allocation algorithm
CC_CHAN_ID_16_ALLOWED	0x00000020	Channel ID #16 is considered as a valid B-channel-id
CC_USE_T1_PRI	0x00000040	PRI interface type is forced to T1
CC_USE_E1_PR	0x00000080	PRI interface type is forced to E1
CC_START_WITH_B_CHAN_OOS	0x00000100	B-channels start in the Out-Of-Service state (OOS)
CC_CHAN_ALLOC_LOWEST	0x00000200	CC allocates B-channels starting from the lowest available B- channel id
CC_CHAN_ALLOC_HIGHEST	0x00000400	CC allocates B-channels starting from the highest available B-channel id
CC_TRANSPARENT_UUI	0x00004000	Allows the application to freely send UUI elements in any primitive
CC_GTD5_TBCT	0x00010000	Implements the GTD-5 Switch variant of the TBCT Supplementary Service, as specified in FSD 01-02-40AG.

Examples:

- BladeParams.TrunkConfig[i].ProtocolSpecific.ISDNTrunk. ISDNGeneralCCBehaviour = 0x104

CC_START_WITH_B_CHAN_OOS and CC_REVERSE_CHAN_ALLOC_ALGO

- BladeParams.TrunkConfig[i].ProtocolSpecific.ISDNTrunk. ISDNGeneralCCBehaviour = 0x0080;

Only CC_USE_E1_PR.

5.3.6.7 Example of Using the ISDN Flexible Behavior Parameters

Let's assume that you require the following behavior from the ISDN on Trunk 2.

- Use your own Interface ID in NFAS
- Always send CONNECT-ACK after CONNECT
- Let the blade answer all calls automatically for you

Recommended Configuration

```
BladeParams.TrunkConfig[2].ProtocolSpecific.ISDNTrunk.ISDNQ931LayerResponseBehavior =       (NS_SEND_USER_CONNECT_ACK | NS_EXPLICIT_INTERFACE_ID)
BladeParams.TrunkConfig[2].ProtocolSpecific.ISDNTrunk.ISDNIncomingCallsBehavior =
(CC_DATA_CONN_RS | CC_VOICE_CONN_RS( BladeParams.TrunkConfig[2].ProtocolSpecific.ISDNTrunk.ISDN_NFAS_InterfaceID = IID_Value /* User value */
```

5.3.7 Performing Manual D-Channel Switchover in NFAS Group

If an NFAS group is configured with two D-channels (Primary and Backup), you can do a manual switchover between these D-channels.

➢ **To manually switchover from active to standby D-channel:**

1. Open the NFAS Group & D-Channel Status page (**Status & Diagnostic** tab > **VoIP Status** menu > **NFAS Group & D-Channel Status**).
2. Select the required NFAS group, and then click the **Switch Activity** button.

Notes:

- The **Switch Activity** button is unavailable (i.e, grayed out) if a switchover cannot be done due to, for example, alarms or unsuitable states.
- This feature is applicable only to T1 ISDN protocols supporting NFAS, and only if the NFAS group is configured with two D-channels.

5.3.8 ISDN Overlapped Digits

Some ISDN variants support *sending overlapped digits*, that is, the capability to send and/or receive called number digits one at a time (or a couple at a time).

5.3.8.1 Sending ISDN Overlapped Digits

To send overlapped digits, the SETUP message that is sent must not have the information element 'Sending – complete' in SETUP (as result of PlaceCall).

Therefore, the 'cc_User_sending_complete' bit must be set in the field BladeParam.TrunkConfig[i].ProtocolSpecific.ISDNTrunk.ISDNOutgoingCallsBehavior.

If this bit is not set, then if you send any digit (one or more), the ISDN stack automatically generates the information element 'Sending-complete' (regardless of the fact that you requested in PlaceCall() not to send it.

- OverLap receiving: When the blade receives a setup message containing no called number, it automatically answers with a setup acknowledge message. All subsequent received digits are reported to the user via acEV_ISDN_OVLP_DIGIT_IN. When the sending complete IE is set (reported as one of the acEV_ISDN_OVLP_DIGIT_IN parameters) or enough digits have been received, invoke proceeding or alerting and continue with the 'standard' call setup.
- OverLap transmitting: Invoke acPSTNPlaceCall() without setting the sending complete IE and no digits (or less than the network needs for routing) for the called number parameter. When acEV_ISDN_SETUP_ACK_IN is received, invoke acSendOverlapDigits() for the called number digit string and set the sending complete parameter to YES if it contains the last digit of the called number. Afterwards, continue with the 'standard' call setup.

5.3.8.2 Example of Using ISDN Overlapped Digits

To use overlap behavior on Trunk 2, the recommended configuration is as follows:

```
BladeParams.TrunkConfig[2].
ProtocolSpecific.ISDNTrunk.ISDNOutgoingCallsBehavior =
(CC_USER_SENDING_COMPLETE | ALL_OTHER_User_Bits(

Remark:

In PlaceCall() the last field is:      enum acTPSTNSendingComplete
SendingComplete

Use only the enum acTPSTNSendingComplete values and no other
values,
as the integer values of 0,1 might confuse.

typedef enum acTPSTNSendingComplete
{
    SENDING_COMPLETE_NO_MORE_DIGITS = 0,
    OVERLAP_MODE_MORE_DIGITS_IN_NEXT_MSG = 1
} acTPSTNSendingComplete;
```

5.3.9 Q.931 Relay Mode

Q.931 relay is a new way to establish or tear down a call in a media gateway Host/channel architecture. It is suitable for users who need to handle a complete Q.931 protocol at the Host level. Normally, ISDN call operation is controlled by PSTN library commands or events. In Q.931 relay mode, users can get or send a packet with Q.931 message information to a specific trunk (LAPD interface). Q.931 data is 'relayed' directly between the link and Host.

Note that D-channel maintenance is still controlled by the integral ISDN stack.

5.3.9.1 Using Q.931 Relay Mode

- **From Blade to Host** – To receive a Q.931 message, users should periodically invoke an acGetPacket() call, to get packets of the defined acPCILapdToHost type.
- **From Host to Blade** – To send a Q.931 message, call acSendPacket with acPCILapdToHost PacketType and pointer to outgoing Q.931 relay packet.

Note : Use the acISDNGetDChannelStatus() command to retrieve the synchronization status of the D-channel before starting to send outgoing Q.931 relay packets.

```
enum acTPacketType
{
    …
    acPCIlapdToHost = 69,   /* packet type for Q931 Relay mode*/
    ….
};
```

5.3.9.2 Q.931 Relay Packet Structure

Incoming or Outgoing packet, containing acTPacketHeader followed by the Q.931 packet.

```
struct Q931Packet
{
unsigned short TrunkId,
unsigned short Q931BuffSize, /* unsigned char units*/
unsigned char  Q931Buff[]
}
```

5.3.9.3 Activating Q.931 Relay Mode

To activate Q.931 Relay Mode, add this value in the blade *ini* file or in structure BladeParams: Trunk protocol must be:

- acPROTOCOL_TYPE_E1_IUA for E1

or

- acPROTOCOL_TYPE_T1_IUA for T1

5.3.10 Q.931 Raw Message Mode

Q.931 raw data message mode provides a mechanism for raw ISDN Q.931 data. The blade allows a Host to send and receive raw Q.931 messages. This is performed by concatenating the Q.931 raw data message to the delivered ISDN messages to the Host.

The blade enables the Host to send Q.931 raw data messages restricted to the rule that the message does not change the state of the call (e.g., Facility, Status).

This feature is suitable for a Host that needs to receive extended information or non-supported messages, and send non-call-state changeable messages to the blade.

5.3.10.1 Using Q.931 Raw Message Mode

- From Blade to Host – When receiving a Q.931 raw data message, event acEV_ISDN_RAW_DATA_Q931_IN, indicating an incoming Q.931 raw data message, is received.
- From Host to Blade – When sending a Q.931 raw data message, invoke an acISDNSendQ931RawData () call.

5.3.10.2 Q.931 Raw Data Message Structure

Incoming or Outgoing Q.931 raw data messages use the following structure:

```
struct acTEvIsdnRawDataQ931InInfo
{
    int         CallHandle;  /* CRV number between host to blade
    int         ConnId;
    int         TrunkId;
    int         Size;        /* size of the Q.931 message
    Int   Type;         /* type of the Q.931 message   (NOTE 1)
    char  Data[256];   /* Q.931 message          (NOTE 2)
};
```

Notes:

- An incoming message uses the field 'Type' to define the type of proprietary stack message. The message type can be recovered from field 'Data', fifth byte. An outgoing message uses field 'Type' to define the Q.931 message type.
- An incoming message starts with the Protocol discriminator and CRV (Call Reference Value), following the message type and the IEs. An outgoing message starts with the message type, following the variant IEs.

5.3.10.3 Activating Q.931 Raw Data Message Structure

Outgoing (from Host to Blade) – this feature is always active. The Host can send any Q.931 raw message that does not change the state of the call. Note that the call must exist in the blade (has a **connId**).

Incoming (from Blade to Host) – In this direction, the blade uses the blade parameter BladeParam.PSTNSettings.ISDNDuplicateQ931BuffMode to filter the message. This parameter is set by the host in the blade configuration parameters.

Q.931 Raw Message File Names

ini File Field Name	Bit	Value	Description
ACT_DUPLICATEQ931_MSG	8	0x80	Activates 'all types' messages
-	7	0x40	Reserved
ACT_FACILITIY_MASK	6	0x20	Activates FACILITY messages
ACT_USER_INFO_MASK	5	0x10	Activates USER INFORMATION messages
ACT_CLEAR_IN_MASK	4	0x08	Activates DISCONNECT and RELEASE messages
ACT_ALERT_IN_MASK	3	0x04	Activates ALERT messages
ACT_PROGRESS_IN_MASK	2	0x02	Activates SETUP ACK, CALL PROC and PROGRESS messages
ACT_CONN_IN_MASK	1	0x01	Activates SETUP messages
ACT_NOT_ACT_MASK	-	0x00	Duplication is inactive.

5.3.10.4 Examples of Using Q.931 Raw Data Message Structure

- Activate None.
  BladeParam.PSTNSettings.ISDNDuplicateQ931BuffMode = ACT_NOT_ACT_MASK.
- Activate All.
  BladeParam.PSTNSettings.ISDNDuplicateQ931BuffMode = ACT_DUPLICATEQ931_MSG.
- Activate SETUP ACK + CALL PROCEEDING + PROGRESS.
  BladeParam.PSTNSettings.ISDNDuplicateQ931BuffMode = ACT_PROGRESS_IN_MASK.
- Activate SETUP + DISCONNECT + RELEASE:
  BladeParam.PSTNSettings.ISDNDuplicateQ931BuffMode = (ACT_CLEAR_IN_MASK || ACT_CONN_IN_MASK)

Note: The validation check of the Q.931 message in the outgoing direction is the Host's responsibility. A non-valid message is discarded.

5.3.11 B-channel Selection in ISDN Protocols

5.3.11.1 B-channel Selection – Outgoing call

In the acPSTNPlaceCall() function, the host indicates one of the following:

- A: B-channel is indicated, the ExclOnOff field is set to 1, no acceptable alternative (Exclusive);
- B: B-channel is indicated, the ExclOnOff field is set to 0, any alternative is acceptable (Preferred);
- C: B-channel is not indicated (-1 = any channel is acceptable). In the user side, Channel identification information element indicates 'any channel' or the Channel identification information element is not present. In the network side, Channel identification information element indicates 'exclusive' with a B-channel indicated.

For 'A' and 'B', if the indicated channel is available, the remote side selects it for the call.

For 'B', if the remote side cannot grant the preferred channel, it selects any other available B-channel.

For 'C', the Remote side selects any available B-channel associated with the D-channel (only if the remote side is network).

- B-Channel selection – Incoming call.

In the acEV_PSTN_INCOMING_CALL_DETECTED event, the host is indicated one of the following:

- A) B-channel is indicated, the Exclusive field is set to 1 , no acceptable alternative (Exclusive);
- B) B-channel is indicated, the Exclusive field is set to '0' any alternative is acceptable (preferred);

For 'A' and 'B', if the indicated channel is available, the host selects it for the call.

For 'B', if the host cannot grant the preferred channel, it selects any other available B-channel.

Otherwise, set B-channel field to -1, this implies the acceptance of the B-channel for the call.

Note: In the incoming SETUP message, the B-channel can have the value 'any'. B-channel is selected by the PSTN software. If it is the user side it is indicated as preferred, otherwise it is indicated as exclusive.

The two tables below summarize the B-channel negotiation.

Outgoing acPSTNPlaceCall

B-channel	Exclusive field	Result
Any (-1)	Don't care	USER: ChanID IE can not be sent or sent with a specific B-channel (Preferred). NET: ChanID IE with a specific B-channel (Exclusive).
N	Exclusive (1)	ChanID IE with a specific B-channel (Exclusive).
N	Preferred (0)	ChanID IE with a specific B-channel (Preferred).

Incoming call acEV_PSTN_INCOMING_CALL_DETECTED

B-channel	Exclusive field	Result	Remarks
N	Exclusive (1)	B-channel must be accepted. The value of the B-channel, in the first respond, can be -1 or the same.	If the network does not accept this B- channel the network can disconnect the call
N	Preferred (0)	B-channel can be accepted or not (see remark). The value of the B-channel, in the first respond, can be -1, the same or change the value.	If the network does not accept this B- channel the network 'can' reply with another B- channel in the response

5.3.12 Clearing Call Values

There are two Clear types:

- NetCause Clear values are those reported in the Cause IE in the Q.931 message
- RetCause values are the Network Layer reasons coming from the blade's software

5.3.12.1 NetCause Clearing Code Values

```
typedef enum acTISDNNetCause
{
    /* Normal event – class 000 */

    UNASSIGNED_NUMBER =                         1,
    NO_ROUTE_TO_TRANSIT_NET =                   2,
    NO_ROUTE_TO_DESTINATION =                   3,
    CHANNEL_UNACCEPTABLE =                      6,
```

```
    CALL_AWARDED_AND =                          7,
    PREEMPTION =                                8,  /* Added in
ETS 300 403 */

    /* Normal event – class 001 */

    NORMAL_CALL_CLEAR =                         16,
    USER_BUSY =                                 17,
    NO_USER_RESPONDING =                        18,
    NO_ANSWER_FROM_USER_ALERTED =               19,
    ACCEPT_DONE =                               20,
    CALL_REJECTED =                             21,
    NUMBER_CHANGED =                            22,
    NON_SELECTED_USER_CLEARING =                26,
    DEST_OUT_OF_ORDER =                         27,
    INVALID_NUMBER_FORMAT =                     28,
    FACILITY_REJECT =                           29,
    RESPONSE_TO_STATUS_ENQUIRY =                30,
    NORMAL_UNSPECIFIED =                        31,
    CIRCUIT_CONGESTION =                        32,
    USER_CONGESTION =                           33,

    /* Resource not available */

    NO_CIRCUIT_AVAILABLE =                      34,
    NETWORK_OUT_OF_ORDER =                      38,
    NETWORK_TEMPORARY_FAILURE =                 41,
    NETWORK_CONGESTION =                        42,
    ACCESS_INFORMATION_DISCARDED =              43,
    REQUESTED_CIRCUIT_NOT_AVAILABLE =           44,
    RESOURCE_UNAVAILABLE_UNSPECIFIED =          47,
    PERM_FR_MODE_CONN_OUT_OF_S =                39,     /* Added
in ETS 300 403 */
    PERM_FR_MODE_CONN_OPERATIONAL =             40,     /* Added
in ETS 300 403 */
    PRECEDENCE_CALL_BLOCKED =                   46,     /* Added
in ETS 300 403 */

    /* Service not available */

    QUALITY_OF_SERVICE_UNAVAILABLE =            49,
    REQUESTED_FAC_NOT_SUBSCRIBED =              50,
    BC_NOT_AUTHORIZED =                         57,
    BC_NOT_PRESENTLY_AVAILABLE =                58,
    SERVICE_NOT_AVAILABLE =                     63,
    CUG_OUT_CALLS_BARRED =                      53,         /*
Added in ETS 300 403 */
    CUG_INC_CALLS_BARRED =                      55,         /*
Added in ETS 300 403 */
    ACCES_INFO_SUBS_CLASS_INCONS =              62,         /*
Added in ETS 300 403 */

    /* Service not implemented */
```

```
    BC_NOT_IMPLEMENTED =                          65,
    CHANNEL_TYPE_NOT_IMPLEMENTED =                66,
    REQUESTED_FAC_NOT_IMPLEMENTED =               69,
    ONLY_RESTRICTED_INFO_BEARER =                 70,
    SERVICE_NOT_IMPLEMENTED_UNSPECIFIED =         79,

    /* Invalid message */

    INVALID_CALL_REF =                            81,
    IDENTIFIED_CHANNEL_NOT_EXIST =                82,
    SUSPENDED_CALL_BUT_CALL_ID_NOT_EXIST =        83,
    CALL_ID_IN_USE =                              84,
    NO_CALL_SUSPENDED =                           85,
    CALL_HAVING_CALL_ID_CLEARED =                 86,
    INCOMPATIBLE_DESTINATION =                    88,
    INVALID_TRANSIT_NETWORK_SELECTION =           91,
    INVALID_MESSAGE_UNSPECIFIED =                 95,
    NOT_CUG_MEMBER =                              87,     /* Added
in ETS 300 403 */
    CUG_NON_EXISTENT =                            90,     /* Added
in ETS 300 403 */

    /* Protocol error */

    MANDATORY_IE_MISSING =                        96,
    MESSAGE_TYPE_NON_EXISTENT =                   97,
    MESSAGE_STATE_INCONSISTENCY =                 98,
    NON_EXISTENT_IE =                             99,
    INVALID_IE_CONTENT =                          100,
    MESSAGE_NOT_COMPATIBLE =                      101,
    RECOVERY_ON_TIMER_EXPIRY =                    102,
    PROTOCOL_ERROR_UNSPECIFIED =                  111,

    /* Interworking */

    INTERWORKING_UNSPECIFIED =                    127,

    /* not Q.931 causes*/
    ACU_CAUSE_ACU_BAD_ADDRESS =                   128, /* 0Xf0,
value 0: bad context addressing info, or no free context available
*/
    ACU_CAUSE_ACU_BAD_SERVICE =                   129, /* 0Xf1,
value 1: bad ACU service value */
    ACU_CAUSE_ACU_COLLISION =                     130, /* 0Xf2,
value 2: incoming call collision */
    ACU_CAUSE_ACU_FAC_REJECTED =                  131, /* 0Xf3,
value 3: Facility request rejected by ACU */
    ACU_NETWORK_CAUSE_NIL =                       255, /* 0Xff,
value F: unspecified */
} acTISDNNetCause;
```

5.3.12.2 RetCause Clearing Code Values

```
    ACURC_BUSY          'b'   /* busy */
    ACURC_INCOMING      'i'   /* incoming call detected while try to
dial */
    ACURC_NOLINE        'l'   /* (analog) line is seized by another
equipment */
          /* (ISDN) Wrong Addressing info, or context already used
*/
    ACURC_HUNGUP        'h'   /* remote has hung up or incident on
connection */
ACURC_BAD_SERVICE       'S'   /* Bad Service-id in ACU_CONN_RQ/RS */
    ACURC_INTERNAL      'I'   /* other internal error */
```

5.3.13 ISDN Service Message

ISDN SERVICE and SERVICE_ACK messages are B-channel maintenance messages used between the network and the user side. Each B-channel can be in one of the following three states:

- In Service (the B-channel is active and can be used for a call).
- Out of Service (the B-channel is not activated and no call can be made on this channel).
- Maintenance (the B-channel is not active due to a protocol problem - i.e., a problem in freeing the B-channel when the call is released).

5.3.13.1 Function acISDNServiceRequest()

The Host issues the function acISDNServiceRequest(). In response, the stack tries to send a SERVICE message to the remote side.

5.3.13.2 Event acEV_ISDN_SERVICE_CHANGE

The Host receives the event acEV_ISDN_SERVICE_CHANGE in the following cases:

- Function – confirmation for a request made by the function acISDNServiceRequest if ServiceReportType = acIsdnReportTypeOK (command succeeded). If ServiceReportType = acIsdnReportTypeError (command failed), refer to ServiceErrorType for the failure cause.
- Event – The stack notifies the Host of an incoming SERVICE message. ServiceReportType = acIsdnReportTypeEvent.

5.3.13.3 Restrictions

ISDN Service Message is only supported by T1 protocols. Not all T1 variants support all B-channel states. Note that a request can be performed per B-channel or for all B-channels in the trunk.

Supported protocols:

- acPROTOCOL_TYPE_T1_NI2_ISDN
- acPROTOCOL_TYPE_T1_4ESS_ISDN
- acPROTOCOL_TYPE_T1_5ESS_9_ISDN
- acPROTOCOL_TYPE_T1_5ESS_10_ISDN
- acPROTOCOL_TYPE_T1_DMS100_ISDN
- acPROTOCOL_TYPE_J1_TRANSPARENT
- acPROTOCOL_TYPE_T1_NTT_ISDN

- acPROTOCOL_TYPE_T1_DMS100_MERIDIAN_ISDN
- acPROTOCOL_TYPE_T1_NI1_ISDN.

The following protocols do not support acBCHANNEL_IN_MAINTANANCE:

- acPROTOCOL_TYPE_T1_NI2_ISDN
- acPROTOCOL_TYPE_T1_NI1_ISDN

Note: When the protocols acPROTOCOL_TYPE_T1_DMS100_MERIDIAN_ISDN and acPROTOCOL_TYPE_T1_DMS100_ISDN are used, the function acISDNServiceReq() first sends a RESTART message and then a SERVICE message. The following restriction applies to the RESTART message: 'no further RESTART messages will be sent until a RESTART ACKNOWLEDGE is received'. Therefore, the host must either wait for a Service_ack before invoking acISDNServiceReq() again on the same trunk, or insert a delay between each call.

5.3.14 Graceful Lock in ISDN US Variants

There is an option to gracefully lock trunks configured as ISDN American variants.

When performing this action at board level, the following is done for each relevant trunk:

- SERVICE messages (changing states to out-of-service) are sent to the remote side for each free B-channel on that trunk.
- If a B-channel is involved in call(s), then the SERVICE message will be sent once the B-channel is released.
- The locked B-channels will not be available for new calls until the unlock action is required.

When unlocking the board, SERVICE messages (changing states to in-service) are sent to the remote side for each B-channel on all the (ISDN US) trunks, and the B-channels become available for call processing.

5.3.15 COLP/COLR Supplementary Service

COLP (Connected Line Identity Presentation) is a supplementary service that allows the calling party to receive the connected party's address information. This information is delivered by the final connected party in the Q.931 CONNECT message and consists of the calling party's national number (ISDN), the country code, international call information and sub-address information.

The COLR (Connected Line Identity Restriction) supplementary service prevents the called party's address information from being presented to the calling user.

When there is an incoming CONNECT message, the blade checks the existence of the service and sends it to the HOST in the event EV_PSTN_CALL_CONNECTED.

In the other direction, the HOST can include the connected party's information in the command acPSTNAnswerCall().

5.3.15.1 Using COLP/COLR

- From Host to Blade – Send a Q.931 connect message.

The Host invokes command acPSTNAnswerCall() to send a connect message. In this command, the Host can insert the connected party's address information.

- From Blade to Host – Receive a Q.931 connect message.

The blade sends event EV_PSTN_CALL_CONNECTED indicating an incoming Q.931 connect message.

5.3.15.2 Structure COLP/COLR

The following structure is a part of the incoming/outgoing connect structure.

```
typedef struct acTISDNConnectedNumberEvent
{
    int     Exists;                              /*!< BOOL – validity
of the structure. If TRUE, the next fields are valid. */
    acTACUNumberType     Type;
    acTACUNumberPlan     Plan;
    acTACUNumberPresentation     Presentation;
    acTACUNumberScreening     Screening;                  /*!<
acTACUNumberScreening */
    char    DestPhoneNum[MAX_PHONE_NUM];       /*!< conn nb phone
number */
    acTACUSubAddressType     SubAddType;
    acTACUSubAddressOddIndicator     SubAddOddIndicator;
    int     Size;                                          /*!<
size of the conn           number*sub-add string in bytes */
} acTISDNConnectedNumberEvent;
```

5.3.15.3 Activating Structure COLP/COLR

This feature can be activated in both directions.

Only when the **existing** field is set to PRES, the structure consists of valid data, otherwise the structure is ignored.

Notes:

- If the existing field is set to PRES, the address information, within the structure, is considered as valid and correct. If it is not set to PRES, this structure should be ignored.
- The Host is responsible for providing the address information for an incoming call (if any), and its validity.

The blade is responsible for taking (from the Connect message) the address information for outgoing call (if any), and its validity.

5.3.16 ISDN User-to-User IE Implementation

The purpose of the User-to-User Information Element (IE) is to convey information between ISDN users. This information is not interpreted by the network but is rather carried and delivered to the remote user. The user-user IE is supported in the following ISDN messages: SETUP, ALERTING, CONNECT, DISCONNECT, RELEASE, RELEASE COMPLETE and USER INFORMATION.

5.3.16.1 How to Send the UUI IE

To include a UUI IE in the sent message: In each command the UUI IE is supported, fill the structure acTUuiElement that includes all of the UUI IE parameters. Otherwise a NULL pointer is sent.

5.3.16.2 How to Receive the UUI IE

Except for the event acEV_ISDN_USER_INFORMATION_MESSAGE, the UUI IE is not part of the event's associated structure. In all other events, the field UuiSequenceNum indicates the UUI IE existence in the following message. If the field UuiSequenceNum is equal to any value other than zero, users should wait for a following UUI IE event, with the same UuiSequenceNum value, before proceeding. If the field is set to 0, there is no UUI IE in the following message.

Note: The 4ESS variant supports the use of an empty UUI in the SETUP message. If the originating user needs a UUI to be transported in the ALERTING, CONNECT, or DISCONNECT message on a call but has no UUI to be sent in the SETUP message, the originating user should still place an 'empty' UUI-type information element in the SETUP message, in order to obtain a suitable network connection to ensure UUI transport.

The SETUP message should contain an empty UUI with UUI length = 1, and null ("") UUI data.

5.3.16.3 UUI IE Examples

```
Send a UUI in the ISDN Connect message:
Incoming call with CallHandle = 511
Set Puui_IE
   UuiProtocolDescription = USER_SPECIFIC_PROTOCOL;
   UuiDataLength = 33;
   UuiData = "USER TO USER INFORMATION ELEMENT \0 " (use copy
function).
acPSTNAnswerCall(511,-1,NULL,Puui_IE,NULL,NULL,NULL)
```

Receive UUI in ISDN Connect message:

```
Outgoing call with callHandle = 1
Receive acEV_PSTN_CALL_CONNECTED event with UuiSequenceNum = 603
Receive acEV_ISDN_UUI_INFORMATION_ELEMENT with:
   UuiProtocolDescription = USER_SPECIFIC_PROTOCOL;
   UuiDataLength = 33;
   UuiData = "USER TO USER INFORMATION ELEMENT \0 "
```

5.3.17 ISDN Set Additional IE

To send an IE that is not directly supported by the VoPLib, build the IE in Q.931 raw data format and send it as an argument (pAdditionalInformationElement) of the function.

The additional IE is supported in the following commands:

- acISDNSendOverlapDigits()
- acISDNSendAlert()
- acPSTNAnswerCall()
- acPSTNReleaseCall()
- acPSTNDisconnectCall()

- acISDNSendCallProceeding()
- acISDNSendCallProgress()
- acIsdnSendQ931NotifyMessage()
- acIsdnSendQ931InformationMessage()
- acIsdnSendUserInformationMessage()
- acIsdnSendFacilityMessage()
- acISDNSendOverlapDigits()
- acISDNSendAlert()

Note: To use this field, the host must understand the way an ISDN message is built and how to build a specific IE. The raw IE is checked for validity by the ISDN stack. If it is invalid, the command is discarded and an error is sent to the host.

5.3.17.1 Examples

Send an additional IE (UUI in this case) in the ISDN ALERT message:

```
/* Incoming call with IncomingCallHandle */
/* Declare the additional IE data structures */
acTAdditionalInformationElement additional_information_element;
char AdditionalInformationElementData[14];
int AdditionalInformationElementDataLength = 14;
/* Fill the additional IE data */
AdditionalInformationElementData[0] = 0x7e;   /*Message Type =
UUI*/
AdditionalInformationElementData[1] = 0x0c;   /*UUI IE content
length*/
AdditionalInformationElementData[2] = 0x04;
memcpy(&(AdditionalInformationElementData[3]), "NetworkData", 11);
/*UUI IE data*/
/* Set the additional IE structure */
additional_information_element.acISDNInfoElementsBuffer =
AdditionalInformationElementData;
additional_information_element.acISDNInfoElementsBufferLength =
AdditionalInformationElementDataLength;
/* send the ALERT message */
acISDNSendAlert(IncomingCallHandle, -1, PROGRESS_LOC_USER,
PROGRESS_DESCR_IN_BAND_NOW, NULL,
&additional_information_element);
```

5.3.18 ISDN Supplementary Services

In addition to basic call control services, ISDN PRI (Primary Rate Interface) and BRI (Basic Rate Interface) support a number of supplementary services.

This subsection:

- Lists the services supported by AudioCodes products
- Briefly explains each service, its purpose and use
- Explains in detail AudioCodes' VoPLib API regarding ISDN supplementary services

- The ISDN supplementary services below are applicable only in PRI, unless indicated otherwise.

Notes:

- Readers should be familiar with the subject of ISDN supplementary services, ASN.1 coding and the relevant standards before proceeding to develop/implement an application using AudioCodes' VoPLib and blades.
- Support for ISDN supplementary services in AudioCodes' products is preliminary and not fully tested and therefore may be subject to future changes (parameters, structures, names, etc.).

5.3.18.1 Supplementary Services and ISDN Variants

'ETSI Variant Supplementary Services Specifications' on page 339 through to '5ESS Variant Supplementary Services Specifications' on page 343 describe the supplementary services supported by AudioCodes' software.

The relevant specifications are mentioned per ISDN variant.

5.3.18.1.1 ETSI Variant Supplementary Services Specifications

HOLD and RETRIEVE Supplementary Service (ETSI 139)

The HOLD supplementary service allows users to interrupt communications on an active call.

After holding the call, the B-channel of the held call can be used by another call, outgoing or incoming.

The RETRIEVE supplementary service allows users to re-establish communications on the held call.

ECT (Explicit Call Transfer) Supplementary Service (ETS-300-367, 368, 369)

The Call Transfer supplementary service is supported for BRI as well as PRI trunks.

This service provides the served user that has two calls, to ask the network to connect them together and release its line. The two calls can be incoming or outgoing calls. This service is similar to NI2 Two B-Channel Transfer (TBCT) Supplementary Service. The main difference is that in ECT, one of the calls must be in HELD state.

The ECT standard defines two methods of work:

- Implicit
- Explicit

When using the Implicit method, the two calls must be on the same trunk. In BRI one will use the Implicit mechanism, and in PRI the Explicit mechanism is used.

AOC (Advice Of Charge) Supplementary Services (ETS-300-178, 179, 180, 181, 182)

This service enables users to receive call rates at the startup stage of a call and to receive recorded charges during the call's active phase and at call termination. Rate data can be sent at the startup stage of the call (Q.931 SETUP or CONNECT messages), during the call with messages containing the accumulative charge (using the Q.931 FACILITY message) or at the end of the call with a message containing the recorded charge.

The service includes:

- Rate data at call setup (AOC-S)
- Rate data during the call (AOC-D)
- Rate data at the end of the call (AOC-E)
- AOC supplementary services are also applicable in BRI

Diversion Supplementary Services (ETSI EN 300 207-1)

The diversion supplementary services comprise:

- Call Forwarding Unconditional (**CFU**) supplementary service
- Call Forwarding Busy (**CFB**) supplementary service
- Call Forwarding No Reply (**CFNR**) supplementary service
- Call Deflection (**CD**) supplementary service

CFU enables a served user to have the network redirect calls (that are addressed to the served user's ISDN number) to another user. The service can operate on all calls or just on those associated with specified basic services. The served user's ability to originate calls is unaffected. After the service has been activated, calls are forwarded independently of the status of the served user.

CFB enables a served user to have the network redirect calls (which are addressed to the served user's ISDN number which is busy) to another user. The service can operate on all calls or just on those associated with specified basic services. The served user's ability to originate calls is unaffected.

CFNR enables a served user to have the network redirect calls (which are addressed to the served user's ISDN number and for which the connection is not established within a defined period of time) to another user. The service can operate on all calls or just on those associated with specified basic services. The served user's ability to originate calls is unaffected.

CD enables the served user to respond to an incoming call by requesting redirection of that call to another user. The service can only be invoked before the connection is established by the served user, i.e., in response to the offered call, or during the period that the served user is being informed of the call. The served user's ability to originate calls is unaffected.

Diversion supplementary services are also applicable in BRI.

Multi-level Precedence and Preemption (MLPP)

Multi-level Precedence and Preemption (MLPP) supplementary service provides a prioritized call handling service. This supplementary service has two parts:

- Precedence - assigning a priority level to a call
- Preemption - seizing of resources which are in use by a call of a lower precedence by a higher level precedence call, in the absence of idle resources.

Users in networks that do not support this service are not affected by this service.

Call Waiting (EN 300 058-1)

ISDN BRI: The Call Waiting (CW) supplementary service is now supported, as defined in EN 300 058-1. It is only applicable when the trunk is configured as BRI, NT with point to multi-point connection.

Malicious Call Identification (MCID) (ETSI EN 300 130)

The MCID supplementary service provides the storage and registration of the call information, as a result of an appropriate user request. Call information consists of the called party number, the calling party number, the time and date of the request, and, as a network option, the calling party subaddress, if provided by the calling user.

To invoke the MCID supplementary service, the called user sends a mCIDRequest invoke component, carried by a Facility information element in a FACILITY message.

To indicate that the service has been accepted, the network sends a mCIDRequest return result component carried by a Facility information element in a FACILITY message.

Support Call Re-arrangements Procedure (ITU-T Q.931 section 5.6)

The Call Re-arrangements Procedure, as described in ITU-T Q.931 Section 5.6, is now supported.

The support includes received SUSPEND and RESUME messages and sent SUSPEND ACKNOWLEDGE, SUSPEND REJECT, RESUME ACKNOWLEDGE and RESUME REJECT messages from the network side.

Message Waiting Indication (MWI) (ETS 300 745-1)

The service is applicable in PRI as well as BRI.

As a result of a request of a controlling user, the MWI supplementary service enables the network to indicate to the receiving user, that there is a message waiting. The indication is delivered to the receiving user.

The service includes:

- Activation of MWI by the Controlling User
- Deactivation of MWI by the Controlling User
- Indication of MWI by the network to the receiving user.

5.3.18.1.2 QSIG Variant Supplementary Services Specifications

Name Identification Supplementary Services (ECMA-163, 164)

This service provides the name of the calling, called or connected party.

The service includes:

- Calling Name Identification Presentation (CNIP)
- Calling Name Identification Restriction (CNIR)
- Connected Name Identification Presentation (CONP)
- Connected Name Identification Restriction (CONR)
- Called Name
- Busy Name

Call Diversion Supplementary Services (ECMA-173, 174)

In this service, served users can ask the network to forward incoming calls to another number. This can be executed for all incoming calls (CFU) or only if the user has an active call (CFB) or if the user does not answer the call (CFNR). The Call Deflection (CD) option allows users to transfer the call before it is connected (during the call setup stage).

The service includes:

- Call Forwarding Unconditional (CFU)
- Call Forwarding Busy (CFB)
- Call Forwarding No Reply (CFNR)
- Call Deflection (CD)

MWI (Message Waiting Indication) Supplementary Service (ECMA-241, 242)

This service allows users to be sent an MWI. The service also enables the MWI to be canceled. MWI is activated and deactivated by the Message Centre with messages to users. MWI interrogation is performed by users.

The service includes:

- Activation of MWI
- Deactivation of MWI
- Interrogation of MWI

Call Transfer Supplementary Service (ECMA-177,178)

SS-CT is a supplementary service which enables a served user (User A) to transform two of that user's calls into a new call between the other two users of the two calls (User B and User C). Each call can either be an incoming call to User A or an outgoing call from User A. After successful invocation of SS-CT, User B and User C will no longer be able to communicate with User A.

One of the calls may be an outgoing call that has not been answered by the other user (User C). After successful invocation of SS-CT User A will no longer be able to

communicate with User B. User B and User C will be in a position to communicate with each other as soon as User C has answered.

Call transfer can be achieved using one of two methods:

- Transfer by Join
- Transfer by Rerouting

Support of "Transfer by Join" is mandatory. Support of "Transfer by Rerouting" is an option which, if not supported by all PINXs (Private Integrated services Network eXchanges) involved in the operation of call transfer, allows a fall back to using "Transfer by Join".

Supported Operations:

- CallTransferIdentify
- CallTransferAbandon
- CallTransferInitiate
- CallTransferSetup
- CallTransferActive
- CallTransferComplete
- CallTransferUpdate
- SubaddressTransfer

Single Step Call Transfer Supplementary Service (ECMA-300)

SS-SSCT is a supplementary service enabling User A to transform an existing call between User A and User B, into a new call between Users B and C. User A does not have a call established with User C prior to the call transfer.

Supported operations:

- SingleStepCallTransferInitiate
- SingleStepCallTransferSetup
- SingleStepCallTransferPostDial
- SingleStepCallTransferDigitInfo

Path Replacement PathReplacePropose ANF SS (ECMA-176)

Path Replacement Additional Network Feature (ANF-PR) according to ECMA-176, is now supported.

Supported Operations:

- PathReplacePropose
- PathReplaceSetup
- PathReplaceRetain
- PathReplaceInvite

The service is also applicable in BRI for the QSIG variant.

5.3.18.1.3NI2 Variant Supplementary Services Specifications

Two B-Channel Transfer (TBCT) Supplementary Service (GR-2865)

This service provides the served user that has two calls, to ask the network to connect them together and release its line. The two calls can be incoming or outgoing calls. This service is similar to the ETSI Explicit Call Transfer (ECT) Supplementary Service.

The service is currently supported in PRI trunks only.

Name Identification Supplementary Service (GR-1367)

This service (similar to the QSIG Name Identification Supplementary Services) provides the name of the calling, called or connected party.

Message Waiting Notification (MWN) Supplementary Services (GR-2942)

This service is similar to the QSIG Message Waiting Indication Supplementary Service.

Multi-level Precedence and Preemption (MLPP) Supplementary Services (ANSI T1.619-1992 and T1.619a-1994)

This service is similar to the ETSI Multi-Level Precedence and Preemption (MLPP) service.

ISDN PRI: E9-1-1 Tandem/End Office to ISDN PSAP Interface (GR-2968)

The present growth in the wireline and wireless telecommunications industry, along with increasing demands on E9-1-1 service, have pushed the analog signaling protocol and some
E9-1-1 tandem capabilities to their limits. Support for using the full 10 digits of the calling station's number at the E9-1-1 tandem and on the E9-1-1 tandem to the Public Safety Answering Point (PSAP) interface is required for E9-1-1 service to continue to work effectively.

The support is compliant with the Telcordia GR-2968-CORE standard.

Currently, only the sending of a SETUP message from an E9-1-1 tandem to an ISDN PSAP with
E9-1-1 parameters is supported. This means that three additional information elements are supported in the SETUP message in this direction:

- Emergency Call Control Information Element.
- Generic Information Element to carry the Location Identification Number information.
- Generic Information Element to carry the Calling Geodetic Location information.

5.3.18.1.4 Nortel DMS Variant Supplementary Services Specifications

RLT (Release Link Trunk) Supplementary Service

This ISDN PRI variant is a Nortel™ propriety variant known as DMS. The RLT supplementary service is similar to NI2 Two B-Channel Transfer (TBCT) and ETSI Explicit Call Transfer (ECT) Supplementary Services. **The service is currently supported in PRI trunks only.**

5.3.18.1.55 ESS Variant Supplementary Services Specifications

Name Identification Supplementary Service (GR-1367)

This service (similar to the QSIG Name Identification Supplementary Services) provides the name of the calling or called party.

5.3.18.2 AudioCodes' ISDN Supplementary Services Implementation

5.3.18.2.1 Using the Q.931 Facility Message

The services supported by this command / event are:

- Call hold
- Call retrieve
- Two B-Channel Transfer (TBCT)
- Explicit Call Transfer (ECT)
- Send Facility IE

The command and the event basically use the same parameter structure.

Send Facility Message Command

The command is:

- Name: acIsdnSendFacilityMessage()

The command includes the following parameters:

- CallHandle – Handle to call
- FacilityCode – Determines the service (HOLD, ECT, TBCT, etc.)

- FacilityAction – Type of action (Activate, Clear or Enquiry)
- OtherCallHandle – This parameter is valid only if the service is ECT or TBCT.

  If the FacilityCode parameter is equal to *acIsdnFacilityAdditionalInformationElement*, the other parameters are not used. Users use the *FacilityCode* parameter to send a Q.931 Facility message containing only Facility IE. This is the case when sending an ISDN Supplementary Service in the Facility message. (For more information, refer to the following section.)

Receive Facility Message Event

The event is:

- Name: acEV_ISDN_FACILITY_MESSAGE
- Parameters structure: acTEvIsdnFacilityMessageInfo

The event includes the following parameters:

- CallHandle – Handle to call
- FacilityCode – Determines the service (HOLD, ECT, TBCT, etc.)
- FacilityAction – Type of action (Ack, Reject or Indication)
- FacilityNetCause – This parameter is valid only if FacilityAction = Reject
- OtherCallHandle – This parameter is valid only if the service is ECT or TBCT

Utilization Example: Activate TBCT/ECT Scenario

In this example, the AudioCodes' blade is the served user. Due to the fact that the two supplementary services are similar, the following explanation combines TBCT and ECT scenarios. It is assumed that the network supports the TBCT/ECT service and that the user has permission to use this service.

➢ **To activate TBCT/ECT services:**

1. Establish an ISDN call (called Call B in the specifications)
2. In ECT, this call **must** be in HELD state. To do this, use the acIsdnSendFacilityMessage() API with the following parameters:
   - CallHandle = Call B handle
   - FacilityCode = acIsdnFacilityCodeHold
   - FacilityAction = acPstnActionCodeActivate

   All other parameters are invalid and can stay at their default values.
3. In ECT, before moving to the next step, the application needs to wait for network confirmation of the HOLD request. Receipt of event acEV_ISDN_FACILITY_MESSAGE with the following parameters means that the network confirms the HOLD request:
   - CallHandle = Call B handle
   - FacilityCode = acIsdnFacilityCodeHold
   - FacilityAction = acPstnActionCodeAck
4. Establish a second ISDN call (called Call C in the specifications). This call can be in active state or in alerting state. In ECT, the second call **must** be on the same trunk.
5. Activate the TBCT/ECT service. Use command acIsdnSendFacilityMessage() with the following parameters:
   - CallHandle = Call B handle (the first established call)
   - FacilityCode = acIsdnFacilityCodeTbct or acIsdnFacilityCodeEct
   - FacilityAction = acPstnActionCodeActivate
   - OtherCallHandle = Call C handle (the second established call)

6. AudioCodes' blade sends confirmation of the TBCT/ECT request, received from the network, to the user host application, using event acEV_ISDN_FACILITY_MESSAGE with the following parameters:
   - CallHandle = Call B handle (the first established call)
   - FacilityCode = acIsdnFacilityCodeTbct or acIsdnFacilityCodeEct
   - FacilityAction = acPstnActionCodeAck
   - OtherCallHandle = Call C handle (the second established call)
7. The network also releases calls B and C. AudioCodes' blade's resources are freed to handle new calls.

5.3.18.2.2 Using Facility IE

Facility IE is the main mechanism used to implement ISDN Supplementary Services. This IE can be sent and received using many ISDN call-related commands and events (PlaceCall, AnswerCall, Alert, ReleaseCall, DisconnectCall, etc.)

5.3.18.2.3 Facility IE Data Structures

The main structure in the AudioCodes' VoPLib that is related to ISDN Supplementary Services is called *acTSuppServComponent*. This structure defines one Supplementary Service component. It contains the following main parameters:

Structure acTSuppServComponent Parameters

Parameter	Description
ComponentTag	Determines the Supplementary Service component type (Invoke, ReturnResult, ReturnError or Reject)
ComponentHeader	Defines the header parameters according to the above Tag. One of the parameters in the header is OperationId (the identifier of the operation). The parameters listed here correspond to the specification.
ss_u	This structure's union includes all specific Supplementary Service operation parameters required, in accord with parameter ComponentHeader. As a rule, this parameter is relevant only if the Supplementary Service component is Invoke or ReturnResult.

5.3.18.2.4 Send Facility IE

There are two parameters in VoPLib ISDN call-related commands:

- pSuppServInformationElement – This pointer to Supplementary Service parameters contains up to three consecutive acTSuppServComponent structures.
- NumOfSuppServInformationElement – The number of acTSuppServComponent structures in the previous parameter.

To send a Supplementary Service (e.g., add Facility IE), locally allocate the *acTSuppServComponent* structure. Define its parameters according the specific Supplementary Service required and call the API command with the two parameters (above) updated. As a result, the VoPLib sends two TPNCP messages to the blade:

- The first message contains the Facility IE. The blade saves the data according to the call identifier (Call Handle).
- The VoPLib then calls the call-related command (e.g., PlaceCall).

In the blade, the software adds the stored Facility IE to the Q.931 message and sends it to the line.

5.3.18.2.5 Receive Facility IE

On the receiving side, the same ISDN Supplementary Services parameter structure *acTSuppServComponent* is used.

The VoPLib features event acEV_ISDN_FACILITY_INFORMATION_ELEMENT. Structure *acTEvIsdnFacilityInformationElementInfo* is related to this event. The main parameters are:

Main Parameters in Structure acTEvIsdnFacilityInformationElementInfo

Parameter	Description
CallHandle	Reference to the call
MessageType	Type of Q.931 message with which the Facility IE is received
FacilitySequenceNumber	Unique number to connect the Facility to the call-related event (refer to the explanation below for details)
IsLast	Flag determining whether this is the last Facility event related to the same call-related event. The software supports up to 3 Facility Ies in one incoming Q.931 message.
SsComponent	Structure acTSuppServComponent contains the Supplementary Service component parameters.

The *FacilitySequenceNum* parameter is used in VoPLib ISDN call-related events (e.g., IncomingCallDetected and Alert), similar to the case of sending Facility IE.

Parameter FacilitySequenceNum Used in VoPLib ISDN Call-Related Events

Parameter	Description
FacilitySequenceNum	Default = 0. Any value other than 0 indicates that a Facility IE has been received. Users should wait for up to three FACILITY events with the same number.

The software supports up to three Facility IEs in one incoming Q.931 message. The blade sends two TPNCP messages to the VoPLib:

1. The ISDN call-related event with a non zero number configured in the *FacilitySequenceNum* parameter.
2. Up to three FACILITY events with the same *FacilitySequenceNum* are supported. The last FACILITY event has the *IsLast* parameter switched on.

5.3.19 T310 Timer in ISDN Variants

1. The T310 is the timer that is set when an ISDN system receives a Call Proceeding message.
2. If no Alerting, Progress, or Connect message is received within the duration of T310, the call clears. The new *IsdnTimerT310 ini* file parameter enables the setting of the T310 timer duration for DMS, Euro ISDN and NI2 variants. This feature was inserted as a result of a customer's request. One example is the requirement of the Italian regulation for ISDN BRI/PRI interfaces T310 timeout to be set to not less than 600 seconds.

Note: This variable replaces the old *IsdnDmsTimerT310* that was used in DMS variants. *IsdnDmsTimerT310* will become obsolete in future releases.

3. If both *IsdnDmsTimerT310* and *IsdnTimerT310* are used, the value of *IsdnTimerT310* will prevail.

5.4 CAS

The following describes the CAS protocol.

5.4.1 General Description

CAS/robbed bit protocols are implemented by specific protocol state machine files.

AudioCodes supports various CAS protocols, starting from the T1 E&M family, including special support for E911, and E1 CAS protocols such as many national MFCr2 protocols over E1-R2D protocol type.

The CAS table file should load to the gateway as auxiliary files using the regular interfaces:

- INI file parameter. Several files can be loaded using the *CasFileName* parameter. For example:

```
CasFileName_0 = 'R2_China_ANI.dat'
CasFileName_1 = 'E1_R2D.dat'
    …..
CasFileName_7 = 'R2_User_Change_ANI.dat'
```

- Auxiliary files wizard in the Web Interface
- EMS Software Upgrade wizard
- Burning the file at the gateway internal flash takes place automatically when the user burns the GW configuration

The user can load the blade up to 8 different CAS variants files.

The following is a list of the variants implemented by AudioCodes.

Variants Implemented by AudioCodes

Variant Group	Variant Table	Notes
E1– E&M	E_M_WinkStart_A-Bit_For_E1	
	E_M_ImmediateStart_For_E1	
T1 – E&M	E_M_DelayStart	
	E_M_FGAImmediateStart	
	E_M_FGBWinkStart	
	E_M_FGDWinkStart	Include also E_M_FGD_BLV
	E_M_ImmediateStart	
	E_M_WinkStart	
LoopStart/Ground Start	GroundStart_FXO	
	GroundStart_FXS	
	LoopStart_FXO	
	LoopStart_FXS	
E1 MFCr2	Mfcr2_Argentina	
	Mfcr2_Bolivia	
	Mfcr2_Brazil	
	MFCR2_Chile	
	Mfcr2_China	
	Mfcr2_Czech_Republic	
	MFCR2_Egypt	
	Mfcr2_India	
	Mfcr2_Indonesia	
	Mfcr2_Israel_Bezeq	
	Mfcr2_ITU	
	Mfcr2_Korea	
	Mfcr2_Malaysia	
	Mfcr2_Mexico	
	Mfcr2_Panama	
	Mfcr2_Philippines	
	MFCR2_Saudi_Arabia	
	Mfcr2_Thailand	
	Mfcr2_Uruguay	
	MFCR2_Venezuela	
Other Files	E1_R2D	

Variants Implemented by AudioCodes

	E911_CAMA_ConnectTo911Switch	
	T1_FCD_channel_bank	
	T1_FGD_E911	
Files for Megaco ONLY	DC5-AC15	
	E1_MELCAS	
	E911_CAMA_ConnectTo911Switch	
	E_M_FGDWinkTable	For T1
	GroundStart_FXS	
	LoopStart_FXS	
	Mfcr2_Brazil_CP	E1 MFCr2
	Mfcr2_Korea_CP	E1 MFCr2
	Mfcr2_Mexico_CP	E1 MFCr2

The trunk protocol type must fit the CAS variant type. T1 variants should run over trunks with protocol type T1_CAS. MFCr2 variants should run over trunks configured as E1_MFCR2, and other E1 variants should run over E1_CAS protocol.

CAS files that are in the MEGACO files group can only run with the MEGACO protocol type, which can run CAS calls with these files only.

User's programmable files: Users can change the protocol's parameters and even entire state machine via two related files:

- A text file describing the protocol state machine (Protocol Table Text File/script file, named xxx.txt), and various initialization parameters.
- The user-defined parameters h file (named xxx_Userdefine.h) that map the text-named parameters in the above Protocol Table Text File to their user-defined numerical values.

After a change has been made, the Protocol Table Text File must be re-compiled, and the updated protocol table/script file xxx.dat file should be downloaded by either the *ini* file, via Web Interface or using the EMS Auxiliary Files wizard.

Take for example the 'E&M wink start' protocol; if no change to the protocol state machine or protocol parameters provided by AudioCodes is needed, then E&M_WinkStart.dat is downloaded without any re-compiling. If a change is made either to E&M_WinkStart.txt or to E&M_WinkStart_Userdefine.h, then recompile according to the instructions described in 'Process CAS Tables' on page 793. The construction of the two files is based on simple rules explained in 'Channel Associated Signaling (CAS) Functions' on page 759. The user has a very flexible environment to define new protocols, or changes to existing protocols.

5.4.2 CAS Trunk Parameters

There are several additional parameters needed to configure a CAS trunk.

CasDelimitersPaddingUsage: This parameter changes the digit's separation of source and destination phone number indication. This parameter can be configured via *ini* file only.

CasTrunkDialPlanName: This parameter selects the Dial Plan which the trunk will work with. This parameter can changed on-the-fly even if the trunk is active, i.e. no need to stop the trunk.

There are two ways for choosing the CAS table(s) relevant for the specific trunk:

- **CasChannelIndex:** This parameter defines which one of the loaded CAS tables will be in use *in each b-channel at the trunk*. This parameter is a string, and can be set in one of two formats:
  - Sets the CAS table per channel. In this format, the user needs to set 31 indexes for E1 trunks (include dummy for b-channel 16), or 24 indexes for T1 trunks.
  - Sets the CAS table per channel group. Every b-channel (including b-channel 16) must be part of a channel group.

  The user must STOP the trunk in order to change this parameter.

  If this parameter is not defined, there will be a single CAS table for the whole trunk, which is selected by the CasTableIndex parameter.
- **CasTableIndex:** This parameter defines which one of the loaded CAS tables will be in use in the *whole* trunk, in case there is a single CAS protocol at this trunk. This parameter can receive values of 0-7. The user must STOP the trunk in order to change this parameter. This parameter is valid only in case CasChannelIndex is an empty string.

Examples:

An example *ini* file for configuring T1_CAS trunk with a single protocol (Trunk 5):

```
ProtocolType_5 = 7
CASFILENAME_0='E_M_FGBWinkTable.dat'
CasTableIndex_5 = 0
CASDelimitersPaddingUsage_5 = 1
```

An example *ini* file for configuring T1_CAS trunks with several protocols (Trunk 5):

```
ProtocolType_5 = 7
CASFILENAME_0='E_M_FGBWinkTable.dat'
CASFILENAME_1='E_M_FGDWinkTable.dat'
CASFILENAME_2='E_M_WinkTable.txt'
CasChannelIndex_5 =
'0,0,0,1,1,1,2,2,2,0,0,0,1,1,1,0,1,2,0,2,1,2,2,2'
CASDelimitersPaddingUsage_5 = 1
```

An example *ini* file for configuring E1_CAS trunk with single protocol (Trunk 5):

```
ProtocolType_5 = 8
CASFILENAME_2='E1_R2D'
CasTableIndex_5 = 2
```

An example *ini* file for configuring E1_CAS trunk with several protocols (Trunk 5):

```
ProtocolType_5 = 8
CASFILENAME_2='E1_R2D'
CASFILENAME_7= E_M_ImmediateTable_A-Bit.txt'
CasChannelIndex _5 = '1-10:2,11-20:7,21-31:2'
```

An example *ini* file for configuring E1_MFCR2 trunk with a single protocol (Trunk 5):

```
ProtocolType_5 = 7
CASFILENAME_0='R2_Korea_CP_ANI.dat'
```

```
CasTableIndex_5 = 0
DialPlanFileName = 'DialPlan_USA.dat'
CasTrunkDialPlanName_5 = 'AT_T'
```

The same parameters are needed in order to configure CAS trunks via the Web Interface or EMS.

➢ **To configure a CAS trunk:**

1. Load the CAS table file.
2. Load the Dial-plan file if there is one.
3. Configure the trunk with the relevant CAS parameters.

While changing *ProtocolType*, the *CasChannelIndex* and *CasTableIndex* parameters are required to stop the trunk. The user can change the *CasTrunkDialPlanName* parameter when the trunk is active.

5.5 SS7 Functionality & Configuration

Note: This SS7 Configuration sub-section is not applicable to **MediaPack, Mediant 1000** and **Mediant 3000**.

Several SS7 network elements are available. This sub-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 the 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.

5.5.1 SS7 Network Elements

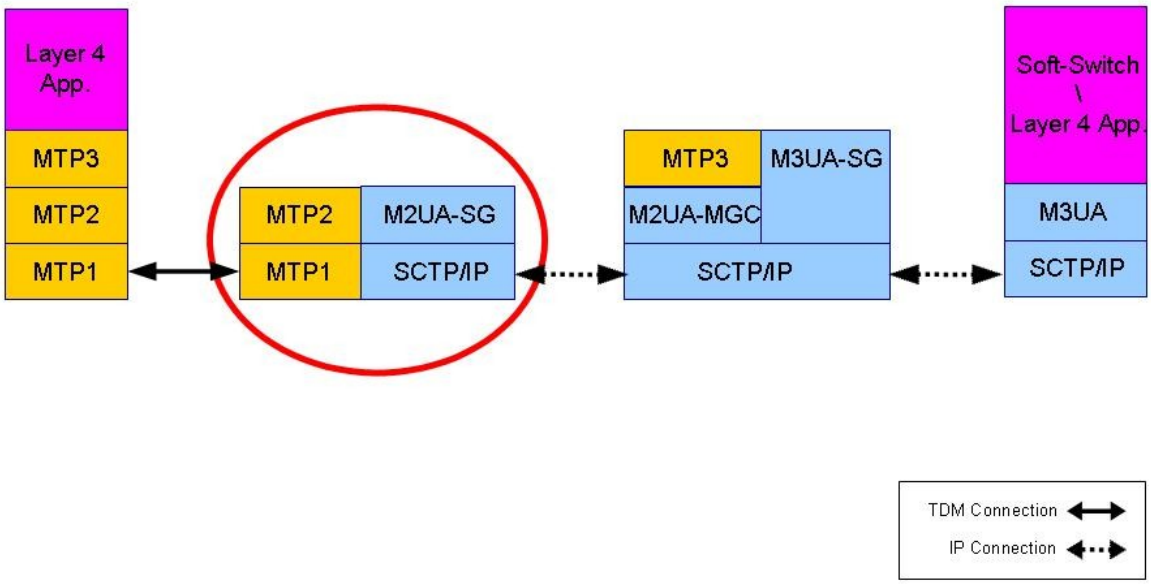
The SS7 network elements include these basic configurations:

- SS7 M2UA - Signaling Gateway (SG) Side
- SS7 M2UA – Media Gateway Controller (MGC) Side
- SS7 MTP3 Node
- SS7 MTP2 Tunneling
- SS7 SN Redundancy - MTP3 Shared Point Code
- SS7 Alias Point Code

5.5.1.1 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 the Media Gateway side.

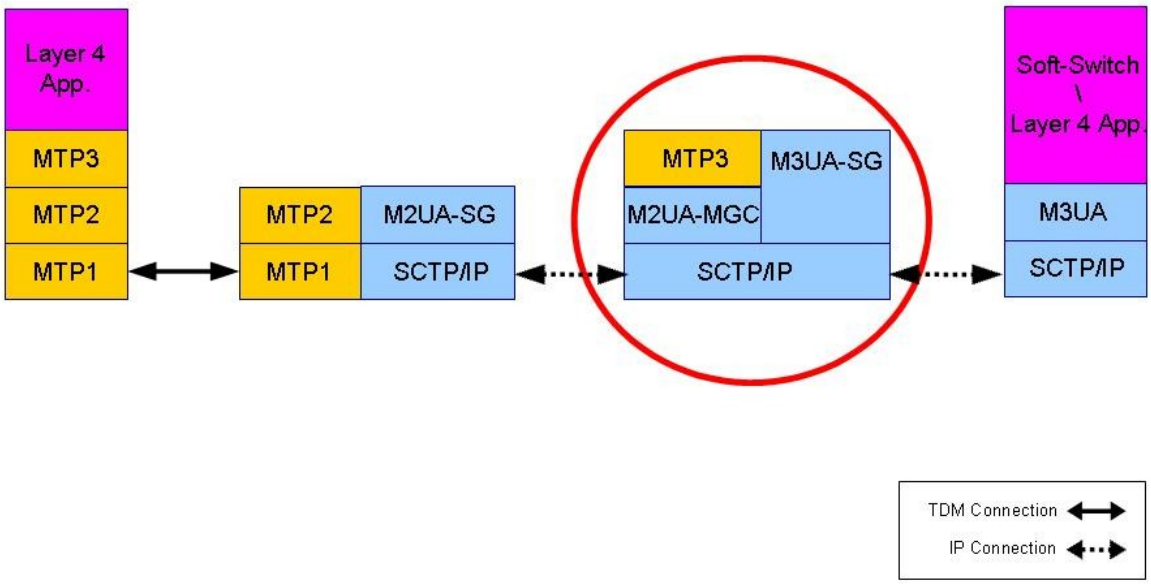
Figure 6: SS7 M2UA - SG Side



5.5.1.2 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 Softswitch. (MTP3 can also route MSUs to other SS7 network elements via other links).

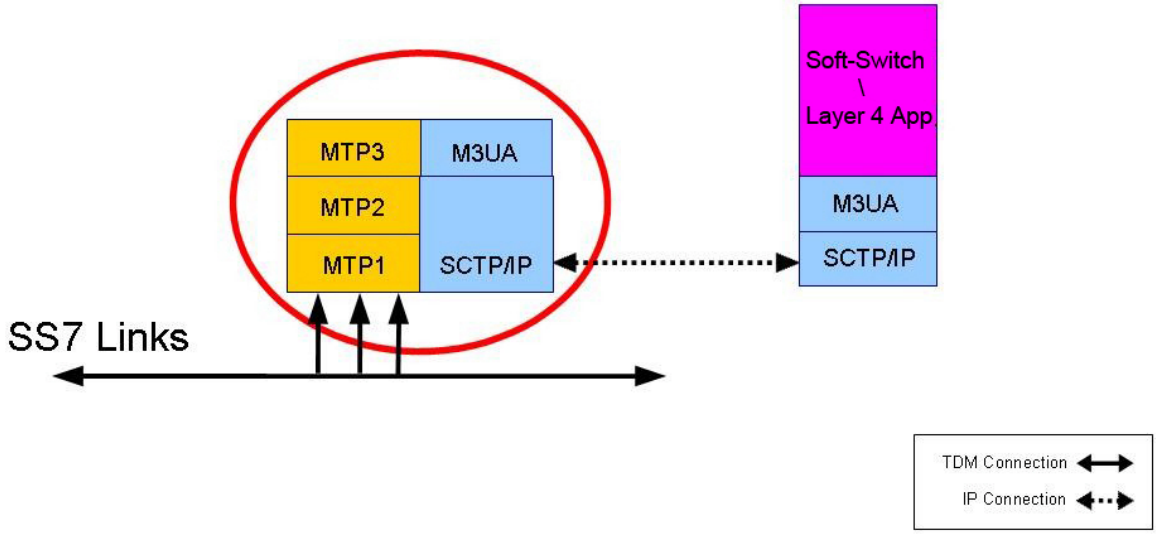
Figure 7: SS7 M2UA - MGC Side



5.5.1.3 SS7 MTP3 Node

The SS7 MTP3 Node is a classic MTP3 over MTP2 configuration. Links are incoming from the SS7 Network. The MTP3 payload is sent via M3UA to the Softswitch or routed to other SS7 network elements according to the MSU headers.

Figure 8: SS7 MTP3 Node

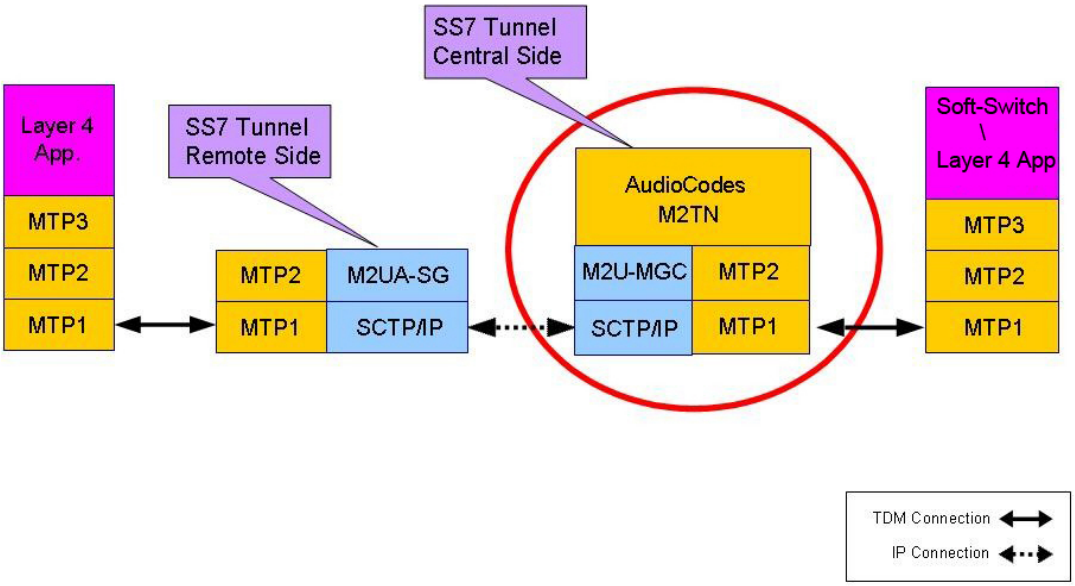


5.5.1.4 SS7 MTP2 Tunneling

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 9: SS7 MTP2 Tunneling



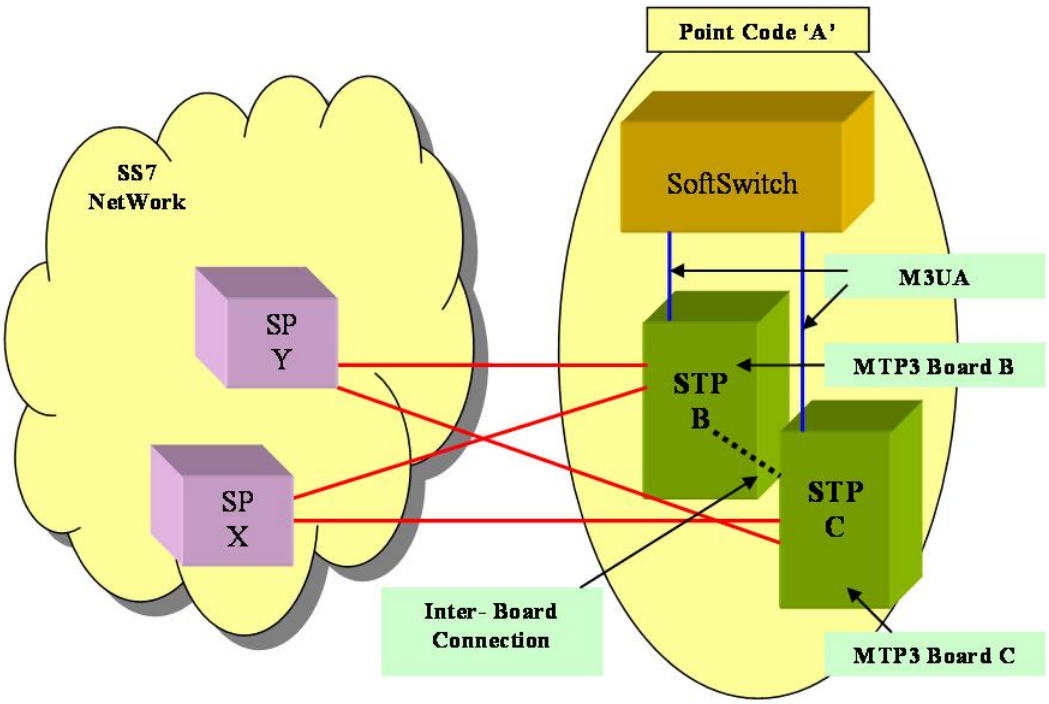
5.5.1.5 SS7 - MTP3 Shared Point Code (SN Redundancy) - Overview

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 two 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**.

Note: Both link-sets are distributed across the two devices.

Figure 10: MTP3 Shared Point Code Configuration Diagram

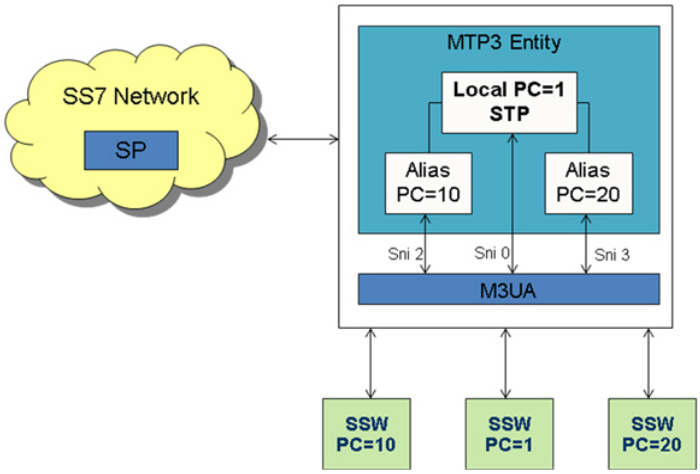


For more information, please refer to 'SS7 MTP3 Shared Point Code (SN Redundancy)' on page 399.

5.5.1.6 SS7 Alias Point Code

Alias Point Code provides the possibility to route signaling messages with a Destination Point Code (DPC) that is different than the board Point Code, from the TDM network to the IP network.

Figure 11: SS7 Alias Point Code Example



In this example, there are two softswitches with Point Codes 10 & 20, that are different from the board Point Code which is equal to 1.

MSUs with a DPC equal to 10 or 20 (that is different from board Point Code equal to 1), will be forwarded by the Gateway to the designated SoftSwitches, by M3UA.

In Alias Point Code architecture, Signaling Nodes must be configured as STP.

Alias Point Codes can be configured in three different ways, depending on the network architecture:

- All Point Codes (Local & Alias) are configured on the same Routing Context value, which means they belong to the same group (Application Server).
- Each Point Code has its own Routing Context value, which means that each Point Code is configured on a different Interface Group. All groups are configured on the same SCTP port.
- Each Point Code has its own Routing Context value, which means that each Point Code is configured on a different Interface Group. Each group is configured on a different SCTP port.

5.5.1.7 Configuration Options

In addition to the basic SS7 configurations described above, the following options are applicable:

- Two SS7 Nodes (SP/STP) can be configured per TPM.
- The device supports mixed SS7 link types, i.e. one device can have few MTP2 links and a number of M2UA links.
- Supported SS7 variants - ITU-T, ANSI and CHINA.
- 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.
- Fully Associated Links (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 below.

5.5.2 SS7 Parameters

The following describes the SS7 parameters.

5.5.2.1 SS7 *ini* File Global 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.

SS7 Parameters

Parameter Name	Description	Default	Range	Notes
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	Can't be changed on-the-fly.
SS7MTP3RdcyMode	Defines the SS7 MTP3-User Adaptation Layer redundancy mode. Determines the redundancy flavor. 0 = Disabled 1 = Enabled	0	0 or 1	Can't be changed on-the-fly.
SS7MTP3RdcyBoardNum	Defines the device number for the 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	Must be set. Can't be changed on-the-fly.
SS7MTP3RdcyKeepAlive Interval	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	Can't be changed on-the-fly.
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	Can't be changed on-the-fly.

SS7 Parameters

Parameter Name	Description	Default	Range	Notes
SS7MTP3RdcyTblSync Interval	In SS7 MTP3-User Adaptation Layer redundancy mode, defines the interval between SS7 tables automatic synchronizations between boards, in minutes. 0 = no automatic synchronization is activated.	0	0 to 30000	Can be changed on-the-fly.
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	2	0 and 2	Can't be changed on-the-fly.

5.5.2.2 SS7 *ini* File Table Parameters

SS7 Table Parameters

The following *ini* file Table Parameters are provided:

5.5.2.2.1 SS7 Signaling Node Timers Table Parameters

SS7 Signaling Node Timers Table Parameters

Parameter Name	Description	Default, msec	Range
SS7_SNTIMERS_INDEX	Index Field for line	0	0 to 4
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 Signaling Node Timers Table Parameters

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
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

5.5.2.2.2 SS7 Signaling LinkSet Timers Table Parameters

SS7 Signaling LinkSet Timers Table Parameters

Parameter Name	Description	Default, msec	Range
SS7_LKSETTIMERS_INDEX	Index Field for line	0	0 to 4
SS7_LKSETTIMERS_NAME	String name for LKSET timer-set	'LKSET_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
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

5.5.2.2.3 SS7 MTP2 Table Parameters

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_ErrorCorrection Method	SLI error correction method: 'B' for basic, 'P' for PCR. 0 is equivalent to 'B'.	'B'	'0', 'B' ,'P'
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', sec	50000	0-100000
SS7Mtp2Parms_T2	Timer T2: Timer 'not aligned', sec	150000	0-200000
SS7Mtp2Parms_T3	Timer T3: Timer 'aligned', sec	2000	0-20000
SS7Mtp2Parms_T4n	Timer T4: Proving period timer normal, period, sec	8200	0-15000
SS7Mtp2Parms_T4e	Timer T4: Proving period timer emergency period, sec	500	0-5000
SS7Mtp2Parms_T5	Timer T5: Timer 'sending SIB', sec	120	0-2400
SS7Mtp2Parms_T6	Timer T6: Timer 'remote congestion', sec	6000	0-10000
SS7Mtp2Parms_T7	Timer T7: Timer 'excessive delay of acknowledgment', sec	2000	0-5000

5.5.2.2.4 SS7 Signaling Nodes Table Parameters

SS7 Signaling Nodes Table Parameters

Parameter Name	Description	Default	Range
SS7_SN_INDEX	Index Field for line	0	0 to 1
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 = OFFLINE • 2 = INSERVICE	OFFLINE	0 or 2
SS7_SN_VARIANT	Variant of signaling node • 1 = ITU • 2 = ANSI • 3 = CHINA	ITU	1 to 3
SS7_SN_NI	Network Indicator of signaling node • 0 = INTERNATIONAL • 1 = INTERNATIONAL_SPARE • 2 = NATIONAL • 3 = NATIONAL_SPARE	INTERNATIONAL	0 to 3
SS7_SN_SP_STP	Routing function of signaling node • 0 = SP • 1 = STP	SP	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_CONGESTION_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 4
SS7_SN_ISUP_APP	Level 4 application that handles ISUP traffic for this signaling node • 0 = NONE • 4 = UAL	NONE (0)	0 or 4

SS7 Signaling Nodes Table Parameters

SS7_SN_SCCP_APP	Level 4 application that handles SCCP traffic for this signaling node • 0 = NONE • 4 = UAL	NONE (0)	0 or 4
SS7_SN_BISUP_APP	Level 4 application that handles BISUP traffic for this signaling node • 0 = NONE • 4 = UAL	NONE (0)	0 or 4
SS7_SN_TUP_APP	Level 4 application handling TUP traffic for this signaling node. • 0 = NONE • 4 = UAL	NONE (0)	0 or 4
SS7_SN_BICC_APP	Level 4 application handling BICC traffic for this signaling node. • 0 = NONE • 4 = UAL	NONE (0)	0 or 4

5.5.2.2.5 SS7 Alias Point Code Table Parameters

SS7 Alias Point Code Table Parameters

Parameter Name	Description	Default	Range
SS7_ALIAS_PC_SN_INDEX	First Index Field for line	0	0 to 1
SS7_ALIAS_PC_INNER_INDEX	Second Index Field for line	0	0 to 1
SS7_ALIAS_PC_NAME	String name for Alias PC	"ALIAS PC"	
SS7_ALIAS_PC_PC	Alias Point Code value	0	Unsigned Integer
SS7_ALIAS_PC_SNI	SNI (Signaling Network Indicator) of Alias Point-Code	2	2 - 5
SS7_ALIAS_PC_ADMINISTRATIVE_STATE	Administrative state of Alias PC • 0 = OFFLINE • 2 = INSERVICE	OFFLINE	0 or 2

5.5.2.2.6 SS7 Signaling Link Table Parameters

SS7 Signaling Link Table Parameters

Parameter Name	Description	Default	Range
SS7_LINK_INDEX	Defines the Index Field for line	0	0 to (MAX_SIGNALING_LINKS_PER_CARD-1)
SS7_LINK_NAME	Defines the String name for Link Params	'LINK'	
SS7_LINK_RDCY_BOARD	Defines the blade number in which the link is physically connected.	0	0 to 2
SS7_LINK_ADMINISTRATIVE_STATE	Defines the Administrative state of signaling link • 0 = OFFLINE • 2 = INSERVICE	OFFLINE	0 or 2
SS7_LINK_TRACE_LEVEL	Defines the Trace level of signaling link (level 2)	0	0 or 1

SS7 Signaling Link Table Parameters

SS7_LINK_L2_TYPE	Defines the Link layer type - defines level 2 media of signaling link • 0 = SS7_L2_TYPE_NONE (default) • 1 = SS7_L2_TYPE_MTP2 • 2 = S7_L2_TYPE_M2UA_MGC • 3 = SS7_L2_TYPE_SAAL	SS7_L2_TYPE_NONE	0 to 3
SS7_LINK_L3_TYPE	Link high layer type - defines Level 3 or L2 high layer of signaling link • 0 = SS7_L3_TYPE_NONE (default) • 1 = SS7_L3_TYPE_M2UA_SG • 2 = SS7_L3_TYPE_MTP3 • 3 = SS7_L3_TYPE_MTP2_TUNNELING • 4 = SS7_L3_TYPE_MTP2Oip	SS7_L3_TYPE_NONE	0 to 4
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 = ITU • 2 = ANSI • 3 = CHINA	ITU	1 to 3
SS7_LINK_MTP2_ATTRIBUTES	MTP2 attributes of signaling link (TDM)	3	0 to _MAX_C7_MTP2_PARAMS_INDEX
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 = NORMAL ▪ 1 = EMERGENCY	EMERGENCY	0 to 1

SS7 Signaling Link Table Parameters

SS7_LINK_TNL_CONGESTION_MODE	MTP2 Tunneling: Congestion mode of signaling links in tunnel ▪ 0 = ACCEPT ▪ 1 = DISCARD	ACCEPT	0 to 1
SS7_LINK_TNL_WAIT_START_COMPLETE_TIMER	MTP2 Tunneling: wait start complete timer, msec	30000	500 to 4294967295
SS7_LINK_TNL_OOS_START_DELAY_TIMER	MTP2 Tunneling: OOS start delay timer, msec	5000	500 to 4294967295
SS7_LINK_TNL_WAIT_OTHER_SIDE_INSV_TIMER	MTP2 Tunneling: Wait for other side to be in service timer, msec	30000	500 to 4294967295

5.5.2.2.7 SS7 Signaling LinkSets Table Parameters

SS7 Signaling LinkSets Table Parameters

Parameter Name	Description	Default Value	Valid Range
SS7_LINKSET_SN_INDEX	First Index Field for line	0	0 to 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 = OFFLINE ▪ 2 = INSERVICE	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 4

5.5.2.2.8 SS7 Signaling LinkSet-Links Table Parameters

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 1
SS7_LINKSETLINK_LINKSET_INDEX	Second Index Field for line: Signaling LinkSet Number	0	0 to (MAX_LINKSETS_PER_SN-1)
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	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

5.5.2.2.9 SS7 RouteSets Table Parameters

SS7 RouteSets Table Parameters

Parameter Name	Description	Default	Range
SS7_ROUTESET_SN_INDEX	First Index Field for line: Signaling Node Number.	0	0 to 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 = OFFLINE ▪ 2 = INSERVICE	OFFLINE	0 or 2
SS7_ROUTESET_DPC	Destination Point-Code of signaling RouteSet.	0	
SS7_ROUTESET_MASK	Mask for routes within signaling RouteSet.	15	0 to 255

5.5.2.2.10SS7 RouteSet-Routes Table Parameters

SS7 RouteSet-Routes Table Parameters

Parameter Name	Description	Default	Range
SS7_ROUTESETROUTE_SN_INDEX	First Index Field for line: Signaling Node Number	0	0 to 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	NIL	0 to MAX_LINKSETS_PER_SN-1
SS7_ROUTESETROUTE_PRIORITY	Priority of route within RouteSet	0	0 to 254

5.5.2.2.11Static Routing Context Table

Routing Context Table Parameters

Parameter Name	Description	Default	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 5
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

Routing Context Table Parameters

Parameter Name	Description	Default	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

5.5.2.2.12SigTran Interface Groups Table Parameters

SigTran Interface Groups Table Parameters

Parameter Name	Description	Default	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	UAL type ((IUA/M2UA/M3UA) of interface group layer. Possible values: ▪ 0 = no_layer ▪ 1 = iua ▪ 2 = m2ua ▪ 3 = m3ua ▪ 4 = m2tunnel ▪ 6 = dua	0	0, 1, 2, 3, 4, 6
SS7_SIG_TRAF_MODE	SigTran group traffic mode.	1	1 to 3
SS7_SIG_T_REC	T(r) - Recovery Timer (in msecs) of interface group.	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

SigTran Interface Groups Table Parameters

Parameter Name	Description	Default	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 (1 = ITU, 2 = ANSI, 3 = 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

5.5.2.2.13SigTran Interface IDs Table Parameters

SigTran Interface IDs Table Parameters

Parameter Name	Description	Default	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. Possible values: ▪ 0 = no_layer ▪ 1 = iua ▪ 2 = m2ua ▪ 4 = m2tunnel ▪ 6 = dua	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

5.5.2.2.14SS7 MTP3 Redundancy SN Table Parameters

SS7 MTP3 Redundancy SN Table Parameters

Parameter Name	Description	Default	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 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	

5.5.2.3 Other Dependencies in *ini* File:

Trunk Protocol Types

Trunks that carry SS7 Links, must be configured with one of the following protocol types:

- 4 = T1_TRANSPARENT
- 5 = E1_TRANSPARENT31
- 6 = E1_TRANSPARENT30

Note: The Trunks Protocol definition must appear in the *ini* file before the SS7/SIGTRAN Table definition.

5.5.2.4 SS7 Constraints

The following lists SS7 limitations.

5.5.2.4.1 SS7 Capacity Limitations

- Up to 2 signaling nodes can be configured per TPM. (When using Shared-Point Code, only one SN is supported.)
- Up to 2 Alias Point Codes can be configured per Signaling Node, but only 1 Signaling Node can be configured with Alias Point Codes
- Up to 64 signaling links can be configured per TPM
- Up to 32 link-sets can be configured per Signaling Node
- Up to 8 links can be configured per link-set
- Up to 30 route-sets can be configured per Signaling Node
- Up to 4 link-sets can be configured per route-set (routes per route-set)

5.5.2.4.2 SS7 MTP3 Signaling Node (SN) Redundancy Limitations

- Up to two blades can be configured as Shared Point Code.
- Blade redundancy parameters must be configured in the *ini* file. They cannot be changed on-the-fly (except for SS7MTP3RdcyTblSyncInterval which can be changed).
- The Add & Delete commands can be performed on the SN-Blade Table Configuration when the SN is set to Offline.
- *ini* file configurations, after reset, are not passed to remote blades.
- SN operations (InService/Start/Stop) are local to the blade they are performed on.
- On-the-fly configurations that are performed on one blade will be passed only to the remote blades that established a TCP connection with that blade. (For this reason, redundancy SN-Blade table configurations are not passed between the new blade that is configured and the other blades.)
- SS7 operations that are performed on one blade will be passed only to the remote blades that have an InService x-link to that blade.
- On-the-fly configuration of a link, (Create/InService/Offline/Delete), must be performed from the blade that the link physically belongs to.
- Only one SCTP port should be configured per blade.
- SS7 and UAL configuration MUST be the same on blades that share the same Point Code.

5.5.2.5 SIGTRAN Constraints

Note: AudioCodes plans to further develop the SS7 and SigTran capabilities in future versions. Some of the changes may not be backward-compatible with the current version (configuration files, API, modes of operation, etc.)

- *ISDNNFASInterfaceID* is not used for IUA. It is only used for the NFAS application.
- The *Q931RelayMode* parameter is not needed for IUA.
- Only SigTran Override Mode is supported.
- Textual Interface Identifier is not supported.
- The maximum number of SigTran interface groups is 32.
- The maximum number of SigTran interface IDs is 84.
- The maximum number of static Routing Context that can be configured is 32.
- Up to three different SCTP ports can be configured on the one blade.
- It is not recommended to configure a SigTran local SCTP port with a value of "0".
- Only in a M3UA configuration, can more than one group be configured on the same SCTP port.
- M3UA Routing Context can be configured statically or dynamically. Only one configuration method can be used on a blade.
- In a dynamic Routing Context configuration, only one RC value can be registered per ASP association.

5.5.3 Examples of SS7 *ini* File Configurations

This section provides examples of *ini* file configurations 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.

5.5.3.1 SS7 M2UA - SG Side *ini* File Example

For the SS7 M2UA - SG Side *ini* file configuration 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).
- An MTP2 Media Gateway Controller device should be connected (using SCTP over IP) to the MTP2 SG device.

The following is an example of SS7 M2UA - SG Side *ini* file configuration.

```
[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_TUP_APP, SS7_SN_BICC_APP;

SS7_SN_TABLE 0 = SN_0, 1, 2, 1, 0, 1, 3,  8, 0, 4, 4, 4, 4, 4;
```

```
[\SS7_SN_TABLE]


[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, 0, 1, 2, 1, 2,  0, 14,  1, 0, 5, 50;
SS7_LINK_TABLE 1 = LINK_1, 1, 1, 2, 1, 2,  0, 14,  1, 0, 5, 50;
SS7_LINK_TABLE 2 = LINK_2, 0, 1, 2, 1, 2,  0, 18,  1, 0, 5, 50;
SS7_LINK_TABLE 3 = LINK_3, 1, 1, 2, 1, 2,  0, 18,  1, 0, 5, 50;

[\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 = lkset_0, 2, 4, 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 = ROUTESET_0, 2, 4;

[ \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;

[ \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 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 ]
```

5.5.3.2 SS7 M2UA - Media Gateway Controller Side *ini* File Example

For the SS7 M2UA - Media Gateway Controller Side *ini* file configuration 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).
- 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).
- There are 2 interface groups:
  - used for the M2UA SG <=> M2UA Media Gateway Controller connection
  - used for the M3UA SG <=> M3UA Media Gateway Controller connection
- There is 1 Routing Context configured, for the M3UA SG.
- There are 4 Interface IDs defined: 1 per link (M2UA Media Gateway Controller side). 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).
- An MTP2 SG device should be connected (over IP) to the MTP2 Media Gateway Controller device.

The following is an example of SS7 M2UA - Media Gateway Controller Side *ini* file configuration.

```
[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_TUP_APP, SS7_SN_BICC_APP;
SS7_SN_TABLE 0 = SN_0, 0, 2, 1, 0, 0, 11, 8, 0, 4, 4, 4, 0, 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, 0, 2, 2, 2, 4, 50;
SS7_LINK_TABLE 1 = link_1, 0, 2, 2, 2, 4, 12;
SS7_LINK_TABLE 2 = link_2, 0, 2, 2, 2, 4, 18;
SS7_LINK_TABLE 3 = link_3, 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, 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 = rtset0, 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 ]

; M3UA SG SIDE DEFINITION:
;


[ 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 2 =  2, 83, 3, 1, 2000, 2000, 30000, 1, 0,
2905, 1, 0, 1, 1;
```

```
[ \SS7_SIG_IF_GROUP_TABLE ]

;
; M2UA MGC SIDE DEFINITION:
;


[ 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, 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_INT_ID_TABLE 1 = 50, Customer1, 4, 2, 0;
SS7_SIG_INT_ID_TABLE 2 = 12, Customer2, 4, 2, 1;
SS7_SIG_INT_ID_TABLE 3 = 18, Customer3, 4, 2, 2;
SS7_SIG_INT_ID_TABLE 4 =  1, Customer4, 4, 2, 3;
[ \SS7_SIG_INT_ID_TABLE ]
```

5.5.3.3 SS7 MTP3 Node ini File Example

For the SS7 MTP3 Node *ini* file configuration 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 Routing Context configured, with 2 filters - a filter per SN.
- There is 1 Interface Group. Both SNs are using it.
- This file is intended for ITU link variant (E1 trunks).

The following is an example of SS7 MTP3 Node *ini* file configuration.

```
[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_TUP_APP, SS7_SN_BICC_APP;

SS7_SN_TABLE 0 = SN_0, 0, 2, 1, 0, 0, 11, 8, 0, 4, 4, 4, 4, 4;
```

```
SS7_SN_TABLE 1 = SN_1, 0, 2, 1, 0, 0,  4, 8, 0, 4, 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, 0, 2, 1, 2, 0, 16, 1, 0, 5, 20;
SS7_LINK_TABLE 1 = link_1, 0, 2, 1, 2, 1, 16, 1, 0, 5, 20;
SS7_LINK_TABLE 2 = link_2, 0, 2, 1, 2, 0, 17, 1, 0, 5, 20;
SS7_LINK_TABLE 3 = link_3, 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;

SS7_LINKSET_TABLE 0, 0 = lkset0_SN_0, 2, 4, 0;
SS7_LINKSET_TABLE 1, 0 = lkset0_SN_1, 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;

SS7_LINKSETLINK_TABLE 0, 0, 0 = 0, 0;
SS7_LINKSETLINK_TABLE 0, 0, 1 = 2, 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;

SS7_ROUTESET_TABLE 0, 0 = RTESET0_SN_0, 2, 4;
SS7_ROUTESET_TABLE 1, 0 = RTESET0_SN_1, 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;

SS7_ROUTESETROUTE_TABLE 0, 0, 0 = 0, 0;
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 ]
```

5.5.3.4 SS7 MTP2 Tunneling ini File Example

For the SS7 MTP2 Tunneling *ini* file configuration 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).
- The Media Gateway Controller gateway connects (over IP) to the SG gateway

The following is an example of SS7 MTP2 Tunneling *ini* file configuration.

```
[ SS7_LINK_TABLE ]
FORMAT SS7_LINK_INDEX = SS7_LINK_NAME, SS7_LINK_TRACE_LEVEL,
SS7_LINK_ADMINISTRATIVE_STATE,SS7_LINK_L2_TYPE, SS7_LINK_L3_TYPE,
SS7_LINK_GROUP_ID, SS7_LINK_M2UA_IF_ID;
SS7_LINK_TABLE 1 = new_link_1, 0, 2, 2, 3, 4, 50;
SS7_LINK_TABLE 3 = new_link_3, 0, 2, 2, 3, 4, 12;
SS7_LINK_TABLE 5 = new_link_5, 0, 2, 2, 3, 4, 18;
SS7_LINK_TABLE 7 = new_link_7, 0, 2, 2, 3, 4,  1;

[ \SS7_LINK_TABLE ]

[ SS7_LINK_TABLE ]
FORMAT SS7_LINK_INDEX = SS7_LINK_NAME, SS7_LINK_TRACE_LEVEL,
SS7_LINK_ADMINISTRATIVE_STATE,SS7_LINK_L2_TYPE, SS7_LINK_L3_TYPE,
SS7_LINK_TRUNK_NUMBER,SS7_LINK_TIMESLOT_NUMBER,
SS7_LINK_LAYER2_VARIANT,SS7_LINK_MTP2_ATTRIBUTES,SS7_CONGESTION_LO
W_M ARK, SS7_CONGESTION_HIGH_MARK, SS7_LINK_TNL_MGC_LINK_NUMBER,
SS7_LINK_TNL_ALIGNMENT_MODE, SS7_LINK_TNL_CONGESTION_MODE,
SS7_LINK_TNL_WAIT_START_COMPLETE_TIMER,
SS7_LINK_TNL_OOS_START_DELAY_TIMER,
SS7_LINK_TNL_WAIT_OTHER_SIDE_INSV_TIMER;


SS7_LINK_TABLE 0 = new_link_0, 0, 2, 1, 3, 0, 15, 1, 0, 5, 50, 1,
1,  0, 30000, 5000, 30000;
SS7_LINK_TABLE 2 = new_link_2, 0, 2, 1, 3, 3, 12, 1, 0, 5, 50, 3,
1, 0, 30000, 5000, 30000;
SS7_LINK_TABLE 4 = new_link_4, 0, 2, 1, 3, 6,  7, 1, 0, 5, 50, 5,
1, 0, 30000, 5000, 30000;
SS7_LINK_TABLE 6 = new_link_6, 0, 2, 1, 3, 7,  3, 1, 0, 5, 50, 7,
1, 0, 30000, 5000, 30000;
[ \SS7_LINK_TABLE ]

[ SS7_SIG_IF_GROUP_TABLE ]
```

```
FORMAT SS7_SIG_IF_GR_INDEX = SS7_IF_GR_ID,SS7_SIG_SG_MGC,
SS7_SIG_LAYER, SS7_SIG_TRAF_MODE, SS7_SIG_T_REC, SS7_SIG_T_ACK,
SS7_SIG_T_HB, SS7_SIG_MIN_ASP, SS7_SIG_BEHAVIOUR,
SS7_LOCAL_SCTP_PORT, SS7_SIG_NETWORK, SS7_DEST_SCTP_PORT,
SS7_DEST_IP, SS7_MGC_MX_IN_STREAM, SS7_MGC_NUM_OUT_STREAM;
SS7_SIG_IF_GROUP_TABLE 4 = 4, 77, 4, 1, 2000, 2000, 30000, 1, 0,
2904, 1,2904,168.100.0.2,3,3;

[ \SS7_SIG_IF_GROUP_TABLE ]

[ SS7_SIG_INT_ID_TABLE ]
FORMAT SS7_SIG_IF_ID_INDEX = SS7_SIG_IF_ID_VALUE,
SS7_SIG_IF_ID_NAME, SS7_SIG_IF_ID_OWNER_GROUP,
SS7_SIG_IF_ID_LAYER, SS7_SIG_IF_ID_NAI;
SS7_SIG_INT_ID_TABLE 7 = 50, Customer1, 4, 4, 1;
SS7_SIG_INT_ID_TABLE 8 = 12, Customer2, 4, 4, 3;
SS7_SIG_INT_ID_TABLE 9 = 18, Customer3, 4, 4, 5;
SS7_SIG_INT_ID_TABLE 10 = 1, Customer4, 4, 4, 7;
[ \SS7_SIG_INT_ID_TABLE ]
```

5.5.3.5 SS7 MTP3 Shared Point Code *ini* File Example

For the SS7 MTP3 shared point code *ini* file configuration example, take into account the following notes:

- There are 2 blades with MTP3 shared point code.
- The 2 blades should have the same SS7 *ini* configuration, except for the *SS7MTP3RdcyBoardNum* parameter, that should be unique in each blade's *ini* file.
- 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).

The following is an example of SS7 MTP3 Redundancy *ini* file configuration:

```
[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_TUP_APP, SS7_SN_BICC_APP;

SS7_SN_TABLE 0 = SN_0, 1, 2, 1, 0, 1, 3,  8, 0, 4, 4, 4, 4, 4;

[\SS7_SN_TABLE]


[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, 0, 1, 2, 1, 2,  0, 14,  1, 0, 5, 50;
SS7_LINK_TABLE 1 = LINK_1, 1, 1, 2, 1, 2,  0, 14,  1, 0, 5, 50;
SS7_LINK_TABLE 2 = LINK_3, 0, 1, 2, 1, 2,  0, 18,  1, 0, 5, 50;
SS7_LINK_TABLE 3 = LINK_2, 1, 1, 2, 1, 2,  0, 18,  1, 0, 5, 50;

[\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 = lkset_0, 2, 4, 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 = ROUTESET_0, 2, 4;

[\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;
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 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]
```

5.5.3.6 SS7 Alias Point Code *ini* File Example

For the SS7 Alias Point Code *ini* file configuration example, take into account the following notes:

- This *ini* file defines one MTP3 SN (signaling nodes). SN must be configured as STP.
- There are two Point Codes configured for in the SN (APC 4, APC 5).
- There are two SS7 links of type: MTP2->MTP3.
- There is one LinkSet per SN with two links.
- There is one RouteSet per SN with one LinkSet.
- There are three entries in RC table - one for each Point Code (1 OPC + 2 APC).
- There are three Interface groups - one for each Point Code (1 OPC + 2 APC).
- This file is intended for the ITU link variant (E1 trunks).

The following is an example of SS7 Alias Point Code *ini* file configuration.

```
[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_TFC, 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_TUP_APP, SS7_SN_BICC_APP;

SS7_SN_TABLE 0 = SN_0, 1, 2, 1, 0, 1, 0, 11, 8, 0, 4, 4, 0, 0, 0;

[\SS7_SN_TABLE]



[SS7_ALIAS_POINT_CODE_TABLE]

FORMAT SS7_APC_SN_INDEX, SS7_APC_INNER_INDEX = SS7_APC_PC,
SS7_APC_SNI;

SS7_ALIAS_POINT_CODE_TABLE 0, 0 = 2, 2;
SS7_ALIAS_POINT_CODE_TABLE 0, 1 = 3, 3;

[\SS7_ALIAS_POINT_CODE_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, 0, 2, 1, 2, 0, 16, 1, 0, 5, 20;
SS7_LINK_TABLE 1 = link_1, 0, 2, 1, 2, 0, 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;

SS7_LINKSET_TABLE 0, 0 = lkset0, 2, 4, 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]


[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, 2, 4;

[\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, 3, -1, -1, -1, -1, -1, -1, -1;
```

```
SS7_ROUTING_CONTEXT_TABLE 1, 0 = 2, 4, 7, -1, -1, 3, -1, -1, -1;
SS7_ROUTING_CONTEXT_TABLE 2, 0 = 3, 4, 7, -1, -1, 3, -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_DEST_SCTP_PORT,
SS7_DEST_IP, SS7_MGC_MX_IN_STREAM, SS7_MGC_NUM_OUT_STREAM,
RdcyBoardNum, SS7_SIG_RC_INDEX, SS7_SIG_RC_VALUE,
SS7_SIG_NETWORK_APPEARANCE;

SS7_SIG_IF_GROUP_TABLE 1 = 1, 83, 3, 1, 2000, 2000, 30000, 1, 0,
2905, 2, 65534, 0.0.0.0, 2, 2, 0, 0, 3, -1;
SS7_SIG_IF_GROUP_TABLE 2 = 2, 83, 3, 1, 2000, 2000, 30000, 1, 0,
2906, 2, 65534, 0.0.0.0, 2, 2, 0, 1, 9, -1;
SS7_SIG_IF_GROUP_TABLE 3 = 3, 83, 3, 1, 2000, 2000, 30000, 1, 0,
2907, 2, 65534, 0.0.0.0, 2, 2, 0, 2, 12, -1;

[\SS7_SIG_IF_GROUP_TABLE]
```

5.5.4 SS7 Tunneling Feature

The following describes the SS7 Tunneling feature.

5.5.4.1 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 12: M2UA Architecture

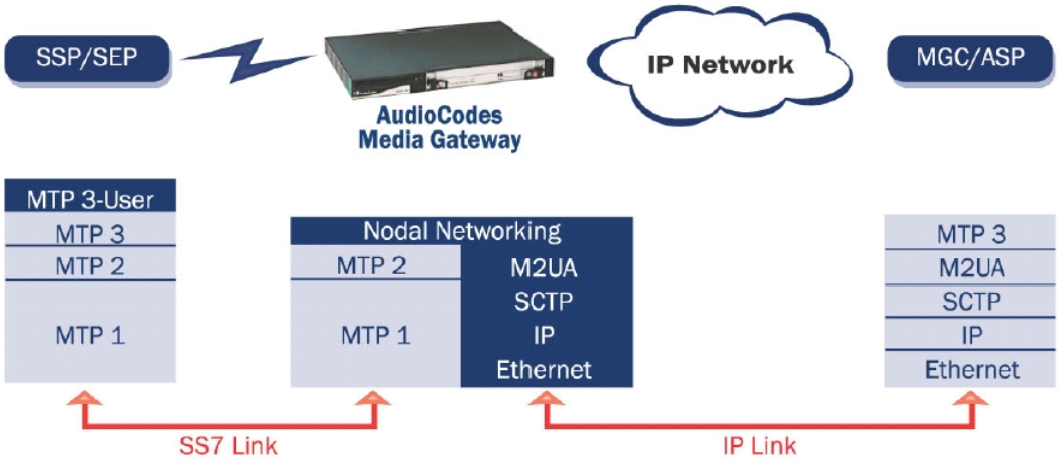
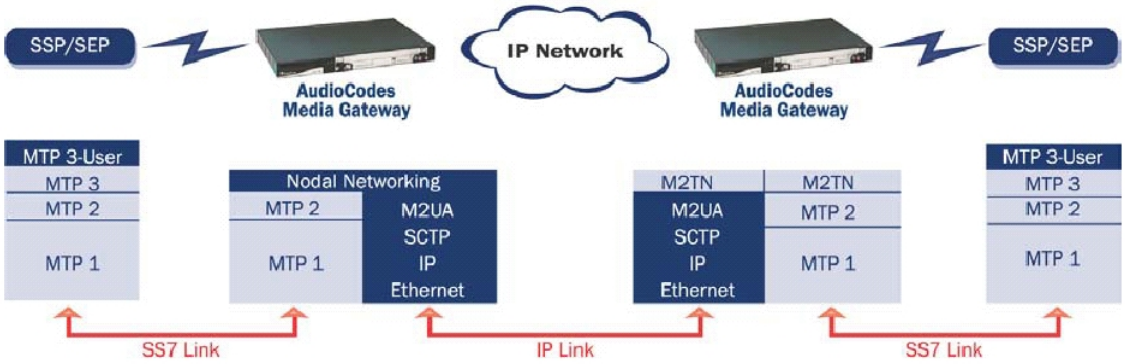


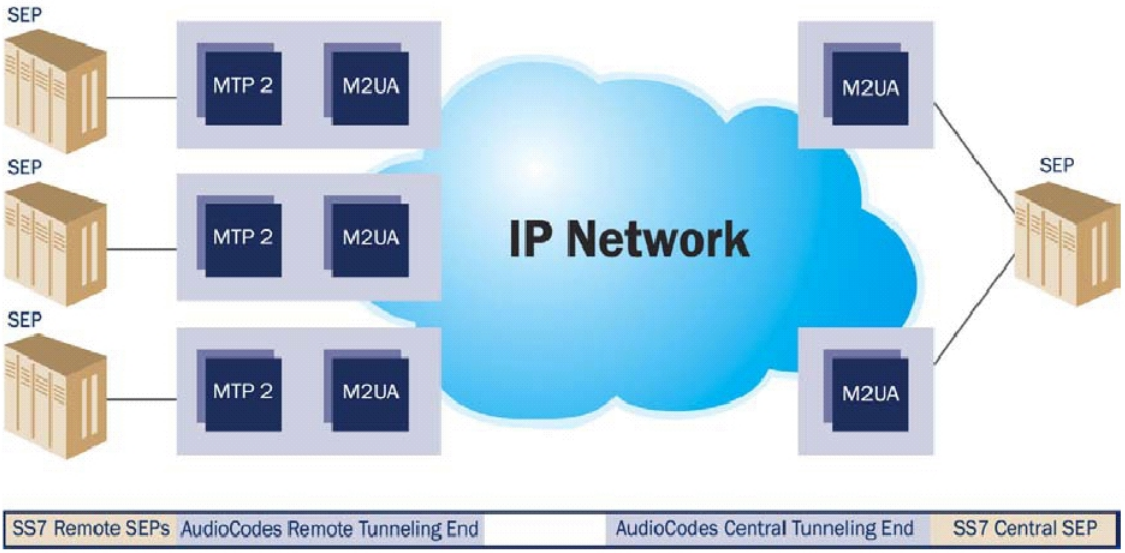
Figure 13: M2TN Architecture



5.5.4.2 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 affected by the fact that they are transported over IP.

Figure 14: Protocol Architecture for MTP2 Tunneling



5.5.4.3 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.

5.5.5 IUA/DUA

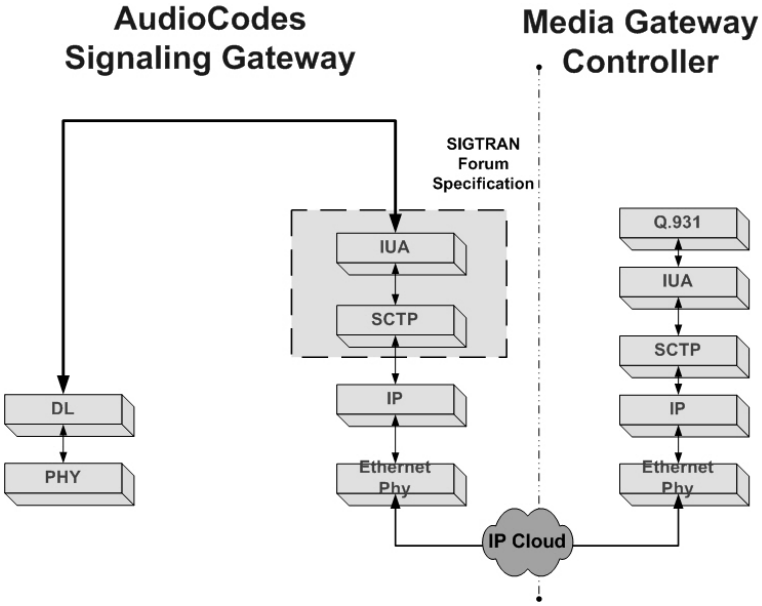
The following AudioCodes devices support the relay of ISDN signaling messages using SIGTRAN IUA/DUA and SCTP protocols. Refer to the figures below:

- TP-8410
- TP-6310

5.5.5.1 IUA (ISDN User Adaptation)

A signaling message entering the device from the ISDN connection goes through the Data Link Layer. The Q.931 PDU is then relayed to the Media Gateway Controller using IUA over SCTP over IP. The Media Gateway Controller performs SCTP and IUA layers on its side and then completes the upper signaling Q.931 layer. The reverse direction functions in the same way.

Figure 15: IUA Signaling Layers in SG and MGC



5.5.5.1.1 Configuring SIGTRAN IUA

Use *ini* file parameter values and SIGTRAN tables to configure an IUA connection.

SIGTRAN IUA Configuration Parameters

ini File Field Name (X is the trunk number)	Valid Range	Description
ProtocolType_X	acPROTOCOL_TYPE_T1_IUA = 28 or: acPROTOCOL_TYPE_E1_IUA = 29	IUA PSTN protocol type causes the IUA layer to be above the DL layer.

The Group and Interface SIGTRAN tables must be configured as shown in the *ini* file example below:

```
[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, 83, 1, 1, 2000, 2000, 30000, 1, 0,
9900, 1, 9900, 0, 3, 3;
[\SS7_SIG_IF_GROUP_TABLE]
```

- The SS7_SIG_SG_MGC value is 83 (ASCII value of 'S' for SG)
- The SS7_SIG_LAYER value is 1 for IUA.

```
[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_INT_ID_TABLE 0 = 100, Interface0, 1, 1, 0;
[\SS7_SIG_INT_ID_TABLE]
```

- The SS7_SIG_IF_ID_OWNER_GROUP should be the group number to which the Interface-Id belongs.
- The SS7_SIG_IF_ID_LAYER value is 1 for IUA.
- The SS7_SIG_IF_ID_NAI must be the number of a trunk configured as IUA.

5.5.5.1.2 Support IUA Behind NAT

In order to support IUA signaling gateway functionality behind a NAT, the Signaling Gateway must initiate SCTP (by sending an SCTP init to the MGC side). After an SCTP association is established, the SG waits for ASP commands from the MGC. This is performed by configuring a Group and an Interface-Id in the Sigtran tables, as shown below.

```
[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]
```

- The SS7_SIG_SG_MGC parameter value for NAT is 1.
- SS7_DEST_IP is the 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.

```
[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_INT_ID_TABLE 0 = 100, Interface0, 1, 1, 0;
[\SS7_SIG_INT_ID_TABLE]
```

- SS7_SIG_IF_ID_OWNER_GROUP must be the SS7_IF_GR_ID value of the group to which the Interface-Id belongs.
- SS7_SIG_IF_ID_LAYER parameter value for IUA is 1.

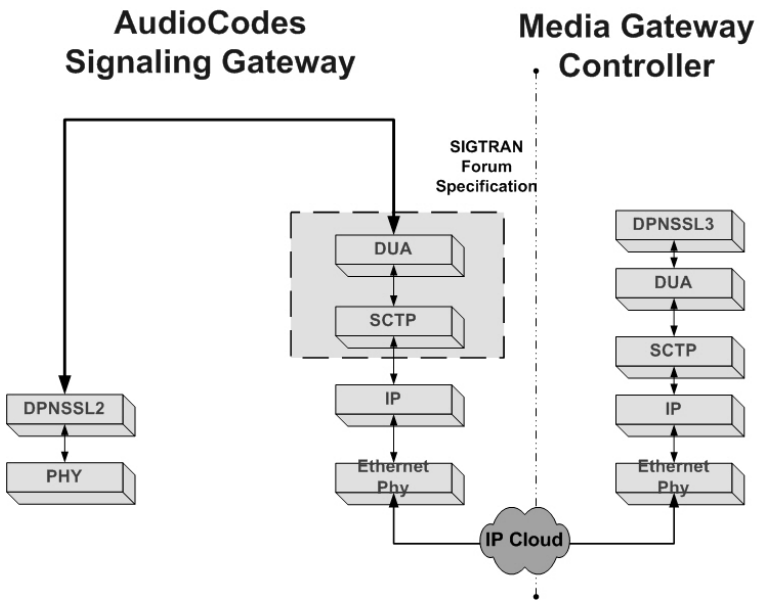
5.5.5.2 DUA (DPNSS User Adaptation)

DUA is based on the draft-ietf-sigtran-dua-08 draft published by the IETF. It is implemented with BTNR 188 DPNSS Layer 2 (an early ISDN protocol in the U.K.; British Telecom Network Requirements). A signaling message entering the device from the DPNSSL2 connection goes through the Data Link Layer. A DPNSSL2 PDU is then relayed to the Media Gateway Controller using DUA over SCTP over IP. The Media Gateway Controller performs SCTP and DUA layers on its side and then completes the upper signaling layers of the **IUA Signaling Layers in SG and MGC** figure above. The reverse direction functions in the same way.

DPNSSL2 Protocol:

1. The link layer can use 30 real channels; an additional 30 virtual channels can be used for services.
2. The link layer uses compelled signaling with the far end.

Figure 16: DPNSS Signaling Layers for SG and MGC



5.5.5.2.1 Configuring SIGTRAN DUA

Use *ini* file parameter values and Sigtran tables to configure a DUA connection.

SIGTRAN DUA Configuration Parameters

ini File Field Name (X is the trunk number)	Valid Range	Description
ProtocolType_X	acPROTOCOL_TYPE_E1_DUA = 37	DUA PSTN protocol type causes the DUA layer to be above the DPNSSL2 layer.
DPNSSBehaviour	Ulong	DPNSS behavior bit field for options implementation.
DPNSSNumRealChannels	Char - Valid range 1-30	Number of real B-channels used for voice. Default = 30.
DPNSSNumVirtualChannels	Char - Valid range 0-30	Number of virtual B-channels used for services. Default = 0.

DPNSSBehaviour Bits Values

DPNSS_BEHAV_STOP_SABMR_AFTER_NL_AND_NT1 bit: (bit #0, bitmask 0x0001

when 1: DPNSS stops repeating SABMR after NL and NT1 limits are exceeded

when 0: DPNSS continues repeating SABMR after NL and NT1 limits are exceeded

Default is 0 (continue repeating SABMR)

DPNSS_BEHAV_FULL_STARTUP_SUCCESS bit: (bit #1, bitmask 0x0002)

when 1: the startup procedure is considered a SUCCESS only when ALL DLCs succeed to reset

when 0: the startup procedure is considered a SUCCESS as soon as 1 DLC succeed to reset

Default is 0: (only partial reset is considered a success)

DPNSS_BEHAV_DLC_OOS_AFTER_NL_AND_NT1 bit: (bit #2, bitmask 0x0004)

when 1: the DLC is declared OOS after NL and NT1 limits are exceeded

when 0: DLC reset occurs after NL and NT1 limits are exceeded

Default is 0

DPNSS_BEHAV_DLC_OOS_WHEN_L3_Q_FULL bit: (bit #3, bitmask 0x0008)

when 1: the DLC is declared OOS when the L3 queue limit exceeds the config value

when 0: simply discard new L3 msgs when the L3 queue limit exceeds the config value

Default is 0

The SIGTRAN tables configuration is very similar to the IUA configuration described in the **Configuring SIGTRAN IUA** sub-section above, except that in both tables, the value of the Signaling Layer parameter should be 6 (for DUA).

5.5.5.2.2 DUA Behind NAT Support

The DUA configuration is very similar to IUA configuration. Follow the instructions of the **Support IUA Behind NAT** sub-section.

In the SS7_SIG_IF_GROUP_TABLE table, the SS7_SIG_LAYER value is 6 for DUA.

In the SS7_SIG_INT_ID_TABLE table, the SS7_SIG_IF_ID_LAYER parameter value is 6 as well.

5.5.5.3 PSTN Behavior in an IP Disconnect Condition

This feature enables a special behavior within the PSTN interface which occurs only when the signaling gateway is controlled by MEGACO (controlled by the *DisconnectBehavior* parameter) or by TPNCP (controlled by the TPNCP parameter).

When an H.248/TPNCP 'Disconnect' event occurs, the following gateway trunk action results:

- For an ISDN trunk, the Q.921 D-channel goes out of service.
- For an IUA trunk, the Q.921 D-channel goes out of service.
- For a DUA trunk, an AIS alarm is sent on the TDM side.
- For a CAS trunk, an AIS alarm is sent on the TDM side.

All other protocol trunks do not change status if an IP disconnect condition occurs.

When an H.248/TPNCP 'Connect' event (MEGACO or TPNCP connection) occurs, all changed status trunks (re)change their state as follows:

- For an ISDN trunk, the Q.921 D-channel is unblocked.
- For an IUA trunk, the Q.921 D-channel is unblocked.
- For a DUA trunk, the AIS alarm is cleared on the TDM side.
- For a CAS trunk, the AIS alarm is cleared on the TDM side.

5.5.5.4 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.

5.5.6 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 32 Static Routing Contexts are supported.
- Routing Context can be configured statically or dynamically.

5.5.6.1 Routing Context Static Configuration

- The static Routing Context parameters should be configured using the "SS7_ROUTING_CONTEXT_TABLE" table. (Please refer to SS7 Static Routing Context Table as described in the Web Interface in the device'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.

5.5.6.2 Routing Context Dynamic Configuration

- User must configure in advance an entry in the "SS7_ROUTING_CONTEXT_TABLE" table for each Routing Key that he is going to configure dynamically. In each entry he should configure in the first INNER INDEX the sni that suits the RK-DPC parameter.
- 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.

Notes:

- The "M3UAROUTINGCLIST" table in no longer supported. Users MUST use the new table and fields, as described above.
- There is no need to configure M3UA in the Interface table. The SN index should be configured in the Routing Context Table.

5.5.7 SS7 MTP3 Shared Point Code (SN 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.

5.5.7.1 General Architecture

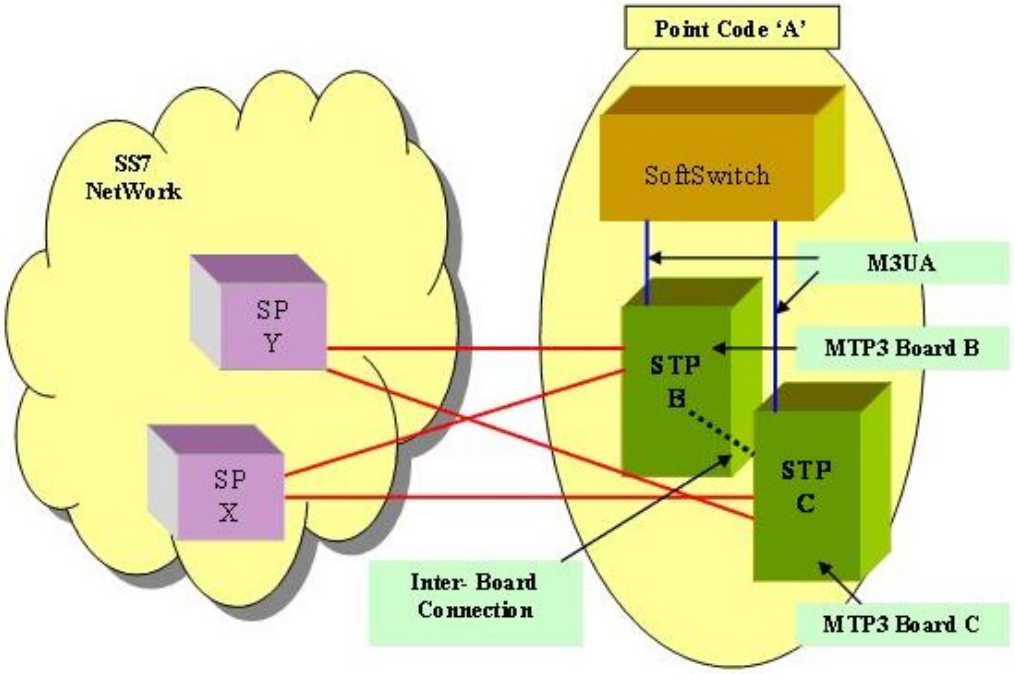
There are two working modes for MTP3:

- 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.
- Shared Point Code (Redundancy) mode - In this mode two devices (i.e. CPUs) may participate in a distributed point-code.

5.5.7.2 SS7 MTP3 Shared Point Code (SN Redundancy) Mode Architecture

Note: The figure below is a private case of Point-Codes-Sharing between two devices.

Figure 17: 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).

5.5.7.3 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.

5.5.7.4 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.

5.5.8 Configuration of SS7 MTP3 Shared Point-Code (SN Redundancy)

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, alias pc, linksets, linksetlinks, linksets timers, routesets, routesetroutes, routing context, interface groups, 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.

Configuration of the redundancy feature can be used only when the device is in MTP3 Redundancy mode.
The following sections describe the redundancy configuration.

5.5.8.1 INI File Global Parameters

The following parameters must be configured in the *ini* file for MTP3 Shared Point Code configuration:

- SS7MTP3RdcyMode
- SS7MTP3RdcyBoardNum
- SS7MTP3RdcyTransferType
- SS7MTP3RdcyKeepAliveInterval
- SS7MTP3RdcyKeepAliveWindow
- SS7MTP3RdcyTblSync Interval

For more details, please refer to 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.

SS7 Parameters

Parameter Name	Description	Default	Range	Notes
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	Can't be changed on-the-fly.
SS7MTP3RdcyMode	Defines the SS7 MTP3-User Adaptation Layer	0	0 or 1	Can't be changed

SS7 Parameters

Parameter Name	Description	Default	Range	Notes
	redundancy mode. Determines the redundancy flavor. • 0 = Disabled • 1 = Enabled			on-the-fly.
SS7MTP3RdcyBoardNum	Defines the device number for the 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	Must be set. Can't be changed on-the-fly.
SS7MTP3RdcyKeepAlive Interval	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	Can't be changed on-the-fly.
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	Can't be changed on-the-fly.
SS7MTP3RdcyTblSyncInterval	In SS7 MTP3-User Adaptation Layer redundancy mode, defines the interval between SS7 tables automatic synchronizations between boards, in minutes. 0 = no automatic synchronization is activated.	0	0 to 30000	Can be changed on-the-fly.
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	2	0 and 2	Can't be changed on-the-fly.

.

5.5.8.1.1 Parameter in Links Table

The *SS7_LINK_RDCY_BOARD* parameter 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 (refer to *S7MTP3RdcyBoardNum* above), then the link is active on that device.

5.5.8.1.2 Parameter in SIGTRAN Interface Group Table

The *RdcyBoardNum* parameter specifies the device number on which the physical SCTP connection should be opened.

5.5.8.1.3 MTP3 Redundancy SNs Table

MTP3 Redundancy SNs Table support issues such as inter-device connectivity and is 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.

5.5.8.1.4 Adding a New Gateway

If one or more gateways are already running in the 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* parameter to the new device number and initiate the new gateway with these *ini* file parameters.

5.6 PSTN Physical Interfaces

The basic digital circuit in the PSTN is a 64-kilobits-per-second channel (DS0). The DS0s are the basic granularity at which switching takes place in a telephone exchange. DS0s are also known as timeslots because they are multiplexed together using time-division multiplexing (TDM). Multiple DS0s are multiplexed together on higher capacity circuits into a DS1 signal, carrying 24 DS0s on a North American or Japanese T1 line, or 32 DS0s (30 for calls plus two for framing and signaling) on an E1 line used in most other countries.

The timeslots are conveyed from the initial multiplexer to the exchange over a set of equipment collectively known as the access network. The access network and inter-exchange transport of the PSTN use synchronous optical transmission (SONET and SDH) technology, although some parts still use the older PDH (E1, T1, T3) technology.

5.6.1 E1

E1 is TDM line in which (usually) 30 voice channels are multiplexed along with their signaling information. Signaling is the control information used to set up or tear down a voice call.

E1 is a 2.048 MHz electrical interface and provides serial synchronous bit stream at 2.048 Mbps.

Basic E1 frame consists of 256 (32 timeslots x 8 bits) bits and is repeated every 125 µs. Timeslot 0 is used for framing and timeslot 16 may be used for CAS signaling or for voice.

There are several framings for E1:

- Basic E1 Double Frame format

- E1 Framing with Signaling Multi-Frame Alignment
- E1 Framing with CRC4 Multi-Frame Format
- E1 Framing with CRC4 Multi-Frame Format Extended (complies with G.706B)
- E1 Framing with CRC4 Multi-Frame Alignment and Signaling (CAS) Multi-Frame Alignment.

5.6.2 T1

T1 is TDM line in which 24 voice channels are multiplexed along with their signaling information. Signaling is the control information used to set up or tear down a voice call.

T1 term is used here interchangeably with DS1.

T1 is a 1.544 MHz electrical interface and provides serial synchronous bit stream at 1.544 Mbps.

Basic T1 frame consists of one framing bit and 192 (24timeslots x 8 bits) payload bits and is repeated every 125 µs. There are several framings for T1:

SuperFrame (or D4) format – 12-frame multiframe

Extended SuperFrame (ESF) format – 24- frame multiframe

SLC-96 (or TR-008) – 72-frame multiframe.

5.6.3 J1

J1 is the first level of the J-carrier – Japanese analog of the US digital transmission T-carrier.

J-1 rate is 1.544 Mbit/s.

Framing method supported for J1 is Extended Superframe with CRC6 (JT G.706).

5.6.4 BRI

The BRI interface implements the ITU-T I.430 standard. The port can supply 48V voltage when it configured as "Network" termination side.

5.6.5 T3

Note: This section is only relevant for PSTN T3/DS3 ports in the product based on TP-6310 hardware.

Digital Signal 3 is a digital signal level 3 T-carrier (referred also as a T3). The main characteristics of this signal are:

- The data rate is 44.736 Mbit/s.
- Can transport 28 DS1 level signals within its payload.
- Can transport 672 DS0 level channels within its payload.
- Two Framing methods are supported:
- M13 Framing method
- C-bit Framing method

For the DS3/T3 configuration parameters please refer to 'DS3 Configuration Table Parameters' on page 271.

5.6.6 Optical OC3 and STM1 Interfaces

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

The device provides *protected* optical fiber transmission interface that can be configured as an *STM1* interface or as *OC3* interface.

5.6.6.1 STM-1 Interface

The product provides short range SDH (Synchronous Digital Hierarchy) STM-1 interface compliant with ITU-T G.707 and ITU-T G.957. The interface operates at 155.52 Mbps over a single-mode fiber span. The product supports VC-4 payload at the STM-1 signal level.

5.6.6.1.1 Synchronous Digital Hierarchy

The Synchronous Digital Hierarchy (SDH), covered by ITU-T recommendations G.707, G.708, and G.709, details the international standards covering synchronous multiplexing and transmission.

SDH suggests some advantages over PDH (E1, T1, T3) transmission:

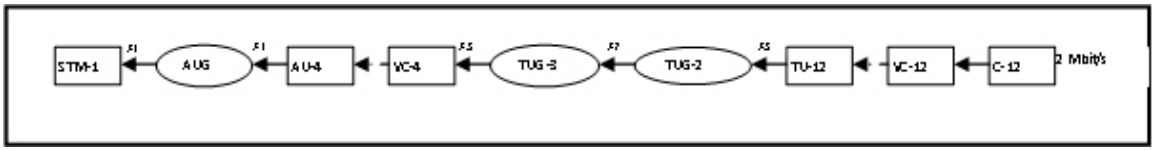
- Simplified multiplexing/demultiplexing techniques.
- Access to lower speed tributaries without the need to multiplex/ demultiplex the entire high speed signal.
- Embedded network management channels which provide enhanced Operations, Administration, and Maintenance (OAM) capabilities.

5.6.6.1.2 SDH Multiplexing Structure

The first level in the SDH hierarchy is at 155.52 Mbit/s and is known as a STM-1 (Synchronous Transport Module 1) signal.

The product supports the following multiplexing structure (asynchronous VC-12 mapping):

Figure 18: SDH Multiplexeing Structure



The STM-1 Interface can be *protected or unprotected*. For more information about protection please refer to 'Automatic Protection Switch' on page 425.

For information about clock management for the product with STM-1 interface please refer to 'Devices with STM1/OC3 Interfaces' on page 416.

5.6.6.1.3 STM-1 Parameters

To configure the STM-1 interface, the following *ini* file parameters are applicable:

- Set the PSTNTransmissionType parameter to "Optical SONET or SDH" ("1").
- Set the SDHFbrGrp_SDHSONETMode parameter to "STM1" ("1")
- Set the SDHFbrGrp_Mapping_Type parameter to "Asynchronous TU12 and E1" ("1")
- Trunks can be E1s only – configure ProtocolType correspondingly.

For more information about parameters, refer to 'PSTN Parameters - SDH/SONET Configuration' on page 222.

For information on SDH-related MIBs, refer to 'TrunkPack-VoP Series Supported MIBs' on page 69.

5.6.6.2 OC-3 Interface

The product provides short range SONET OC-3 interface compliant to GR-253-CORE and ANSI T1.105.

SONET (synchronous optical network) is a standard for optical telecommunications transport.

The interface operates at 155.52 Mbps over a single-mode fiber span.

Two mapping types in OC3 are supported:

- Asynchronous VT1.5 (DS1 ->VT1.5->STS-1->OC3)
- Asynchronous mapping of DS3 in STS1, DS3 channelized to DS1s (DS1->DS3->STS-1->OC3)

5.6.6.2.1 OC-3 Parameters

To configure OC-3 interfaces, the following *ini* file parameters are applicable:

- Set the PSTNTransmissionType parameter to "Optical SONET or SDH" ("1").
- Set the SDHFbrGrp_SDHSONETMode parameter to "OC3" ("2").
- Set the SDHFbrGrp_Mapping_Type parameter to:
  - 'Asynchronous VT1.5 and DS1' (0) for Asynchronous VT.15 application
  - 'Byte-synchronous TU11 and DS1' (2) for Byte-synchronous VT.15 application
  - 'Asynchronous mapping of DS3 in STS1, DS3 channelized to DS1s' (3) for Asynchronous mapping of DS3 in STS1.
  - Trunks can be DS1s only – configured using the parameter ProtocolType

For more information about parameters, refer to 'PSTN Parameters - SDH/SONET Configuration' on page 222.

For information on SDH-related MIBs, refer to 'TrunkPack-VoP Series Supported MIBs' on page 69.

5.6.6.3 SDH/SONET APS Configuration

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 Revertive mode of APS setting.
  - 0 = Non-revertive switching, default
  - 1 = Revertive switching
- SDHFbrGrp_APS_WTR parameter allows setting the Wait-To-Restore time (5 to 12 minutes with a default of 5 minutes). This is effective only if the Revertive mode was selected by the SDHFbrGrp_APS RevertMode parameter.

5.6.6.4 Trunk Numbering Schemes

The product 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).

E1 / T1 Trunks are numbered sequentially in the product while corresponding SDH/SONET instances (timeslots/columns) have 3 referenced numbers built hierarchically. This complex triple numbering is called sometimes KLM-numbering.

Selection of KLM-numbering scheme is important when interconnecting different equipment, e.g., the product based on TP-6310 hardware and an add-drop 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 product based on TP-6310 hardware.

The SDHKlmNumberingScheme *ini* file parameter is used to select the scheme. Refer to 'PSTN SDH/SONET Parameters' on page 222.

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).

5.6.6.4.1 E1 Trunk Enumeration ("SDH" Mappings)

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

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
26	2	2	2	2	5	1	2	2	2
27	2	2	3	2	6	1	3	2	2

STM-1 Numbering Conversion Table

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

STM-1 Numbering Conversion Table

Trunk	ETSI			GR-253			Timeslots		
62	3	7	2	3	6	3	2	7	3
63	3	7	3	3	7	3	3	7	3

5.6.6.4.2 T1 Trunk Enumeration ("Sonet" Mappings)

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

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
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

OC3 Numbering Conversion Table

Trunk	ETSI			GR-253			Timeslots		
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
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

OC3 Numbering Conversion Table

Trunk	ETSI			GR-253			Timeslots		
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

5.7 PSTN Low-Layer Applications

5.7.1 Clock Management

Clocking is one of the most delicate matters in the network element setup.

AudioCodes products are usually designated to work in a point-to-point configuration that simplifies the issue. But having several clock layers and a number of configuration parameters, requires attention for the proper configuration.

5.7.1.1 Devices with E1/T1 External interfaces

Devices like TP-8410, Mediant 1000, TP-1610 and other based on this hardware, have E1 or T1 physical interfaces.

There are two important configuration parameters:

- The blade clock source selected by means of *TDMBusClockSource* parameter
  - **Network** - the blade will use a clock recovered from one of its physical E1/T1 interfaces as a clock source for transmit on all interfaces.

Note: The device terminates the E1/T1 down to 64-kbps timeslots and hence can't tolerate different clock sources for its physical interfaces. Breaking these rules can lead to performance errors and voice disturbances.

  - **Internal** / **BITS -** the blade won't use the clock recovered from its physical interfaces for transmitting, but will rather use the internal oscillator or BITS clock for its transmit.
- **"MASTER ON / OFF" -** the *ClockMater* parameter per E1/T1 interface. Applicable when the device uses the recovered clock (*TDMBusClockSource* parameter set to "Network").

Note: Only those interfaces that are configured as MASTER OFF are *candidates* to supply clock for the device.

  - When the bus type is H.110, SC or MVIP, selecting MASTER_OFF is followed by a warning. In this case AutoClock mode must be disabled, and it is the user's responsibility to change the sync source to another synchronized link.

5.7.1.1.1 PSTN Auto Clock Enable

The TrunkPack-VoP series supports an Automatic and a Manual mode for fallback of T1/E1 clocking. This allows the TrunkPack to use alternate timing references when the reference clock fails. The feature is supported only when the clock is derived from the E1/T1 trunk. The feature is not related to the MASTER/SLAVE clock (for H.110/MVIP and others).

Assume that the clock is derived from a specific E1/T1 trunk and that the trunk fails (or is plugged out). The feature ensures that the blade derives the clock from another E1/T1 trunk.

If no E1/T1 trunk is connected, the blade derives the clock from an internal source (an event is printed - Trunk = -1 for internal clock).

There are two modes:

- **Automatic Mode** - Each time an E1/T1 trunk (that the clock is derived from) fails, the blade automatically selects the highest priority (see below) E1/T1 trunk that is in service.
- **Manual Mode -** The HOST/GUI is notified with an event of the Alarm and the Host can take an action to change the clock source through an API request. The clock can be derived only from an E1/T1 trunk. (In Manual mode, it is not possible to request an internal clock).

Trunk Priority for Auto-Clock Fallback

This feature defines the trunk priority for auto-clock fallback (per trunk parameter), with the *AutoClockTrunkPriority* parameter. Fallback is enabled when parameter TDMBusPSTNAutoClockEnable is set to 1.

Valid values:

- 0 to 99 - Priority (0 = highest = default)
- 100 - The software never falls back to that trunk (usually used to indicate an untrustworthy clock source)

Enable Auto-Clock Reverting

The valid values of blade parameter TDMBusPSTNAutoClockRevertingEnable are:

- 0 = disable (default)
- 1 = enable

The parameter is valid only when parameter TDMBusPSTNAutoClockEnable is set to 1. The parameter enables/disables the PSTN trunk auto-fallback reverting feature. If a trunk with a higher priority than the current LocalReference is being synchronized, the blade LocalReference can be changed to the new trunk.

There are 2 modes:

- **Automatic Mode**: Each time a E1/T1 (that the clock is derived from) fails, the blade automatically selects the highest priority E1/T1 that is in service.
- **Manual Mode:** The HOST/GUI is notified with an event of the Alarm and the Host can take an action to change the clock source through an API request. The clock can be derived only from an E1/T1 (in Manual mode, it is not possible to request an internal clock).

Note: The *PSTNAutoClockEnable* blade parameter is only used when the following parameters are configured:

TDMBUSTYPE	TDMBUSCLOCKSOURCE	TDMBusNetrefOUTPUTMODE
USE_FRAMERS = 2	4 (Network)	Enable
USE_FRAMERS = 2	15 (BITS)	Enable

The same applies to the API of acPSTNSetClockSourceFromTrunkId() which is used only when the above parameters are configured.

5.7.1.1.2 Using BladeParams

➢ **To enable Automatic Mode, set the following parameter :**

```
TDMBUSPSTNAUTOCLOCKENABLE = 1
```

Automatic reverting mode is available using the *TDMBusPSTNAutoClockRevertingEnable* parameter. Note that this parameter is valid only when the *TDMBusPSTNAutoClockEnable* parameter is set to 1.

➢ **To use Manual Mode:**

- Set the Trunk Number from which the clock is derived in the *TDMBusLocalReference ini* file parameter

If you get an acEV_PSTN_ALARM on that trunk, you can derive the PSTN clock from another T1 using the following API function:

```
int acPSTNSetClockSourceFromTrunkId (acTBladeHandle BladeHandle,
int TrunkId)
```

where TrunkId is the new trunk to derive the clock.

5.7.1.2 Device with DS3 External Interfaces

In a device like TP-6310, IPM-6310 or Mediant 3000, that has DS3 physical interfaces, there are two levels of line clock sources:

- DS3 Clock - DS3 is point-to-point connection where one peer should be defined as driving clock (Clock Master) and the second peer as recovering clock (Clock slave).

  Any of three DS3 interfaces can be configured independently.

  To configure the interface clock as "Clock Master" set DS3CONFIG_Clock Source parameter to LOCAL_BOARD ("1")

  To configure the interface clock as "Clock Slave" set DS3CONFIG_Clock Source parameter to EXTERNAL ("0")
- Trunk DS1 Clock - All described in 'Devices with E1/T1 External interfaces' on page 413 is applicable for these products as well.

The blade clock can be supplied from one of the following sources:

- Line clock - The clock is recovered from *one of DS1 trunks* when the *TDMBusLocalReference* parameter is set to "Network". Recovered DS3 Clock can't be used for the blade synchronization.
- Internal Clock - *TDMBusLocalReference* parameter is set to "Internal"
- BITS clock - *TDMBusLocalReference* parameter is set to "BITS"

5.7.1.3 Devices with STM1/OC3 Interfaces

In devices like TP-6310, IPM-6310 or Mediant 3000, that have STM1/OC3 optical interfaces, there are two levels of clock sources:

- STM1 / OC3 Clock. SDH / Sonet systems are "synchronous" in comparison to PDH systems (e.g., based on DS3 interfaces) in the sense that all the network elements are synchronized to a single clock source.
  - STM1 / OC3 Clock can be configured as driving clock (Clock Master) or as recovering clock (Clock slave).
- Trunk E1 / DS1 Clock. 'Devices with E1/T1 External interfaces' on page 413 is NOT applicable for these devices. E1/DS1 can't be the clock source for the blade.

The only "Line" clock used in the STM1 / OC3 – based product is the clock recovered from STM1 / OC3 line rather than E1 / DS1 –recovered clock.

Note: All E1/DS1 trunks should have the same clock source as STM1 / OC3.

The blade clock can be supplied from one of following sources:

- *Line Clock* - Set TDMBusLocalReferenceSource parameter to "Network" ("4"). The product clock will be synchronized to clock recovered from STM1 / OC3 interface. Transmit of all the product interfaces – STM1 / OC3 and E1 / DS1 (multiplexed into STM1 / DS1 but still having their clocks) – will be done with this clock.
- *Internal* - Set TDMBusLocalReferenceSource parameter to "Internal" ("1"). STM1 / OC3 interface clock will be "Clock Master". Transmit of all the product interfaces – STM1 / OC3 and E1 / DS1 – will be done with this clock.
- *BITS* - Set TDMBusLocalReferenceSource parameter to "BITS" ("15"). STM1 / OC3 interface clock will be "Clock Master". Transmit of all the product interfaces – STM1 / OC3 and E1 / DS1 – will be done with this clock.

5.7.2 Alarms

As part of operation, administration, maintenance and provisioning (OAM&P), transmission interfaces on communication equipment needs to be monitored for Defects and Alarms.

- Defects are "raw" failures.
- Defects then are passed through integration/de-integration state machine to become "Alarms".
- Only alarms are reported to Management (EMS, Web Interface) and Control servers.
- If a defect is present, the device waits for its persistence for some "integration time" before reporting the alarm set.
- If a defect has cleared, the device waits for clear persistence for some "de-integration" time before reporting alarm clear.

5.7.2.1 E1 / DS1 Alarms

To indicate problems in the receive or transmit directions for E1 or T1, different alarms can be raised:

- Loss-of-signal (LOS), Loss-of-frame (LOF), Alarm Indication Signal (AIS), Remote Alarm Indication (RAI).
- E1/DS1 alarms are reported to Management entities (EMS, Web Interface) and Control entities (SIP, MEGACO, MGCP Controllers).

For Integration / De-integration time for products based on TP-1610, TP-8410, Mediant 1000 and TP-6310, refer to the tables below.

E1/DS1 Alarms Integration/De-integration time (for TP-1610, TP-8410, Mediant 1000)

Alarm	Control				Management			
	E1		T1		E1		T1	
	Integ. ms	De-integ. ms	Integ. ms	De-integ. ms	Integ. ms	De-integ. ms	Integ. ms	De-integ. ms
LOS	50	50	2000	10000	50	50	2000	10000
LOF	50	50	2000	10000	50	50	2000	10000
RAI	50	50	1000	10000	50	50	1000	10000
RAI & CRC	50	50	1000	10000	50	50	1000	10000
AIS	50	50	2000	10000	50	50	2000	10000

E1/DS1 Alarms Integration/De-integration Time (for TP-6310)

Alarm	Control				Management			
	E1		T1		E1		T1	
	Integ. ms	De-integ. ms	Integ. ms	De-integ. ms	Integ. ms	De-integ. ms	Integ. ms	De-integ. ms
LOS	2500	10000	2500	10000	2500	10000	2500	10000
LOF	2500	10000	2500	10000	2500	10000	2500	10000
RAI	2500	10000	2500	10000	2500	10000	2500	10000
RAI & CRC	2500	10000	2500	10000	2500	10000	2500	10000
AIS	2500	10000	2500	10000	2500	10000	2500	10000

5.7.2.2 DS3 Alarms

To indicate problems in the receive or transmit directions for T3, the following alarms are supported:

- DS3 LOS – DS3 Loss of Signal
- DS3 LOF – DS3 Loss of Frame
- DS3 AIS – DS3 Alarm Indication Signal
- DS3 RAI – DS3 Remote Alarm Indication

Integration time for the alarms is 2.5 sec and de-integration time is 10 seconds.

5.7.2.2.1 Sending Trunk Alarms Only on the DS3 Level

Support is now available for sending trunk alarms on the DS3 level. This is in addition to the already supported alarms on the trunk level. The alarm level is configured by the parameter, DS3AlarmConsolidation. When enabled, only SDH alarms are raised and no alarms are raised for trunks (even if they exist). When the SDH alarm is cleared, trunk alarms are raised (if they exist). This is applicable to Mediant 3000 with TP-6310.

5.7.2.3 STM-1/OC-3 Alarms

To indicate problems in the receive or transmit directions for STM-1/OC-3, the following alarms are supported:

Supported STM-1/OC-3 Alarms

SDH Name	SONET Name	Description	Supported SDH	Supported SONET	Supported by Standard MIB
LOS	LOS	Loss of Signal	Yes	Yes	Yes
Regenerator Section					
RS-LOF	LOF	Loss of Frame	Yes	Yes	Yes
RS-TIM	TIM-S	Trace Identifier mismatch	Yes		No
Multiplex Section					
MS-AIS	AIS-L	Alarm Indication Signal	Yes	Yes	Yes
MS-RDI / MS-RFI (K2 bits 6,7,8)	RDI-L / RFI-L	Remote Failure Indication	Yes	Yes	Yes
Administrative Unit					
AU-LOP	LOP-P	Administration Unit, Lost of Pointer	Yes	Yes	Yes
AU-AIS	AIS-P		Yes	Yes	Yes
High Order Path					
HP-Uneq	Uneq-P	Unequipped	Yes	Yes	No
HP-TIM	TIM-P	Trace Identifier mismatch	Yes	Yes	Yes
HP-PLM	PLM-P	Payload Label	Yes	Yes	Yes

Supported STM-1/OC-3 Alarms

SDH Name	SONET Name	Description	Supported SDH	Supported SONET	Supported by Standard MIB
		Mismatch			
HP-RDI / HP-RFI (G1, bit 5)	RDI-P / RFI-P (G1, bit 5)	Remote Defect Indication	Yes	Yes	Yes
HP-LOM	NA	Loss of Multiframe (H4)	Yes	No	No
Tributary Unit					
TU-LOP	LOP-V	Loss of Pointer	Yes	Yes	Yes
TU-AIS	AIS-V	Alarm Indication Signal	Yes	Yes	Yes
Low Order Path					
LP-Uneq	Uneq-V	Unequipped	Yes	Yes	Yes
LP-TIM	TIM-V	Trace Identifier mismatch	Yes		No
LP-PLM	PLM-V	Payload Label Mismatch	Yes	Yes	Yes
LP-RDI / LP-RFI	RDI-V / RFI-V	Remote Defect Indication	Yes	Yes	Yes

Detection, declaration and clearing of alarms is compliant to ANSI T1.231 and ITU-T G.821, G.826 and G.829. Integration time for the alarms is 2.5 sec and de-integration time is 10 sec.

5.7.2.4 BRI Alarms

BRI Alarms Integration / De-integration Time

Alarm	Control		Management	
	Integration ms	De-integration ms	Integration ms	De-integration ms
LOS	50	50	50	50
LOF	50	50	50	50

5.7.3 Performance Monitoring

As a part of operation, administration, maintenance and provisioning (OAM&P), transmission interfaces on communication equipment needs to be monitored to ensure proper operation of the medium. Performance is monitored by means of collection and report of anomalies, defects and errors.

In AudioCodes, Performance Monitoring is done on all PSTN interfaces: E1, DS1, DS3, STM1 and OC3.

Main standards for PM are:

- ANSI T1.231
- ITU-T G.821, G.826 and G.829

5.7.3.1 E1/DS1 Performance Monitoring

Performance Monitoring functions enable users to monitor E1/T1 trunks and produce quality reports regarding errors and statistics values. They can be accessed using TPNCP or SNMP mode. TPNCP access is through the PCI bus or through a remote IP connection. SNMP (acPMPSTN MIB) mode is described in 'Using SNMP-based Management' on page 67.

Note: The Performance Monitoring applies only to PRI trunks.

The following is a list of Performance Monitoring reports:

- **AlarmIndicationSignal**
  - Blue alarm
- **LossOfSignal**
  - Red alarm
- **RemoteAlarmReceived**
  - Yellow Alarm
- **LossOfFrame**
  - Occurrence of a particular density of framing error events
  - T1 - more than 2 FERs in 3 msec
  - E1 - 3 consecutive frame alignment signals (FASs) have been received with an error.
- **FramingErrorReceived**
  - Incorrect FAS - bits are received
- **CRC_ErrorReceived**
  - CRC error in the remote side
- **EBitErrorDetected**
- **BitError**
- **LostCRC4multiframeSync**
  - Loss of CRC4 multiframe synchronization

- **Line Code Violation**
  - Too many successive zeros
  - Bipolar Violation (BPV) or Excessive Zeros (EXZ)
  - BPV: T1 AMI occurrence of a pulse of the same polarity as the previous pulse
  - B8ZS or HDB3 occurrence of a pulse of the same polarity as the previous pulse without being a part of the zero substitution
  - EXZ: T1 AMI occurrence of more than 15 contiguous zeros
  - B8ZS occurrence of more than 7 contiguous zeros
- **ControlledSlip(CS)**
  - T1 replication or deletion of the payload bits of a frame (does not cause an OOF)
- **Errored seconds**
  - (ESF and E1-CRC) A second with one or more PCV or one or more LFA or one or more CS or a detected AIS defect.
- **Controlled Slip Seconds**
  - A second containing one or more CS
- **SeverelyErroredFramingSeconds**
  - A second with one or more LFA or a detected AIS defect
- **Severely Errored Seconds**
  - A second with 320 or more PCV or one or more LFA or a detected AIS defect
- **Bursty Errored Seconds**
  - A second with less than 320 and more than one PCV, no Severely Errored Frame defects and no detected incoming AIS defect. CS are not included in this parameter.
- **Unavailable Seconds**
  - UAS are calculated by counting the number of seconds that the interface is unavailable. The interface is unavailable if 10 contiguous SESs or the onset of the condition leading to a failure.
  - Failure states includes this alarms: Far End Alarm(Yellow alarm), RAI, AIS, LOF, LOS
  - Once unavailable, the DS1 interface becomes available at the onset of 10 contiguous seconds with no SESs if no failure is present, or if a failure is present and its clearing time is less than or equal to 10 seconds.
  - While the interface is deemed unavailable, the only count that is incremented is UASs
- **PathCodeViolation**
  - ESF and E1-CRC format: CRC or FER
  - D4 and E1 - no CRC format: FER
- **Line Errored Seconds**
  - A second in which one or more LCVs were detected.
- **Degraded Minutes**
  - A minute in which the estimated error rate exceeds 1E-6 but does not exceed 1E-3.
  - Determined by collecting all the available seconds, removing any SES, grouping the result in groups 60 second long (minutes) and counting a minute as degraded if the cumulative errors during the seconds present in the group exceed 1E-6.
- **AssesedSeconds**
  - Counts the seconds in intervals, where Performance Monitoring is enabled.

5.7.3.2 DS3 Performance Monitoring

DS3 Performance Monitoring is defined in the following standards:

- ANSI T1.231 Layer 1 In-Service Transmission Performance Monitoring
- RFC 3896. Definitions of Managed Objects for the DS3/E3 Interface Type. 2004

DS3 Performance Monitoring Parameters

T1.231 Parameters	DS3 MIB Analogy	Notes
Line Coding Violations (CV-L)	dsx3CurrentLCVs	
Line Errored Seconds (ES-L)	dsx3CurrentLESs	
Line Type A Errored seconds (ESA-L)		
Line Type B Errored seconds (ESB-L)		
Line Severely Errored Seconds (SES-L)		
-	dsx3CurrentUASs	Limitation – doesn't count during DS3 AIS
Line Loss of Signal Seconds (LOSS-L)		
Path Coding Violations (CVP-P)	dsx3CurrentPCVs	
DS3 (C-bit application): Path CPbit Coding Violations (CVCP-P)	dsx3CurrentCCVs	
Path Errored Seconds (ESP-P)	dsx3CurrentPESs	
DS3 (C-bit application): Path CPbit Errored Seconds (ESCP-P)	dsx3CurrentCESs	
Path Type A Errored Seconds (ESAP-P)		
DS3 (C-bit application): Path Type A Errored Seconds (ESACP-P)		
Path Type B Errored Seconds (ESBP-P)		
DS3 (C-bit application): Path Type B Errored Seconds (ESBCP-P)		
Path Severely Errored Seconds (SESP-P)	dsx3CurrentPSESs	
DS3 (C-bit application): Path Severely Errored Seconds (SESCP-P)	dsx3CurrentCSESs	
Path SEF or AIS Seconds (SAS-P)	dsx3CurrentSEFSs	Not supported
Path Ais Seconds (AISS-P)		
Path Unavailable Seconds (UASP-P)		
DS3 (C-bit application): Path		

DS3 Performance Monitoring Parameters

T1.231 Parameters	DS3 MIB Analogy	Notes
Unavailable Seconds (UASCP-P)		
Line Coding Violations (CV-L)	dsx3CurrentLCVs	
Line Errored Seconds (ES-L)	dsx3CurrentLESs	
Line Type A Errored seconds (ESA-L)		
Line Type B Errored seconds (ESB-L)		
Line Severely Errored Seconds (SES-L)		
Line Loss of Signal Seconds (LOSS-L)		
Path Coding Violations (CVP-P)	dsx3CurrentPCVs	
DS3 (C-bit application): Path CPbit Coding Violations (CVCP-P)	dsx3CurrentCCVs	
Path Errored Seconds (ESP-P)	dsx3CurrentPESs	
DS3 (C-bit application): Path CPbit Errored Seconds (ESCP-P)	dsx3CurrentCESs	
Path Type A Errored Seconds (ESAP-P)		
DS3 (C-bit application): Path Type A Errored Seconds (ESACP-P)		
Path Type B Errored Seconds (ESBP-P)		
DS3 (C-bit application): Path Type B Errored Seconds (ESBCP-P)		
Path Severely Errored Seconds (SESP-P)	dsx3CurrentPSESs	Y
DS3 (C-bit application): Path Severely Errored Seconds (SESCP-P)	dsx3CurrentCSESs	
Path SEF or AIS Seconds (SAS-P)	dsx3CurrentSEFSs	dsx3CurrentSEFSs not supported
Path Ais Seconds (AISS-P)		
Path Unavailable Seconds (UASP-P)	dsx3CurrentUASs	One MIB counter against 3 in T1.231.3
DS3 (C-bit application): Path Unavailable Seconds (UASCP-P)	dsx3CurrentUASs	One MIB counter against 3 in T1.231.3

5.7.3.3 STM1/OC3 Performance Monitoring

STM1/OC3 Performance Monitoring is defined in the following standards:

1. ANSI T1.231. Layer 1 In-Service Transmission Performance Monitoring

2. ITU-T G829. Error performance events for SDH multiplex and regenerator sections
3. RFC 3592. Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Interface Type. 2003

OC3 / STM-1 Performance Monitoring Parameters

ANSI T1.231 (2004) and ITU-T G.829	SONET MIB Analogy
SONET: Section CV Counter (CV-S)	sonetSectionCurrentCVs
SONET: Section Errored Seconds (ES-S)	sonetSectionCurrentESs
SONET: Section Errored Seconds Type A (ESA-S)	
SONET: Section Errored Seconds Type B (ESB-S)	
SONET: Section Severely Errored Seconds (SES-S)	sonetSectionCurrentSESs
SONET: Section Severely Errored Frame Seconds (SEFS-S)	
SONET: Line Code Violations (CV-L)	sonetLineCurrentCVs
SONET: Line Errored Seconds (ES-L)	sonetLineCurrentESs
SONET: Line Errored Seconds type A (ESA-L)	
SONET: Line Errored Seconds type B (ESB-L)	
SONET: Line Severely Errored Seconds (SES-L)	sonetLineCurrentSESs
SONET: Line Unavailable Seconds (UAS-L)	sonetLineCurrentUASs
SONET: Path Coding Violations (CV-P)	sonetPathCurrentCVs
SONET: Path Type A Errored Seconds (ESA-P)	
SONET: Path Type B Errored Seconds (ESB-P)	
SONET: Path Errored Seconds (ES-P)	sonetPathCurrentESs
SONET: Path Severely Errored seconds (SES-P)	sonetPathCurrentSESs
SONET: Path Unavailable Seconds (UAS-P)	sonetPathCurrentUASs
SDH: Regenerator Section Background Block errors (RS-BBE)	sonetLineCurrentCVs
SDH: Regenerator Section Errored Seconds (RS-ES)	sonetSectionCurrentESs
SDH: Regenerator Section Severely Errored Seconds (RS-SES)	sonetSectionCurrentSESs
SDH: Multiplex Section Errored Seconds (MS-ES)	sonetLineCurrentESs
SDH: Multiplex Section Severely Errored Seconds (MS-SES)	sonetLineCurrentSESs
SDH: Multiplex Section Unavailable Seconds (MS-UAS)	sonetLineCurrentUASs
SDH: VC-4 path Background Block Errors (BBE-P)	sonetPathCurrentCVs
SDH: VC-4 path Errored Seconds (ES-P) Same as SONET ES-P	sonetPathCurrentESs
SDH: VC-4 path Severely Errored Seconds (SES-P)	sonetPathCurrentSESs

5.7.4 Automatic Protection Switch

Automatic Protection Switch (APS) is relevant only for optical OC-3 and STM1 physical interfaces. Please refer to Optical OC-3 and STM1 Interfaces on page , for more information.

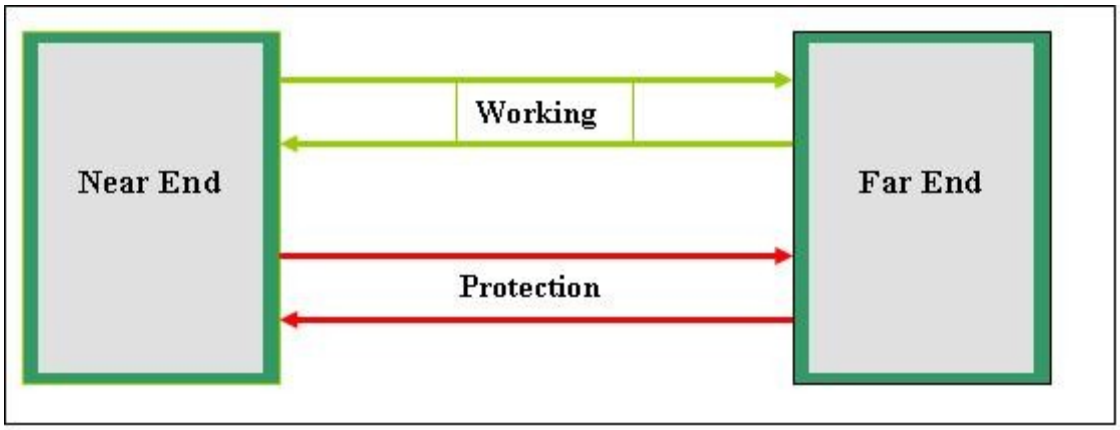
5.7.4.1 APS Common Description

APS has many schemes to provide redundancy in the event of transmission line failure. Three main standards, defining these schemes are:

1. GR-253-CORE. Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria. Issue 4. Sep 2005
2. T1.105.01-2000. Synchronous Optical Network (SONET) - Automatic Protection Switching. Mar 2000.
3. G.841. Types and characteristics of SDH network protection architectures. 10/98.

The simplest scheme is 1+1 line protection (or 1+1 facility protection). This scheme is the simplest way to protect against the most common cause of failures in transmission systems - a break in the fiber line.

Figure 19: 1+1 APS



The above figure shows a near-end and a far-end, connected by a pair of transmit and receive fibers. One bidirectional line is called "Working" and the other is called "Protection".

Two linear APS architectures are defined in the GR-253. These are the 1+1 Architecture and the 1:n Architecture. We are only considering the 1+1 Architecture. It is also called 1+1 linear multiplex section protection (G.841 terminology).

5.7.4.2 1+1 Architecture

5.7.4.2.1 1+1 Unidirectional Mode

For 1+1 unidirectional switching, the working channel is continuously bridged to the protection line, and the channel selection is based only on the local conditions and requests. Therefore, each end operates independently of the other end, and the K1 and K2 bytes are not needed to coordinate switch actions. However, the K1 byte is still used to inform the other end of the local action, and the K2 byte is set to indicate that the K1 byte is being received (i.e., by indicating the same channel number as the received K1), and to inform the other end of the provisioned architecture and mode of operation.

5.7.4.2.2 1+1 Bidirectional Mode

For 1+1 bidirectional systems, the K1 and K2 bytes are exchanged to complete a switch. Head-end to tail-end signaling is accomplished using the APS channel. Therefore each end operation depends on the other end.

The head end maintains a continuous bridge of the working channel to the protection line, and therefore separate bridging actions are not performed for each request.

In addition, each end is allowed to switch immediately, before receiving a bridge confirmation from the other end.

5.7.4.2.3 Revertive and Non-Revertive Switching

A 1+1 system uses, as a default, non-revertive switching where the switch to the protection line is maintained even after the working line has recovered from the failure that caused the switch.

In revertive switching, the traffic is switched back to the working line when the working line has recovered from the failure.

5.7.4.2.4 Switch Initiation Criteria

The following two automatic switch initiation criteria are defined for linear APS:

Signal Fail (SF): A "hard failure" condition detected on the incoming OC-3 / STM-1 signal. Loss of Signal, Loss of Frame and AIS-L/MS-AIS defects and a Line BER exceeding 10-3 on an incoming OC-3 /STM-1 shall be detected as SF conditions on that line.

Signal Degrade (SD): A "soft failure" condition resulting from the Line BER exceeding a pre-selected threshold (user-provisionable over the range of 10-5 to 10-9).

5.7.4.3 APS Modes Supported by the Device (PSTN Interface)

In the device on the PSTN optical interface:

- Both 1+1 unidirectional and 1+1 bidirectional protection modes are supported; unidirectional is default mode
- Both revertive and non-revertive switching are supported; by default non-revertive switching is activated
- Automatic switch initiation criteria implemented in the device are Loss of Signal, Loss of Frame and AIS-L/MS-AIS defects only. BER-based initiation criteria are not supported.

5.7.4.3.1 APS INI-file Parameters

To configure the APS mechanism for PSTN optical interface on the device, a number of *ini* file parameters should be used as follows:

- To activate APS functionality on the PSTN STM-1 / OC-3 interfaces, the SDHFbrGrp_Protected parameter should be set to "Protected". This parameter default value is (1). All APS parameters below are relevant only if "Protected" is selected.

The SDHFbrGrp_APS_DirMode parameter allows selection between Unidirectional and Bidirectional APS modes (0: Unidirectional APS mode, default; 1: Bidirectional APS mode).

The SDHFbrGrp_APS_RevertMode parameter allows selection of the Revertive mode of APS (0: Non-revertive switching, default; 1: Revertive switching).

The SDHFbrGrp_APS_WTR parameter allows setting the Wait-To-Restore time (in minutes) when the Revertive mode was selected by the SDHFbrGrp_APS_RevertMode parameter (5 to 12 minutes with default of 5 minutes).

If a system uses revertive switching, then frequent automatically initiated switches could occur as the result of an intermittent failure. To prevent this, a Wait-to-Restore (WTR) period is defined when revertive switching is used. After the SF condition is cleared, a WTR period is allowed to elapse before the switch-back is done.

Note: The WTR period is not used after an SF condition on the protection line clears.

5.7.4.4 APS Events and Queries

The following is a short description of APS-related events and queries. For their full description please refer to the VoPLib API Reference Manual.

5.7.4.4.1 acEV_SDH_APS_SWITCH_OVER event

The acEV_SDH_APS_SWITCH_OVER event is issued when an automatic protection switchover occurs on one of the Fiber Groups on the blade - PSTN or ATM.

The event reports two fiber links for the fiber group of the specified interface type: the source link (FromFiberLink) and the destination link (ToFiberLink).

5.7.4.5 acSdhQueryApsStatus Query

The acSdhQueryApsStatus function queries information on the Active Fiber Link for a group of optical PSTN and ATM interfaces. The information is returned via the acEV_SDH_APS_STATUS event.

5.7.4.5.1 acEV_SDH_APS_STATUS

The acEV_SDH_APS_STATUS event is issued in response to the acSdhQueryApsStatus function. It provides information on the Active Fiber Link for the fiber group of the specified interface type (PSTN or ATM). The Active Fiber Link can be changed in the blade as a result of an APS in the fiber group.

5.7.5 Trunks Maintenance

The following describes Trunks Maintenance.

5.7.5.1 Management Functions

The following describes Management functions.

5.7.5.1.1 PSTN on-the-fly Changes

This feature provides users the capability to configure the trunk after the blade has been started. The feature provides users with the option to reconfigure the physical layer and/or choose a different protocol type from the same line type which the blade was already configured with. (For E1/T1 trunks, there is no option to move from E1 line type to T1 line type and vice versa).

One of the configuration options is disabling the physical layer by choosing acPROTOCOL_TYPE_NONE.

➢ **To perform an on-the-fly trunk re-configuration:**

1. Open the Trunk Settings page from the PSTN Settings folder, and choose the relevant trunk.
2. Stop Trunk Traffic by using the **STOP TRUNK** button; this action drops all active calls on the trunk.

   EMS users can use "lock" which is same as STOP command in the Web Interface.

Notes:

- If there is active SS7 link on a trunk it can`t be stopped. In order to stop such trunk user need to de-activate the link and then set it to offline and delete it.
- User can`t stop trunk if this trunk is the current clock local reference. In order to stop such trunk user need to change the local clock reference.

3. Change the relevant trunk's parameters.
4. Set these changes by pressing the APPLY TRUNK button. EMS users can use "unlock" which is same as APPLY command at the WEB GUI interface.
5. If all trunk's parameters are valid – the trunk will configured with the requested configuration, and the parameters will lock from changes. STOP TRUNK button will appear.

Notes:

- The user should configure at least one trunk (either E1 or T1 in board initialization) to enable the no-line trunk configuration.
- The user should close all active calls including voice channels on the RTP side before apply trunk's new configuration.

5.7.5.1.2 Deleting trunk

➢ **To perform an on-the-fly trunk deletion, do the following:**

1. Open the Trunk Settings page from the PSTN Settings folder, and choose the relevant trunk.
2. Stop Trunk Traffic by using **STOP TRUNK** button; this action drops all active calls on the trunk.

   EMS users can use "lock" which is the same as the **STOP** command in the Web interface.

Notes:

- If there is an active SS7 link on a trunk, it can't be stopped. In order to stop such trunk user need to de-activate the link and then set it to offline and delete it.
- User can`t stop trunk if this trunk is the current clock local reference. In order to stop such trunk user need to change the local clock reference.

3. Change the trunk protocol type to acPROTOCOL_TYPE_NONE.
4. Set these changes by pressing the APPLY TRUNK button. EMS users can use "unlock" which is same as APPLY command in the Web interface.

   As a result, the "Trunk Configuration State" field will be changed to "Not Configured". If it is a T1/E1/BRI trunk at the Tx interface of the trunk, there will be no signal. The LED color at the Trunk Settings page will change to gray.

Note: The user should close all active calls which is including the voice channels on the RTP side before deleting trunk.

5.7.5.1.3 Changing an NFAS-Related Trunk Configuration On-the-Fly

The procedures for creating and deleting an NFAS group on-the-fly must be performed in the correct order (shown below).

➢ **To Create an NFAS Group:**

1. If there's a backup ('secondary') trunk for this group, it must be configured first.
2. Configure the primary trunk before configuring any NFAS ('slave') trunks.
3. Configure NFAS ('slave') trunks.

➢ **To Stop / Delete an NFAS Group:**

1. Stop / delete all NFAS ('slave') trunks.
2. Stop / delete the backup trunk if a backup trunk exists.
3. Stop / delete the primary trunk.

Notes:

- All trunks in the group must be configured with the same values for trunk parameters TerminationSide, ProtocolType, FramingMethod and LineCode.
- After stopping or deleting the backup trunk, delete the group and reconfigure it.

5.8 Tracing PSTN Protocol Messages

CAS, ISDN, IUA and SS7 protocols are traceable. The protocol's messages can be monitored using the Debug recording tool.

Each of these protocols support several trace levels that filter unnecessary messages.

For detailed information on collecting ISDN trace messages, refer to 'Tracing ISDN Protocols' on page 431.

For detailed information on collecting SS7 trace messages, refer to 'Tracing SS7 Protocols' on page 431.

For detailed information on collecting CAS trace messages, refer to 'Tracing CAS Protocols' on page 432.

5.8.1 Tracing ISDN Protocols

Tracing of PSTN telephony protocol layer messages is supported. A different tracing level can be set per trunk using the TrunkConfig.TraceLevel blade configuration parameter.

The possible trace levels are:

Tracing ISDN Protocols – Trace Levels

Trace Level	Description
NO TRACE	No trace packets are sent from the blade to users.
FULL TRACE	Messages from all layers and protocol entities are reported.
LAYER 3 ISDN TRACE	Only messages from Layer 3 (Q.931) and management entities are reported (these messages are duplicated at both the sending and receiving sides for debugging reasons).
acONLY_ISDN_Q931_MSGS_TRACE	Only Q.931 messages are seen (management entities reports are excluded). Recommended for field tracing.
acLAYER3_ISDN_TRACE_NO_DUPLICATION	Same as acLAYER3_ISDN_TRACE without message duplication.

5.8.2 Tracing SS7 Protocol Messages

Tracing SS7 messages is supported in the same mechanism as for PSTN. The trace result is not a 'pure' SS7 trace, but an internal trace between blade tasks for debugging purpose. A different tracing level can be set per **sli** using the blade configuration parameter SS7Settings.SS7Mtp2LinkConfig[sli].TraceLevel.

The possible trace levels are:

Tracing SS7 Protocol Messages - Trace Levels

Trace Level	Description
NO_TRACE	No trace packets are sent from the blade to users.

Tracing SS7 Protocol Messages - Trace Levels

FULL_TRACE	Messages from all layers and protocol entities are reported (ONLY for laboratory purposes; not recommended in the field).
acSS7_MTP2_SL_L3_no_MSU	Only messages between MTP2 and the above layer (excluding MSUs)
acSS7_MTP2	Messages from all layers sending to or from MTP2.

5.8.3 Tracing CAS Protocols

Tracing CAS is supported. A different tracing level can be set per trunk using the blade configuration parameter TrunkConfig.TraceLevel.

The possible trace levels are:

Tracing CAS Protocols – Trace Levels

Trace Level	Description
NO TRACE	No trace packets are sent from the blade to users.
FULL TRACE or LAYER 3 ISDN TRACE	Trace messages are sent from the blade to users.

The textual trace file is composed of text lines. Each line depicts an event (refer to Note 2 below) that occurs in the protocol for a specific B-channel.

The trace line includes the following fields:

Tracing CAS Protocols – Trace Line Fields

Field	Description
Time	The time of the event in msec.
From	The source of the event that can be From_DSP (CAS ABCD change), From_User (acEV_PACE_CALL), or From_Table (SET_TIMER table function).
CurrentState	The current state of the protocol.
Event	The event that occurred, for example, acEV_PLACE_CALL From_User or acEV_TIMER_EXPIRED1 From_Table. The reserved word 'FUNCTION' in the event's field is used for lines describing the table's function execution.
NextState	The next state in the table only for events whose source is table (From_table). When the event source is From_DSP or From_User, the value is as NO_STATE.
Function	The function that is used by the table. When the event comes From_DSP or From_User, the value is as NONE.
Parameter, #1, #2, #3	The table's function parameters as in the table.
TrunkNum	The trunk number the event is associated with.
BchannelNum	The B-Channel number the event is associated with.

Tracing CAS Protocols – Trace Line Fields

Field	Description
ConnId	The connection ID the event is associated with. The ConnId is related to the CallHandle.

Notes:

- In the text file, there are two events which seem to be duplicated but which are from different sources, which means that it came from one source and was sent to another. For example, EV_CAS_0_1 From_DSP and then the same event From_Table, means that the DSP sends this event and the Table receives it later.
- An event in this section means any change of any kind that occurs in the protocol's state machine, i.e., protocol table actions.

In order to read the CAS trace you should use the AudioCodes Wireshark plug-in which convert the data to text as part of the Wireshark. The Wireshark plug-in can be found in .\Utilities\WiresharkPlugins.

5.8.4 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 (DR)' on page 56). 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.

5.8.5 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:**

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.

5.9 Call Flows

The following describes Call Flows.

5.9.1 PSTN Protocol Implementation Example

Following is an implementation example (users do not need to strictly follow this example). The example assumes a configuration of ISDN PRI telephony protocol or T1 Robbed bit protocols. The example demonstrates the unified API for both protocol families. The differences between those families, regarding the host application, are emphasized.

5.9.1.1 Outgoing Calls

- Invoke acPSTNPlaceCall(BladeHandle, TrunkID, B-channel, …) with the relevant specific parameters for the call (destination phone number, etc.). acPSTNPlaceCall() initiates the call and returns a unique CallHandle for further referencing this call.
- Dependent on the actions taken by the PSTN network as a result of the call initiation, the following two events can be issued by the blade. The event acEV_PSTN_REMOTE_ALERTING is issued if the PSTN network sends an alert message (in-band Ringback tone for CAS protocols), and an acEV_PSTN_PROCEEDING_INDICATION event is issued if the PSTN network sends a call-proceeding message.
- For CAS protocols, in-band Call Progress Tones are reported via the events acEV_TONE_DETECTED and acEV_END_TONE_DETECTED. The type and index describing the specific tone are according to the user's Call Progress Tones *ini* file.
- If the call is set up successfully, an acEV_PSTN_CALL_CONNECTED event follows when the blade receives the connect message (ISDN), or one of the connect events (CAS) from the PSTN network.
- It is then assumed that the call is active and a voice path should be cut through; i.e., Users can open the DSP channel with the call's TrunkId and B-channel using acOpenChannel() if they have not done so before.
- For ISDN protocols, there is no timing restriction imposed by the TrunkPack-VoP series software between the call set-up function and events, and Users applying acOpenChannel(). (There could be restrictions imposed by the telephony protocol).
- Note that to re-open or to newly open for CAS protocols, impose a couple of tenths of msec of blanking the DSP channel. Open it when not expecting a CAS change in this specific time frame. acOpenChannel() can be applied after EV_PSTN_CALL_CONNECTED when the call is already set up.

5.9.1.2 Incoming Calls

- Incoming call processing starts when the user has retrieved the acEV_PSTN_INCOMING_CALL_DETECTED event (for ISDN) or acEV_CAS_SEIZURE_DETECTED (for CAS). Store the unique CallHandle in the EventInfo structure for further referencing the call. The {TrunkId,Bchannel} trunk number/ID and B-channel information parameters in the EventInfo identify the physical channel of the call.
- ISDN: Optionally, invoke acISDNSendCallProceeding(CallHandle, Bchannel, ...) and/or acISDNSendAlert(ChannelHandle, Bchannel, ...) for sending call-proceeding message and/or alert message.
- CAS: Users can invoke acPlayCallProgressTone() when appropriate to play Ringback or dial tone.

- Invoke acPSTNAnswerCall(CallHandle, BChannel) when accepting the offered incoming call (only after: acEV_PSTN_INCOMING_CALL_DETECTED).
- Invoke acPSTNDisconnect(CallHandle …) when rejecting the offered incoming call.
- On receipt of connection confirmation by acEV_PSTN_CALL_CONNECTED event for the call, it is assumed that the call is active and a voice path should be cut through. That is, Users can open the DSP channel with the call's TrunkId and B-channel using acOpenChannel() if they had not done it before.
- For ISDN protocols, there is no timing restriction imposed by the TrunkPack-VoP Series software between the call set-up function and events, and Users applying acOpenChannel(). (There could be restrictions imposed by the telephony protocol).
- For CAS protocols it is strongly recommended that acOpenChannel() is not applied while the call is setup; i.e., between acEV_CAS_SEIZURE_DETECTED and acEV_PSTN_CALL_CONNECTED.

5.9.1.3 Release Procedure

For an active near-end call clearing:

- Invoke acPSTNDisconnectCall(CallHandle, …) with the call's CallHandle for clearing the call.
- On receipt of clear confirmation by acEV_PSTN_CALL_RELEASED event for the call, release all call resources including the CallHandle. (Users could then invoke acCloseChannel() for DSP resources release).

For an active far-end (remote) call clearing:

- On receipt of clear indication by acEV_PSTN_CALL_DISCONNECTED event, respond with acPSTNReleaseCall(CallHandle, …) to clear the call.
- On receipt of clear confirmation by acEV_PSTN_CALL_RELEASED event, release all call resources including the CallHandle. (Users can then invoke acCloseChannel() for DSP resources release).

5.9.2 ISDN Call Setup and Tear-down Diagrams

The following describes ISDN Call Setup and Tear-down Diagrams.

5.9.2.1 Outgoing Calls (En-Bloc Sending Mode)

The Host initiates a call with command acPSTNPlaceCall(), causing the SETUP message to be sent to the line. When the blade detects a CALL_PROCEED, ALERT or CONNECT message it generates acEV_PSTN_PROCEEDING_IN, acEV_PSTN_REMOTE_ALERTING or acEV_PSTN_CALL_CONNECTED respectively.

The following figure and table illustrate the above scenario.

Figure 20: Outgoing calls Implementation Example

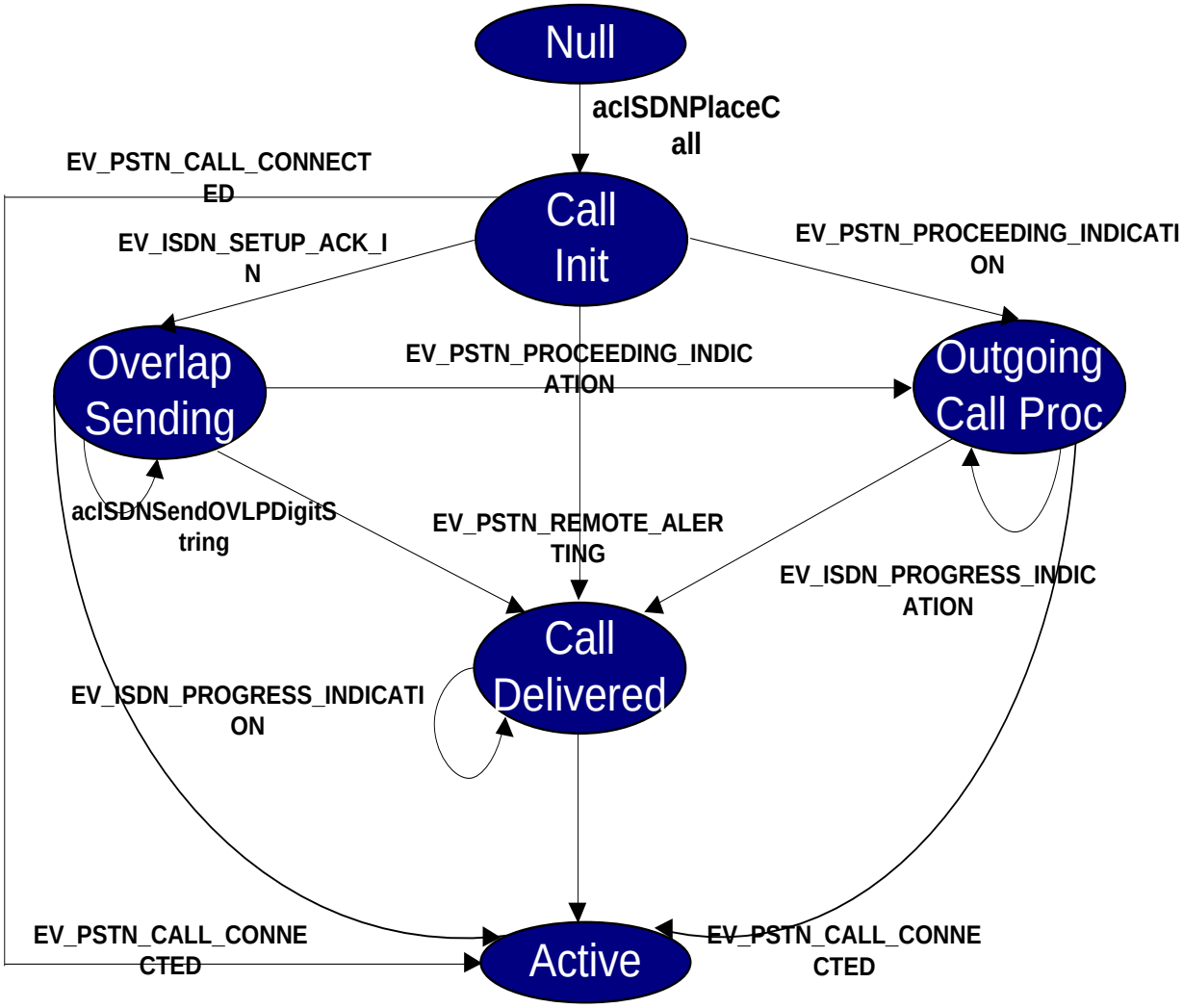
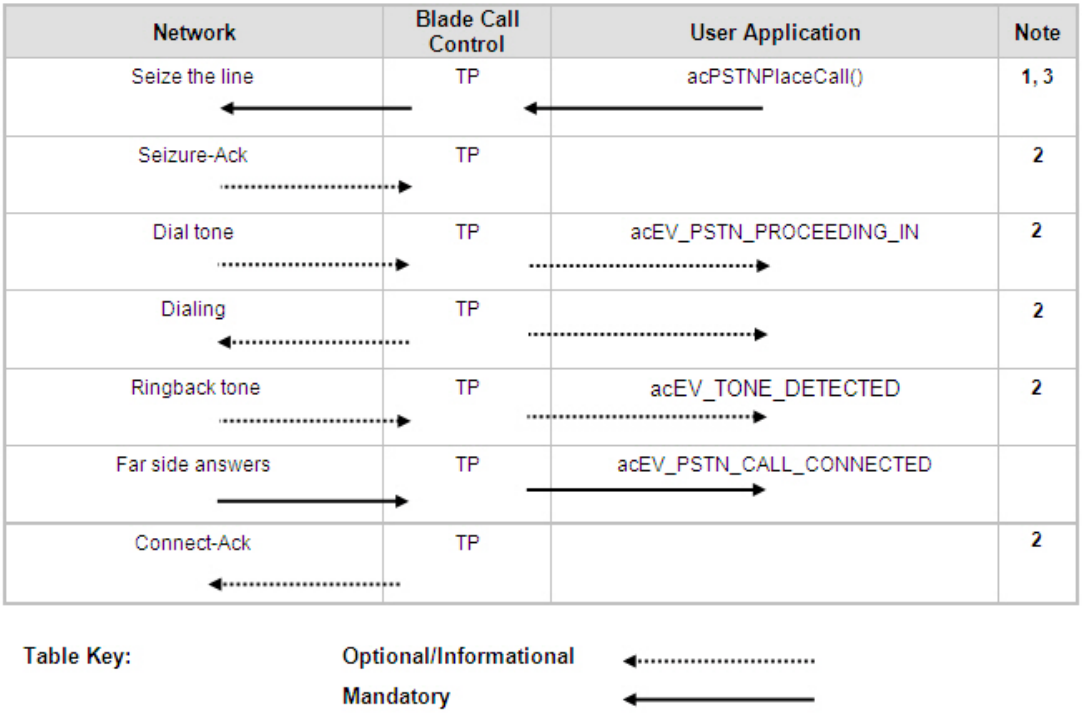


Figure 21: Outgoing Calls - Block Sending Mode



Notes:

1. acPSTNPlaceCall primitive must contain the complete Called Number and sub-address.
2. Optional; contains the provided B-channel ID if the B-channel negotiation procedure is used.
3. CONNECT ACK is generated automatically (optionally).

5.9.2.2 Incoming Calls

In this case, the Host sends command acISDNSendCallProceeding(), forcing the blade to issue the CALL PROCEED message to the trunk.

The Host can issue command acISDNSendAlert() followed by command acPSTNAnswerCall(), causing the ALERT and CONNECT messages to be sent to the line.

The CONNECT is answered with the CONNECT_ACK message which is translated to the event acEV_PSTN_CALL_CONNECTED.

The following figure and table illustrate the above scenario.

Figure 22: Incoming Calls Implementation Example

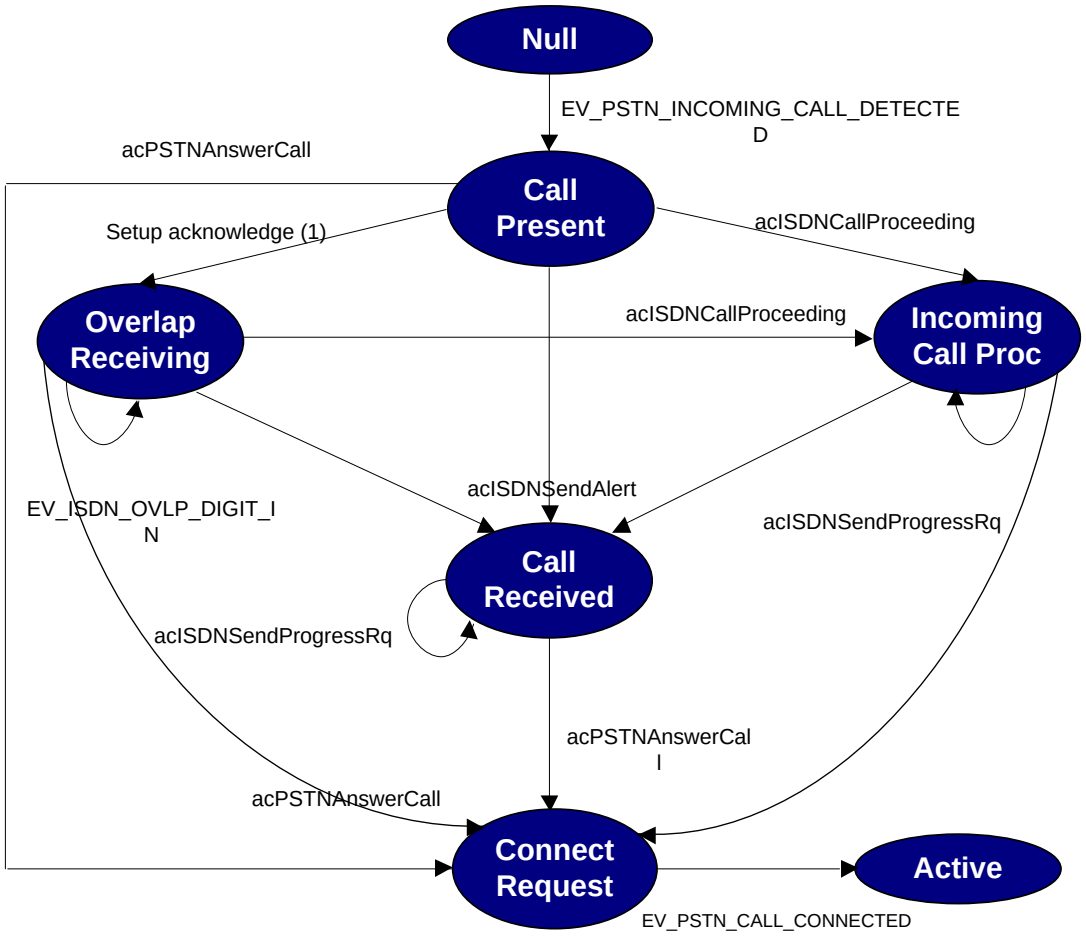
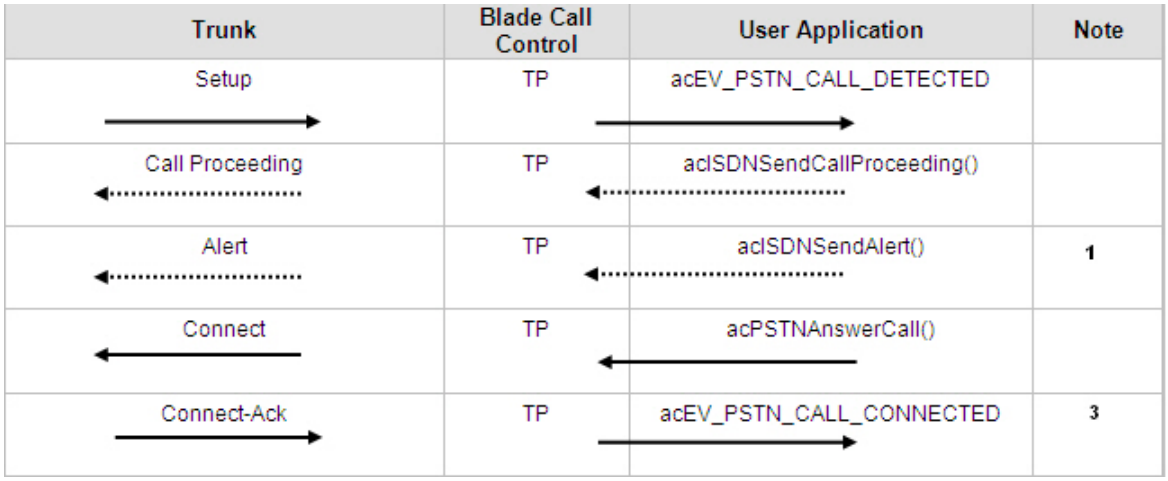


Figure 23: Incoming Calls



Notes:

- The acISDNPlaceCall primitive must contain the complete Called Number.
- Behavior bits may change the scenario.
- CONNECT ACK is generated automatically (optionally).

Note that in overlap mode, when the SETUP message is detected the blade automatically sends SETUP_ACK with the event acEV_PSTN_CALL_DETECTED. The other digits are detected in the INFO message and delivered to the Host in acEV_ISDN_OVLP_DIGIT_IN.

5.9.2.3 Call Clearing

5.9.2.3.1 Outgoing Clearing

The host invokes command acPSTNDisconnectCall() when it needs to terminate the call. The blade sends the DISCONNECT message.

When the RELEASE message is detected, acEV_PSTN_CALL_RELEASED is sent response to the Host and a RELEASE COMPLETE is sent to the line.

The figure and table below illustrate the above scenario.

Figure 24: Clear Calls Implementation Example

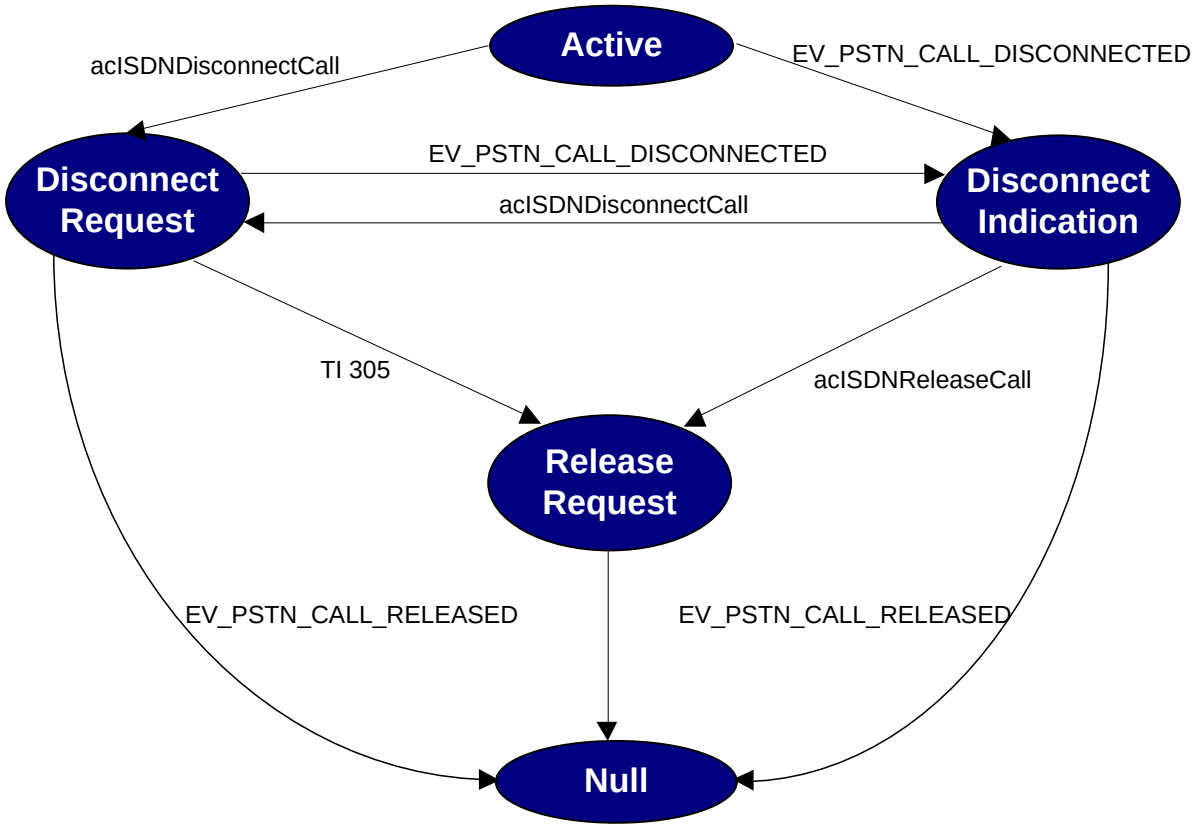


Figure 25: Outgoing Clearing

ISDN Network Layer 3 (Q.931)	Blade Call Control	User Application	Note
Disconnect ←	TP	← acPSTNDisconnectCall()	
Release →	TP	→ acEV_PSTN_CALL_RELEASED	
Release Completed ←	TP		1

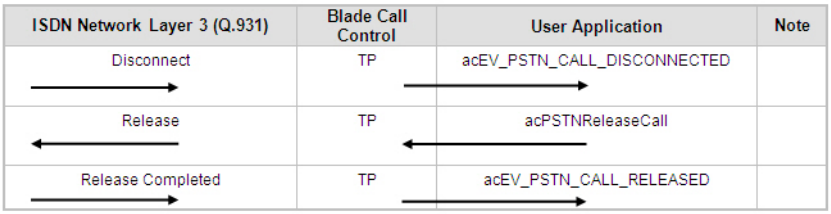
Note : Release complete is sent automatically by the blade.

5.9.2.3.2 Incoming Clearing

If the PSTN Network side terminates the call, it sends a DISCONNECT message which initiates event acEV_PSTN_CALL_DISCONNECTED. The Host replies with acPSTNReleaseCall() in order to send RELEASE to the line. When the RELEASE COMPLETE is detected, acEV_PSTN_CALL_RELEASED is sent to the host.

The table below summarizes the above scenario.

Figure 26: Incoming Clearing

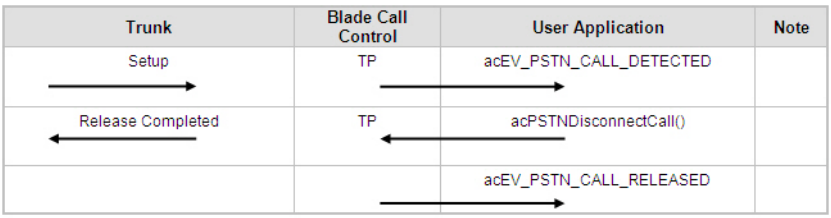


5.9.2.3.3 Rejecting Incoming Call

If the Host decides to reject the call, it does so with command acPSTNDisconnectCall(). This issues the RELEASE COMPLETE message and event acEV_PSTN_CALL_RELEASED.

The following table summarizes the above scenario.

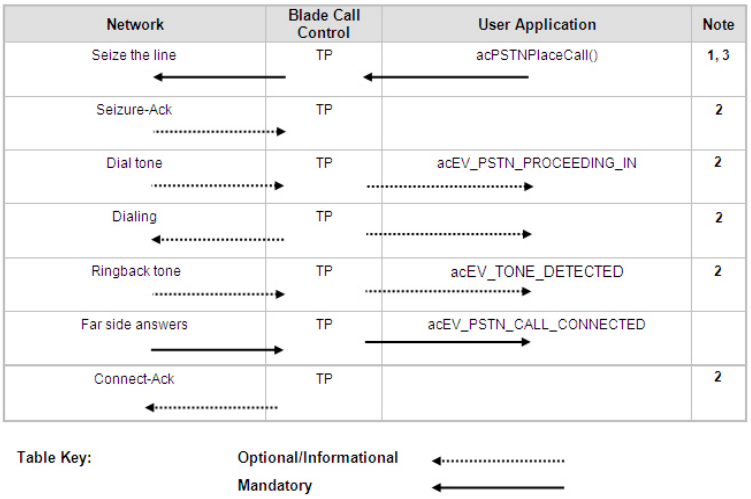
Figure 27: Rejecting Incoming Call



5.9.3 CAS Call Setup and Call Tear-down Diagrams

5.9.3.1 Outgoing Calls (Block Sending Mode)

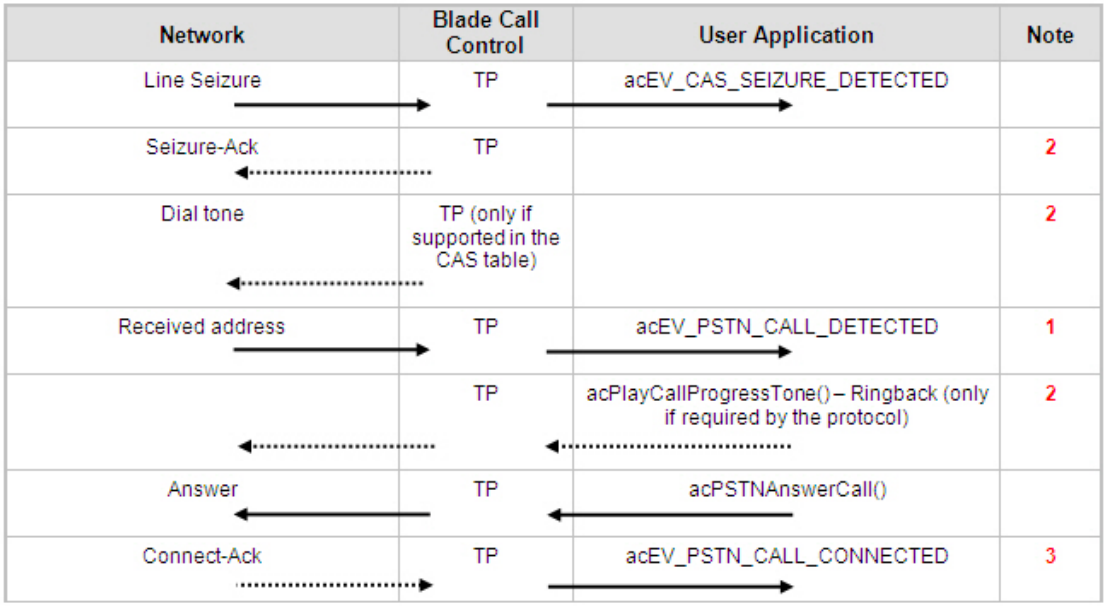
Figure 28: Outgoing Calls - Block Sending Mode



Notes:

1. acISDNPlaceCall primitive must contain the complete Called Number and Source (Calling) Number (overlap digits not supported).
2. Handled by the CAS table (the variant) automatically as configured.
3. MFC-R2 protocols (variants) must include the information for the category in the *TransferCap* and *SourceNumberType* parameters. (Refer to the acPSTNPlaceCall command).

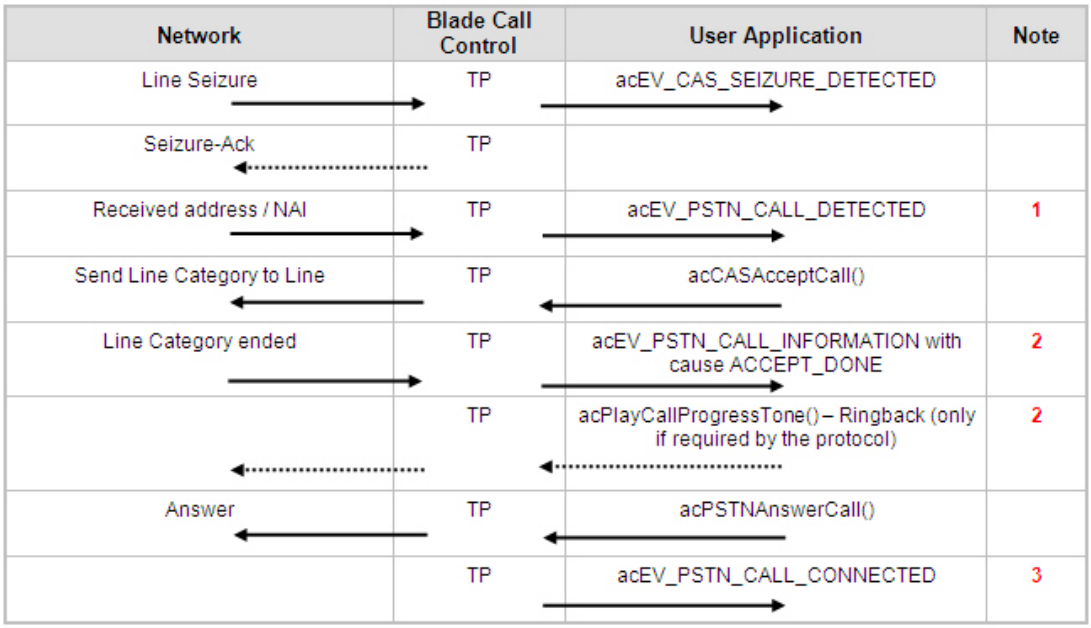
Figure 29: Incoming Calls T1 CAS



Notes:

1. acEV_PSTN_CALL_DETECTED contains the complete Called Number and Source (Calling) Number (overlap digits not supported).
2. Automatically handled by the CAS table (the variant) as configured. Users can handle the Ringback tone (in case the table is not) while opening the channel for it only after acEV_PSTN_CALL_INFORMATION.
3. The connect Ack is not supported in most variants so the table automatically provides the event acEV_PSTN_CALL_CONNECTED.

Figure 30: Incoming Calls E1 MFC-R2



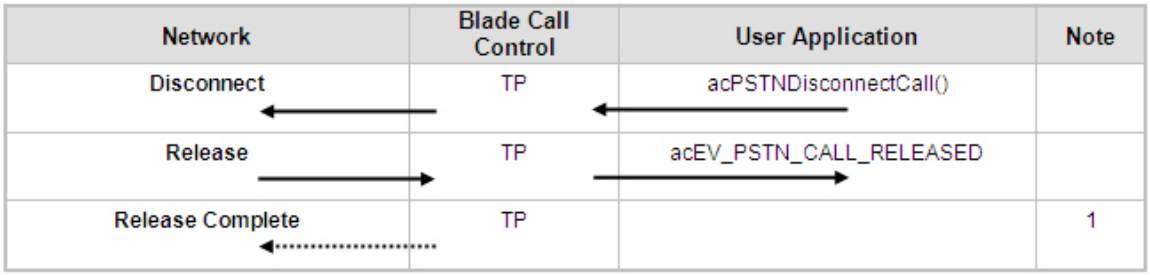
Notes:

1. acEV_PSTN_CALL_DETECTED contains the complete Called Number and Source (Calling) Number (overlap digits not supported).
2. Automatically handled by the CAS table (the variant) as configured. Users can handle the Ringback tone (in case the table is not) while opening the channel for it only after acEV_PSTN_CALL_INFORMATION.
3. The connect Ack is not supported in most variants so the table automatically provides the event acEV_PSTN_CALL_CONNECTED.

5.9.3.2 Call Clearing

5.9.3.2.1 Near End Clearing

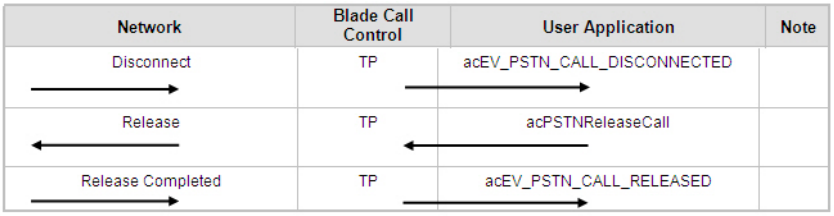
Figure 31: Near End Clearing



Note: Release Complete is automatically sent by the blade.

5.9.3.2.2 Far End Clearing

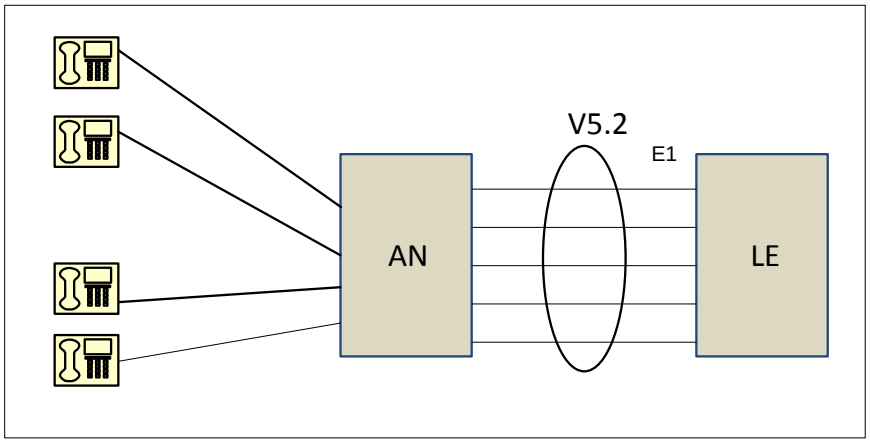
Figure 32: Far End Clearing



5.10 Overview of V5.2

A V5.2 Interface connects an "Access Network" (AN) to a "Local Exchange" (LE).

Figure 33: V5.2 Entities



A V5.2 Interface comprises up to 16 physical 2048 kbps links (E1 links) supporting both bearer channels and logical "communications channels" which convey signaling and control information over the interface. In general the communications channels occupy timeslot 16 on one or more of the 2048 kbps links. When the V5.2 Interface comprises two or more E1 links, two of these links are designated respectively the "primary" and the "secondary" and the communications channels are provisioned on timeslot 16 of these two links. A protection protocol is used to switch from the Primary to the Secondary link in event of failure of the primary link (and vice versa).

The communications channels supported over the V5.2 interface transport the following protocols between AN and LE:

PSTN signaling which conveys supervisory line states to and from end-user analog POTS ports.

ISDN D-channel signaling to and from end-user ISDN ports.

Control protocol which is used to block, unblock, activate and de-activate both PSTN and ISDN user ports and to provide a mechanism to supervise the V5.2 Interface (restart , re-provision and other common control commands).

Bearer Channel Connection (BCC) protocol which is used for dynamic assignment of timeslots to both PSTN and ISDN user ports.

Protection protocol which is used to support the switchover of communications channels between primary and secondary links in the event of communications channel failure.

Link control protocol which is used to identify and control the state of the physical E1 links that makes up the V5.2 interface.

5.10.1 Protocol Architecture

The V5.2 implements three layers (with reference to OSI seven layer protocol stack) as follows:

The Physical Layer converts between the signal of the physical medium and the data stream of the Data Link Layer. For this layer the ITU-T Recommendations I.430, I.431 applies.

The Data Link Layer presents the data to the higher layers in an organized, error-free manner. This layer is sub-divided into two sub-layers as follows:

LAPV5-EF sub-layer uses an Envelope Function Address (EFAddr) to identify the destination process of the encapsulated message. EFAddr in range 0-8175 are reserved for ISDN user port address. The ISDN D-Channel protocols are not terminated by the AN.

They are forwarded from this sub-layer to the ISDN end user terminals .EFAddr values in the range 8176-8180 are assigned to the PSTN protocol, the Control protocol, the BCC protocol, the Protection protocol and to the Link Control protocol. Messages conveyed by these protocols are terminated in the AN and in the LE. Each of those protocols runs over its own data link layer using the LAPV5-DL sub-layer.

LAPV5-DL – Link Access Protocol V5.2 Data Link (LAPV5-DL) is a subset of the Q.921 and I.441 which is called the Line Access Protocol for the D-channel (LAPD).

The Network Layer provides the transparency between the Data Link Layer below and the Transport layer above. It is supports the Layer 3 (L3) processing for the PSTN protocol, the Control protocol, the BCC protocol , the Protection protocol and the Link Control protocol . For this layer the ITU-T Recommendations G.964 and G.965 and ETSI EN 347-1 and EN 324 apply

5.10.2 PSTN Protocol

The PSTN protocol conveys the supervisory state of all analog POTS end-user lines between the AN and the LE. The port with which any given PSTN protocol message is associated is identified by means of L3 Address. The L3 Address is effectively the User Port number of the POTS line.

The V5.2 PSTN protocol is a "stimulus" protocol. In that it conveys information about analog line state changes in the upstream direction, from AN to LE, and requests to change state in the direction from LE to AN.

The PSTN protocol uses ESTABLISH, SIGNAL, DISCONNECT and other primitives for handling the call's signaling.

5.10.3 Control Protocol

This protocol is divided into two groups:

PORT CONTROL group which is used to take individual user ports in and out of service, mainly for maintenance and testing purposes. The PORT CONTROL uses the PORT CONTROL and the PORT CONTROL ACK messages to block, unblock, activate and de-activate both PSTN and ISDN user ports.

COMMON CONTROL group which is used to maintain and to supervise the interface state. It uses the COMMON CONTROL and the COMMON CONTROL ACK messages to start, restart and re-provision the interface.

5.10.4 BCC Protocol

The BCC protocol is used to control the dynamic assignment of timeslots within the E1 links that make up a V5.2 Interface to end-user bearer channels that may be analog POTS lines or ISDN B-channels.

The V5.2-BCC protocol V5.2-BCC protocol provides the means for the LE to request the AN to establish and release connections between specified AN user ports and specified V5.2-interface time slots.

It is based on two kinds of messages from LE to AN: ALLOCATE and DE_ALLOCATE, and two messages from AN to LE: COMPLETE or REJECT.

5.10.5 Protection Protocol

The Protection protocol is used to control the switchover of communications channels from a failed E1 link to a backup E1 link.

Two of the links in the V5.2 interface are designated respectively primary and secondary. The protection protocol operates over timeslot 16 of both the primary and the secondary links. Other V5.2 protocols also operate over timeslot 16 of the primary link, with failure to

the secondary when needed (or vice versa). Timeslots 16 of the primary and the secondary link compromise Protection Group 1.

In some circumstances, there maybe not sufficient capacity on timeslot 16 of the primary link to carry all of the message traffic needed to support the V5.2 interface. In this case, additional communications channels maybe provisioned on further E1 links, also using timeslot 16. Timeslots 15 and 31 may also be used if necessary to provide additional capacity for message traffic. A similar number of backup communication channels may be provisioned. This group of communication channels compromises Protection Group 2.

5.10.6 Link Control Protocol

The Link Control protocol is used to manage the E1 Physical links that make up a V5.2 interface. Two kinds of control operations are supported: taking links in and out of service, and checking the link identity.

5.10.7 Standards Conformance

The V5.2 AudioCodes protocol implementation conforms to the following ETSI standards:

ETS 300 324-1 "Signaling Protocols and Switching (SPS); V interfaces at the digital Local Exchange (LE); V5.1 interface for the support of Access Network (AN); Part 1: V5.1 interface specification"

ETS 300 347-1 "Signaling Protocols and Switching (SPS); V interfaces at the digital Local Exchange (LE); V5.2 interface for the support of Access Network (AN); Part 1: V5.2 interface specification"

EN 300 324-1 "V interfaces at the digital Local Exchange (LE); V5.1 interface for the support of Access Network (AN); Part 1: V5.1 interface specification"

EN 300 347-1 "V interfaces at the digital Local Exchange (LE); V5.2 interface for the support of Access Network (AN); Part 1: V5.2 interface specification"

6 V5.2 Access Gateway

6.1.1 Overview of the V5.2 Access Gateway

This chapter provides a description of the AudioCodes V5.2 Access Gateway solution. An overview of the product is presented in this document. A step by step explanation of the configuration process and maintenance can be found in the *Mediant 3000 and TP-8410 MGCP-MEGACO User's Manual*.

6.1.1.1 General

The AudioCodes V5 Access Gateway application aggregates legacy circuit-switched voice from the subscriber side, converts V5.2 protocol messages to H.248 IP protocol and then hands them off to a softswitch and vice versa. The softswitch replaces the traditional Class 5 switch.

6.1.1.2 About the AudioCodes Product

The AudioCodes V5.2 Access Gateway is a carrier grade product developed on the Mediant 3000 Media Application chassis. This platform can host two TP-8410 cPCI blades.

The TP-8410 blade supports up to 84 DS1 (63 E1 / 84 T1) PSTN spans with a capacity of up to 2,016 DS0 channels. For Mediant 3000, the blade provides redundancy protection functionality when two blades are installed, and the standby blade takes over from the active blade, should it fail.

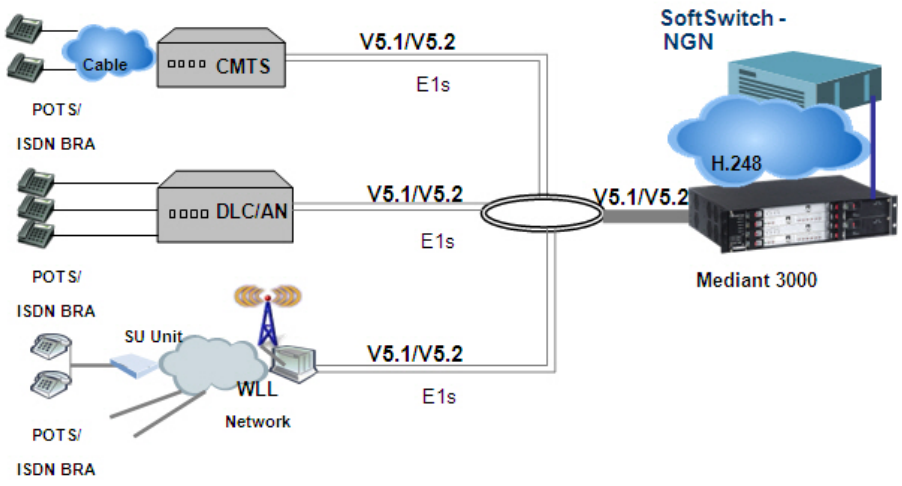
In the solution, only E1 trunks are supported. In each E1, the first DS0 is used as a synchronization timeslot, implying that 1,953 out of the 2,016 DS0s are used as voice or signaling channels.

The V5.2 Access Gateway in the TP-8410, converts messages from the legacy TDM protocol V5.2 to H.248 ITU standard analog packages and then hands them off to a softswitch and vice-versa. The V5.2 Access Gateway also converts the voice from TDM payload to VoIP RTP, and vice versa.

The gateway, together with the softswitch, replaces the traditional Class 5 switch, and therefore implements the LE side of the V5.2 Protocol.

The following diagram illustrates the use of the Mediant 3000 in a Next Generation Networking environment. Only PSTN (POTS) user ports are currently supported, although the diagram shows ISDN and PSTN user ports.

Figure 34: Mediant 3000 as a V5.2 to MEGACO Solution



6.1.1.3 About the V5.2 Protocol

V5.2 is a standardized protocol suite for the connection of Access Networks (AN) to the Local Exchange (LE) usually implemented in a traditional Class-5 switch. The Access Network itself typically has PSTN and ISDN user ports to the customer. The V5.2 interfaces are based on G.703/G.704 interfaces at 2,048 kbit/s (E1).

V5.1 is a single 2,048 kbit/s interface, whereas V5.2 may consist of up to sixteen 2,048 kbit/s links, configurable by the operator.

Time slot 0 (TS0) of the 32 time slots is always used for frame alignment, error reporting and error performance monitoring using cyclic redundancy check procedures. In the case of V5.2 links, TS0 is additionally used to verify the correct physical connection of a 2,048 kbit/s link. Up to three time slots on each 2,048 kbit/s link may be assigned as so-called communication channels (C-channels). C-channels carry PSTN signaling, ISDN D-channel information, control information and, in case of V5.2 the bearer channel connection protocol (BCC) and the protection protocol.

All 64 kbit/s time slots not provisioned as C-channels are available as PSTN or ISDN bearer channels or may carry analogue or digital leased lines.

For more details on V5.2 Protocol, refer to 'V5.2 Interface Activation/Deactivation' on page 452.

For a V5.2 protocol overview, please refer to Overview of V5.2.

6.1.1.4 About the MEGACO/H.248 Protocol

MEGACO (synonymous with H.248) is a control protocol that enables switching of voice, fax and multimedia calls between the PSTN and IP networks.

It is based on the term Media Gateway Controller and is the official industry standard protocol for interfacing between external call agents called **Me**dia **Ga**teway **Co**ntrollers (MGCs) and Media Gateways (MGs) in a VoIP network.

The standard is the result of a unique collaborative effort between the IETF and ITU standards organizations. Derived from MGCP (which, in turn, was derived from the combination of SGCP and IPDC), MEGACO draws heavily from MGCP, additionally introducing several enhancements.

For MEGACO implementation details refer to 'MEGACO (Media Gateway Control) Protocol' on page 560.

6.1.2 General Features

- Two Blades for Redundancy: The AudioCodes V5.2 Access Gateway platform - the Mediant 3000 - hosts two TP-8410 blades, running as active and standby.
- V5.2 Links: Each TP-8410 blade may have in its TDM side, up to 63 E1 trunks that may be configured as V5.2 links or as E1 transparent trunks.
- V5.2 Interfaces: These E1 trunks are divided into groups or V5.2 interfaces of 2-16 E1s each. There may be no more than 31 V5.2 interfaces in a V5.2 Access Gateway.
- Signaling Channels: A V5.2 interface uses 2 timeslots for signaling (active and standby) in separate V5.2 links called Primary and Secondary links (Protection Group). In both signaling links, the timeslot 16 is dedicated to the V5.2 signaling. In normal conditions, a V5.2 interface starts up with the signaling on the Primary link.
- Voice Channels: In all the links, timeslots 1 to 31 (except for timeslot 16 in signaling links) are used as voice channels. The number of simultaneous calls in a V5.2 Access Gateway is derived from the number of timeslots available for voice, after subtracting the signaling channels: 1945 simultaneous calls.
- User Ports: A V5.2 Access Gateway supports 10000 PSTN user ports distributed among the V5.2 interfaces so that each interface may have up to, but not more than 4,800 user ports.

Notes:

- Only E1 trunks that are configured as V5.2 links can be a part of a V5.2 interface. An E1 trunk that is configured as Transparent can carry a reference clock.
- Refer to the Release Notes regarding the number of supported user ports.

6.2 Principles

6.2.1.1 Blade Configuration

In order to use the Mediant 3000 and establish calls, the user must configure the V5.2 and MEGACO resources/parameters immediately after reset.

This configuration session may be performed in either of two ways:

- via the EMS
- by downloading the V5.2 parameters in the *ini* file. (Refer to 'V5.2 INI File Configuration' on page 460 for more details.)

In either option, the user must download a User Ports Configuration file which contains the configuration of all the PSTN user ports. Both the *ini* file and the User Ports Configuration file may be burnt to the flash so that they need not be downloaded again in subsequent resets. Once burnt to the flash, these files are automatically read at reset time. For more details on the V5.2 configuration, refer to 'Configuration' on page 456.

6.2.2 V5.2 Interface Activation/Deactivation

A V5.2 interface has two state fields:

- Admin State (Administration) - It is changed by the user from Offline to In-Service or vice versa in order take the V5.2 interface up or down.
- Oper State (Operational) - It reflects the actual state of the V5.2 Interface. Its values are:
  - Offline
  - Busy
  - In-Service

The figure below shows how the operational state changes according to occurring events or to user's commands.

The initial value of both the Administration and Operational states is "Offline". Once the blade has been configured, the user may activate a V5.2 interface. This is done by changing the administration state to In service. This immediately causes the Operational state to switch to "Busy" for a duration of 95 seconds after which the startup process begins. If the startup is successful, the Operational state becomes "In Service".

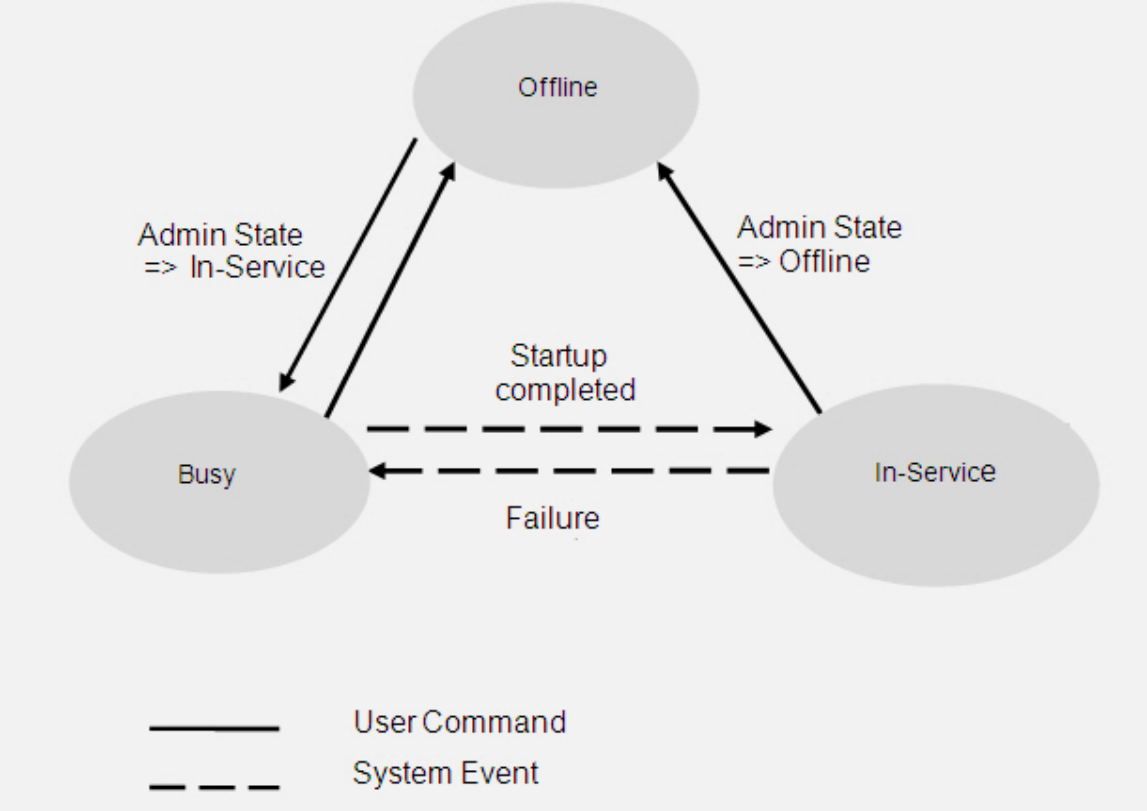
Only Layer 1 and 2 failures can set the operational state back to the "Busy" state, and only user's commands can set it to the "Offline" state.

Once the V5.2 interface is set to "In-Service", the user may establish calls either from the softswitch or from the Access Node.

No change in the configuration is allowed when the V5.2 interface administration state is set to "In-Service".

If you wish to change, add or delete a configuration parameter related to a V5.2 interface, you must terminate this interface, before performing the change (i.e. change the Admin state from "In-Service" to "Offline"). This action affects the existing calls. See also 'Starting and Stopping a V5.2 Interface' on page 454 and 'EMS V5.2 Commands' on page 462.

Figure 35: V5.2 Interface Operational State Transitions



6.2.3 Protection Switch-Over

6.2.3.1 Protection Switch-Over

This action is allowed only when the V5.2 interface Admin State is set to "In-Service". It is applied on a V5.2 interface to move the PSTN, Control, BCC and Link Control c-paths from Timeslot 16 of the primary link, to Timeslot 16 of the secondary link or vice versa.

This action is automatically performed whenever a problem occurs in the physical or data link layers. It may also be initiated by the user at the V5.2 interface level. (Refer also to 'EMS V5.2 Commands' on page 462).

Note: A protection switch over is allowed only if the destination link is in Operational state. An automatic protection switch over may occur when there is a failure on the active link. But there will be no switch back when the failed link recovers.

6.2.4 Link-Id Check

This action is allowed only when the V5.2 interface Admin State is In-Service. The procedure is used in order to check the link identification for a specific link to ensure that there is no mismatch between the ends of the link. It is automatically applied on a V5.2 link by either peer at startup time or after the link recovers from a physical failure. The user may also activate the procedure as described in 'EMS V5.2 Commands' on page 462.

6.2.5 Blocking/Unblocking a V5.2 Link

This action is allowed only when the V5.2 interface Admin State is In-Service. The user may block and unblock a V5.2 link, but these actions may also be initiated by the peer entity. A block action requested by the AN may be deferred or non-deferred. The LE always performs block actions (non-deferred). The non-deferred block action takes down all the calls existing on that link, while a deferred block takes place only after the last call existing on that link goes down. New calls will not have their voice channel allocated on blocked links. If the link being blocked carries the signaling, a protection switch-over will occur.

Note: The whole V5.2 interface may terminate if the blocked link is the only operational link in the interface.

6.2.6 Blocking/Unblocking a PSTN User Port

The blocking and unblocking actions are performed in the soft-switch or by the AN. The user cannot block or unblock a PSTN user port from the V5.2 Access Gateway. A block action may be deferred or non-deferred. In case the user port is busy (in call), the deferred block action will be granted only after the port has gone on-hook.

6.2.7 Starting and Stopping a V5.2 Interface

In order to establish calls the V5.2 interface must be started. When the configuration is completed, the user may start the interface by changing the Admin State to "In-Service" and initiate a start-up process. Some validations are performed. In case of failure, errors are displayed and the Admin and Oper states remain "Offline". Otherwise, the Admin state is changed to "In-Service", and the Oper state to "Busy".

After a while, the startup process begins. During this process, the application synchronizes with the AN and restarts all the configured user ports.

In case of start-up failures, alarms are set (displayed in EMS) and the system retries periodically to start-up again. If the process completes successfully, the Oper sate becomes "In-Service" and calls may be established. The user may stop a V5.2 interface by changing the Admin state to "Offline". This action affects the call processing. It also causes the "Oper" state to change to "Offline".

See also 'V5.2 Interface Activation/Deactivation' on page 452 and 'EMS V5.2 Commands' on page 462.

6.3 Redundancy

The Mediant 3000 application hosts two TP-8410 blades for High Availability.

One of the blades acts as the active blade and the other one as the standby blade. The active blade constantly updates the standby by sending it all the configuration and dynamic information, via the Auto-update mechanism (AUDP). In case of failure, the standby blade takes over and becomes active.

Typically there are two kinds of "Redundancy":

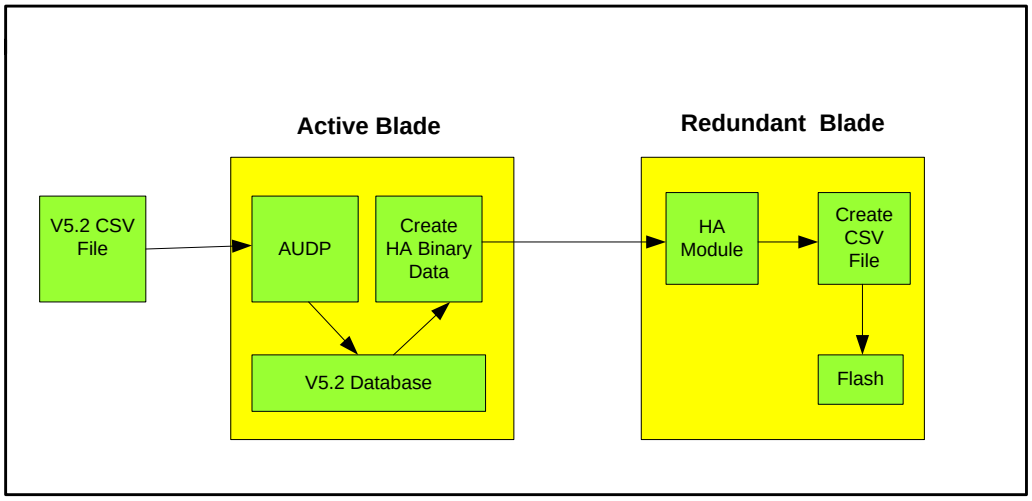
- **Full Redundancy** – The active blade constantly updates the standby blade with static and dynamic data. In case of failure, the switch–over is performed smoothly so that the existing calls are not affected.
- **Warm Redundancy** – The standby blade is updated with static data and when the active blade fails, the E1s are automatically switched to the standby blade. The latter is reset and restarts the V5.2 protocols. Existing call are interrupted. After the restarts complete, the users may initiate new calls, using all E1 capacity.

The V5.2 Access Gateway uses a Mediant 3000 chassis and therefore inherits all the platform features.

The Ports Configuration file is sent from the active blade to the redundant blade as a binary file. In the redundant blade, the file is converted to a text file and burnt to the flash.

After reset, the redundant blade comes up as a regular blade, reading its configuration from the flash.

Figure 36: Configuration File in Redundancy



6.4 Configuration

This sub-section describes the configuration of the V5.2 Access Gateway. To configure the H.248 protocol, refer to 'MEGACO-Specific Parameters' on page 253 and 'Individual ini File Parameters' on page 164 . The configuration is enabled via an *ini* file or by using the EMS.

The V5.2 configuration consists of:

1. Loading an *ini* file, at reset time, as done on TP-8410 blades. This file may contain V5.2 configuration tables.
2. Configuring on-the-fly V5.2 tables (if not already done in Step 1 above).
3. Downloading a User Ports Configuration file.

Note: The entire configuration can be burnt to the flash so that no configuration will be necessary at subsequent resets.

6.4.1.1 Configuration Tables, Files and Parameters

The resources that need to be configured in the V5.2 Application are the following:

- Trunks - There is no change in the way a trunk is configured, but additional parameters must be considered when using a V5.2 trunk. Trunks can be configured in *ini* files or through EMS.
- V5.2 interfaces - The V5.2 interfaces are configured in a table. Refer to the description of the V5.2 Interfaces table in V5.2 Interfaces Configuration.
- V5.2 User ports - User ports are configured by downloading a V5.2 User Ports Configuration file to the blade. Refer to 'PSTN User Ports' on page 457 for more information.
- V5.2 Links - The V5.2 links are configured in a table. Refer to the description of the V5 Links table in V5.2 Links Configuration.
- Other Parameters

Notes:

- A V5.2 link cannot be configured unless the corresponding V5.2 trunk and V5.2 interface have been previously configured.
- If the blade has no V5.2 trunk configured, the first V5.2 trunk configured on-the-fly necessitates a blade reset.
- When a V5 trunk has been defined, the blade becomes a V5 blade. This means that no trunk with the protocol type other than V5 or Transparent, can be configured.
- Similarly, if the blade has trunks with a protocol type other than V5 or Transparent, no V5 trunk can be configured.

6.4.2 PSTN User Ports

6.4.2.1 General

The V5.2 user ports are configured through a V5.2 User Ports Configuration File. This is the only way the user can configure PSTN user ports to V5.2 interfaces.

The file contains user ports belonging to all the V5.2 interfaces. It is downloaded to the TP-8410 blade after reset (before or after other V5.2 configurations).

If user ports are to be added or removed, the relevant lines need to be modified and the entire file downloaded again to the blade.

6.4.2.2 Configuration File Creation

The file is a CSV comma separated file. Each line in the file might be either:

- The Header or Version Line - It is the first line in the file. It describes the version number as follows:

```
; 1.0 version
```

- A Comment Line - It should start with ";" and may be placed anywhere in the file, but not before the header line.
- A Command Line - A command line configures one PSTN user port or line. This line format is:

```
<command>,<Interface number>, <port number>,<Layer3 address>
```

It should observe the following rules:

- The command may be "add" or "del". But in the current version only the "add" command is currently supported.
- Up to 10,000 user ports may be configured through this file.
- An Interface number should be in the range of 0 to 30.
- A Port number should be unique within the V5.2 interface and in range 1 to 4800.
- An L3 address should be unique within the V5.2 interface and in range 0 to 32766.

Note: The user must **not** download the configuration file unless all the V5.2 interfaces are set to the "Offline" state.

The following is an example of a User Ports Configuration File:

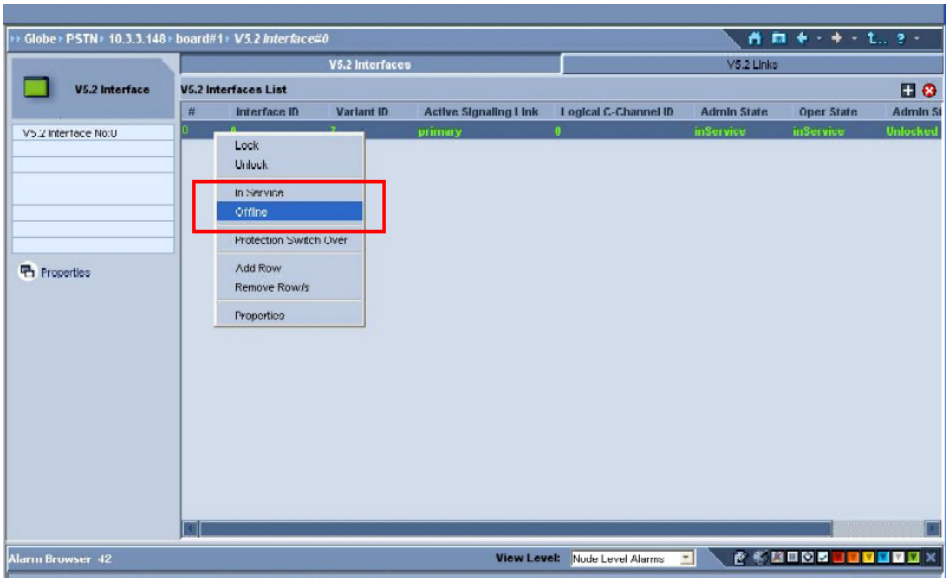
```
 ; 1.0 version
; add to V5.2 Interface 12 the port 35 with L3 address 4000
add,12,35,4000
; add to V5.2 Interface 17 the port 22 with L3 address 2345
add,17,22,2345
```

6.4.2.3 User Ports Configuration File Download

This section provides instructions for the User Port Configuration file download using EMS.

1. Create the file as described in the Configuration File Creation paragraph above.
2. Put the file on the EMS server.
3. Make sure that all the V5.2 interfaces are set to "Offline". If they are not, change their Admin state to "Offline" as follows:
   - On the TP-8410 blade screen, left-click on the **Signaling** tag and then on the V5.2 tag.
   - When the V5.2 tables screen is displayed, right-click on the V5.2 Interfaces to get the list of V5.2 interfaces.
   - Select an interface and right-click to display the Commands menu as shown in the figure below.

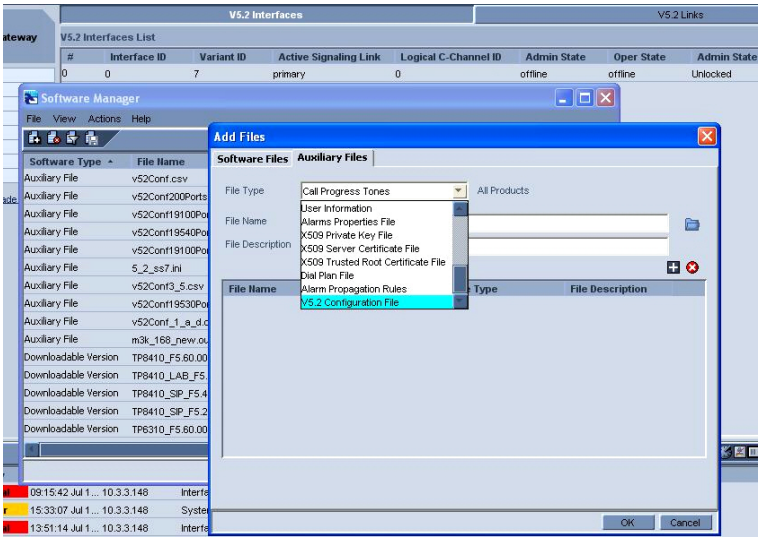
Figure 37: Setting a V5.2 Interface Admin State to "Offline"



Note: The above screen is only for illustration purposes. The actual EMS screen may vary. For the current screen view please refer to the EMS User's Manual.

4. Use the EMS Software Manager (Tools > Software Manager) to add the configuration file to the server before upgrading the software (download process).

Figure 38: Putting the V.2 Configuration File on the Server



Note: The above screen is only for illustration purposes. The actual EMS screen may vary. For the current screen view please refer to the EMS User's Manual.

5. After the Software Manager has successfully completed this step, use the **Software Upgrade** command to download the file to the blade.

Note: For more information about the download process, see the "Software Manager" chapter in the EMS User's Manual Version 5.6 document.

6. Download Results:

If the process ends successfully, the following message will be displayed in the EMS:

Lines Added = ___, Removed = ___, Duplicated = ___, Not exist = ___, check the Syslog for more information.

If no error occurred, the number of added ports should be the number of the **Add** commands in your file. If you tried to add a port that already exists, the **Duplicated** counter will be incremented. The **Removed** and **Not exist** fields are for future use.

If the download process fails, one of the following errors may appear in the EMS:

- **Malloc failed while loading V5.2 port configuration file** – Blade memory problem.
- **Bad file version ID. No configuration takes place, check the Syslog for more information** – Error at the file header. Check the Syslog for more information.
- **File validation error. No configuration takes place, check the Syslog for more information** – Syntax error at one of the lines in the file, such as too many or too few parameters. Check the Syslog for more information.
- File validation error. Download is in progress but no file found. File name = ____ **– internal error.**
- There are more ADD commands in the file than the V5.2 can have. There are only ___ free ports when the file has ___ **– You tried to add ports above the maximum allowed (310 * number of trunks).**

- File was loaded to the blade but the blade configuration is not V5 **– You tried to load the V5.2 port configuration file to a blade that is not configured as V5.2.**

6.4.2.4 Other Configuration Parameters

In a V5.2 Access Gateway, the following parameters must be configured in the *ini* file:

- CallerIDType = 1
- CallerIDTransportType = 1
- ETSICallerIDTypeOneSubStandard = 0
- CASProtocolEnable = 0

6.4.3 V5.2 *ini* File Configuration

Configuring the V5.2 Access Gateway through the *ini* File requires the definition of trunks, V5.2 Interfaces and Links tables and MEGACO parameters. The User Ports Configuration file may also be downloaded at reset time. In this case, the file URL should be provided in the *ini* file.

6.4.3.1 INI File Example

1. Miscellaneous Parameters.

```
CallerIDType = 1
CallerIDTransportType = 1
ETSICallerIDTypeOneSubStandard = 0
CASProtocolEnable = 0
```

2. V5.2 Parameters in trunks configuration. 4 trunks are configured as V5.

```
; V5.2 Protocol type = 43
ProtocolType_0 = 43
ProtocolType_1 = 43
ProtocolType_2 = 43
ProtocolType_3 = 43
V5NumberOfCChannels_0 = 1
V5NumberOfCChannels_1 = 1
V5NumberOfCChannels_2 = 0
V5NumberOfCChannels_3 = 0
; V5ProtocolSide. 0 for AN, 1 for LE (only LE is supported)
V5ProtocolSide_0 = 1
V5ProtocolSide_1 = 1
V5ProtocolSide_2 = 1
V5ProtocolSide_3 = 1
```

3. V5.2 Interfaces table: In this example, one interface is defined.

```
[V5Interfaces]
FORMAT V5Interfaces_Index = V5Interfaces_V5InterfaceId,
V5Interfaces_VariantId, V5Interfaces_LogicalCchannelId,
V5Interfaces_V5AdminState, V5Interfaces_TraceLevel;
V5Interfaces 0 = 17, 7, 0, 2, 0;
[\V5Interfaces]
```

4. V5.2 Links table:4 links are defined, all belonging to the V5.2 interface #0.

```
[V5Links]
FORMAT V5Links_Index= V5Links_LinkId, V5Links_V5InterfaceIndex,
V5Links_LinkType;
V5Links 0 = 20, 0, 1;
V5Links 1 = 21, 0, 2;
V5Links 2 = 22, 0, 0;
V5Links 3 = 23, 0, 0;
[\V5Links]
```

5. V5.2 User Ports Configuration file (should be in ftp directory).

```
V5PortConfigurationFileName = 'V5Ports.csv'
```

Notes:

- Only configurable parameters should appear in *ini* files. Read-only parameters are ignored.
- The Row Status is never configured in *ini* file tables.

6.4.4 Configuration Constraints

Note: When SS7 is configured, it is not possible to add a V5 configuration on the same blade. Similarly, there can be no SS7 configuration on a V5 board.

6.5 EMS V5.2 Commands

6.5.1 Commands Applicable at V5.2 Interface Level

Several types of commands may be applied in EMS on a V5.2 interface:

- Commands on the Table Row
  - Add – Adds a new row
  - Remove – Deletes an existing row, when its row status is locked.
- Commands on the Row Status
  - Unlock – Activates a row after its configuration. This action has no impact on the V5.2 interface itself, but only on the table row.
  - Lock – Moves the row to a configuration state. The row parameters cannot be changed unless the row status has been moved to the locked state.

Note: A V5.2 interface row cannot be locked if the Admin State is set to "In Service".

- Commands on the V5.2 interface as a V5.2 entity.

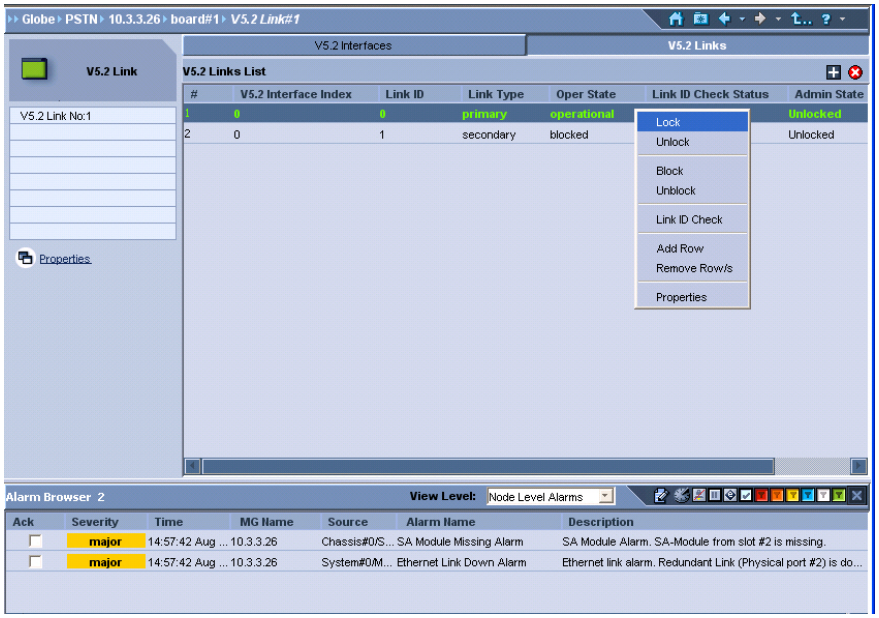
The following commands cannot be used when the row is in a "Lock" state.

- In Service – Changes the Admin State of the V5.2 interface to In-Service. This command initiates a start-up as mentioned above.
- Offline – Deactivates the V5.2 interface.
- Protection Switch Over – Transfers the signaling from the Primary Link of the V5.2 interface to the Secondary or vice versa.

6.5.2 Commands Applicable at Link Level

All the commands are shown in the following figure. To activate these commands, select a V5.2 link in the V5.2 Links table, and right-click on it.

Figure 39: V5.2 Link Commands in EMS



Note: The above screen is only for illustration purposes. The actual EMS screen may vary. For the current screen view please refer to the EMS User's Manual.

The following types of commands may be applied in EMS on a V5.2 link:

- Commands on the table row
  - Add – Adds a new row.
  - Remove – Deletes an existing row, when its row status is locked.
- Commands on the Row Status
  - Unlock – Activates a row after its configuration. This action has no impact on the V5.2 link itself, but only on the table row.
  - Lock – Moves the row to a configuration state. The row parameters cannot be changed unless the row status has been moved to the locked state.

Note: A V5.2 link row cannot be locked if the Admin State of the associated V5.2 interface is set to "In Service".

- Commands on the V5.2 link as a V5.2 entity. These commands may not be used when the row is in locked state.
  - Block –Blocks the V5.2 link. This affects the signaling and the calls on this link.
  - Unblock – Unlocks the V5.2 link.
  - Link-ID Check. Performs a Link-Id Check process on the V5.2 link.

6.5.3 Gateway Maintenance Actions

Graceful Management (i.e. Locking / Unlocking the device with either Graceful or Forced setting) can be implemented in the following ways:

- SNMP (refer to 'Administrative State Control' on page 107)
- H.248 protocol (refer to 'Graceful Management via MEGACO' on page 575)
- Web Interface (refer to Maintenance **Actions** in the Web Interface chapter of the device's User Manual).

The following explains the implementation of these management commands in the V5 protocol. Please note that these operations can be used for the whole Gateway only (and not on a specific V5 Interface or trunk).

6.5.3.1 Force Lock

A reset starts regardless of traffic, and any existing traffic is terminated at once. The implementation in the V5 protocol is to send an AIS_ALARM on all links. As a result, all the V5 signaling will be stopped and after a V5 standard time out, the V5 interfaces will change their state to BUSY.

6.5.3.2 Graceful Lock

The device reset starts only after a user-defined time expires (i.e., timeout) or after no active traffic exists (which ever is earliest). In this state, the Gateway rejects any start of a new call (receiving an Establish message) by replying with the Disconnect message. The V5 interfaces remain in the IN_SERVICE state.

6.5.3.3 UnLock

The behavior is dependent on the previous management state:

- If the previous management state is Force Lock, the physical alarm (AIS_ALARM) is removed. As a result, the V5 interfaces perform re-synchronization.
- If the previous management state is Graceful Lock, the Gateway stops rejecting the new incoming calls.

6.5.4 CLI Commands

The following CLI commands provide complimentary information about the V5.2 environment.

They are displayed in the following table. Some terms should be clarified before using the commands:

- **Line** - Rfers to a user port or phone when it is not used in a V5.2 interface context. There are up to 10,000 lines in a V5.2 Access Gateway.
- **Port** – Refers to a user port index within a V5.2 interface, as configured in the User ports configuration file. There are up to 4,800 ports in a V5.2 interface
- **L3 Address** – The V5.2 identifier of a PSTN user port within a V5.2 interface, as configured in the User ports configuration file.

Conversion commands are provided to convert a line number to a V5.2 interface and Port number (or L3 address) and vice versa.

CLI Commands

Command	Short Format	Arguments	Description
/PStn/V5/V5DataBase/ v5ConvertPorttoLine	/ps/v5/vdb/cpl	<V5.2 Interface 0-30> <Port 1-4800>	Converts a V5.2 interface and port number to a line number.
/PStn/V5/V5DataBase/ v5ConvertLinetoPort	/ps/v5/vdb/clp	<Line number 0-19999>	Converts a line number to a V5.2 interface and port number.
/PStn/V5/V5DataBase/ v5Convertl3AddrtoLine	/ps/v5/vdb/cal	<V5.2 Interface 0-30> <L3Address 0-32767>	Converts a V5.2 interface and L3Address to a line number.
/PStn/V5/V5DataBase/ v5ConvertLinetol3Addr	/ps/v5/vdb/cla	<Line number 0-19999>	Converts a line number to a V5.2 interface number and L3-address.
/PStn/V5/V5DataBase/ v5PortCountAll	/ps/v5/vdb/pca	None	Counts and displays the number of user ports in all the V5.2 interfaces.
/PStn/V5/V5DataBase/ v5PortDisplay	/ps/v5/vdb/pd	<V5.2 Interface 0-30> <from port 1-4800> <to port 1-4800>	Displays a range of user ports belonging to a specific V5.2 interface.
/PStn/V5/V5Bchannels/ v5GetAllocTrunkBch	/ps/v5/vb/gatb	<Line number 0-19999>	Displays the Trunk and B-channel allocated to the line.
/PStn/V5/V5Bchannels/ v5GetAllocLine	/ps/v5/vb/gal	<Trunk number > <BChannel 1-31>	Displays the line attached to the Trunk and B-channel.
/PStn/V5/V5Maintenance/ v5GetLinkInfo	/ps/v5/vm/gli	<V5.2 Interface 0-30> <Trunk number>	Displays information on a V5.2 link
/PStn/V5/V5Maintenance/ v5GetPortInfo	/ps/v5/vm/gpi	<V5.2 Interface Id> <User Port L3Add>	Displays information on user port

6.5.5 V5.2 H.248 Solution

This section describes the migration of the V5.2 Local Exchange and access network nodes to Next Generation Network Media Gateway and Media Gateway Controller architecture. It provides guidelines for mapping the V5.2 Protocol to the H.248.1 protocol and H.248 analog packages as published in H.248.23 and H.248.26 sub-series.

6.5.5.1 Termination Naming

The Termination ID structure is provisioned in the Media Gateway Controller and Media Gateway and is known by both of them at start-up or before.

Two hierarchical naming structures can be used for physical terminations:

- The first structure is a two level physical termination naming:

```
al/<interface id>/<port>
```

Where “al/" is a fixed prefix, <interface id> and <port> are non-zero integer values.

According to this naming scheme, an analog line connected on port 2 of interface 1 would be referred to as al/1/2.

The mapping between the V5.2 protocol address (Interface/L3Address) and H.248 termination naming (Interface/Port) is done by provisioning. (Refer to 'PSTN User Ports' on page 457 for more information.)

The Interface Id must be the same and each L3Address is mapped to unique Port Id, according to the Physical location of this user at the AN box.

- The second structure is a three level physical termination naming as defined in the ETSI ES 283 002 profile

```
al/<subrack>/<card>/<port>
```

Where, "al/" is a fixed prefix and <subrack>, <card> and <port> are non-zero integer values.

According to this naming scheme, an analog line connected on Port 2 of Card 1 in Subrack 22, would be referred to as al/22/1/2.

The mapping between the V5.2 protocol address (Interface/L3Address) and H.248 termination naming (Subrack/Card/Port) is done in the following way:

- The Interface ID is a subrack id.
- The port in the interface is calculated according to (card#) x (Nmax) + port, where Nmax is the maximum number of ports in each card and the first card number is assumed to be 0. Nmax value can be configured by the *NumberOfV5PortsInCard* parameter in the V5Interface table.

Notes:

- The maximum number of ports in each card is uniquely defined within the subrack.
- No mixes are allowed within the subrack. (We can have different cards, with different number of ports, between the subrack only).
- The number of cards in the subrack is limited to 253.
- By default, the <subrack> value starts from 0. In order to change it, the user should configure the EP_Num value at level 0.

6.5.5.2 Mapping V5 Protocol to H.248

This section provides guidelines on mapping the V5 protocols to the H.248 protocol and H.248 analog packages as published in the H.248.1, H.248.23 and H.248.26 sub-series.

The following table describes the mapping of V5 PSTN Protocol messages to H.248 commands.

Mapping of V5.2 PSTN Protocol Messages

Message Type	Mapping Direction MGC =>MG/LE=>AN	Mapping Direction AN=>LE/MG=>MGC	Mapping Details
Path Establishment Messages			
ESTABLISH	Add/Modify command Signal Descriptor Ringing Signals (al/ri,alert/ri,andisp/dwa)	Notify command Observed Event Descriptor Off Hook event (an/of)	See note below
ESTABLISH ACK	Not mapped. It is used internally in order to continue ringing process.	Not mapped. It is done automatically if Off Hook event is requested on this termination.	
SIGNAL	Add/Modify command Signal Descriptor Detailed signals mapping is defined in Table 3	Notify command Observed Event Descriptor Detailed event mapping is defined in Table 3.	V5.2 signals can be carried within an Events Descriptor and Signals Descriptor as defined in Table 3.
SIGNAL ACK	No mapping	No mapping	This V5.2 message is specific to the V5.2 PSTN protocol and support is not required within H.248.

Mapping of V5.2 PSTN Protocol Messages

Path Clearing Messages			
DISCONNECT	Subtract command or automatically when termination in Null context and upon receiving V5 ON HOOK steady signal	Notify command Observed Event Descriptor On Hook event (an/on)	The V5.2 "Disconnect" is mapped as follows: If the termination hook's state is ON hook, then H.248 Subtract command is mapped to V5.2 "Disconnect". And if the termination is in OFF hook state and termination is in the Null context, then the V5.2 "Disconnect" is done automatically upon receiving ON_HOOK signal from AN side
DISCONNECT COMPLETE	Not mapped. It is used internally in Off Hook Reject and when call is released before the Off Hook.	Transaction Reply	This is an acknowledgement to the V5.2 Disconnect Message. The subtract reply is sent after receiving this message from AN.
Other Messages			
STATUS ENQUIRY	No mapping	No mapping	This V5.2 message is specific/internal to the V5.2 PSTN protocol error handling. It is used to request the V5.2 PSTN protocol state of the AN and therefore no direct mapping is required. The H.248 protocol has an Audit Value command and Audit Descriptor for auditing the MG.
STATUS	No mapping.	No mapping	This V5.2 message is specific/internal to the V5.2 PSTN protocol error handling. It carries the state and cause information elements and therefore no direct mapping is required. The H.248 protocol has its own mechanism for conveying protocol errors by using the Error Descriptor.
PROTOCOL PARAMETER	No mapping.	No mapping	Not supported in current release

The following table provides a mapping of V5 information elements to H.248 concepts.

Mapping V5.2 PSTN Protocol Signal Information Elements

V5.2 PSTN Protocol Information Elements	H.248 Mapping Details	Notes
Line Information	Not mapped.	Not supported in current release.
Autonomous Signaling Sequence	Not mapped.	Not supported in current release.
Sequence Response	Not mapped.	Not supported in current release.
Cadenced Ringing	It is directly mapped to the ring signal defined in "analog line supervision" and "enhanced alerting" packages defined in H.248.1 and H.248.23 sub-series, and also used as part of the "display with alert" signal defined in "Analogue Display Signaling" package in H.248.23 sub-serial. The V5.2 cadence ring type information element is mapped to the "pattern" parameter in these signals.	AN to LE direction
Pulsed Signal	In the MG to MGC direction, only "register recall" type has direct map to the "Flash hook' event defined in "Analog Line Supervision' package in H.248.1. Other types have no direct mapping. In the MGC to MG direction, some of the information elements, like Initial Ring, are used internally as part of the ringing state machine.	AN to LE direction LE to AN direction
Steady Signal	In the MGC to MG direction, some of the types of this information element are mapped to the signals (signal descriptor) defined in H.248.26 as follows: Normal polarity is mapped to "Network disconnect signal" defined in "Extended Analog Line Supervision" package. Reversed polarity is mapped to "Line-side answer supervision signal" defined in "Extended Analog Line Supervision" package. Other types either have no mapping or they are used internally as part of the ringing or display with alert state machine In the MG to MGC direction, some of the types of this information element are mapped to the ObservedEvent defined in "Analog Line Supervision" package as follows: Off hook (loop closed) is mapped to the "Off hook" event. On hook (loop open) is mapped to the "On hook" event. Other types either have no mapping or they	LE to AN direction AN to LE direction

Mapping V5.2 PSTN Protocol Signal Information Elements

V5.2 PSTN Protocol Information Elements	H.248 Mapping Details	Notes
	are used internally as part of the ringing or display with alert state machine	
Digits Signal	In the direction of MG to MGC. The digits detected by the MG are reported as an event as part of H.248 digit detection package. In the direction of MGC to MG. No mapping	AudioCodes V5.2 Access Gateway cannot generate digits over V5. Only detection is available.
Recognition Time	This is no mapped to H.248 but recognition time is configured in MG.	Not supported in current release.
Enable Autonomous Acknowledge	To enable autonomous acknowledgement, an embedded Signals Descriptor is added to the event specifying the signal to be detected. The Signals Descriptor shall contain the signal that is to be autonomously sent in response to the detected event.	Not supported in current release.
Disable Autonomous Acknowledge	An autonomous acknowledgement in the MG can be disabled by replacing the events descriptor containing an embedded Signals Descriptor with an Events Descriptor, which does not contain embedding.	Not supported in current release.
Resource Unavailable	This is mapped to the H.248 .8 Error Code – 510, "Insufficient Resources".	Not supported in current release.
Enable Metering	Currently it is not used.	Not supported in current release
Metering Report	Currently it is not used.	Not supported in current release
Attenuation	This is mapped to the gain control property in the TDM Circuit Package (ITU-T H.248.1).	Not supported in the current release. The same functionality can be implemented by control the GW gain

The following table describes the mapping of the V5 Bearer Channel Connection protocol to H.248 commands.

Mapping of V5.2 Bearer Channel Connection (BCC) Protocol

V5.2 BCC Message	Mapping Direction MGC =>MG/LE=>AN	Mapping Details
Allocation	Add/Modify command (see note below)	An "Allocate" maps to H.248 Add commands, which specify an association between analog termination and the allocated b-channel or to the H.248 Modify command with DSP required signals in signal descriptor or DSP required events in event descriptor.
Deallocation	Subtract/Modify command (see note below)	The "Deallocate" maps to H.248 "Subtract" commands that return analog termination to the NULL context while disconnecting and de-allocated the b-channel or to the H.248 Modify command which release DSP required signals and events

The following table provides a mapping of V5.2 PSTN user port control protocol to H.248 concepts.

Mapping of V5.2 PSTN Port Control Protocol

V5.2 Port Control Message	H.248 Mapping	Notes
Port Control messages are mapped to the H.248 service change command. It can be used in both directions. When the MGC sends a command to the MG, it requests to change the state of the port according to the method type. When the MG sends a service change command to the MGC, it reports a change in the port state.		
Unblock	H.248 Service Change command with Force method	
Block	H.248 Service Change command with Restart method	

The above table contains an overview of the mapping of H.248 packages to V5.2 messages based on Table 1, 2 and 3. It provides an additional view of the V5.2 protocol mapping but this time from the H.248 packages perspective.

Mapping of H.248 Packages

Signal/Event/Property Name	Descriptor Parameter	V5.2 PSTN Protocol Mapping Details	Notes
"Analog Line Supervision" package defined in H.248.1			
"of" off hook event	not relevant	It is mapped to STEADY-SIGNAL message with OFF HOOK type in incoming call or ESTABLISH message in outgoing call	
"on" on hook event	not relevant	It is mapped to STEADY-SIGNAL message with ON HOOK type or DISCONNECT message	
"fl" flash hook event	not relevant	It is mapped to PULSED-SIGNAL message with REGISTER RECALL type	
"ri" ringing signal		There is no direct mapping of this signal, but it uses ESTABLISH message with Initial Ring and SIGNAL message with CADENCE-RINGING information element in order to ring. Default cadence type 1 is used for this signal.	Cadence and Frequency parameters are not mapped. Instead default pattern, with value 1, is used.
"Enhanced Alerting" Package defined in H.248.23			
"ri" ring signal	"pattern"	There is no direct mapping of this signal. It uses ESTABLISH message with Initial Ring and CADENCE-RINGING message in order to ring. The cadence type information element is mapped to the "pattern" parameter.	
"rs" ringsplash signal		It is mapped to the ESTABLISH message	
"Analogue Display Signaling" Package defined in H.248.23			
"dwa" display with alert	"ddb" - Display Data Block	There is no direct mapping of this signal. It uses the ESTABLISH message with Initial Ring and CADENCE-RINGING message in order to ring. The cadence type information element is mapped to the "pattern" parameter.	

Mapping of H.248 Packages

Signal/Event/Property Name	Descriptor Parameter	V5.2 PSTN Protocol Mapping Details	Notes
		In addition, PULSED-SIGNAL message is used as initial ring before "display data block"/caller-id generation.	
"Extended Analog Line Supervision" package defined in H.248.26			
"las" Line-side answer supervision signal		It is mapped to PULSED-SIGNAL message with REVERSE POLARITY type	
"nd" Network disconnect signal		It is mapped to PULSED-SIGNAL message with NORMAL POLARITY type	
"Automatic Metering" Package defined in H.248.26			
"pr" periodic report event		It is mapped to METERING-REPORT SIGNAL message	Not supported in current release
"em" enabling metering	Pulse count	It is mapped to ENABLE-METERING SIGNAL message	Not supported in current release
'mpb" metering pulse burst signal			Not supported in current release
"Digit Detection" Package defined in H.248.1			
D0-DF events		It is mapped to DIGIT-SIGNAL message.	
"TDM Circuit" Package defined in H.248.1			
Gain control property		It is mapped to ATTENUATION message.	Not supported in the current release. The same functionality is implemented by controlling the gain of the TDM side.

6.5.5.3 Permanent Off-hook Indication

Permanent off-hook is the state in which the phone handset was picked up and was never returned to the on-hook position. (This may occur after a call was made, or if the handset was un-intentionally moved). In such a state, the MGC/Exchange Response would be to play a warning tone which should alert the user to this state. However, after this tone has timed-out, some action should be taken, as the line is considered temporarily out of service until the handset has been returned to the on-hook position.

To indicate that state to the MGC, the gateway will send a *serviceChange* with the "Forced" method, whenever the line is detracted from a call and there is no second party on this call. (Either it is already on-hook or never existed). When the handset has been returned to the on-hook position, a *serviceChange* with the "restart" method will be sent to indicate that the line is back in service, and will be followed by the on-hook event.

Some MGCs do not need these service change commands, as they are handling the line state themselves, according to hook-state. Therefore, the behavior is controlled by the 5th bit (value 128) of the *MGCPCompatibilityProfile ini* file parameter.

6.5.5.4 Configuring Register-Recall Duration Type in the Access Network (AN)

This feature enables the V5.2 Access Gateway to configure the register-recall recognition time in the AN by sending the V5.2 PROTOCOL PARAMETER message.

The recognition time is the time that the signal should be stay active before being recognized. This is necessary for the AN to report the hook flash to the LE (Local Exchange - V5.2 Access GW). The configuration is done at the beginning of each the new call.

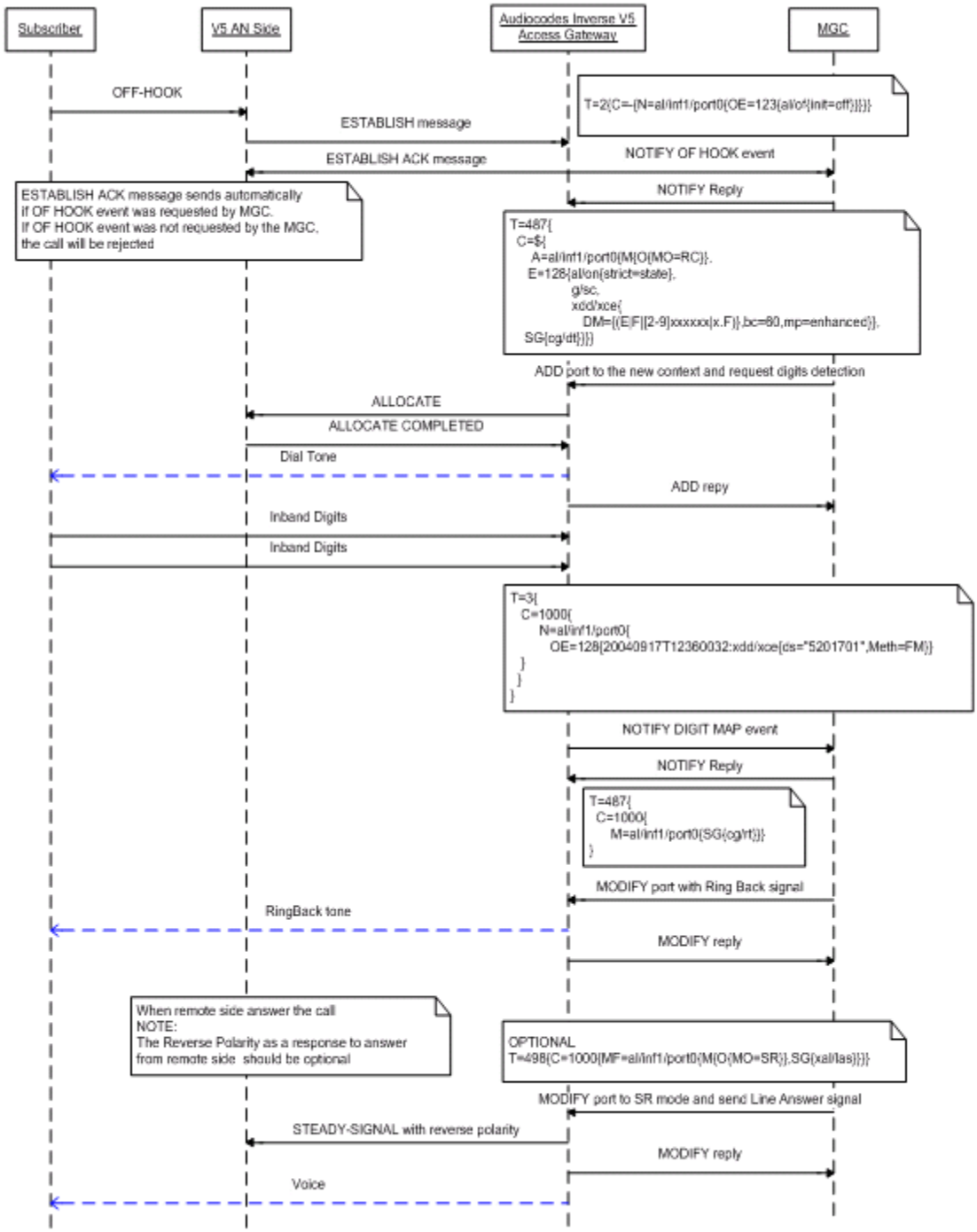
In order to enable it, the *V52EnableRegisterRecallConfiguration* configuration parameter should be set to Enable (1). Its default value is 0 - Disable. The *V52RegisterRecallDurationType* configuration parameter should be set with the value to be sent in the register-recall duration type. The register-recall recognition time is an index to a pre-defined table within the AN. The table in the AN contains the actual value of the duration of the register-recall recognition time.

6.5.6 Call Flow Examples

This section provides call flow examples showing the mapping of the V5.2 protocol to H.248.

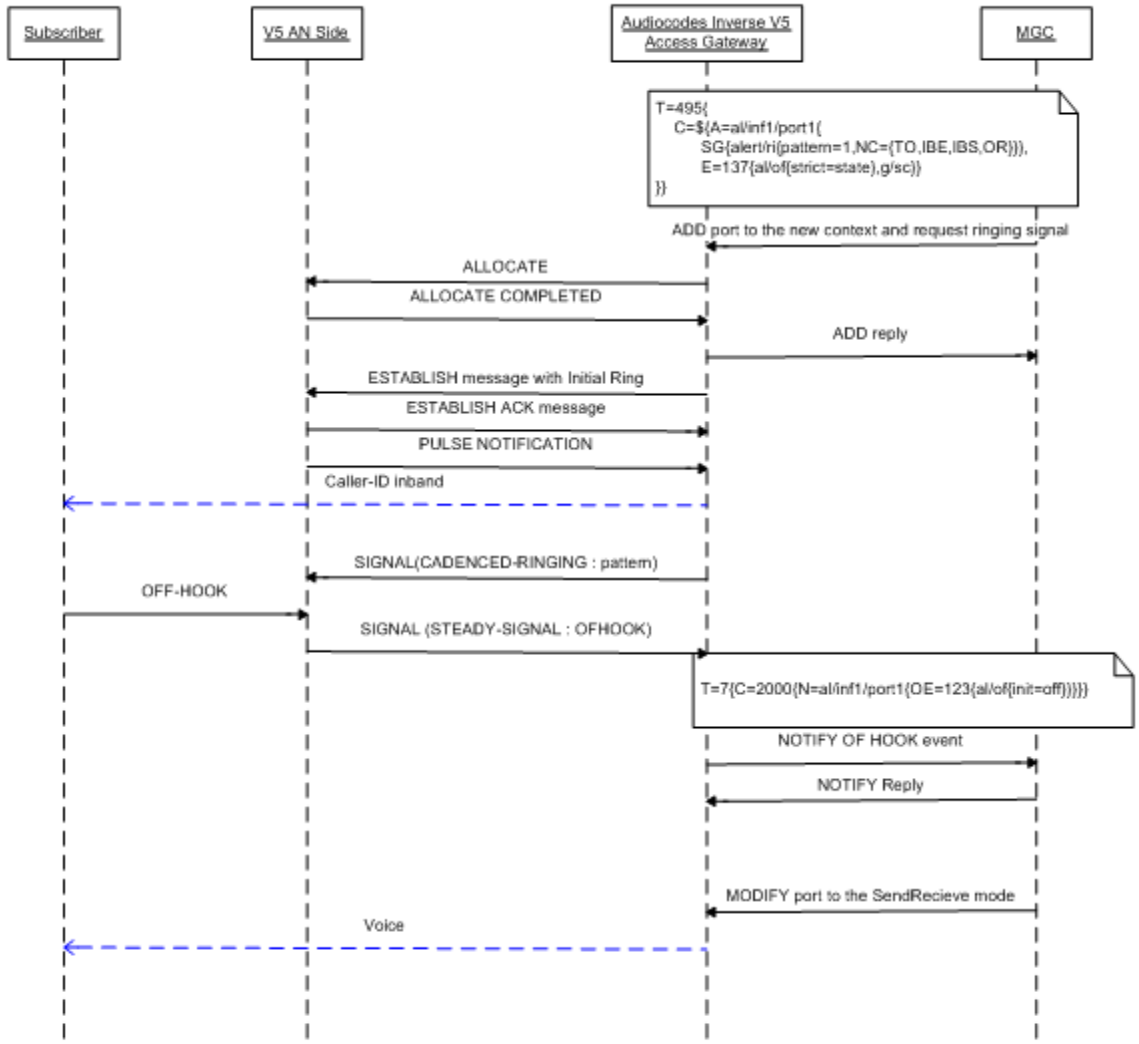
6.5.6.1 Incoming Call

Figure 40: Incoming Call



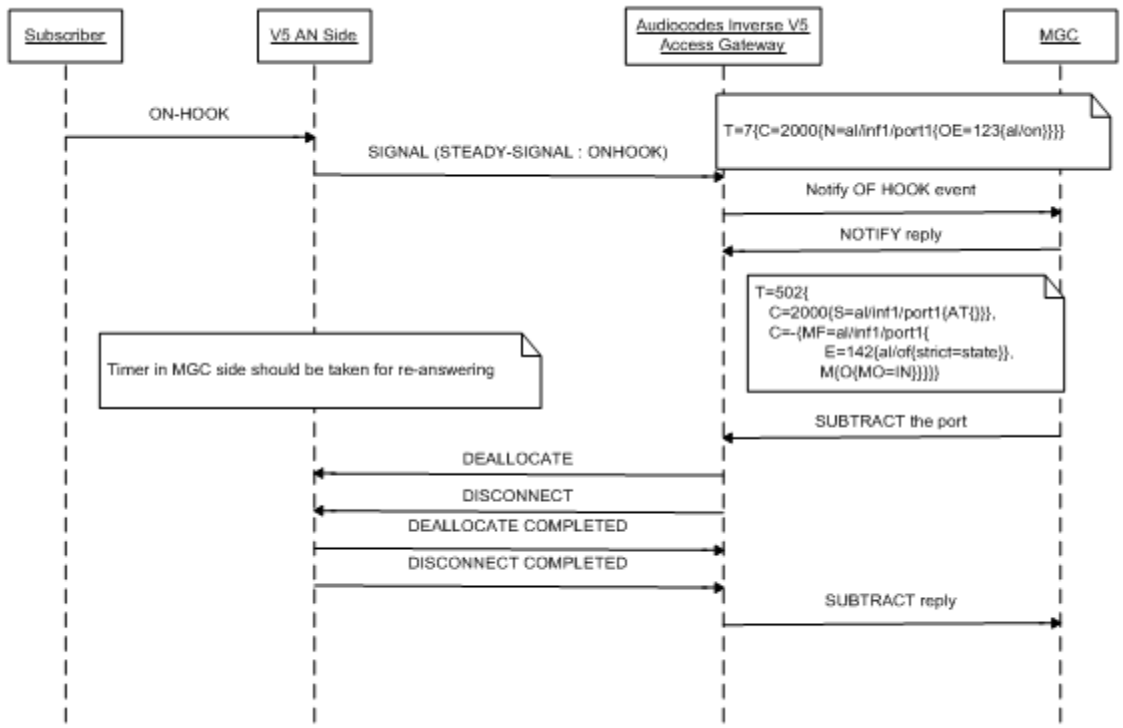
6.5.6.2 Outgoing Call

Figure 41: Outgoing Call



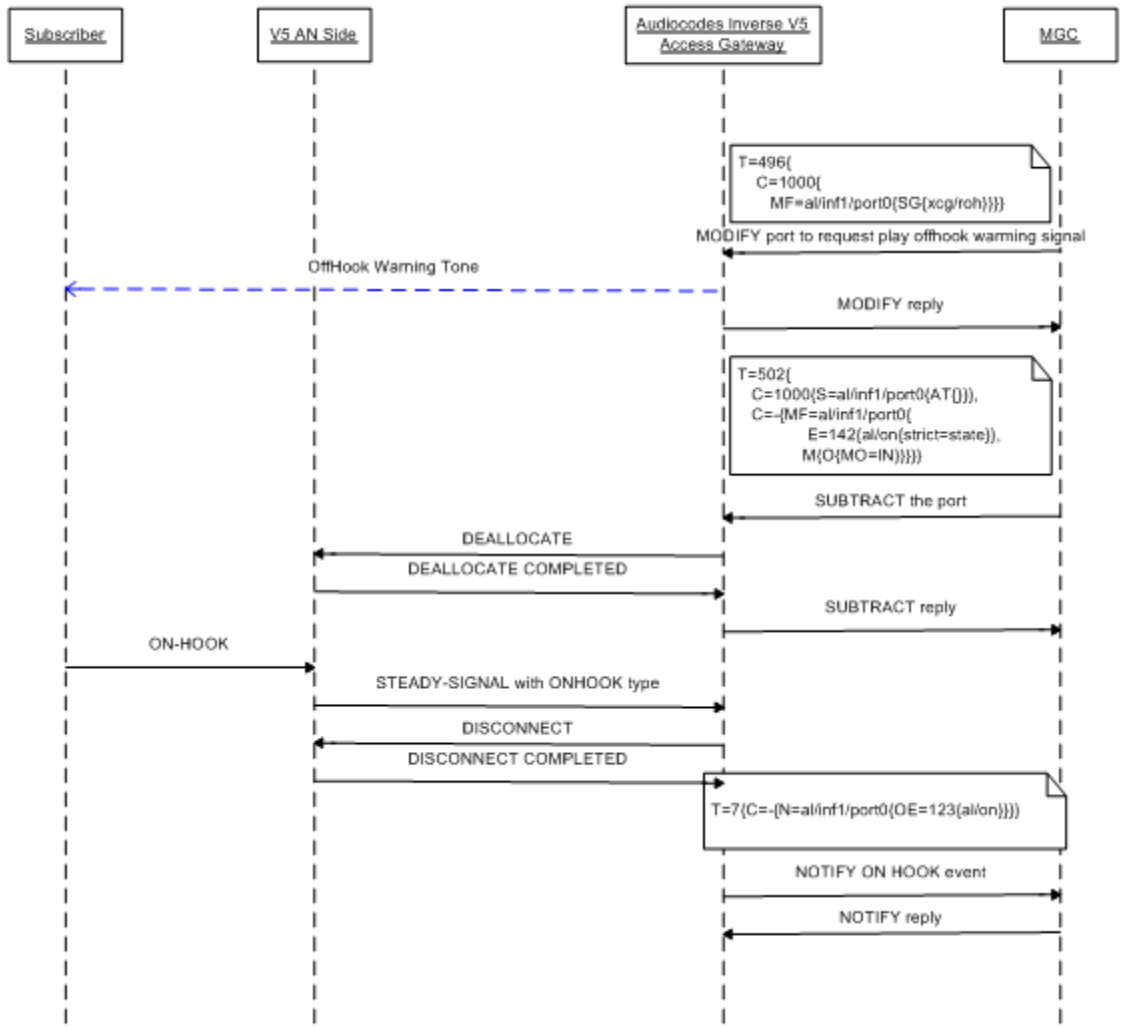
6.5.6.3 Call Disconnected by Access Network (AN)

Figure 42: Call Disconnected by AN



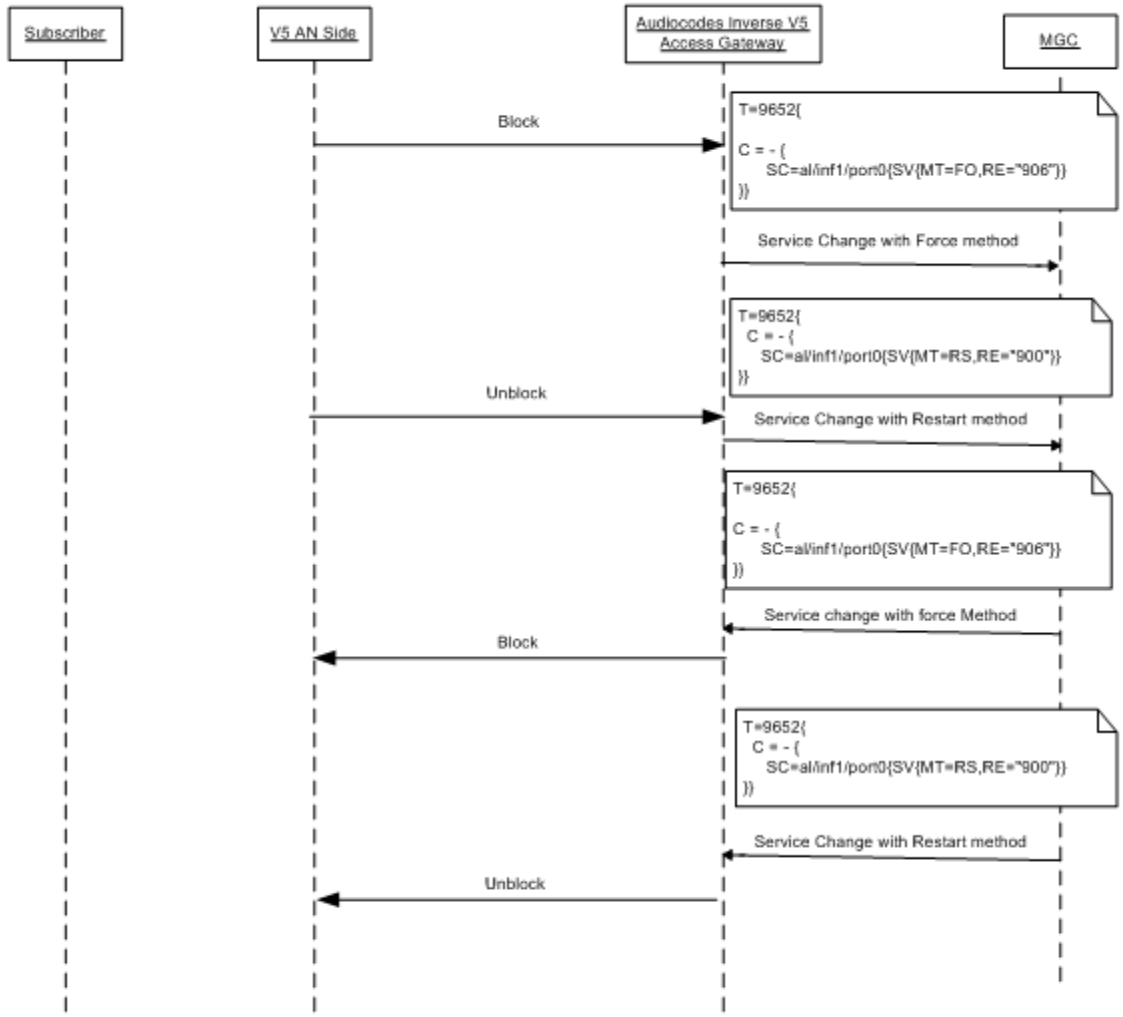
6.5.6.4 Call Disconnected before On-Hook by AN

Figure 43: Call Disconnected before On-Hook by AN



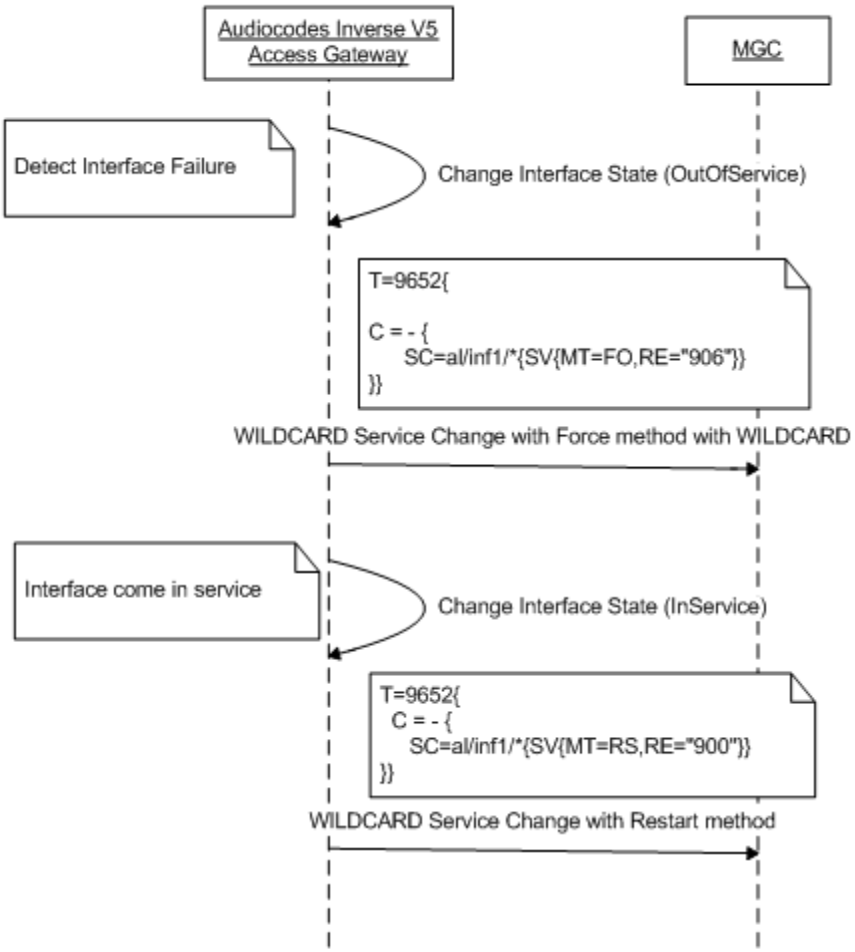
6.5.6.5 Port Control

Figure 44: Port Control



6.5.6.6 Interface Control

Figure 45: Interface Control



7 Standard Control Protocols

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: Some IETF URL links referred to in this document may not be active and in such a case may require an Internet search for the text required.

7.1 MGCP Control Protocol

7.1.1 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.

7.1.2 MGCP Operation

7.1.2.1 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.

7.1.2.2 MGCP Call Agent Configuration

The Call Agent can be configured using three different methodologies:

7.1.2.2.1 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.

7.1.2.2.2 Configuring Primary and Secondary IP Addresses

Up to four IP addresses are configured - one for the primary call agent and the other IP addresses are optional for the redundant call agents. If the primary CA stops responding, the re-transmission mechanism tries sending its commands to the first redundant CA IP (if redundant CA's have been configured). If the connection to the first redundant CA (if configured) fails, the board will send the commands to the second redundant CA and so on. When the connection to the last configured redundant CA fails, the board will switch again to the primary CA.

In case one of the redundant CA's is configured to a zero address, the subsequent redundant CA will not be used.

For example, if the second redundant IP address is configured to 0.0.0.0 or not configured, if the connection to the first redundant CA fails, the board will switch its connection to the primary CA.

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_0 = IP address B
RedundantAgentIP_1 = IP address C
RedundantAgentIP_2 = 0.0.0.0

CallAgentPort = Port A
RedundantAgentPort_0 = PORT B
RedundantAgentPort_0 = PORT C
```

7.1.2.2.3 Using a Configuration Table to Assign Endpoints

The third option defines the relationship between a group of trunks (endpoints in Analog gateways), and primary and secondary MGCs. The CPCallManagerGroups matrix is used for this purpose. The 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.

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'

The following matrix appears as follows:

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.

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.

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]
```

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.

7.1.3 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 and STR1 should contain one slash sign ("/").
- 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 a free text string.

7.1.4 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.

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

7.1.5 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.

7.1.6 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 747.

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).

7.1.7 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 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.

7.1.8 IPv6 Support for Media

The device supports dual IPv4 and IPv6 network stacks for media channels.

A user may simultaneously open different channels on the blade, some using IPv6 and some using IPv4.

The MGCP stack can offer SDP with local IP addresses of IPv4 only, IPv6 only and both IPv6 and IPv4. It can also answer an incoming SDP with remote IP addresses of IPv4 only, IPv6 only and both IPv6 and IPv4 according to the interface table and the *CpMediaIPVersionPreference ini* file.

On an outgoing call, when the MGCP stack offers a local SDP as a response to a MGCP command without RCO : (offer case)

- If both IPv4 and IPv6 address types are offered, the *CpMediaIPVersionPreference* parameter determines the order in which the addresses will be offered. This configuration parameter may have one of the following values:
  - Value 0 - Prefer IPv6 – if the parameter has this value the IPv6 address will be offered before the IPv4 address.
  - Value 1 - Prefer IPv4 - if the parameter has this value the IPv4 address will be offered before the IPv6 address.
  - Value 2 - IPV6 Only - if the parameter has this value only IPv6 address type will be offered.

- Value 3 - IPV4 Only - if the parameter has this value only IPv4 address type will be offered.

The default if to offer both and to prefer the IPv6 (Prefer IPv6 - 2)

- If the *CpMediaIPVersionPreference* offers both IPv6 and IPv4 (values 0 or 1), the returned local SDP will contain an IPv4 line for each media type, if the interface table contains IPv4 media entry, and an IPv6 line if the interface table contains an IPv6 media entry.
- In case both IPv6 and IPv4 addresses are offered, the local SDP returned will include grouped media lines, according to RFC 4091.

For example:

For the following MGCP command:

```
CRCX 23954 ds/tr0/1@[10.4.10.101] MGCP 1.0 tgcp 1.0
C: 1
M: recvonly
```

and

Interface Table is as follows:

Index	Application Type	Interface Mode	IP Address	GW	VLAN ID	Interface Name
0	OAMP + Media + Control	IPv4 Manual	10.4.10.101	10.4.0.1	1	O+M+C
1	Media	IPv4 Manual	10.3.10.101	10.3.0.1	3	second
2	Media	IPv6 Manual	1:290:8fff:fe09:909c	1:290:8fff:fe09:0000	3	third

The *CpMediaIPVersionPreference ini* file parameter has its default value as **cpPreferIPv6**.

The local SDP returned from the blade will be:

```
v=0
a=group:ANAT 1 2
a=group:ANAT 3 4
m=audio 4010 RTP/AVP 0
c=IN IP6 2000::1:290:8fff:fe09:909c
 a=mid:1
m=audio 4010 RTP/AVP 0
c=IN IP4 10.3.10.101
a=ptime:20
a=silencesupp:off - - - -
a=mid:2
m=image 4002 udptl t38
c=IN IP6 2000::1:290:8fff:fe09:909c
a=mid:3
m=image 4002 udptl t38
 c=IN IP4 10.3.10.101
 a=mid:4
a=T38FaxVersion:0
 a=T38MaxBitRate:14400
```

When a MGCP message is received with RCO, the local address type is selected according to the remote address in the RCO. If more than one address is offered in the RCO, the first compatible address is selected.

The blade does not support different media address versions for different media types (e.g., T.38 remote address and RTP remote address must be of the same address version).

After the first compatible address is chosen, the following media lines with the different addresses are ignored.

For example:

If the Interface Table is the one above, and the received command is:

```
CRCX 23954 ds/tr0/1@[10.4.10.101] MGCP 1.0 tgcp 1.0
C: 1
M: recvonly

v=0
a=group:ANAT 1 2
m=audio 4010 RTP/AVP 0
c=IN IP6 2000::1:290:8fff:fe09:919c
a=ptime:20
a=silencesupp:off - - - -
 a=mid:1
 m=audio 4010 RTP/AVP 0
 c=IN IP4 10.3.10.102
a=ptime:20
a=silencesupp:off - - - -
a=mid:2
```

The returned SDP will be

```
v=0
m=audio 4010 RTP/AVP 0
c=IN IP6 2000::1:290:8fff:fe09:909c
a=ptime:20
a=silencesupp:off - - - -
```

7.1.8.1 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 first bit (value 1) of the *ini* file parameter 'CPSdpProfile' must be turned on.

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=T38FaxVersion:0
a= cpar: a=T38FaxUdpEC:t38UDPRedundancy
a= cpar: a=T38MaxBitRate:14400
a= cpar: a=T38FaxMaxBuffer:1024
a= cpar: a=T38FaxMaxDatagram:238
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
```

7.1.8.2 RFC 3264 Offer-Answer Model Support

RFC 3264 is made up of three SDP negotiation bahaviors. Each behavior is configurable by turning on a different bit of the *CPSdpProfile ini* file parameter (Refer to 'SDP Support Profiling' on page 590).

In order to fully support this RFC, all the three bits should be turned on.

The required functionality and bits are as follows:

- Prefer the Remote Coder

If the 12h bit (Value 4096) of the *CPSdpProfile ini* file parameter is turned on during coder negotiation, the coders in the command are processed in the order they are listed in the RCO. (The remote coder order is preferred.)

- Symmetric Payload Types

If the 8th bit (Value 256) of the *CPSdpProfile ini* file parameter is turned on, we use (and return in the Local SDP in the response) the same payload received in the remote offer (and not the default payload type from the coder table as used by default).

- Multiple M-lines

If the second bit (Value 4) of the *CPSdpProfile ini* file parameter is turned on, the blade will return for each "m=" line in the command, a corresponding "m=" line in the response. The response will contain exactly the same number of "m=" lines as the command. If a m=line in the command is not supported, the m-line will be returned with Port 0.

For example, if the command contained the following:

```
v=0
o=- * * IN IP4 10.17.2.100
s=-
c=IN IP4 10.17.2.100

m=audio 4020 RTP/AVP 0
m=image 4002 udptl t38
```

and T.38 is not supported, the blade will return the following:

```
v=0
o=- * * IN IP4 10.4.10.101
s=-
c=IN IP4 10.17.2.100
m=audio 4010 RTP/AVP 0
a=fmtp:18 annexb=yes
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=ptime:20
a=silenceSupp:off - - - -
m=image 0 udptl t38
```

Note: The response will NOT contain the same number of "m=" lines as the command, if the fax package is used in the command (since both RFC 3264 and the Fax Package conflict with each other). The Fax Package standard requires in some cases, not to include a fax "m="line in the response, even though T.38 is supported. An "m="line with a zero port is not returned (since T.38 Fax is supported).

7.1.9 MGCP Fax

7.1.9.1 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 498) – 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 503). enables negotiating the T.38 fax port without using the T.38 fax package.

Note: **The following table is NOT applicable to MediaPack.**

MGCP Fax Package Gateway Mode

Gateway CH 0	Call Agent	Gateway CH 1
200 17501 OK	RQNT 17501 ds/tr0/1@[10.4.4.129] MGCP 1.0 X: 12 S: L/dl R: D/X(D) D: 2xxx	
NTFY 2075 ds/tr0/1@[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 ds/tr0/1@[10.4.4.129] MGCP 1.0 C: 1 L: a:PCMA , fxr/fx:gw M: recvonly X: 12 R: fxr/gwfax	
	CRCX 17503 ds/tr0/2@[10.4.4.129] MGCP 1.0 C: 1 L: a:PCMA , fxr/fx:gw	200 17503 OK I: 35 v=0

MGCP Fax Package Gateway Mode

	M: sendrecv X: 12 R: fxr/gwfax v=0 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	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
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 ds/tr0/1@[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 ds/tr0/1@[10.4.4.129] MGCP 1.0 X: 12 R: fxr/gwfax	
	RQNT 17506 ds/tr0/2@[10.4.4.129] MGCP 1.0 X: 12 R: fxr/gwfax	200 17506 OK
	200 2076 OK	NTFY 2076 ds/tr0/2@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/gwfax(start)
NTFY 2077 ds/tr0/1@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/gwfax(start)	200 2077 OK	

MGCP Fax Package Gateway Mode

NTFY 2078 ds/tr0/1@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/gwfax(stop)	200 2078 OK	
	200 2079 OK	NTFY 2079 ds/tr0/2@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/gwfax(stop)
250 17507 OK	DLCX 17507 ds/tr0/1@[10.4.4.129] MGCP 1.0	
	DLCX 17508 ds/tr0/2@[10.4.4.129] MGCP 1.0	250 17508 OK

Note: **The following table is applicable to MediaPack only.**

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 O: 2580	200 2096 OK	
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	

MGCP Fax Package Loose Mode

	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 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	MDCX 16826 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=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	200 2099 OK	

MGCP Fax Package Loose Mode

O: FXR/t38(start)		
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 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	200 16828 OK 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)

MGCP Fax Package Loose Mode

200 16829 OK	RQNT 16829 ACgw0@[10.4.4.129] MGCP 1.0 X: 12 R: fxr/t38	
	RQNT 16830 ACgw1@[10.4.4.129] MGCP 1.0 X: 12 R: fxr/t38	200 16830 OK
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 ACgw0@[10.4.4.129] MGCP 1.0 X: 12 O: FXR/t38(stop)	200 2105 OK	
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
250 16831 OK	DLCX 16831 ACgw0@[10.4.4.129] MGCP 1.0	
	DLCX 16832 ACgw1@[10.4.4.129] MGCP 1.0	250 16832 OK

7.1.10 Fax Transport Type Setting with Local Connection Options

In addition to the T.38 Fax package described in 'Fax Package Definition - FXR' on page 533, the parameter, “x-faxtranstype” can set the Fax Transport Type of each connection to Transparent, Relay, Bypass 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.

Fax Transport Type

Fax Mode	Description
x-faxtranstype:transparent	Fax events are ignored
x-faxtranstype:relay	Faxes are transmitted on T.38
x-faxtranstype:bypass	Fax is transmitted with bypass coder
x-faxtranstype:transparentwithevents	Fax is transmitted in-band and fax events are detected

7.1.10.1 Fax Attributes

In case fax m-line is returned, the returned SDP contains the following T.38 attributes, as defined in the T.38 spec:

- T38FaxVersion - This will have the values 0 or 3.
- T38MaxBitRate - The value of this attribute is according to the FaxRelayMaxRate configuration parameter and can have the the following values: 2400, 4800, 7200, 9600, 12000, 14400, 16800, 19200, 21600, 24000, 26400, 28800, 31200, 33600
- T38FaxRateManagement - This attribute will always have the value "transferredTCF".
- T38FaxMaxBuffer - This attribute will always have the value 1024.
- T38FaxMaxDatagram - This attribute will always have the value 238.
- T38FaxUdpEC - This will always have the value of *T38UDPRedundancy*.

The same attributes will also be added to a T.38 capability line, when it is returned.

7.1.10.2 Fax Version and Max Bit Rate Negotiation

The device supports Fax Versions 0 and 3 over UDP and Max Bit Rate between 2400 and 33600. The Max Bit Rate is up to 14400 for Version 0.

Supported Fax Version and Max Bit Rate are negotiated between the Media Gateways using SDP T38FaxVersion and T38FaxMaxBitRate attributes. The negotiation is required in order to transfer high rate faxes like V.34 over T.38.

The following is a description of the Fax Version and Max Bit Rate negotiation and default values.

- T38FaxVersion (a= T38FaxVersion:value) negotiation
  - The Local Connection Option (LCO) does not have a T.38 version parameter, so the reply will be according to T38Version board *ini* file parameter
  - If the remote SDP doesn't contain the T38FaxVersion attribute, Version 0 will be used (backward compatibility).
  - If the remote SDP has a T38FaxVersion attribute and this version is supported, then this version will be used and replied. Otherwise Version 0 will be used.
- T38FaxMaxBitRate (a= T38FaxMaxBitRate:value) negotiation
  - The LCO does not have a fax rate parameter, so the reply will use the FaxRelayMaxRate board *ini* file parameter adjusted to the highest supported version.
  - If the remote SDP does not contain the Max Bit Rate attribute, we will work according to the LCO.
  - If the remote SDP has a Max Bit Rate attribute and bit rate is supported, then this Max Bit Rate will be used. Otherwise the highest supported Max Bit Rate will be used.

Note: The T38FaxMaxBitrate values are given according to the SDP standard - i.e., 14400, 33600 etc. The values of FaxRelayMaxRate *ini* file parameter are given according to AudioCodes internal standards – i.e., 0,1,2 etc.

FaxRelayMaxRate to T38FaxMaxBitrateValues Conversion

FaxRelayMaxRate	T38FaxMaxBitrate
0	2400
1	4800
2	7200
3	9600
4	12000
5 (Max for Version 0)	14400 (Max for Version 0)
6	16800
7	19200
8	21600
9	24000
10	26400
11	28800
12	31200
13 (Max for Version 3)	33600 (Max for Version 3)

Notes:

- Fast fax (Version 3) is supported only when the fax session is opened at the beginning of the call. Re-negotiation from Audio to Fast Fax is not yet supported.
- DSPVersionTemplateNumber value should be '10' to support Version 3.

7.1.10.3 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.

7.1.11 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 (value 2) to enable the VBD functionality.

There are 2 coders supporting Bypass and therefore can be used for VBD:

- PCMU
- PCMA

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.

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.

7.1.11.1 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.

7.1.11.2 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:

- A *gpmd* field for each of the coders:

```
L: a:codec1;codec2, md/gpmd:"codec1 vbd=yes",
                    md/gpmd:"codec2 vbd=yes"
```

- One *gpmd* field for all of the coders:

```
L: a:codec1;codec2, md/gpmd:"codec1 vbd=yes" ;
   "codec2 vbd=yes"
```

- Reference to a specific coder if same coder appears a number of times:

```
L: a:codec1;codec1, md/gpmd : "codec1:2 vbd=yes".
(The vbd coder is the second codec1).
```

Note that it is possible to use the "gpmd" attribute without the package "md" prefix , that is replacing all occurrences of "md/gpmd" with "gpmd" alone.

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 8 a=rtpmap:96 PCMA/8000/1 a=gpmd:96 vbd=yes (The whole 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 8 a=rtpmap:96 PCMA/8000/1 a=gpmd:96 vbd=yes a=rtpmap:97 PCMU/8000/1 a=gpmd:97 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 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
CRCX 29747 ACgw1@[10.4.4.123] MGCP 1.0 C: 1 M: recvonly L: a:PCMA,md/gpmd:"PCMA:1 vbd=yes"	534 29794 FAIL (There must also be one voice coder which doesn't support VBD.)

7.1.12 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.

7.1.13 TGCP Compatibility

To use Trunking Gateway Control Protocol (TGCP) conventions, the user must set the device to the TGCP profile, e.g., turn on the 6th bit (value 32) of the *MGCPCompatibilityProfile 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.

7.1.14 TDM Hairpin

Hairpin connection occurs when two TDM endpoints are connected in one call without the outside packet network involved.

In the MGCP standard, there are two methods to achieve this connection:

7.1.14.1 TDM Hairpin By Using "Z2"

The MGCP "Z2" attribute, in which the second endpoint is defined, enables two endpoints to be connected by using a single CRCX command.

The connection can be done 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 a 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.

Another option is to connect the endpoint by using soft IP loopback. In this option, the voice is sent between the two endpoints by an internal RTP connection. The RTP is not sent to the outer network. The default coder used for this method is PCMU. Activating it is the same as above, without adding the "clearmode" coder.

Example:

```
CRCX 19278 ds/tr0/1@[10.31.4.93] MGCP 1.0
M: sendrecv
z2:ds/tr0/2@[10.31.4.93]
```

7.1.14.2 TDM Hairpin By Using NT:LOCAL

When using the "LOCAL" network type, three commands are needed to create a connection between the two endpoints, as shown in the example below.

Two connection options are available using this method:

- Soft IP loopback, in which events and signals can be applied to each endpoint.
- TDM level connection, in which the two endpoints are connected in the TDM side, and no data processing in available. This option is enabled by setting the *MegacoTdmHairPinningMode* configuration file parameter to 1.

Example:

```
CRCX 15995 ds/tr0/1@[10.4.4.46] MGCP 1.0
C: EC6F2B0
L: e:off,nt:LOCAL
M: sendrecv
R:
S:
X: 2308CC87
```

```
200 15995 OK
I: 21

v=0
c=LOCAL EPN ds/tr0/1
m=audio 0 LOCAL 0

CRCX 15996 ds/tr0/2@[10.4.4.46] MGCP 1.0
C: EC6F2B0
L: e:off,nt:LOCAL
M: sendrecv

v=0
o=- 1441227873 0 LOCAL EPN ds/tr0/1
s=-
c=LOCAL EPN ds/tr0/1
t=0 0
m=audio 0 LOCAL 0


200 15996 OK
I: 22

v=0
c=LOCAL EPN ds/tr0/2
m=audio 0 LOCAL 0

MDCX 15997 ds/tr0/1@[10.4.4.46] MGCP 1.0
C: EC6F2B0
I: 21
L: e:off,nt:LOCAL
M: sendrecv
R:
S:
X: 2308CD09

v=0
o=- 820074714 0 LOCAL EPN ds/tr0/2
s=-
c=LOCAL EPN ds/tr0/2
t=0 0
m=audio 0 LOCAL 0

200 15997 OK

v=0
c=LOCAL EPN ds/tr0/1
m=audio 0 LOCAL 0
```

7.1.15 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.

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 %
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, :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.

MGCP AMR Alert Policy

Gateway	Call Agent
200 4 OK	mdcx 4 ds/01/05@TPM0Slot6.M5K MGCP 1.0 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 rqnt 5 ds/01/05@TPM0Slot6.M5K MGCP 1.0 X:2 R:r/x-qac

MGCP AMR Alert Policy

Gateway	Call Agent
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

7.1.16 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.

7.1.16.1 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”.

7.1.16.2 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.

7.1.16.3 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.

7.1.16.4 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 511.) 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.

7.1.16.5 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.

7.1.16.6 Conference Configuration

7.1.16.6.1 *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.

7.1.16.6.2 Conference Parameter Package

Package Name: cnf

Version: 0

Local Connection Option Parameters:

While using this package, the user can set each conference call properties individually.

- MaxConfusers - defines the maximum number of the conference parameters to be reserved for this conference call. 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.

  Maxconfusers Number = 3 to 64

- Confusertype - defines the type of the conference participant

  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.

Format Example:

L: cnf: maxconfusers =number confusertype= confusertypevalue

L: cnf: maxconfusers =number, cnf: confusertype= confusertypevalue

7.1.16.7 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.

7.1.16.7.1 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
```

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
```

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
```

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
```

7.1.17 CALEA (Communications Assistance for Law Enforcement Act)

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 affected while Electronic Surveillance is performed. The feature is described in the PacketCable specification, Appendix H of 'http://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
```

7.1.18 RTP Media Encryption - RFC 3711 Secure RTP

The SRTP (RFC 3711) media encryption standard is supported according to the following description.

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.

Note: SRTP support is according to the selected DSP version template.

7.1.18.1 Supported Suites

SRTP implementation is limited to the following four suites:

- AES_CM_128_HMAC_SHA1_32
- AES_CM_128_HMAC_SHA1_80
- ARIA_CM_128_HMAC_SHA1_80
- ARIA_CM_192_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.

7.1.18.2 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.

7.1.18.3 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 *AriaProtocolSupport* parameter defines ARIA encryption algorithm support when set to Enable, e.g. AriaProtocolSupport = 1.
5. ***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.
6. ***ini* file parameter** – The *RTPEncryptionDisableTx* parameter can be set in order to work with non-encrypted RTP.
7. ***ini* file parameter** – The *RTCPEncryptionDisableTx* parameter can be set in order to work with non-encrypted RTCP.
8. ***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.

7.1.18.4 SRTP Local Connection Option Format

Use of SRTP LCO parameters is described below, in Secured Connection Negotiation.

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, ARIA_CM_128_HMAC_SHA1_80, ARIA_CM_192_HMAC_SHA1_80, SRTP_SUITE_NULL

7.1.18.5 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.

The length of key string should be exactly 40 symbols for 128 bit encryption suites and 51 symbols for 192 bit encryption suites.

The only 192 bit encryption suite is ARIA_CM_192_HMAC_SHA1_80.

Suites which have a key string with invalid length will be ignored by the SDP parser.

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;G72
7_24;G727_32_16;G727_32_24;G727_32;G727_40_16;G727_40_24;G727_40_3
2;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;
ARIA_CM_128_HMAC_SHA1_80;ARIA_CM_192_HMAC_SHA1_80;SRTP_SUITE_NULL
```

7.1.18.6 Secured Connection Negotiation

The examples below show the creation of a secured connection via CRCX and MDCX commands.

7.1.18.6.1 SRTP Negotiation

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/SAVP 8
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=crypto:3 ARIA_CM_128_HMAC_SHA1_80
inline:MKHEBFC/PMKHEB+CJfvspnkheifcZW
a=crypto:4 ARIA_CM_192_HMAC_SHA1_80
inline:9630xvspsqnkhecZEC/8520xurolpmugffdfer45984
```

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/SAVP 8
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
```

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/SAVP 8
a=crypto:1 AES_CM_128_HMAC_SHA1_32
inline:EbYVSPNKNLIFC/966j030xvspmjheh
```

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/SAVP 8
a=crypto:1 AES_CM_128_HMAC_SHA1_32
inline:EbYVSPNKNLIFC/966j030xvspmjheh
```

Update succeeded.

```
200 15940 OK
```

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/SAVP 8
a=crypto:1 AES_CM_128_HMAC_SHA1_32
inline:NLIFC/QNKHEB/8520tqtrolifcaXnk
```

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/SAVP 8
```

```
a=crypto:1 AES_CM_128_HMAC_SHA1_32
inline:pA9631yvspnkhebZKPMJGEB+OLJGDA
```

7.1.18.6.2 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.

7.1.18.6.3 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].
```

7.1.18.6.4 SRTP "Crypto" Attribute Grammar

Below is an Augmented BNF [RFC 2234] 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" /
                          "AES_CM_128_HMAC_SHA1_80" /
                         "ARIA_CM_128_HMAC_SHA1_80" /
```

```
                          "ARIA_CM_192_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
("-")
```

7.1.19 MGCP Coder Negotiation

7.1.19.1 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 4566. 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 779 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.

7.1.19.2 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.

7.1.19.3 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 752). 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.

7.1.19.4 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.

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
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

7.1.20 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 interface.

7.1.20.1 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.

7.1.20.2 Generic Media Package - G

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

7.1.20.3 DTMF Package - D

DTMF Package - D

Symbol	Definition	R	S	Duration
0	DTMF 0	x	BR	
1	DTMF 1	x	BR	
2	DTMF 2	x	BR	
3	DTMF 3	x	BR	

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

7.1.20.4 Line Package - L

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	
oc	Report on completion of	x		

	TO			
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 only.**

7.1.20.4.1 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.

7.1.20.4.2 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.

7.1.20.5 Handset Emulation Package - H

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		

7.1.20.6 Trunk Package - T

Note: Trunk Package - T is NOT applicable to **MediaPack.**

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		

7.1.20.7 PacketCable (NCS) Line Package - L

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		
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	

7.1.20.8 Announcement Package - A

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		

7.1.20.9 RTP Package - R

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(4000)
```

Notes: **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.

7.1.20.10 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)

7.1.20.10.1 MF FGD Operator Services Package - MO

MF FGD Operator Services Package - MO

Code	Description	Event	Signal	Additional Info
ans	Call Answer	P	-	
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	-	

7.1.20.11 ISUP Trunk Package - IT

Note: The ISUP Trunk Package - IT is NOT applicable to **MediaPack.**

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		-	
of	Operation failure		-	
ro	Reorder tone	-	TO	Time-out = 30 sec
rt	Ring back tone	-	TO,C	Time-out = 180 sec

7.1.20.12 Media Format Parameter Package - FM

Supported FMTP Formats

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

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

Redundancy

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

AMR Family

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

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)

7.1.20.13 Fax Package Definition - FXR

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

7.1.20.14 Conference Package - CNF

Note: The Conference Package - CNF is only applicable to **IPmedia**.

Conference Package - CNF

Symbol	Definition	R	S	Duration/Comment
maxconfusers	Maximum number of conference participants			local connection option parameter
confusertype	Conference participant type			local connection option parameter

7.1.20.15 Extended Line Package - XL

Extended Line Package - XL

Symbol	Definition	R	S	Duration/Comment
rev	activates or switches off line reversal on an endpoint	x	TO	Infinite

7.1.20.16 V5 Package Definition X-v5

Note: The following V5 Package Definition table is NOT applicable to **MediaPack.**

V5 Package Definition

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

7.1.20.17 Base Audio Package - BAU

Note: The following Base Package Definition table is only applicable to **IPM.**

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.

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

7.1.20.18 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.

Signal List Package Definition

Symbol	Definition	R	S	Duration/Comment
oc	Operation complete	x		
of	Operation failed	x		
s(list)	Signal List		TO	variable

7.1.20.19 NCS V5 SCN Line Package - E (Applicable to MediaPack only)

Note: The following NCS V5 SCN Line Package definition is 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.

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 rep=1 (one repetition).
cr(parm)	Cadence Ringing		TO	parm is the Cadence Ringing index, which can be 0-7.

7.1.21 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 752 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).

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”

Compression Coders

Coder	MGCP Textual Name
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”
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"

Compression Coders

Coder	MGCP Textual Name
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"
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
```

7.1.22 Connection Statistics (CDR)

Call Detail Report can be generated when a voice call terminates.

The generation can be either to the Syslog server or to the internal file which can be viewed in the CLI or transferred to an NFS hosting 'cdr send' command shell command. For more information refer to Call Detail Reports (CDR) Commands.

The *CPConnectionStatistics ini* file parameter, or 'cdr start' command shell command starts generating CDR records.

Record Format and Fields Example:

```
CALL_STATISTICS:
    Endpoint Name: ds/Tr0/1,
    Connection deleted by Call Agent,
    Coder: PCMA,
    ConnectionID: 21,
    CallId: 00004d760000052f0000000,
    Call duration: 2 seconds,
    Local RTP address: 10.4.4.35 port 5440,
    Remote RTP address: 10.4.4.35 port 5410,
    Tx/Rx bytes 22720/22720,
    Tx/Rx packets 142/142,
    Call start time: 2000/01/01 01:30:36,
    DSP device: 0,
    Packetization: 20 ms,
    Packet Loss: 0,
    Call Type: Voice, Fax Type: NA, Pages: NA
    Connection Mode: sendrecv,
    Jitter: 0 ms,
    Echo Canceller: On, length 128 ms,
    Call number 1 closed
```

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).

Call Type possible values are 'Voice' or 'Fax'. When the Call Type is 'Fax', Fax Type may be either 'T38' or 'Other'. Otherwise the Fax Type and Pages value is 'NA'. Pages field is applicable for T38 Fax calls only. In all other cases, the value is 'NA'.

7.1.23 Disabling the Delete Connection Functionality from the Gateway Side

The *DisableDLCXByGW ini* file parameter, 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.

7.1.24 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.

In order to use the RTCP-XR feature, it should be enabled in the feature key. For TP-6310/TP-8410 blades, the VQMONENABLE *ini* file parameter should be set to a value greater than 0.

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.

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		

RTCP XR Example Flow

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 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	

RTCP XR Example Flow

	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, 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	

RTCP XR Example Flow

	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

7.1.25 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

7.1.26 DigitMap Special Handling

7.1.26.1 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 length, then an Error Message is generated.

7.1.26.2 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).

7.1.27 Digest Authentication

Note: The following sub-section on Digest Authentication is only applicable to **MediaPack**.

7.1.27.1 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.

7.1.27.2 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="SdytTxTy
REBm0GvEMLcyO6Ei9iKRneGL"
```

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="/Jb1RTsT
DYkKHNuuMiCaUO5AeDb9Ekky9p59",uri="MGCP",qop=auth-
int,nc=00000001,cnonce="",response="5171b9dca748868813ce004b147c96
25",opaque="SdytTxTyREBm0GvEMLcyO6Ei9iKRneGL"
```

The call agent validates the gateway.

```
200 2019 OK
```

7.1.27.3 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="SdytTxTy
REBm0GvEMLcyO6Ei9iKRneGL"
```

And the gateway replies with:

```
X+Authorization: Digest
username="[10.4.4.138]",realm="actestvoiceservice",nonce="/Jb1RTsT
DYkKHNuuMiCaUO5AeDb9Ekky9p59",uri="MGCP",qop=auth-
int,nc=00000002,cnonce="",response="adf6df39a6d0ac44fc57b4096542f9
da",opaque="SdytTxTyREBm0GvEMLcyO6Ei9iKRneGL"
```

7.1.28 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 107.
- **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 485.

7.1.29 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).

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

MGCP Compliance Matrix

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		
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		

MGCP Compliance Matrix

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	
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**			

MGCP Compliance Matrix

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		
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		

MGCP Compliance Matrix

69	ObservedEvents	Yes		
70	**Create Connection command**			2.3.5
71	CallID	Yes		
72	Endpoint	Yes		
73	NotifiedEntity	Yes		
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		

MGCP Compliance Matrix

92	S:	Yes		
93	Encapsulated Endpoint Configuration	Yes		
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**			

MGCP Compliance Matrix

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 (from gateway)	Yes		2.3.8
133	CallID	Yes		
134	EndPointID	Yes		
135	ConnectionID	Yes		

MGCP Compliance Matrix

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		
152	NotifiedEntity	Yes		
153	Connection Identifiers	Yes		
154	DetectEvents	Yes		
155	ObservedEvents	Yes		
156	EventStates	Yes		

MGCP Compliance Matrix

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		
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		

MGCP Compliance Matrix

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 & return code)	No		
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	

MGCP Compliance Matrix

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	
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	

MGCP Compliance Matrix

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	
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	

MGCP Compliance Matrix

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	
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		

MGCP Compliance Matrix

258	Receiving Duplicated transaction IDs	Yes		
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		
288	**Security requirements**			5
289	MGCP IP security (RFC 1825)	No		

7.2 MEGACO (Media Gateway Control) Protocol

Note: The following section on MEGACO is NOT applicable to MediaPack and the TP/IPM-260 product family.

7.2.1 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.

7.2.2 Gateway Operation

Virtual Gateway (VGW) is defined in H.248.1 (Refer to H.248.1, Section 11.1 "Multiple Virtual MGs"). One physical gateway can appear to the outer world (the call manager) as a set of different gateways. This means that each such virtual gateway registers to its controlling MGC by sending a serviceChange. Subsequently, the VGW responds to MGC commands and sends Event notifications if needed independently from the other VGWs.

Note: GEN3 supports up to three Virtual Gateways (VGW). Mediant 1000, Mediant 2000 and IPM support only one gateway. MP-1xx supports up to two VGWs.

- Each VGW can be controlled by a different set of MGCs. Each VGW has its own individual MGC list (an Active MGC and several Redundants) and each VGW maintains its own connectivity state.
- Each VGW might be connected to different control networks/VLANs and may work with a different transport type.
- Each VGW may have its own pre-defined/default media network/VLANs.

The trunks are divided between the VGWs according to the configuration. The user may configure some VGWs with trunks and others without trunks (for IP to IP calls only). In analog gateways, each VGW has the list of the analog ports connected to it.

All system resources except Trunks and Analog ports are common for all VGWs (resources such as: context, ephemeral terminations, DSP, conference resources, CPU etc). As a result Context and ephemeral terminations naming are common and sequential between the VGWs.

This means for example, that when a context is opened, one VGW will have Context ID 1, the next context opened on a second VGW will have Context ID 2, and so on.

The Transaction ID is unique to each VGW. The CPU resource is common to all the VGWs. This means that the CPU handles a command from the first VGW, and then from the second and so on (the commands are not executed in parallel). The user may configure load weight to each VGW. The load weight is the proportion of commands processed by each VGW.

The VGW Configuration Table replaces some parameters from the Web Interface General Parameters page:

- *ServiceChangeProfile* replaces the *CPServiceChangeProfile ini* file parameter.
- *MegacoVersion* has been moved from General Parameters page.
- *MID* replaces the *MEGACO_MID ini* file parameter.

Up to 10 MGCs can be configured for each VGW.

Each VGW is identified either by a different local address (a combination of an interface, and a local port) or by a different MGC (defined by transport type, MGC address and MGC Port),

The combination of the following parameters from VGW and MGC tables should have uniqueness for Transport Type, Local Port, IP Interface Name, MGC address and MGC port.

Note: When using SCTP TransportType, the VGWs should differ by port.

The MGC Address format should match the IP Interface Name at VGW configuration table. For example, if IPv4 is set to "MG Controller Address Format ", IPv4Interface Name must be set in the VGW configuration table.

The MGC Table replaces some parameters from the Web Interface Basic Configuration page:

- *MGControllerAddress* replaces the *ProvisionedCallAgents ini* file parameter.
- *MGControllerPort* replaces the ProvisionedCallAgentsPorts *ini* file parameter.
- *TransportType* replaces the *CPTransportType ini* file parameter.

Notes:

- Backward compatibility is maintained.
- If the VGW Configuration and MGC tables are not configured, a configuration will be constructed automatically from the old parameters.
- When the MGC address is configured as DNS, only a single VGW can be configured. Any other VGW can not be operational.
- Administrative operations are performed for the whole gateway and can not be applied to a specific VGW.
- V5.2 Access Gateway and the AMS Feature Package support only one VGW.

For more information on VGW, refer to the Web Interface section of the decice's User Manual.

7.2.2.1 Executing MEGACO Commands

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

Several commands that support the basic operations are 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.

7.2.2.2 KeepAlive Notifications From the Gateway

The keep-alive notifications from the gateway to the MGC are implemented either using an AudioCodes proprietary mechanism via a NOP ServiceChange command (controlled by *ini* file parameters), or using the standard Inactivity Timer Package (H.248.14).

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

- *KeepAliveEnabled* - activates or de-activate the keep-alive function
- *KeepAliveInterval* - defines the inactivity period in seconds

If the KeepAlive mechanism is enabled, the device sends a NOP ServiceChange command when it detects a defined period without commands from the MGC. The *no op service* change is sent on an undefined termination.

Example:

```
MEGACO/2 [10.4.5.125]:2944
T=27882{
C = - {
SC=UNDEFINED{
SV{
MT=RS,RE="900 Service Restored"}}}}
```

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.

The standard inactivity timer package (H.248.14) is fully supported. The activation is done 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.

7.2.2.3 Loss of H.248 Connectivity

Loss of H.248 connectivity (caused by 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”.

The *DisconnectBehavior INI* file 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).

7.2.2.4 Setting MEGACO Call Agent IP Address and Port

Users can provide the device with up to 10 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.

7.2.2.5 Authorization Check of Call Agent 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:

- No authorization check is performed. This mode specifies that every command is accepted and executed.
- 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 *MEGACOCheckLegalityOfMGC ini* file parameter, for which the default value is 1.

7.2.2.6 "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.

7.2.2.7 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 cpTransportType *ini* file parameter 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 reply with an Error Code 533.

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.

7.2.2.8 Support of DiffServ Capabilities

The Media Gateway can support the Quality of Service (QoS) for control flow by providing the default Differentiated Service Code Point (DSCP) value.

The range of the DiffServ parameter is between 0 and 63. The value of the control path is controlled by the *PremiumServiceClassControlDiffServ ini* file parameter, with a default value of 40. If this parameter is not set, and the old parameter *ControlDiffServ* is set, the old parameter is used.

7.2.2.9 Overload Report

Note: The Overload Report section is only applicable to TP-6310, TP-8410, IPM-6310, IPM-8410 and Mediant 3000 devices.

Gateway support overload reporting can be done by using either:

- Detail Congestion Report (DCR) package as defined in H.248.32
- Congestion Handling package (CHP) as defined in H.248.10
- Overload Control Package (OCP) as defined in H.248.11

These packages enable the MGC to control the Media Gateway load. Based on the report received from the Media Gateway, the MGC may take corrective action to improve the efficiency of the entire system.

7.2.2.9.1 Congestion Handling Package

The package makes it possible for the Media Gateway to control its load. It enables the Media Gateway to request that the MGC starts or finishes load reduction by raising an event. The event is notified when CPU usage is over 80% or returns back to less than 70%. When CPU usage is over 80%, this event is notified with a reduction value equal to 10. It represents that the MGC is requested to block (reject or deflect) 10% of the calls. When the CPU usage is reduced to less than 70%, the event with a reduction value equal to 0 is notified. It means that no reduction shell be applied to the Gateway and the Gateway will return to its normal functionality.

7.2.2.9.2 Detail Congestion Report Package

This package definition allows the Media Gateway to report the current state of its resource usage to the MGC. The resources whose usage may be reported with this package include:

- General Resources – CPU and MEGACO Socket Buffer usage
- Digital Signal Processor (DSP) resources – DSP usage
- Internet Protocol (IP) resources – Media IP resources usage
- Asynchronous Transfer Mode (ATM) – ATM resources usage
- Extension Resources – AudioCodes defined ext1 resource as Conference DSP resources – Conference legs. usage

Note: When working in full CAS mode, in which all channels are opened automatically, the DSP resource is irrelevant.

The MGC can determine at which threshold level(s) the Media Gateway sends reports. The threshold values are defined as percentages. The MGC can also determine when periodical reports are issued, i.e., in the absence of any change in threshold level, a report is issued at a defined interval (optional).

Note: The threshold set for a resource is limited to 3 thresholds. This threshold setting should be in an ascending order.

Based on the Resource Usage report received from the Media Gateway, the MGC may take corrective action to improve the efficiency of the entire system (e.g., re-route calls, reduce possible message intensive audits, throttle calls, etc.). Once the event is enabled, the Media Gateway is expected to report the corresponding resource usage observed any time a threshold is crossed (either escalating, or deescalating).

To avoid a flood of messages when the resource usage oscillates across a threshold level, a hysteresis is employed within the Media Gateway. This hysteresis is determined via the *PMCongestionHysteresis* ini file parameter. The default value is 2%. This hysteresis allegedly creates “low” and “high” values for each threshold from the one to three thresholds received at setup. For example, for the default hysteresis value (2%) and the threshold 50%, a report is generated at 52% and 48%.

The following example shows a command that requests reports for 2 resources. Each resource has its own threshold setting. An interval report is requested. When the threshold has crossed or the interval expires, a report is generated.

```
MEGACO/1 [172.16.8.88]
 T = 31307{
 Context = -{
 Modify = root{
   Events = 12345 {dcr/conrep {eresname = [gen, dsp],
                               rptthresh  = [0, 10, 50,  0, 20],
                               rptint  =  10}}}
 }}
```

Note: Although the resources appear in the SNMP MIB, these parameters are under the jurisdiction of MEGACO. This means that SNMP cannot update or change them.

7.2.2.10 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 20 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 when the event is detected.

7.2.2.11 Playing Signals

Signals requests 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.

General Signal Combination Options

Group 1	Notes	Group 2	Notes
ToneGen/*	Including all the inheriting packages	al/ri	
an/*		alert/ri	
ct/*		alert/rs	
alert/cw	Only when analog device	xal/*	
andisp/*	Only when analog	gb/*	(3G)

General Signal Combination Options

	device		
aasb/*	Including all the inheriting packages	bt/*	(3G)
nttrk/*			
ctyp/*			

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).

7.2.2.12 Support Profiling

Profiling of various MEGACO functionality is controlled via the *MGCPCompatibilityProfile* ini file parameter. Initially, only value 2 has been supported. (Value 0 is obsolete). Value 1 and 2 are the same and are used for supporting MEGACO version 1. Value 2 is the default value.

- Bit 2 (Value 4) – Enables the following features:
  - Controls the type of support for Fax T.38 negotiation (refer to 'Fax T.38 and Voice Band Data Support (Bypass Mode)' on page 596 and controls the type of RFC 2833 negotiation (refer to 'RFC 2833 Support' on page 591).
  - Uses the packetization period of the voice for Bypass Fax. If this bit is not set, the packetization period for the fax bypass is taken from the *ini* file parameters.
  - Uses the G.711 Coder as the Bypass coder as well.

  This bit has been deprecated as of Version 5.2 and replaced by Bit 3 (Value 8) and Bit 7 (Value 128) of the *CPSDPProfile* parameter. (It will 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 589.) This bit has been deprecated as of Version 5.2 and replaced by Bit 4 (Value 16) of the *CPSDPProfile* parameter. (It will 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.)

- When sending a notification transaction request, the device does not mark it as optional.
- When responding to a command with a wildcarded termination (*), the command will be handled as if the command was wildcarded,
- The SDP negotiation does not fail on illegal PTIME when more than 2 coders configured in SDP. This functionality has been deprecated as for Version 5.4 and replaced by Bit 11 (Value 1024) of CPSDPProfile parameter. (It is to be removed in a later software version.)

- Bit 6 (Value 64) – Controls the number of the context replied to the context=*{av=root{audit{}}} command. When this bit is turned on, the number the context list length is limited to 69.
- Bit 7 (Value 128) – Enables the permanent off-hook indication to the MGC. For more information refer to Permanent off-hook indication in the PSTN - V5.2 chapter. This profile is applicable on Access Gateways only.
- Bit 8 (Value 256) – For MFCR2 implementation, sends the dialed number without the HASH (#) mark and without double quote.
- Bit 9 (Value 512) – Enables sending only one notification in a message.
- Bit 10 (Value 1024) – For CAS implementations, sends the block and unblock notifications even if the event was not requested by the MGC. The event number will be 0.
- Bit 11 (Value 2048) – When the gateway sends a FailOver service change, it will use Reason 909 (MGC Impending Failure) instead of Reason 907 (Transmission Failure).
- Bit 12 (Value 4096) – Prevents printing the RTPMAP attribute in the SDP part of the MEGACO reply.
- Bit 14 (Value 16284) – Configures whether NULL termination should be added at the end of the MEGACO message or not. When this profile is ON, No NULL termination is added to the end of the MEGACO message.
- Bit 15 (Value 32768) – When this bit is set, the notification messages will not be marked as optional.
- Bit 17 (Value 131072) – Sets the DSCP value in reversed order according to clause B.3 in H.248.1.
- Bit 18 (Value 262144) – For CAS implementation, report the basic CAS events with the BCAS package even if the event descriptor requested an inheriting package such as ICAS.
- Bit 19 (Value 524288) – This bit prevents cleaning the events from a physical termination being subtracted from a context.

7.2.2.13 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.

7.2.2.13.1 Termination Name Patterns

Each termination type name pattern is defined by an *ini* file or SNMP parameter. The pattern may contain acceptable characters as defined in MEGACO. The '*' and '%' characters are safe characters which are used to represent where a digit or name should be appear. Therefore, it cannot be part of the name itself. All other characters including '/', except for the character after '%' which is used as the delimiter, are considered text and can be used as part of the termination name pattern.

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.

Physical terminations can be structured to up to 5 groups, when the minimum is 2 FixedPrefixes/TrunkIDs/ChannelIDs. Other groups are used for hierarchical grouping of the trunks and can be used, for example for DS3 numbering of the trunks, i.e. FixedPrefix/DS3/TrunkID/ChannelID.

Terminations structure is set using the *ini* file parameters:

PHYSTermNamePattern describes the structure of the pattern. e.g., a 3 level hierarchy can be set as: gwDS3/*/*/*. The extreme right '*' character is used for channel numbering, while the other '*' characters are used for hierarchical grouping of the trunk numbering.

EP_Num_0 - EP_Num_4 specifies the start index within each group. It goes from the higher hierarchical group (e.g., trunks) to the lower (e.g., channels).

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.

Example of Setting *ini* File Parameters for DS3 Numbering of Trunks:

```
PHYSTermNamePattern=gwDS3/*/*/*
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
```

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 terminations can be structured into either:

- 2 fields - FixedString/Id
- 3 fields - FixedString/InterfaceName/Id

where **InterfaceName** is a logical interface to a network expressed as an alphanumerical character string and **Id** is the termination specific identifier.

The starting number for the RTP terminations is controlled by RTP_NUM (The default value is 0).

RTP termination naming example:

- gwRTP/* - 2 level termination fields pattern
- gwRTP/%/* - 3 level termination fields pattern

The '*' indicates where the digit should be placed and '%' character indicates where the string should be placed.

With the 3 level termination fields, the MGC should always use CHOOSE in the *InterfaceName* field in the ADD command. For example:

```
gw/$/$
```

If not, the Media Gateway will reply with an error descriptor using Error Code #501.

In the current implementation, the termination interface name is selected according to the realm this termination is associated with. For more information about the realm and termination media IP address allocation refer to 'Control Protocol Media Realm Table' on page 587.

The ADD command can be received either with or without the realm identifier. If the realm is explicit in the MEGACO command, the realm name will be returned as the interface. This can be done only if the user loaded the blade with the *ini* file that contains the Interface table and realm table. Otherwise, a default realm will be returned as the interface.

Notes:

- The following patterns at the command are not supported - gw/RTP/interface/$ gw/RTP/$/id, gw/RTP/$/*.
- The character following the '%' is the delimiter and cannot appear as part of the termination string.

7.2.2.13.2 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".

7.2.2.14 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.

If the Media Gateway Controller replies with no parameters at all, this implies support of the negotiated version.

```
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.

7.2.2.14.1H.248.1 V2 - Main Changes

These are the main changes of MEGACO 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.

7.2.2.14.2H.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 }}}}
```

7.2.2.15 Management Commands

Management functionality (i.e. Locking / Unlocking the device with either Graceful or Forced method) can be implemented via the H.248 protocol, using the ServiceChange message on the ROOT termination sent from the call agent.

Note: Management functions can also be performed using the Web Interface. Refer to the Web Interface section of the device'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)

7.2.2.15.1 Graceful Shutdown

Graceful shutdown is performed when MT = Graceful (GR). Graceful Shutdown timeout is activated with the delay set in the DL and it is a non-zero value. If DL is absent or set to zero, the delay value is considered to be infinite.

7.2.2.15.2 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 dependent 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 and 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
```

7.2.2.15.3Restart

To restart the device, set the MT field to Restart (MT = RS). If the RE field is not equal to 918 (Cancel Graceful), the gateway will reset.

However, if the device is in a locked state (either by a MEGACO command or from a management interface), it can be unlocked by sending a Service Change message with the MT field set to Restart (MT = RS), and the RE field is set to 901 (Cold Boot) or 902 (Warm Boot).

7.2.2.15.4Force 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.

7.2.2.15.5Configurable 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 is configurable using the *cpServiceChangeProfile ini* file parameter: (Type: string, max length: 63 chars, default value: "TGW"). The parameter can also be configured using the Web. Each VGW has it's own value of the profile parameter in the VGW table.

7.2.2.16 Call Detail Report (CDR)

A Call Detail Report can be generated when a voice call terminates.

The generation can be either to the Syslog server or to an internal file, which can be viewed in the CLI or transferred to an NFS host using the 'cdr send' command shell command. For more information refer to Call Detail Reports (CDR) Commands.

The *CPConnectionStatistics ini* file parameter, or 'cdr start' command shell command enables generating CDR records.

Record Format and Fields Example:

```
CALL_STATISTICS:
    Terminations:  gws0c1 gwrtp/0,
    Connection deleted by Call Agent,
    Coder: PCMU,
    ContextID: 1,
    Call duration: 5 seconds,
    Local RTP address: 10.4.4.35 port 4000,
    Remote RTP address: 10.4.4.36 port 4000,
    Tx/Rx bytes 22880/22880,
    Tx/Rx packets 143/143,
    Call start time: 2000/01/01 01:35:10,
    DSP device: 0,
    Packetization: 20 ms,
    Packet Loss: 0,
    Call Type: Voice, Fax Type: NA, Pages: NA
    Connection Mode: sendrecv,
    Jitter: 0 ms,
    Echo Canceller: On, length 128 ms,
    Call number 1 closed
```

Notes:

- A CDR record is generated when the **Subtract** command is received by a RTP termination. No CDR is generated when a **Subtract** command is received by a physical termination.
- Call Type possible values are 'Voice' or 'Fax'. When the Call Type is 'Fax', Fax Type may be either 'T38' or 'Other'. Otherwise the Fax Type and Pages value is 'NA'. Pages field is applicable for T38 Fax call only. In all other cases, the value is 'NA'.
- The Tx/Rx bytes and packets contain an informative value only if **Subtract** command was sent with an audit statistics descriptor. Otherwise, the value is 0.
- DSP Device and Echo Canceller - On TDM-to-IP calls, this field will contain an informative value only when the RTP termination is subtracted before the physical termination. On other calls, the field will be informative if a DSP resource was allocated for this call. If the field is not informative, the 'NA' value is printed.
- The Terminations field will contain the physical termination name, in addition to the subtracted RTP termination, only if RTP termination is subtracted before the physical termination.
- CDR records are not generated on the Force Lock.

7.2.3 Feature Operation

7.2.3.1 Call Progress Tone Signals

In order to use Call Progress Tone generation signal packages the tones must be defined by the user in a Call Progress Tones (CPT) file. An off-line utility (DCONVERT) is supplied to convert this file from a text form to a binary file. Each tone has a toneId in the file, used by MEGACO when playing the signal. For more information about the CPT file, refer to Call Progress Tone and User-Defined Tone Auxiliary Files section and Process Call Progress Tones File(s). 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 the following signals by default:

- Dial tone
- Ringing tone
- Busy tone

When in play command the duration parameter is missing the default value is used. The default value is taken from the CPT file. When the value in the CPT file is -2, MEGACO hard coded default is used according to the “MEGACO Call Progress Tone Signals table”

The following table describes the correlation between MEGACO signals and CPT File IDs, default signal type and default tone duration.

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
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

MEGACO Call Progress Tone Signals				
Symbol	**Definition**	**Type**	**Duration**	**CPT ID**
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
bcg/bcw	Call Waiting tone	BR	1 sec	9
bcg/bcr	Caller Waiting tone	TO	180 sec	15
carr/cdt	Carrier Dial tone	BR	0	216
carr/ans	Carrier Answer tone	BR	0	217

MEGACO Call Progress Tone Signals				
Symbol	**Definition**	**Type**	**Duration**	**CPT ID**
carr/chg	Carrier Charging tone	BR	0	218
carr/ldi	Long Distance Indicator tone	BR	0	219
prectn/preconf	preset conference notification tone	BR	0	42
prectn/pcprec	preset conference precedence notification tone	BR	0	41
prectn/precrt	precedence ringing tone	TO	180 sec	44
prectn/preempt	pre-emption tone	BR	0	43
conftn/enter	conference entrance tone	BR	0	33
conftn/exit	conference exit tone	BR	0	34
conftn/lock	conference lock tone	BR	0	35
conftn/unlock	conference unlock tone	BR	0	36
conftn/timelim	a time limit warning tone	BR	0	37
alert/cw	Enables playing up to 4 distinctive patterns of call waiting. Note that if no distinctive call waiting is loaded, the default tone (9) will be played.	BR	1 sec	17, 30, 31, 32 mapped to pattern 1,2,3,4 accordingly. Or 9 if pattern 1 was requested and none of the above tones are loaded.

7.2.3.2 Announcement Signals

In order to play announcement signals by using AN package, voice prompt (VP) file should be prepared offline by users and downloaded to the device. For more information refer to Process Voice Prompts File(s) section.

The announcement value parameter of the an/apf signal (an/apf{an=2}) is directly mapped to the Voice Prompt ID.

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}
             }
     }
  }
}
```

7.2.3.3 Digits Collection Support

The following digit collection methods 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.
- **Collection according to a pre-defined digit map -** This option uses the basic and advanced digit map collection events as described in the previous section. Digit Map can be defined by the following INI file parameters: *DigitMapName* and *DIGITMAPPING* and GWC should use the *DigitMapName* in the event descriptor, in order to activate it.
- **Collection according to a pre-defined dial plan** - This option uses the basic and advanced digit map collection events as described in the previous section, but instead of supplying the digit map data in the command itself, a digit map name only is used. This name refers to an entry in the auxiliary file "Dial Plan" (Refer to 'Dial Plan File' on page 757 for more information.). The collection timers in this case cannot be configured in the command itself, as they are part of the digit map value. Therefore, the application uses the pre-configured timers defined in the INI file: *DigitMapTimeoutTime*r for the start timer, *cpDigitMapShortTimer* for the short timer and *cpDigitMapLongTimer* for the inter-digit long timer. The following assumptions have been made regarding the timers logic in the dial plan:
  - When the dial plan line in question has ended with a range of digits (i.e. 09,[4-8]) the short timer is used, while the upper limit (8 in this case) of digits has not been reached. This is applicable for both basic and advanced matches.
  - When the dial plan was fully matched (or has reached the upper limit of digits), no timer is activated and the event will be sent immediately.
- **Collection by using a combined digit map and dial plan -** This is a proprietary method which can be used whenever there is a need for both regular dial plan and extra special patterns to satisfy special functions. When working in this mode, the digit map descriptor will contain in its value, part of the special patterns in question, according to the known rules. The use of both digit map and dial plan is indicated in the digit map name according to the following ABNF rules:
  - DigitMapName = Prefix DPPriority DialPlanName
  - Prefix = " IntACDialPlan"
  - DPPriority = 1-2(DIGIT)
  - DialPlanName = ALPHA *11(ALPHA / DIGIT / "_")

  DPpriority defines where to place the dial plan in the digit map patterns. For example, if six patterns are used, and the priority is set to 3, it is handled as if the digit map patterns 0-2 were written first, followed by a single pattern containing the dial plan and then all the remaining digit map patterns. This might be useful in case of a shortest match when one or the other is needed to be given a higher priority. Typically the dial plan should be given the lowest priority (0) as the special functions take precedence. However, it is up to the user to decide how to handle this.

7.2.3.4 Reporting Fax Events

Fax and modem events are reported using the packages from H.248.2: "CTYP", "FAX" and "IPFAX".

The "CTYP" package is used to report Fax and Modem tones using the "ctyp/dtone" event whenever the mode is identifies as Fax or Modem. The following tone types are supported:

- "V21flag"
- "CNG"
- "ANS"
- "ANSbar"
- "ANSAM"
- "ANSAMbar"
- "V8BIS", followed by the parameter "v8bist" which can have the values "CRe" or "CRdr".
- "JM"
- "BellHi"
- "SIG" – This will be sent for V.32 calling modem tone, V.22 bis tone, and other cases in which the mode is defined to be modem.

The "FAX" and "IPFAX" packages are used to report Fax state changes. 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.

Note: This number of the FAX pages statistics collected only on T.38 fax call. In other Fax type the value is always 0.

7.2.3.5 Reporting Media Failure

7.2.3.5.1 Reporting Media Establishment 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 (eventid:cause) defined in generic package (packageID: g) with reasons FT (“Failure, Temporarily”) and FP (“Failure, Permanent”).

The following are the possible errors:

- **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”.
- **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”.

7.2.3.5.2 Reporting Application Inactivity Detection

The gateway has ability to detect no media application data flow (Broken Media connection/Application inactivity). The application inactivity is detected when the termination hasn't received either RTP or RTCP packets for more than the specified timeout.

The gateway has two ways to notify application media inactivity.

1. The first way is by using the MEGACO netfail event (eventid: netfail) defined in network package (packageID: nt). The detection time interval is defined by the *BrokenConnectionEventTimeout* ini file parameter.
2. The second way is by using application data inactivity detection package (H.248.40). This package allows the MGC to request the MG to detect that after a certain period of time no IP packet has flowed on a particular termination/stream and notify an event (adid/ipstop) to the MGC. The default detection time interval is defined by the *BrokenConnectionEventTimeout* ini file parameter.

Notes:

- The detection is not available when termination is in SendOnly or Inactive modes.
- Only "IN" direction is supported. This is also a default direction. If "OUT" direction will be required an error will be generated. If "BOTH" directions will be required, the gateway will generate a notice message and will only detect the "IN" direction.

7.2.3.5.3 Reporting Media Quality Alert

The gateway has the ability to measure the quality of the media connection and indicate the loss of quality to the Media Gateway Controller.

Our quality measurement algorithm is based on the 'R Factor - General Voice Quality Grade' method defined in RTCP-XR (RFC 3611). Quality Alert functionality is supported only on the channels where RTCP-XR is enabled. For RTCP-XR implementation details refer to 'RTCP-XR support (H.248.30)' on page 616.

On channels without RTCP-XR functionality, the event will be handled but no event will be raised.

Both Quality Alert Event of network packages defined in H.248.1 and the Quality Alert Ceasing package defined in H.248.13, are supported.

The default value of the 'threshold' parameter can be configured by the *VQMONRVALTHR ini* file parameter and mapped to the H.248 definition in the following way:

```
Threshold = 100 – VQMONRVALTHR
```

Note: When both events are specified for the same termination, their threshold value should be the same. No error is raised in case the threshold is different and in such a case, the last value will be used.

7.2.3.6 Media Path QoS Support

The Quality of Service (QoS) for egress media flows can be either configured with the default values for the MG or set per call by MGC by using The Differentiated Service package.

The default Differentiated Service Code Point (DSCP) value for the egress media path is configured via the *PremiumServiceClassMediaDiffServ* ini file parameter. If this parameter is not set, and the old parameter *IPDiffServ'*is, the old parameter is used.

The Differentiated Service package enables the MGC to set the Quality of Service (QoS) for egress media flows without having to provision the default Differentiated Service Code Point (DSCP) value and enables configuring different values for each media flows.

7.2.3.7 Supporting Network Address and Port Translation

The Network Address and Port Translation Processing feature, allows the device to support media flows that have passed through an unknown number of CPE or network-based NAT devices. The device can latch to an address provided by an incoming Internet protocol (IP) application data stream rather than the address provided by the call/bearer control.

When this functionality is enabled, the device automatically compares the source address of the incoming RTP/RTCP/T.38 stream with the IP address and UDP port of the configured remote address. If the two are not identical, the transmitter modifies the sending address to correspond with the address of the incoming stream. The RTP, RTCP and T.38 can thus have independent destination IP addresses and UDP ports. The process of comparing and updating the remote address is executed each time the RTP stream connection breaks.

EnableIpAddrTranslation and *EnableUdpPortTranslation* allow you to specify the type of compare operation.

To compare only the IP address, set *EnableIpAddrTranslation* to "1", and *EnableUdpPortTranslation* to "0". In this case, if the first incoming packet arrives with only a difference in the UDP port, the sending addresses won't change. If both the IP address and UDP port need to be compared, then both parameters need to be set to "1".

This feature can be controlled (disabled/enabled) in two ways:

- It can be enabled it by setting *DisableNAT* to "0"
- Using MEGACO IP NAT Traversal Package (H.248.37)

  When the package had been used and a latch signal has been received, it will be handled according to its mode:

  - "OFF" mode – the NAT mechanism will be disabled.
  - " LATCH" mode or "RELATCH" - the NAT mechanism will be enabled.

7.2.3.8 Media IP Address Allocations

In order to support the interconnection of a packet network with another network, the MGC should have ability to instruct the MG which IP address to allocate. The AudioCodes gateway provides two ways of allocating media addresses:

- IP Domain Connection Package
- VLAN Package

7.2.3.8.1 IP Domain Connection Package

The IP Domain Connection (IPDC) package, as defined in H.248.41, is supported and allows for setting the address space in which the MGC can allocate an IP address. This is achieved by setting the IP realm identifier in the IP Domain Connection package. It enables a MGC to specify a realm for a termination.

The association of the "IP realm" and the "IP address space" is based on the cpMediaRealm table. For more information, refer to 'Control Protocol Media Realm Table' on page 587.

The IP realm identifier corresponds with the realm name in the cpMediaRealm table and the IP address and Port allocated for this termination/stream is based on the media interfaces configured for this realm. The following is an example of a MEGACO command using the IPDC package:

```
MEGACO/1 [10.4.10.84]:2944
Transaction = 1237 {
       Context = $ {
          Add  = $ {Media {
                     LocalControl {
                        Mode = Receiveonly, ipdc/realm=MasterRealm
                     },
         Local {
   v=0
   c=IN IP4 $
   m=audio $ RTP/AVP 0},
         Remote {
   v=0
   c=IN IP4 10.4.10.84
   m=audio 4000 RTP/AVP 0}}}}}
```

If a termination was created using a specific realm, the termination may not be modified to use a different realm. This is equivalent to stating that once a termination has been given a local IP address, the address may not be changed.

7.2.3.8.2 VLAN Package

The VLAN package, as defined in H.248.56, allows the MGC to select the interface (IP address) 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 and corresponding interface row of the Interface Table.

If a termination was created using a specific VLAN or default, the termination may not be modified to use a different realm or interface. This is equivalent to stating that once a termination has been given a local IP address, the address may not be changed.

Note: This implementation supports only one VLAN tag. The Ethernet Priority property is not in use.

7.2.3.8.3 Combination of the IPDC and VLAN Packages

A combination of the above two packages can also be used but is not recommended, as the same can be done by using one of the packages.

The following table summarizes the connection between the packages (assuming the Realm supplied and VL supplied really exists):

Combination of the IPDC and VLAN Packages

Using Realms	Using Vlan Tags	Outcome
x	x	First media interface from the default realm will be used.
x	❿	Interface corresponding to the VLAN tag in the default realm will be used.
❿	x	First free media interface corresponding to the given realm will be used.
❿	❿	If we have an interface that is both in the realm and uses the VLAN tag, it will be used else the command will fail.

- Control Protocol Media Realm Table

The Control Protocol (CP) Media Realm table enables the user to group up interfaces to better support the interconnection of different packet networks and use IPDC package as well as multi-level RTP termination naming.

- **IP Address Realm or IP Realm**

The CP Media Realm can be defined as a list of the IPv4 and IPv6 media interfaces. Each interface is associated with a VLAN and an IP subnet. This enables offering (offer in SDP reply) an IPv4 and an IPv6 address while using the same realm.

- **Media Realm Table Overview**

The Media Realm table is used to associate media interfaces (IPv4/IPv6/both) with realms. It allows the user to define up to 16 different realms in the table format, as shown below.

Currently, a user may configure up to two interface pair CP media realms: an IPv4 interface and an IPv6 interface (one of each).

Media Realm Table

Table Index	Realm Name	IPv4IF	IPv6IF	Port Range Start	Media Sessions	Port Range End
0	Default_NW	Media1IF		-1	-1	-1
1	Verizon_NW	Media2IF	Media10If	-1	-1	-1

Media Realm Table

2	HOT_NW	Media3IF		-1	-1	-1

Each row of the table defines a logical Media Realm, with its own Name, list of the media interfaces, and possible configuration of the port which allowed to be used for this realm.

- **Table Index**

This column holds the index of each interface. Possible values are 0 to 15. Each interface index must be unique.

- **Realm Name**

This column allows the configuration of a short string (up to 40 characters) representing name this realm. This name should be then used by the IPDC/Realm property of the LocalControl descriptor in order to allocate IP address from the specific realm. This column must have a unique value for each realm (no two realms can have the same name) and must not be left blank.

- **IPv4IF and IPv6IF Columns**

These columns allow the user to associate IPv4 and/or IPv6 interfaces to the realm. The IPv4/6IF columns should contain the name of the IPv4/6 interface corresponding to the one appearing in the interface table.

- **Port Range Start, Media Sessions and Port Range End Columns**

These columns allow the user to associate the UDP port regions with the realm.

- **Port Range Start** - is the starting port for the range of media ports.
- **Media Sessions** - is the number of media sessions associated with the range of ports. It is the actual number of media session available to the user in the port regions. For example, 100 ports correspond to 10 media sessions, since ports are allocated in chunks of 10.
- **Port Range End** – is the ending port for the range of media ports. It is a read only field which is calculated by adding the Media Session field (multiplied by the port chunk size) to the Port Range Start field. A value appears once a row has been successfully added to the table.

  A user may not configure some of the rows with a Port Range and some without. Port ranges over 60000 should not be used. Ranges of realm ports should not overlap.

Defining the Default Realm

A default realm is used if no other realms are supplied by the CA command. The default media realm is defined as follows:

- The user can configure the new *cpDefaultMediaRealmName* ini file parameter for explicitly defining a default realm from any of the realms appearing in the table.
- If the *cpMediaRealm* table was configured and the *cpDefaultMediaRealmName* parameter was not defined, then the first realm appearing in the table will be used as the default.
- If the *cpMediaRealm* table has not been configured, then the default realm will contain all media interfaces in the system.

7.2.4 Media Operation

7.2.4.1 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 *inii* 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)
- 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).

7.2.4.2 SDP Coder Negotiation Rules

The H.248 standard does not clearly define rules regarding the SDP coder negotiation, as was done in the MGCP RFC. Therefore, when first implementing it, it was decided to prefer the offer side (SDP of the remote side) to the local SDP. The reasoning was that the remote side already opened the call with its preferred coder. So if the local side also supports this coder there is no reason to force the other side changing it.

It appears, however, that some call agents require preferring the local coders. To support this requirement, a profile bit was added (Refer to 'SDP Support Profiling' on page 590). When this profile bit is set, the local coders are preferred to the remote coders in the answering side, while the offering side will give up and accept the choice of the answering side.

It is important, when using this profile, to understand the way the offering and answering sides are defined. The H.248 standard does not define the offering and answering side of the termination. The following is our definition:

- A command with only local SDP or with Local SDP and empty Remote SDP. This is the offering side.
- A command with only remote SDP. This is the offering side.
- A command with both local and remote SDP, and the termination did not have a remote SDP. This is the offering side.
- A command with both local and remote SDP, and the termination already had remote SDP. This is the answering side.

7.2.4.3 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 591 and 'Fax T.38 & Voice Band Data Support' on page 596.)
- 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 589.)
- Bit 5 (Value 32) - Default packetization period (ptime) for the transparent coder is 10 msec. Using the SDP attribute ptime overrides it.
- Bit 6 (Value 64) - The reply always contains SDP even if no change was made in the call configuration (Applicable to MGCP only).
- Bit 7 (Value 128) – Uses G.711 Coder as Bypass too.
- 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.
- Bit 9 (Value 512) – Defines that audio port should be used for T.38 call.
- Bit 10 (Value 1024) – Allows receiving illegal PTIME when more than 2 coders are configured in the SDP. In this case, the default PTIME will be selected.
- Bit 11 (Value 2048) – Defines the default fax mode as T.38 relay without T.38 activation. If T.38 was not negotiated in a call, the default configuration of the fax will be T.38 but no T.38 will be sent to the network until T.38 is negotiated. This configuration is required for V.34 fax over T.38 or for sending T.38 through a firewall.

- Bit 12 (Value 4096) – Defines the SDP negotiation mode to be as defined in MGCP. This means that the local coders are preferred to the remote coders.
- Bit 13 (Value 8192) – Defines that the MKI field size in the SRTP negotiation should be the same in the local and remote SDPs.
- Bit 14 (Value 16384) – Enables using coders from a local descriptor as if they were received in a remote descriptor. Since the remote descriptor is not fully saved, when a local descriptor is received without a remote descriptor, assume that the old remote descriptor contains the coders that are now in the local descriptor.
- Bit 15 (Value 32768) - Enables using G.711 for both voice and FAX bypass (Applicable to MGCP only).
- Bit 16 (Value 65536) - Replaces the negotiated ClearMode (transparent) coder with the G.711 coder. This is done for users that need to detect events, a function not active with the ClearMode coder.
- Bit 17 (Value 131072) - Corrects the behavior of the “reserveValue" parameter to be according to the H.248 standard. The default value is OFF. When the value is set to OFF, only the first voice coder will be included in the reply, but other non-voice coders are also included in the reply.
- Bit 18 (Value 262144) – The coder negotiation assumes vbd=yes for the local side, if it was received for the same coder in the remote side.
- Bit 19 (Value 524288) – When receiving an image line in the remote SDP with Port 0, reply with a valid image line instead of ignoring it.
- Bit 21 (Value 2097152) – Expands Bit 9. When both bits (9 and 21) are set, the SDP response contains the same port for both audio and fax, even when both audio and image (T.38) lines are returned. in such a case, the channel is opened in T.38 mode and not in audio mode.

7.2.4.4 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 RFC 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.

For backward compatibility, the same functionality can be enabled by the value of bit 2 (value 4) for the *MGCPCompatibilityProfile* MEGACO profiling parameter.

7.2.4.5 Silence Suppression Support

Silence suppression can be enabled in two ways:

- Configure the “SCE” ini file parameter to ON through one of the configuration tools. This is a static way, and applies to all calls. The value of the SCE ini file parameter is “ON” by default.
- 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:

- Setting the “SCE” *ini* file parameter to OFF. This is a static way, and applies to all calls.
- 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.
- 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 or the remote side is capable of processing.

The table below summarizes the operation of silence suppression:

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 OR - SDP does not contain Payload Type 13	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

7.2.4.6 Under-Specified Local Descriptor

The supported under-specified fields in the SDP are:

- IP address
- Port
- Payload
- Profile
- Ptime

For example:

```
c=IN IP4 $
m=audio $ $ $
a=ptime:$
```

The reply for such command is a list of all supported coders with default ptime for the first coder, RTP/AVP protocol and allocated IP address and port. The first coder in the supported coder list is the defined default 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. In this case, 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:$
```

7.2.4.7 Support of Asymmetric Tx/Rx Payloads

In the MEGACO commands, up to two SDP sessions are received - one for the local side and one for the remote side. Each SDP session 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. The payload type to be used is taken from 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 should be the payload type defined in the local SDP.

This functionality can be disabled by turn on “Symmetric payloads” in SDP Profile. For more information refer to SDP Support Profiling section.

7.2.4.8 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.

Note: The VBD Payload Type is always symmetric. If the remote descriptor contains VBD, the Payload Type will be taken from the remote descriptor.

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

7.2.4.9.1 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 turned off.
- FaxTransportType is set to T.38 Relay.
- Fax Redundancy can be controlled using the *FaxRelayRedundancyDepth* configuration parameter. This parameter controls only non-V.21 packets. For V.21 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 *FaxModemBypassBasicRTPPacketInterval* parameter.
- The payload to be used is configured by the *FaxBypassPayLoadType* and *ModemBypassPayloadType* parameters.

7.2.4.9.2 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.

The returned SDP contains the following T.38 attributes, as defined in the T.38 spec:

- T38MaxBitRate – The value of this attribute is according to the *FaxRelayMaxRate* configuration parameter and can have the following values:
  - 2400
  - 4800
  - 7200
  - 9600
  - 12000
  - 14400
  - 16800
  - 19200
  - 21600
  - 24000
  - 31200
  - 33600

Note: T38MaxBitRate values above 14400 are allowed only when T38FaxVersion is equal to 3.

  - T38FaxRateManagement - This attribute will always have the value "transferredTCF".
  - T38FaxVersion – The value of this attribute is according to T38Version configuration parameter and can have the following values:
    - 0
    - 3

- A value of '3' enables real V.34 fax transfer over T.38 fax relay, which allows transfer bitrates of up to 33600 bps.

Note: DSP Template No. 10 should be used to support T.38 version 3.

- A value of '0' forces a fallback of a V.34 fax to T.30, allowing a transfer bitrate up to 14400 bps over T.38 fax relay.
- T38FaxMaxBuffer - This attribute will always have a value of '1024'.
- T38FaxMaxDatagram - This attribute will always have a value of '238'.
- T38FaxUdpEC - This attribute only appears if T.38 redundancy is used (see above). In that case the value will be "T38UdpRedundancy".

Two modes of fax negotiation 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 only 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 G.711 coder is not negotiated and T.38 is not negotiated, the Fax 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.

Another SDP profile bit that controls the fax configuration is the 7th bit (value 128). It enables using the negotiated G.711 coder and payload as the bypass (VBD) fax/modem coder and payload type.

If this bit is set and the G.711 coder is negotiated, the Fax (in case T.38 is not negotiated) and Modem Transport mode is set to Bypass (VBD), and the G.711 coder and payload type are used for the fax and modem. Note that this is a proprietary way to define a VBD coder. It is recommended to use the standard way by using the V.152 VBD attribute (see next section).

To pass V.34 Fax over T.38 the following SDP negotiation rules are required:

- T38FaxVersion attribute (a= T38FaxVersion:value)
  - If it is not defined in the local descriptor, use the T38Version configuration parameter value in both strict and non-strict SDP negotiation modes.
  - If it is not defined in the remote descriptor, use T38FaxVersion=0.
  - An attribute defined in the remote descriptor is a master attribute and has the highest priority during negotiation.
- T38FaxMaxBitRate (a= T38FaxMaxBitRate:value)
  - If it is not defined in the local descriptor, use the FaxRelayMaxRate configuration parameter in both strict and non-strict SDP negotiation modes.

- If T38FaxMaxBitrate in the local descriptor doesn't match T38FaxVersion, use the highest supported version (actually 14400 should be supported by both Versions 0 and 3).
- If it is not defined in the remote descriptor, set the attribute's value according to one defined in the local descriptor.
- An attribute defined in the remote descriptor is a master attribute and has the highest priority during negotiation.

For backward compatibility purposes, the functionality of both bits can be enabled by the value of bit 2 (value 4) for the *MGCPCompatibilityProfile* MEGACO profiling parameter.

When the gateway is behind a Network Address Translation (NAT), and the call starts as a voice call and is moved by the MGC to T.38 after detecting the fax, it is important to use the same port for audio and T.38 transports. In order to activate this option, CPSDPPROFILE should have bit #9 set to 1 (add 512 to the value of profile).

7.2.4.10 V.152 - VBD Attribute Support

The V.152 defines a way to declare support of VBD (Voice Band Data), and define which coder and payload will be used for it. This is done by using a new SDP attribute 'gpmd'.

(See 'http://potaroo.net/ietf/idref/draft-rajeshkumar-mmusic-gpmd/' 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 a local descriptor that contains coders is received from the call manager, the VBD attribute will be included if a G.711 coder exists in the request (for backward compatibility) or if a VBD was specified in the request. In the first case (only G.711), the G.711 will be returned as a normal voice coder. Also a new dynamic payload with the same G.711 coder will be added to indicate VBD support. (See example 1 below) In the second case, the VBD coder returned is the one from the offerer.
- If both local and remote descriptors are received (Answerer) and the G.711 coder is negotiated with the VBD, the payload of the remote descriptor will be adopted for our VBD. (See example 2 below.)
- If the coders are also under-specified, then in case the Fax Transport Type is configured to Bypass (VBD) or the modem is configured to Bypass (VBD), the

supported coder list will be returned with a new dynamic payload with VBD mapping for each G.711 coder in the device coder list.

```
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
}
```

7.2.4.11 Fax and Modem Operation Recommendation

This section provides guidelines for configuring the device to pass Fax over T.38 and Fax SG3/Modems over Voice Band Data.

7.2.4.11.1T.30 Fax over T.38

To pass T.30 Fax over T.38, the MGC should negotiate a T.38 stream.

It is recommended to configure the modems to Voice Band Data (VBD) as explained below.

When the gateway is behind a Network Address Translation (NAT), and the call starts as a voice call and moved by the MGC to T.38 only after detecting the fax, it is important to use the same port for audio and T.38 transports.

In order to activate this option, CPSDPPROFILE should have the bit #9 set to "1" (add 512 to the value of profile).

7.2.4.11.2V.34 FAX

To pass V.34 Fax over T.38, the MGC should negotiate a T.38 stream.

It is important to notice that in order to support V.34 fax over T.38 the fax should be configured to T.38 at the beginning of the call, and the fax mode cannot be changed during the call. If the T.38 stream may not be negotiated at the beginning of the call but the support for V.34 fax relay is desired, then CPSDPPROFILE should have the bit #11 set to 1 (add 2048 to the profile).

When this bit is set, the default FAX configuration as set by the INI transport parameters is ignored and if no explicit directions for fax configuration are received from the MGC, the fax is configured to T.38 ”dormant” mode, until the T.38 stream is negotiated.

In this mode, fax is never configured to transparent (with or without events), so it is not possible to pass faxes transparently on a best effort basis using voice coders. Besides that, when bit 11 is set and the VBD fax transportation is desired, the fax should be configured by the MGC on a per call basis (as explained below). Again, any INI file default transport type parameters are ignored.

To pass V.34 Fax with a bitrate above 14400 over T.38, the following should be done:

- DSP Template 10 should be used
- T38Version configuration parameter should be set to '3'
- SDP attribute a=T38FaxVersion:3 should be defined in the remote descriptor

Note: If T38Version=0, the V.34 fax will be forced to fall back to T.30 (up to 14400 bps).

7.2.4.11.3SG3 Fax & Modems over Voice Band Data (VBD)

To pass SG3 fax and modems over VBD, the user can either configure the gateway to operate in this mode by default or can configure the gateway dynamically to use VBD on each call establishment. To use the VBD as a default option the user should configure the gateway as follows:

Turn off the bit #3 (value 8) of the CPSDPProfile and configure the following parameters:

- V21ModemTransportType = 2
- V22ModemTransportType = 2
- V23ModemTransportType = 2

- V32ModemTransportType = 2
- V34ModemTransportType = 2
- FaxModemBypassCoderType = [same as configured in the remote gateway]
- FaxBypassPayloadType = [same as configured in the remote gateway]
- ModemBypassPayloadType = [same as configured in the remote gateway]

There are three options for using VBD for each call separately:

- Use the standard V.152 protocol.

Turn on the bit 1 (value 2) of the CPSDPPROFILE parameter.

The call establishment should follow the recommendation per V.152. The gateway will send its VBD capabilities in its local descriptor during call establishment process. If the remote side will agree to the gateway capabilities of using the VBD, the SG3 fax & Modem will pass over VBD using the VBD selected coder & VBD selected Payload Type. If the remote side does not support VBD, then the SG3 fax & Modems will pass according to the gateway default configuration.

- Configure the gateway as follows:

Turn on Bit 7 (Value 128) of the CPSDPPROFILE. When this bit is set, In case the set of coders negotiated in the call establishment will contain G711 coder/payload type, it will also be used as the VBD coder/payload type. If, on the other hand, no G711 coder was negotiated, the modem will be configured according to the configuration.

- Both options can be used simultaneously – i.e. if both bits are set to 1, if the remote descriptor contains the VBD attributes, it will be used, otherwise, if G711 coder is negotiated it is used as the VBD otherwise the modem will be configured according to the default configuration.

Using one of the above options will configure the modems as bypass. The faxes will be configured as bypass in such cases only if no T.38 stream was negotiated.

7.2.4.11.4SG3 Fax & Modems over High Bit Rate coder using NSE Mode

In order to interoperate with vendors that do not support the standard VBD mode but are compatible with Cisco's proprietary pass-through transport type, there is another possibility to work using the NSE (Named Service Event) mode. In this mode the gateway is being configured to always use NSE mode on all SG3 & Modem calls. Upon SG3 fax or modem detection, the gateway will send the remote side NSE packet to move to a preconfigured Bypass coder (E.g. G.711Alaw), & NSE packet to disable the Echo Canceller on the remote side. The gateway will act similar to VBD mode in respect to Jitter Buffer, Echo Canceller etc manipulation. This is done to make sure that SG3 Fax & modems will have best conditions to pass through the gateway. The following configuration should be made in order to operate the NSE option:

Turn off the bit #3 of the CPSDPProfile and configure the following parameters:

- V21ModemTransportType = 2V22ModemTransportType = 2
- V23ModemTransportType = 2
- V32ModemTransportType = 2
- V34ModemTransportType = 2
- NSEPayloadType = [same as configured in the remote gateway]
- NSEMode = 1
- FaxModemBypassCoderType = [same as the NSE coder configured in the remote gateway]

7.2.4.11.5SG3 Fax & Modems Transparently over the RTP Stream

It is not recommended to use this option. If possible, VBD should be the preferred option.

Passing the fax and modems transparently using the LBR coder is only on a best effort basis and cannot be guaranteed.

Use either of the following options:

- Option 1: Set bit #3 (value 8) of the CPSDPProfile to 1. It is a strict SDP negotiation mode and any default INI file configuration is ignored. Fax will be opened as "Transparent".
- Option 2: Turn off the bit #3 (value 8) of the CPSDPProfile and configure the following parameters:
  - V21ModemTransportType = 0
  - V22ModemTransportType = 0
  - V23ModemTransportType = 0
  - V32ModemTransportType = 0
  - V34ModemTransportType = 0

7.2.4.12 Media Encryption (SRTP) using RFC 3711

Note: SRTP support is according to the selected DSP version template.

SRTP (RFC 3711) details the media encryption standard. The device partially implements it. RFC 3711 defines a new media profile “RTP/SAVP” to negotiate 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.

7.2.4.12.1Supported 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.

7.2.4.12.2Supported 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.

7.2.4.12.3Configuration 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. *(It can be overridden by session params.)*
6. *ini* file parameter – The RTCPEncryptionDisableTx parameter can be set in order to work with non-encrypted RTCP. *(It can be overridden by session params.)*
7. *ini* file parameter – The RTPAuthenticationDisableTx parameter can be set in order to work with non-authenticated RTP. *(It can be overridden by session params.)*

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.

7.2.4.12.4SDP 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 choose 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

  When the key params are fully specified, it means that the sender wants a specific suite, and wants the device to use the given key (i.e., not to produce a new one).

  The length of key string should be exactly 40 symbols for 128 bit encryption suites and 51 symbols for 192 bit encryption suites.

  The only 192 bit encryption suite is ARIA_CM_192_HMAC_SHA1_80.

  Suites with key string with invalid length will be ignored by the SDP parser.

- 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.

7.2.4.12.5Connection 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 = Inactive},
                    Local {
v=0
c=IN IP4 $
m=audio $ RTP/SAVP 0
a=ptime:20
 }}}}}
```

The device answers:

```
MEGACO/1 [10.4.4.46]:2944
   P = 1 {
```

```
       C = $ {
          A = $ {M {O {
                    MO = Inactive},
                    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 = Inactive},
                    Local {
v=0
c=IN IP4 $
m=audio $ RTP/SAVP 0
m=audio $ RTP/AVP 0
a=ptime:20
 }}}}}
```

The device answers:

```
MEGACO/1 [10.4.4.46]:2944
   P = 2{
       C = 2 {
          A = GWRTP/2 {M {O {
                    MO = Inactive},
                    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 = Inactive},
                   Local {
v=0
c=IN IP4 $
m=audio $ RTP/SAVP 0
a=crypto:1 $ $
a=ptime:20
 }}}}}
```

The device answers:

```
MEGACO/1 [10.4.4.46]:2944
   P = 3 {
       C = 3 {
          A = GWRTP/3 {M {O {
                   MO = Inactive},
                   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 = Inactive},
                    Local {
v=0
c=IN IP4 $
m=audio $ RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_32 $
a=ptime:20
 }}}}}
```

The device answers:

```
MEGACO/1 [10.4.4.46]:2944
   P = 4 {
       C = 4 {
          A = GWRTP/4 {M {O {
                    MO = Inactive},
                    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
 }}}}}
```

Oferrer – Suite is fully defined

In the example, the MGC wants the device to work with a specific suite and key. The device returns with the same specified key:

The MGC sends:

```
MEGACO/1 [10.2.1.228]:2944
   Transaction = 4 {
       Context = $ {
          Add = $ {Media {LocalControl {
                    Mode = Inactive},
                    Local {
v=0
c=IN IP4 $
m=audio $ RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_32
inline:MKHEBFC/PMKHEB+CJfvspnkheifcZW
a=ptime:20
 }}}}}
```

The device answers:

```
MEGACO/1 [10.4.4.46]:2944
   P = 4 {
       C = 4 {
          A = GWRTP/4 {M {O {
                    MO = Inactive},
                    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
 }}}}}
```

Note: The key sent in the remote descriptor is used to decrypt the incoming RTP/RTCP packets. It is therefore not recommended to set the mode of the stream to receive packets until the remote descriptor with the remote key is sent to the terminating side.

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 device 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.

7.2.4.13 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.

7.2.4.14 IPv6 Support for Media Streams

MEGACO supports the usage of IPv6 for opening and manipulating media streams. This includes Audio, Video, Fax and Data streams.

A user may simultaneously open different channels on the blade, some using IPv6 and some using IPv4. In order to enable this feature, the user should configure at least one IPv6 interface.

An IPv6 interface may be selected either by the VLAN package or by IPDC package. If packages are not used to select the IPv6 interface, the default IPv6 interface (first IPv6 interface) will be used.

A user may use both a long and a short IPv6 command representation when issuing a command. For example, both of the following IPv6 addresses are valid:

- 2000::1:290:8fff:fe09:909c
- 2000:0000:0000:0001:0290:8fff:fe09:909c

The following are two examples of **Add** commands using IPv6:

Add Command Examples

Command	Response

MEGACO/1 [10.4.4.119]:2944
Transaction = 1237 {
Context = $ {
Add = te/physpool0/t0/c1 { Media {
LocalControl {Mode = SR}}},
Add = te/tepool1/$ {Media {
LocalControl {
Mode = SR
},
Local {
v=0
c=IN IP6 $
m=audio $ RTP/AVP 0},
Remote {
v=0
c=IN IP6 2000::1:212:79ff:feae:d9ca
m=audio 4010 RTP/AVP 0}}}}}
	MEGACO/2 [10.4.4.119]:2944
	P=7600{
	C=1{
	A = te/physpool0/t0/c1,A = te/tepool1/0{
	M{
	L{
	
	v=0
	c=IN IP6 2000::1:290:8FFF:FE09:909C
	m=audio 4000 RTP/AVP 0
	a=ptime:20
	a=silencesupp:off - - - -
	}}}}}
	
	(An IPv6 was requested and returned).

MEGACO/1 [10.4.4.119]:2944
Transaction = 1237 {
Context = $ {
Add = te/tepool1/$ {M{O{MO = SR},
Local {
v=0
c=IN IP6 $
m=audio $ RTP/AVP 0
v=0
c=IN IP4 $
m=audio $ RTP/AVP 0}
}}}}
	MEGACO/2 [10.4.4.119]:2944
	P=7715{
	C=2{
	A = te/tepool1/1{
	M{
	L{
	
	v=0
	c=IN IP6 2000::1:290:8fff:fe09:909c
	m=audio 4010 RTP/AVP 0
	a=ptime:20
	a=silencesupp:off - - - -
	v=0
	c=IN IP4 10.4.4.119
	m=audio 4010 RTP/AVP 0
	a=ptime:20
	a=silencesupp:off - - - -
	}}}}}
	
	(Both IPv4 and IPv6 were offered)

7.2.4.15 EVRC Family Coders

7.2.4.15.1 EVRC Coders

The EVRC coders are supported according to the following standards:

- RFC 3558 - two types of coders exist in the EVRC family:
  - EVRC0 - This is the Header-free format. Each frame can include only one packet and each can be of a different rate.
  - 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.
- Draft-ietf-avt-compact-bundled-evrc-11.txt - defines one more EVRC coder:
  - 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:

- X-EVRC-TTY - This is an EVRC coder with TTY support.
- 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.

7.2.4.15.2 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).

7.2.4.15.3 Silence Suppression Support in EVRC Coders

RFC 4788 defines four new parameters to support Silence Suppression in EVRC. It also defines that the default value for the Silence Suppression is "ON". Here is an example for an 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 return the Silence Suppression as "ON", it will be turned on in the call. Note that we use the default value for the "hangover" parameter.

7.2.4.16 AMR Coders

7.2.4.16.1 AMR Supported Modes

AMR can work in one of two modes:

- Octet aligned
- Bandwidth efficient

The mode can be set per call by using the SDP FMTP attribute with the value 'octet-align'. If this value is not included, we will use the configuration parameter 'AmrOctetAlignedEnable'. The octet aligned value must be symmetric. Therefore, if it is included in the remote SDP but not in the local SDP, use the value from the remote SDP.

7.2.4.17 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.

7.2.4.18 Microsoft RTA coders

7.2.4.18.1 Microsoft RTA NB coder

The following parameters must be specified to support the MS/RTA NB coder:

- m=audio $ RTP/AVP 114
- a=rtpmap:114 x-msrta/8000
- a=fmtp:114 bitrate=29000

Bitrate should be set between 8800 and 29000.

Clock rate must be set to 8000.

7.2.4.19 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 overwritten by the external CoderTable. For more information refer to 'Coder Table File' on page 752.

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
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

MEGACO Mapping Payload Numbers to Coders

Default Payload Number	Encoding Name	Coder
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
114	"X-MSRTA"	Microsoft RTA NB

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.

7.2.4.20 RTCP-XR support (H.248.30)

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
- XRBM

Note that the two newer packages - RECRTCPXR and RECXRBM - are not supported.

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. Note that ini file parameter configuration is only effective when the working profile is non-strict mode.

In order to use RTCP-XR functionality, it should be enabled in the feature key with the RTCP-XR feature. For TP-6310/TP-8410 blades, the VQMONENABLE *ini* file parameter should be set to a value greater than 0.

When the RTCP-XR feature key is not enabled, there is no way to have any RTCP-XR reference. That means that no statistics will be reported, and no RTCP-XR line will be shown in the SDP.

In the following description the term "RTCP-XR line" stands for (a=rtcp-xr: voip-metrics), while the term "empty RTCP-XR line" stands for (a=rtcp-xr:).

Except for the first two paragraphs, all the rest assume that the feature key contains the RTCP-XR feature.

7.2.4.20.1 Including RTCP-XR Line in the SDP Response

The RTCP-XR line is included in the SDP response in the following case:

- The feature key contains the RTCP-XR feature, and
- VQMONENABLE is set to a value greater than 0 (only for TP-6310, TP-8410 and Mediant 3000)
- We receive an ADD or MODIFY command in which there is an RTCP-XR line in the local SDP. (See below in section "Working in a non-strict SDP negotiation" for an exception)

7.2.4.20.2 Including an Empty RTCP-XR Line in the SDP Response

Not all gateways are able to handle the empty RTCP-XR line format. Therefore, the only time that we will return the empty RTCP-XR line is when it was included in the SDP sent to our gateway. When no RTCP-XR in included in the SDP, the assumption is that no RTCP-XR support is needed.

7.2.4.20.3 Sending RTCP-XR Packets

Packets are sent to the remote side under the following conditions:

- The returned local SDP includes the RTCP-XR line
- When a remote SDP is received which also includes the RTCP-XR line

7.2.4.20.4 Stop Sending RTPC-XR packets

Deactivation takes place sending RTCP-XR packets if:

- A remote SDP is received with an empty RTCP-XR line

or

- A remote SDP is received without any RTCP-XR line

7.2.4.20.5 Statistics Report

The statistics for RTCP-XR are reported if the returned local SDP includes the RTCP-XR line.

The report is issued when the call manager requests statistics, usually when closing a call (Subtract), but also during the call (Audit).

7.2.4.20.6 Working in a non-strict SDP Negotiation

If the working mode is set to non-strict SDP negotiation, and the ini file RTCP-XR is set to ON we behave as if the MGC included the RTCP-XR line in the local SDP. That means that we will return that line in our response and send statistics. However, we will NOT send RTCP packets unless the line was also included in the remote SDP.

7.2.5 Call Types and Connection Model

7.2.5.1 CAS Calls Support

7.2.5.1.1 MFCR2 Support

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 Start call flow).

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 46: MEGACO-R2 Call Start Flow Diagram

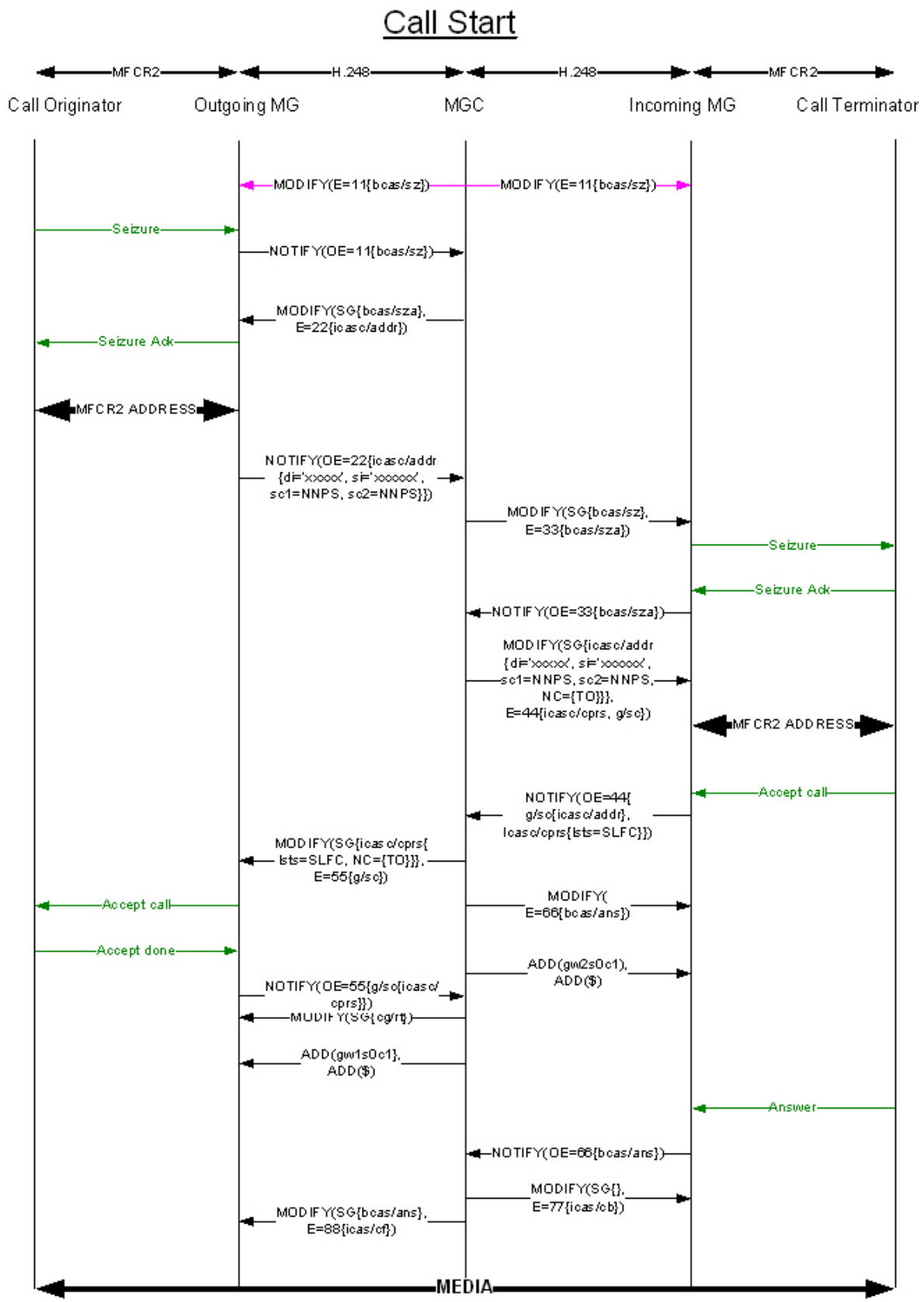
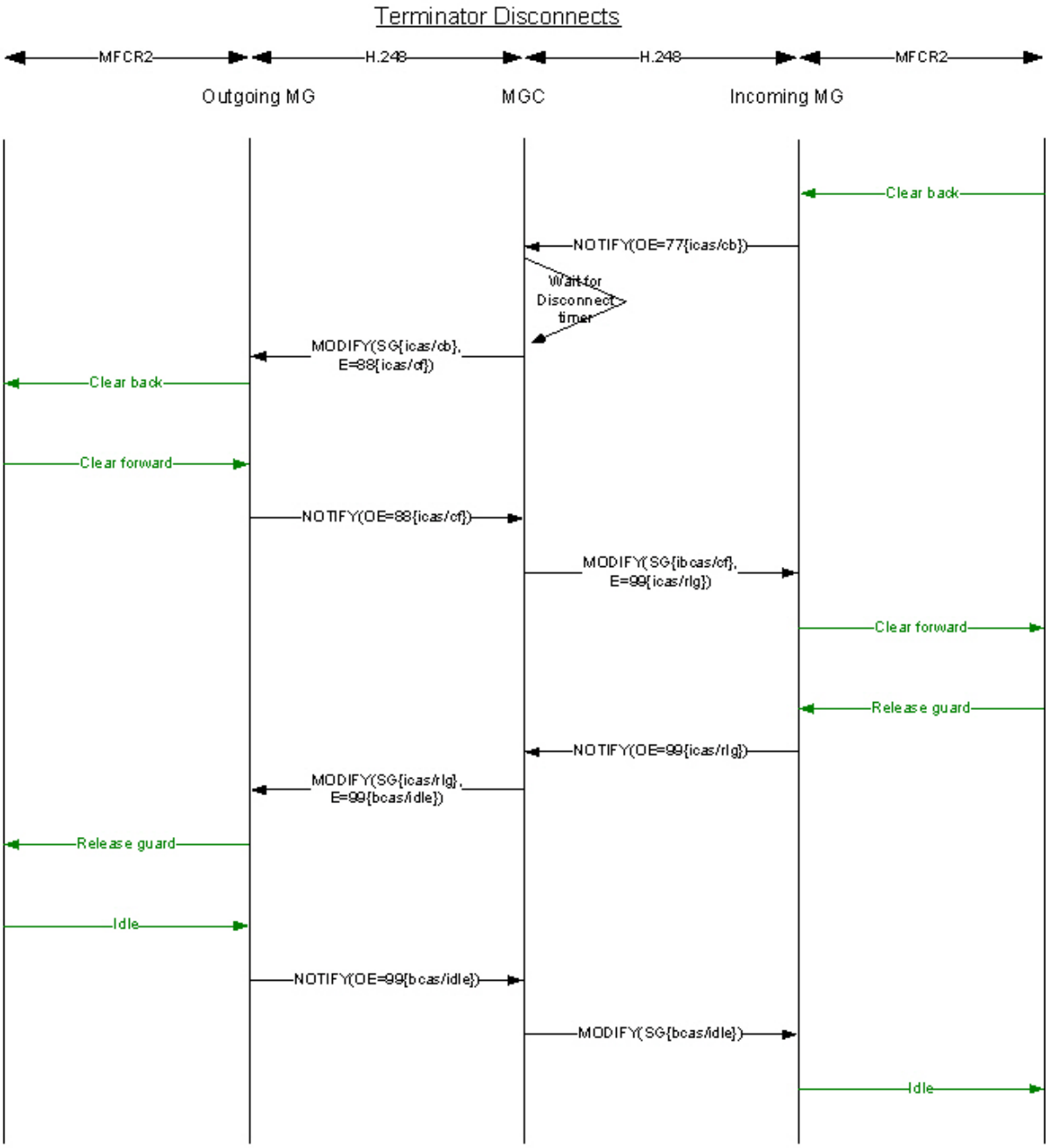


Figure 47: MEGACO-R2 Call Disconnect Flow Diagram



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.

7.2.5.1.2 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 48: MEGACO-911 Call Start Flow Diagram

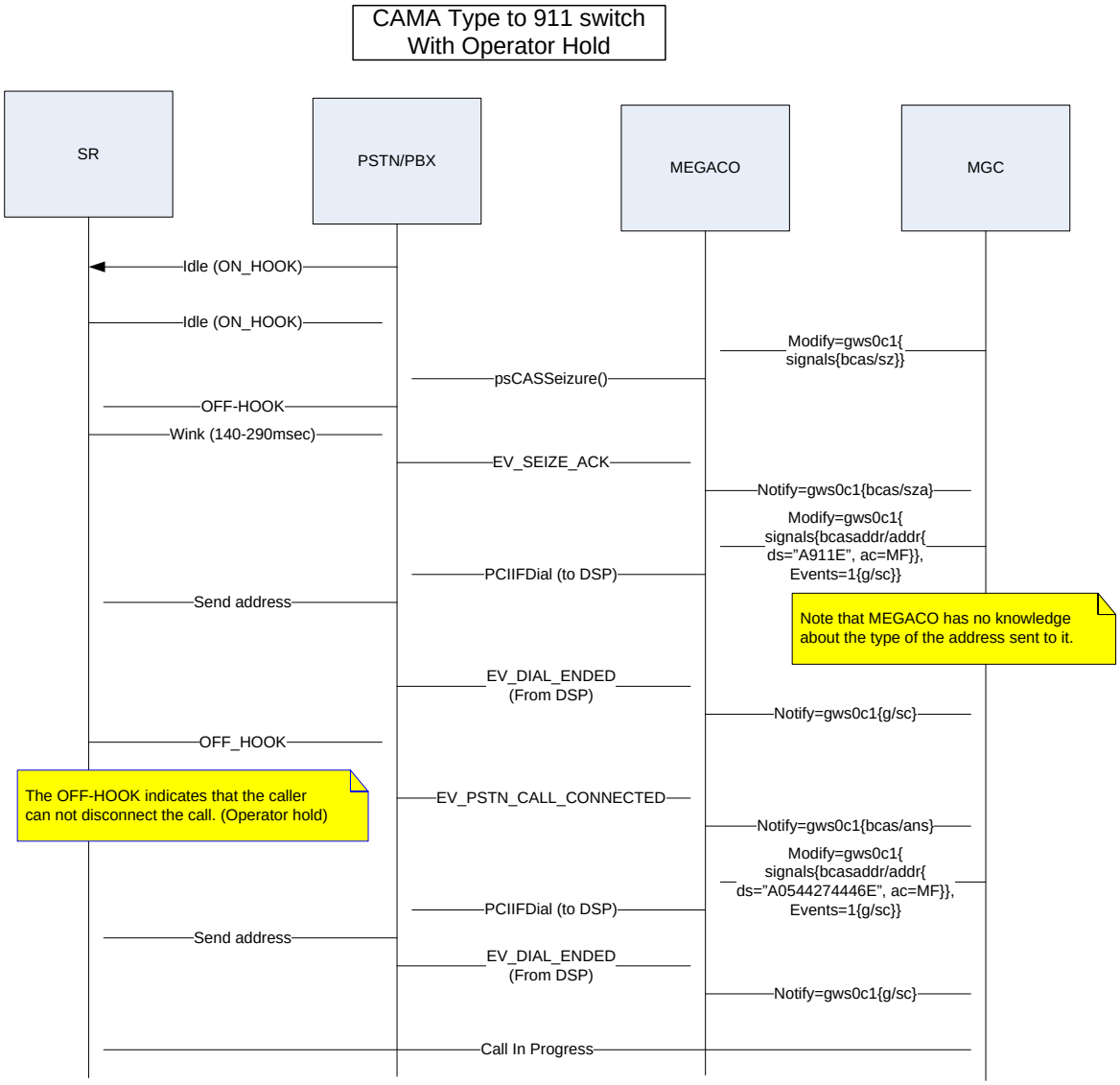
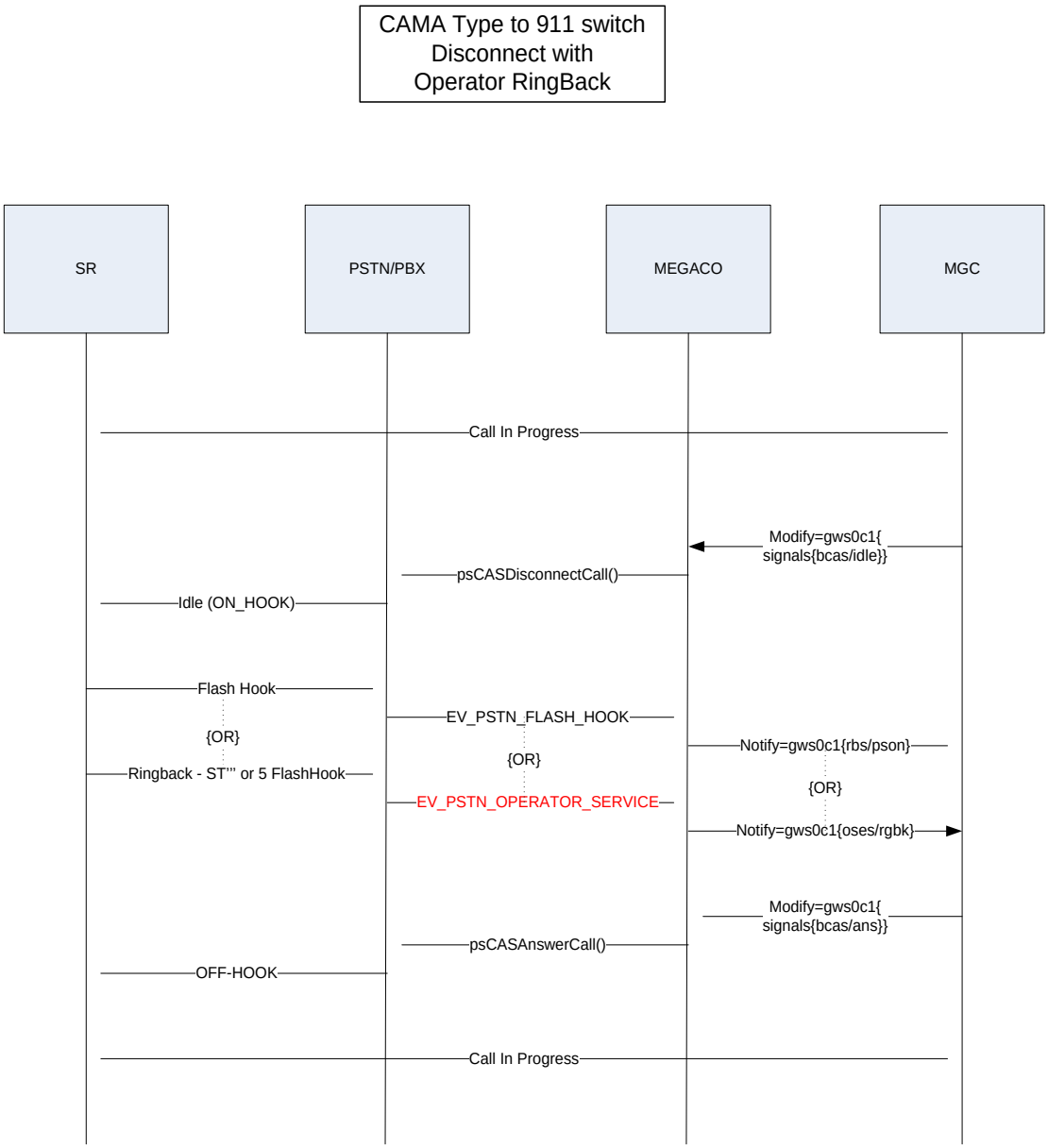


Figure 49: MEGACO-911 Operator Ringback Flow Diagram



7.2.5.1.3 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.

7.2.5.2 TDM Hairpinning

TDM Hairpinning is a scenario whereby a call is being made between two TDM endpoints on a single media gateway. The media comes in on one TDM channel and goes out on another TDM channel through the use of an internal loop in the media gateway.

In H.248 terms, hairpinning is done by using a single Context with two TDM Terminations.

Two modes of the internal loop are available:

1. Connect two TDM terminations through PSTN. In this mode no DSP is used and the two TDM time slots are directly connected via PSTN. In this mode, any termination configuration is ignored and both signal generation and event detection are unavailable.
2. Connect two TDM terminations through internal IP software loopback. In this mode two DSPs are opened with the default coder and default configuration (according to the board configuration). The mode selection is done by using the *MegacoTdmHairPinningMode* ini file parameter.

7.2.5.3 Conferencing

7.2.5.3.1 3-way Conferencing

For devices which do not support N-way conferencing (see the next section) there is an option to support a 3-way conferencing.

The following configurations are available:

- TDM-IP-IP
- TDM-IP-TDM
- TDM-TDM-TDM

Pure IP conferencing is not supported.

When closing one of the participants (subtracting from the conference), the conference is closed and the call goes back to the original state, except in the following cases:

- The conference had TDM(CID1)-IP1(CID1)-IP2(CID2)
- IP1(CID1) was subtracted
- In this case, there is no option to connect CID1 and CID2 without the conference, so we leave both CIDs and the conference open.

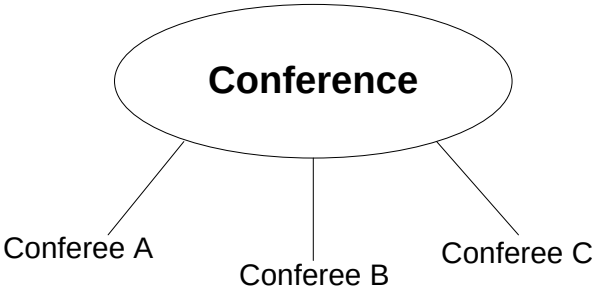
In order to obtain a CID for a second IP participant, we look for an idle channel. We start from the highest available channel and go down. When such a channel found, the CID will be used for the second IP user, and the termination which was connected to that channel will not be available for use.

Note: This sub-section is only applicable to IPM.

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 50: 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.

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. *ConferenceMaxSimultaneousSpeakers* ini file parameter defines the number of users that can speak together in a conference.

Each conference participant needs a DSP resource, which is allocated internally.

There are three options to create a conference:

1. Use proprietary context attribute package, as described in AudioCodes H.248 Proprietary Packages. Using this package enables reserving context and resources like Conference Bridge, Conference Legs, Terminations and DSPs for Conference call.
2. Add three or more terminations with both way topologies to the same context in one command. A conference creation command is shown in the following example.

By using this option Conference Bridge will be allocated and conference legs will be reserved according to the number of the termination in the command.

```
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 = Inactive }}},
    Add  = $ {Media {LocalControl {Mode = Inactive }}},
    Add  = $ {Media {LocalControl {Mode = Inactive }}}
  }
}
```

3.

4. Add three terminations to the same context one by one. Using this option, conference will be created only when third termination is added to the context. Adding third termination may fail due to conference out of resources. When using this option, the *CPMediaResourceOptimization ini* file parameter can't be configured with value 1 (ENABLE_SINGLE_DSP_ALLOCATION).

7.2.5.4 Interactive Voice Response (IVR)

Note: This sub-section is only applicable to IPM.

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'.) 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. The amount of the resident storage is dependent on the device. 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 local 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.

7.2.5.4.1 Segment Description Matrix

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

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

Segment Descriptor Variables

Variables			
	gen	Generic	Yes
	ndn	North American DN	Yes(Only in TD-51)
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

7.2.5.4.2 Configuring the Device

- **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.

- **Configuration**

The audio bundle can be uploaded using either FTP, NFS or HTTP. For more information see 'Automatic Update Facility' on page 26.

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.

- **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 165 and 'Automatic Update Facility' on page 26.

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 63.

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.

- **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 268.

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 108.

7.2.5.5 Test Trunk Support

Note: This sub-section is only applicable to IPM.

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.

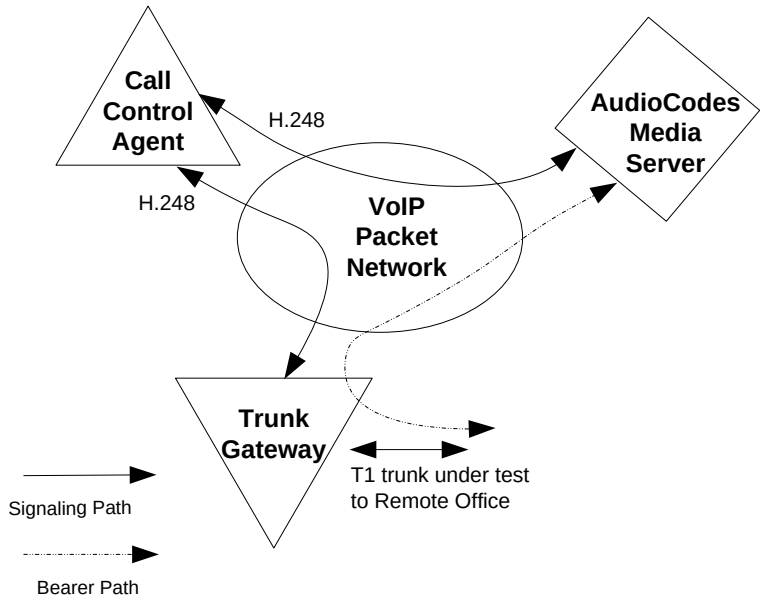
Digital trunks interconnect a trunk gateway and the PSTN and consist of multiple timeslots carrying digitized voice traffic, multiplexed at a T1 or higher rate. Individual timeslots correspond to 'trunk circuits' in the trunk database of the call agent. These circuits may connect directly to a Telco 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.

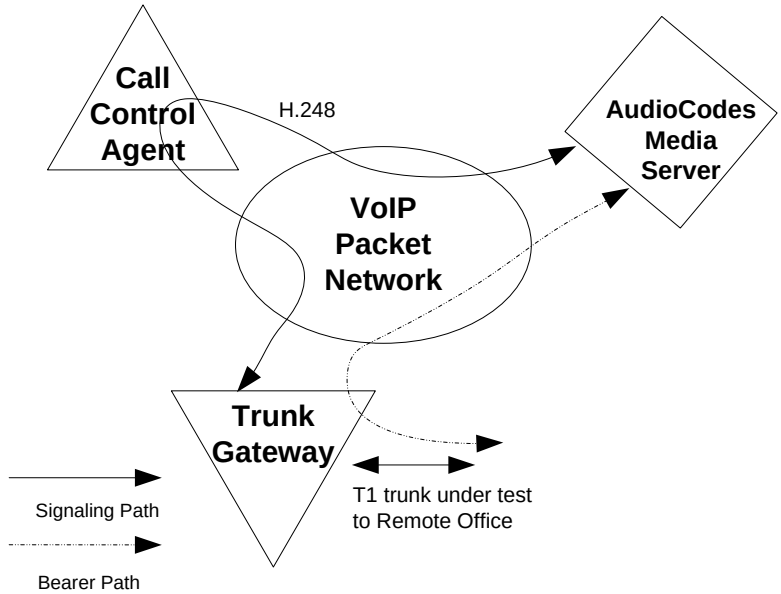
The following two diagrams below show the network connections plus the signaling and bearer paths for the device based TDM trunk testing. Note that in this configuration the network supports the exchange of call control messaging between the Media Gateway Controller of the call control agent and the 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 51: 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 52: 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.

7.2.5.5.1 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.

7.2.5.6 IP-to-IP Interworking Support

Note: Applicable to BGF configurations with an SBC session Feature Key.

IP-to-IP Interworking in MEGACO is a context with two ephemeral terminations. It can be used for:

- Interworking between two subnets (Translation of IP address and Port Number - NAPT).
- Transcoding (Connect users with different coders and coder settings)
- Interworking between IPv4 and IPv6 networks (NAPT-PT)

A termination can contain three different types of streams:

- **“Media Aware” Streams**

  These are streams where the MG knows the transported "media" information and the underlying transport protocol type. As a result, the function applied on the IP flow may include media processing, (for example: trans-coding).

  The following is a SDP description of a "Media Aware" stream"

```
v=0
c=IN IP4 10.4.2.38
m=audio 4000 RTP/AVP 0
```

  Currently, only coders that are supported by our DSP can be used in a "Media Aware" stream.

- **"Media-agnostic" Streams**

  In these streams, the values of media description are not allowing the MG to conclude the transported "media" information. As a result, the function applied on IP flow is limited to Layer 3 (IP ; Network layer) and Layer 4 (transport layer only). The inter-working is done on the RTP/SRTP level (RTP,RTCP manipulations, including SRTP encryption manipulations).

  The following is a SDP description of a "Media-agnostic" stream:

```
v=0
c=IN IP4 10.4.2.38
m=- 4000 RTP/AVP –
```

  The following is an example of a "Media-agnostic" SRTP stream:

```
v=0
m=- 4000 RTP/SAVP –
a=crypto:1 AES_CM_128_HMAC_SHA1_80
inline:Uvq8XAa9xflgN3SJW7ico2u4eHDllAmkMC8dtEE6
c=IN IP4 10.4.2.38
```

  A termination with a "Media-agnostic" stream, may contain additional T.38 media line and can be connected only to a termination with a "Media-agnostic" stream.

- **"Transport protocol-agnostic"** (or briefly "Transport-agnostic") on these streams - the MG may not conclude from signaled SDP information on the transported protocol. As a result, the connection between the terminations is on the UDP level and no manipulation is applied to the media streams at all.

No signals/events can be requested on a "Transport-agnostic" stream and no RTP events (such as statistics, netfail etc) are notified on these streams.

Note: Only UDP layer transport agnostic streams are currently supported.

The following is a SDP description of a "Transport-agnostic" stream:

```
v=0
c=IN IP4 10.4.2.38
m=- 4000 udp –
```

A termination with a "Transport-agnostic" stream may contain additional T.38 media lines and can be connected only to a termination with "Transport-agnostic" stream.

The AudioCodes gateway supports three types of connections between the IP-to-IP context's terminations:

- **UDP level IP-IP connection:** The connection between the context's ephemeral terminations is mediated on the UDP layer. No manipulation is applied to the media streams. This type of connection is used for "Transport-agnostic" streams. This type of a connection does not support::
  - RTP/RTCP statistics report to be requested on the termination
  - Signals/Events
  - SRTP encryption manipulations
- **RTP level IP-to-IP connection (RTP Forwarding):** The connection between the context's ephemeral terminations is mediated on the RTP layer. No transcoding is applied on the media streams. This type of connection is used for "Media-agnostic" streams and for media aware streams, if no transcoding is required. In any case that transcoding is required, (e.g.,when transcoding is needed or media level events such as RFC 2833 are to be detected) a TDM based IP-to-IP connection is made. Supported media capabilities for this connection are:
  - RTP level basic validation - size, version, sequence numbers etc.
  - Media broken connection (Packages: Net fail, ADID)
  - Media latching (ipnapt package)
  - RTP statistics calculation and reporting

- **TDM based IP-to-IP connection:** The connection between two ephemeral terminations which is used in order to make full transcoding between the termination streams. This connection type is used if media transcoding is required. The interworking of the media streams that have the following media characteristics requires full transcoding:
  - When the streams have different coders
  - DTMF/MFdetection or generation is required
  - Silence detection or generation is required
  - Different RTP characteristics exist on each stream - redundancy, p time, specific coders attributes
  - Different transport characteristics (transport types) for DTMF, FAX, modem or caller-ID

Note: When an IP-to-IP context UDP connection is used, only the duration (net/dur) statistics are available. Other statistics are reported with a value equal to 0.

The IP-to-IP interworking context 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.

7.2.5.7 Narrow Band IP Interface Support on IMS System

Support for Narrow Band (NB) IP interfaces has been added to the IP Multimedia Subsystem (IMS). The transport supported is only IP. Both Narrowband (NB) and Wideband versions of 3GPP AMR are supported.

The supported configurations are:

- Gateway (IP to TDM)
- Transcoding (IP to IP): 3GPP AMR NB/WB to G.711/G.722/AMR RFC 3267 and vice versa

The following H.248 features are now supported:

- Q.1950 Bearer Characteristics Package (BCP)
- Q.1950 Generic Bearer Connection
- Q.1950 Bearer Network Connection Cut Through
- Q.1950 Bearer Control Tunneling
- 3GUP Package (ThreeGUP)
- IPBCP tunneling for SDP delivery as defined in Q.1970

The following is an example of a call flow which transcodes between 3GPP AMR and the IETF AMR:

```
MEGACO/1 [10.3.3.41]:2944
    Transaction = 20 {
Context = $ {
            Add = $
                    {
                    Media {
                        LocalControl {
                            mo=sr,
                            BCP/BNCChar = IP/RTP,
                            BT/TunOpt = 1,
                            threegup/mode = Supp,
                            threegup/upversions = [4],
                            threegup/delerrsdu = NA,
                            threegup/interface = cn,
                            threegup/initdir = out
                        },
                        Local {
                            v=0
                            c=IN IP4 $
                            m=audio $ RTP/AVP 125
                    a=rtpmap:125 AMR/8000/1
                    a=fmtp:125 mode-set=7
```

```
                            a=silencesupp:off
                        }
                    },
                    Events = 1111 {gb/bncchange,
                                   bt/tind,
                                   g/cause
                    }
                    },
            Add = $
                    {
                    Media {

                        Local {
                            v=0
                            c=IN IP4 $
                            m=audio $ RTP/AVP 125
                  a=rtpmap:125 AMR/8000/1
                  a=fmtp:125 mode-set=7
                            a=silencesupp:off
                        }
                    }

                    }
    }}



MEGACO/1 [10.3.3.41]:2944
    Transaction = 20 {
Context = $ {
            Add= gws1c3,
            Add = $
                    {
                    Media {
                        LocalControl {
                            mo=inactive,
                            BCP/BNCChar = IP/RTP,
                            BT/TunOpt = 1,
                            threegup/mode = Supp,
                            threegup/upversions = [4],
                            threegup/delerrsdu = NA,
                            threegup/interface = CN,
                            threegup/initdir = in
                        },
                        Local {
                            v=0
                            c=IN IP4 $
                            m=audio $ RTP/AVP 125
                  a=rtpmap:125 AMR/8000/1
                  a=fmtp:125 mode-set=7
                            a=silencesupp:off
```

```
                        }
                    },
                    Events = 1111 {gb/bncchange,
                                   bt/tind,
                                   g/cause
                },
                    Signals {
                    BT/BIT {
                     BIT = 2020"
v=0
c=IN IP4 10.4.10.101
a=ipbcp:1 Request
m=audio 4060 RTP/AVP 125
a=rtpmap:125 AMR/8000/1
a=fmtp:125 mode-set=7
a=rtcp-xr:voip-metrics
a=silenceSupp:off - - - -
a=ptime:20
a=ptime:20

"

                    }
                }
                    }
    }}


MEGACO/1 [10.3.3.41]:2944
    Transaction = 23 {
Context = * {
            Modify = gwrtp/6
                {

                Signals {
                    BT/BIT {
                     BIT=2020"
```

```
v=0
c=IN IP4 10.4.10.101
a=ipbcp:1 accepted
m=audio 4080 RTP/AVP 125
a=rtpmap:125 AMR/8000/1
a=fmtp:125 mode-set=7
a=rtcp-xr:voip-metrics
a=silenceSupp:off - - - -
a=ptime:20
"
                    }
                }
            }
    }}
```

7.2.5.8 Lawful Interception Support

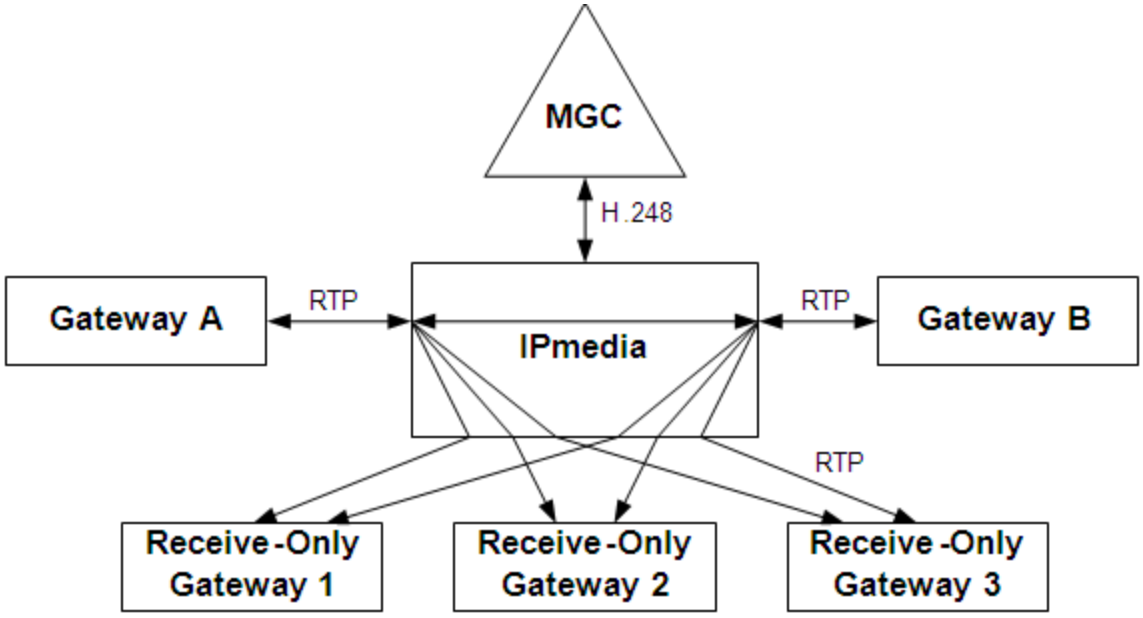
Note: Activating this feature requires the BCT/CALEA Feature key.

7.2.5.8.1 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 53: 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.

- Perform eight ephemeral add operations for BCT endpoints/terminations. 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.)
- Define the following topology between the endpoints as follows:
  - One 'two-way' link command to bridge Gateway A and Gateway B together ("Bothway" topology between endpoints for Gateway A and endpoint for Gateway B).

- Three 'one-way' link commands for enabling the sending of packets from Gateway A to each of the receive-only gateways ("OneWay" between the endpoints for Gateway A and three endpoints in send-only mode for the three receive only gateways).
- Three 'one-way' link commands for enabling the sending of packets from Gateway B to each of the receive-only gateways ("OneWay" between the endpoints for Gateway B and additional three endpoints in send-only mode for the three receive only gateways).

When all terminations have been set up with the same configuration (i.e. coder, payload types, DTMF transport type) and Fax Transport Type is "transparent", terminations are connected via the UDP layer. In this case, no jitter or transcoding processing is performed, which eliminates any latency or signal distortion. Otherwise, DSP resources are allocated for this call which may cause jitter, latency and signal distortion.

Another option for creating BCT is to use the proprietary context attribute package, as described in 'AudioCodes H.248 Proprietary Packages' on page 675. This package enables reserving context and resources for BCT. In this reserved mode, terminations are always connected via the UDP layer, discarding termination/media configurations. Additionally, SDP negotiation is always successful even on unsupported/unknown coders.

Notes:

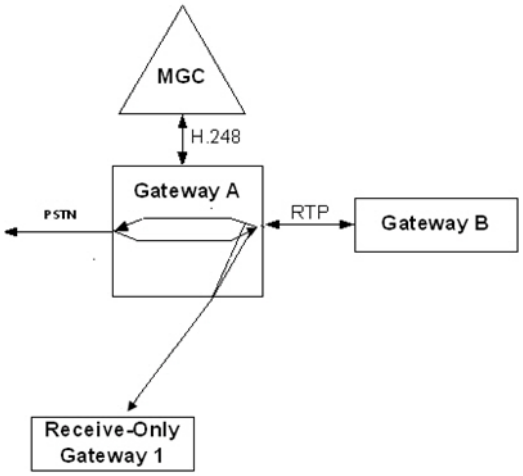
- Media-agnostic SDP, where Media Type and Coder are omitted, is not yet supported.
- BCT is applicable to IPM blades only.

7.2.5.8.2 Communications Assistance for Law Enforcement Act (CALEA)

CALEA Electronic Surveillance enables the conduct of lawfully-authorized electronic surveillance and implemented by AudioCodes proprietary "CALEA" package, as described in the 'AudioCodes H.248 Proprietary Packages' on page 675 or by using the "CCI" package as defined in H.248.60.

A CALEA Termination is an RTP termination with an electronic surveillance call content connection identifier. The CCCID property of the CALEA package and the CCID property of CCI can define the electronic surveillance call content connection identifier of the termination/stream. CALEA Electronic Surveillance is implemented by adding a CALEA termination to an existing call and connecting it to the already existing RTP termination in a "OneWay" topology. The connection sides are not affected while Electronic Surveillance is performed.

Figure 54: Basic Call with CALEA Electronic Surveillance



While Electronic Surveillance is activated, both bi-directional connection RTP streams are duplicated and sent to the remote destination (LEA server). The copied packets are transmitted as a stream of UDP/IP datagram to the IP address and port specified in the remote descriptor and includes the call connection identifier (CCID) configured in the CALEA termination and the original received and sent RTP packets. For the format of the CALEA packets see Appendix H of 'www.packetcable.com/downloads/specs/PKT-SP-TGCP-I10-050812.pdf' or the official CALEA web site.

Note: CALEA is implemented on both TGW and BGF (IP-to-IP context and PSTN-to-IP context) solutions.

7.2.5.9 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.

7.2.5.9.1 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.

7.2.5.9.2 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.

7.2.5.9.3 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.

7.2.5.9.4 The PoC Events

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.

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.

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.

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.

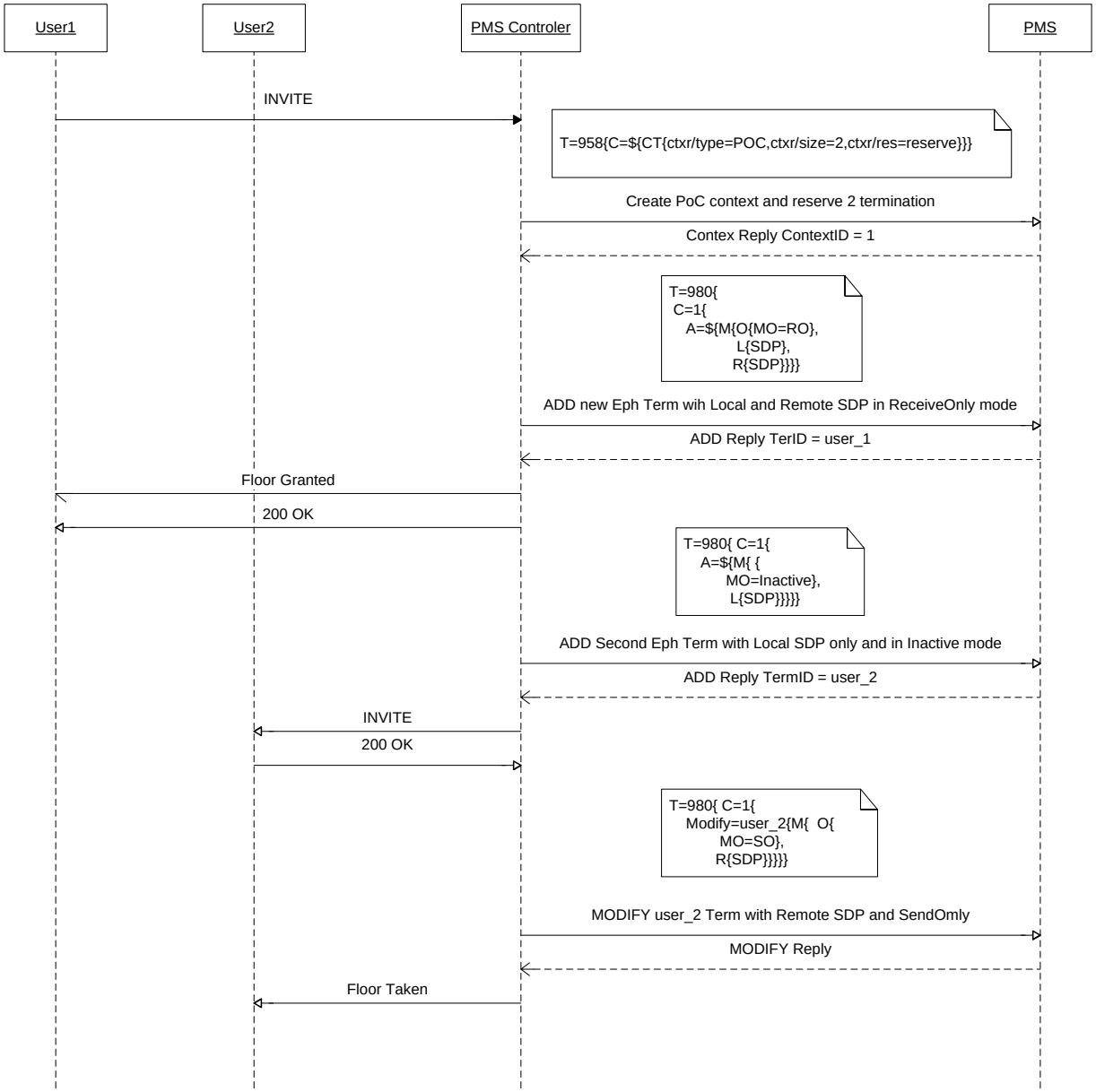
The event will be requested using the *acpoc* proprietary package with the First Media Packet (fmp) Event Id. The event should be requested on the active speaker ephemeral termination (termination in receive only mode) in the PoC context.

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.

7.2.5.9.5 Call Flows

Call Establishment with 2 Participants

Figure 55: Call Establishment With 2 Participants



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**

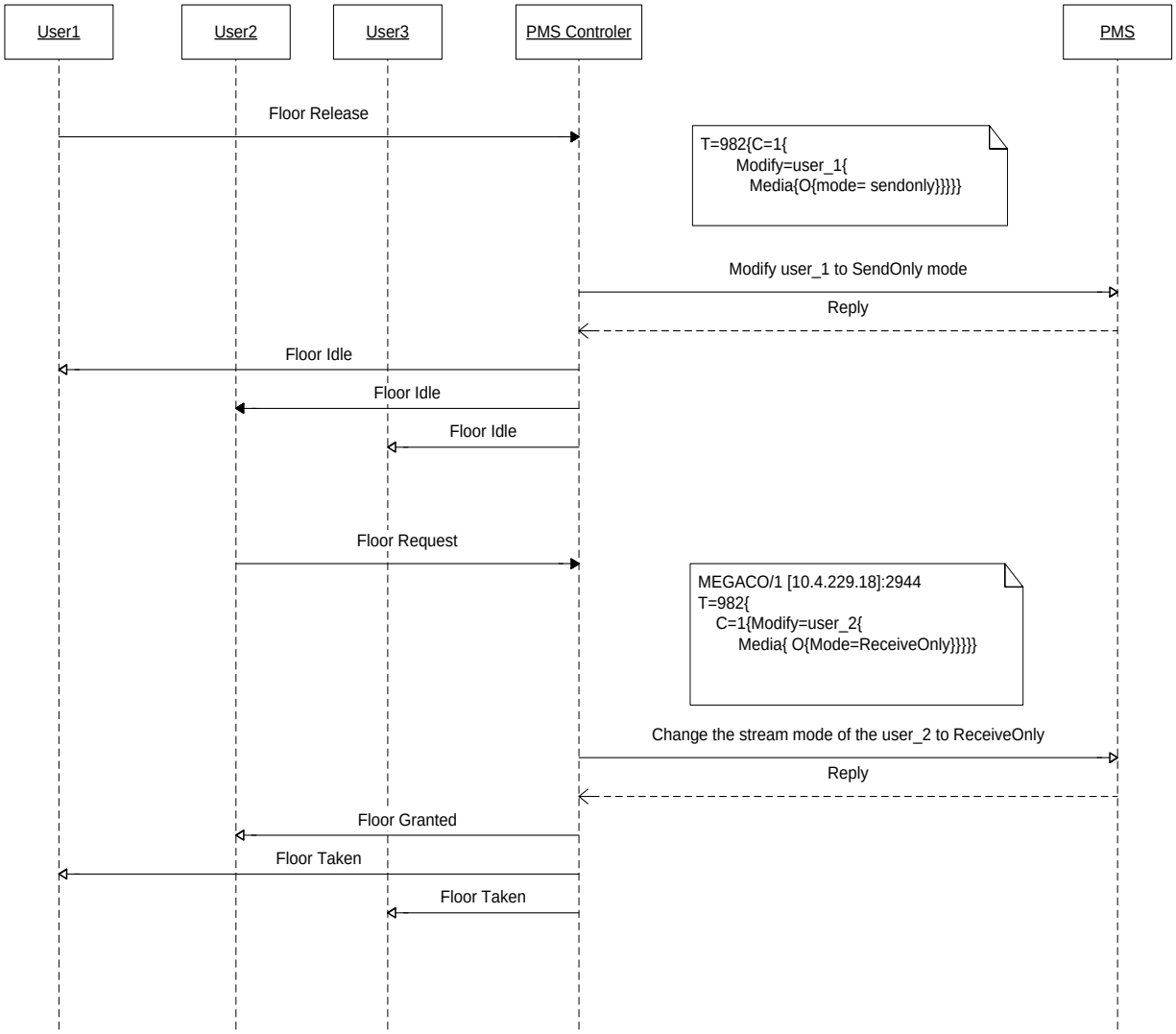
```
;;Command
MEGACO/2 [10.4.10.226]:2944
T=980{
    C=1{
    Modify=te/tepool1/1{
        Media{
            O{MO=SendOnly},
            R{v=0
                c=IN IP4 10.4.4.34
                m=audio 4010 RTP/ AVP 18
a=fmtp:18 annexb=yes
a=ptime:10
a=silencesupp:off - - - -
            }
```

```
        }
    }
}}

;;Reply
MEGACO/2 [10.4.4.32]:2944
P=369{
C=1{
MF = te/tepool1/1
}}
```

7.2.5.9.6 Floor Changing

Figure 56: Floor Changing



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
}}
```

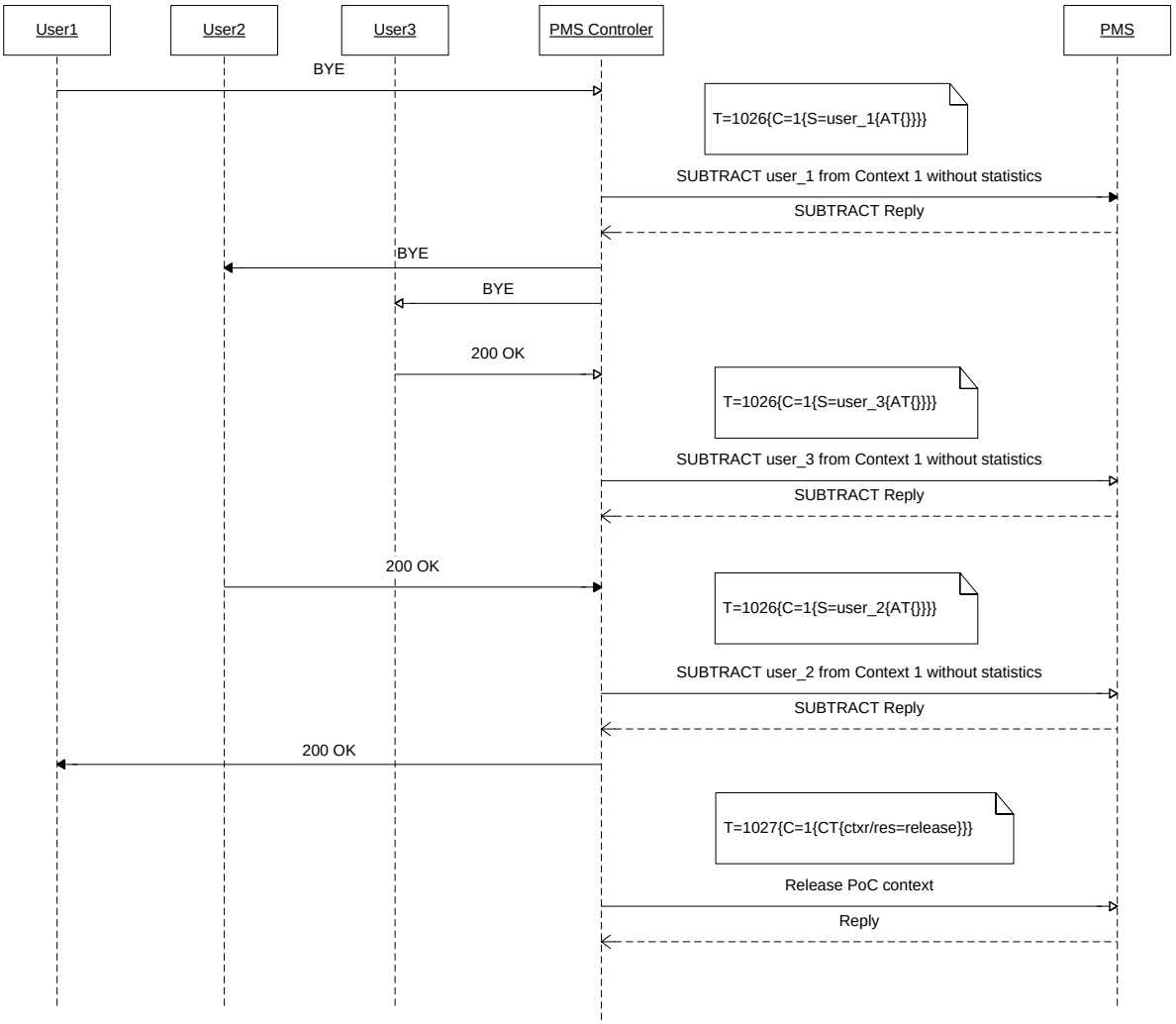
Modify the stream mode of the user_2 to ReceiveOnly

```
;;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
```

7.2.5.9.7 Call Release

Figure 57: Call Release



Notes:

- The subtraction should be done without statistics.
- 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}}}
```

7.2.5.10 Garbage Collection

In some rare cases, Contexts may not be properly released, hence becoming invalid objects. In order to release such objects back to the contexts pool, a Garbage Collection mechanism scans the Contexts List every 24 hours, looking for invalid contexts to release. In order to use this feature, set the configuration parameter cpGarbageCollectionTime. It has a default value of "0" (no automatic deletion). Possible values (in days) are from 1 to 100. Only invalid contexts which were created more than cpGarbageCollectionTime days before the daily scan, are released.

7.2.6 Compliance

7.2.6.1 Supported 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 'http://www.ietf.org/' - refer to 'Search RFC Ed 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.

7.2.6.1.1 General Packages

General Packages

Package Name	Pkg Id	Standard	Supporting Devices	First Supported Version	Note
Generic Package	g	H.248.1	All	3.8	For more information about g/cause event support, refer to 'Reporting Media Failure' on page 583.
Base Root Package	root	H.248.1	All	3.8	
Tone Generator Package	tonegen	H.248.1	All	3.8	For more information. refer to 'Call Progress Tone Signals' on page 578.
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	For more information. refer to 'Call Progress

General Packages

					Tone Signals' on page 578.
Call Progress Tones Detection Package	cd	H.248.1	All	3.8	
Network Package	nt	H.248.1	All	3.8 6.2 for Quality Alert event	For more information about net/netfail event support, refer to 'Reporting Media Failure' on page 583. For more information about Quality Alert events, refer to 'Reporting Media Quality Alert' on page 584.
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	For more information, refer to 'Reporting Fax Events' on page 583.
IP Fax Package	ipfax	H.248.2	All	4.2	
Generic Announcement Package	an	H.248.7	All	3.8	For more information. refer to 'Call Progress Tone Signals' on page 578.
Congestion Handling Package	chp	H.248.10	6310, 8410, Mediant 3000	5.2	
Overload Control Package	ocp	H.248.11	6310, 8410, Mediant 3000	5.2	
Inactivity Timer Package	it	H.248.14	All	4.6	

General Packages

Extended DTMF Detection Package	xdd	H.248.16	All	4.2	For more information, refer to 'Digits Collection Support' on page 582.
Enhanced DTMF Detection Package	edd	H.248.16	All	4.2	For more information, refer to 'Digits Collection Support' on page 582.
Multi-frequency tone generation package	mfg	H.248.24	All	4.2	For more information. refer to 'Call Progress Tone Signals' on page 578.
Multi-frequency tone detection package	mfd	H.248.24	All	4.2	
Conferencing Tones Generation Package	conftn	H.248.27	All	4.8	For more information. refer to 'Call Progress Tone Signals' on page 578.
Carrier Tones Generation Package	carr	H.248.27	All	5.0	For more information. refer to 'Call Progress Tone Signals' on page 578.
RTCP XR Base Package	rtcpxr	H.248.30	All	5.0 6.2 – for TP-6310/TP-8410/Mediant 3000 products	For more information, refer to 'RTCP-XR Support (H.248.30)' on page 616.
RTCP XR Burst Metrics Package	xrbm	H.248.30	TP-1610, IPM-1610, Mediant 2000	5.0 6.2 – for TP-6310/TP-8410/Mediant 3000 products	For more information, refer to 'RTCP-XR Support (H.248.30)' on page 616.

General Packages

Detailed Congestion Reporting Package	dcr	H.248.32	TP-1610, IPM-1610, Mediant 2000	5.0	For more information, refer to 'Reporting Congestion in Performance Monitoring' on page 68.
Hanging Termination Detection Package	hangterm	H.248.36	All	5.2	
IP NAT Traversal Package	ipnapt	H.248.37	All	5.2	For more information, refer to 'Supporting Network Address and Port Translation' on page 585.
Basic Call Progress Tones Generator with Directionality	bcg	Q.1950	All	4.6	For more information. refer to 'Call Progress Tone Signals' on page 578.
Basic Services Tones Generator Package	srvtn	Q.1950	All	4.6	For more information. refer to 'Call Progress Tone Signals' on page 578.
Expanded Services Tones Generation Package	xsrvtn	Q.1950	All	4.6	For more information. refer to 'Call Progress Tone Signals' on page 578.
Expanded Call Progress Tones Generator Package	xcg	Q.1950	All	4.6	For more information. refer to 'Call Progress Tone Signals' on page 578.

General Packages

Enhanced Alerting	alert	H.248.23	All	6.0	Only alert/cw signal is supported. For more information. refer to 'Call Progress Tone Signals' on page 578.
Differentiated Services Package	ds	ETSI TS102.333 Annex A H.248.52	All	5.0	For more information, refer to 'Media Path QoS Support' on page 585.
VLAN Package	vlan	ETSI TS102.333 Annex G H.248.56	All	5.6	For more information, refer to 'Media IP Address Allocations' on page 585.
CALEA Package	calea	AudioCodes Proprietary Package	All	5.8	For package definition refer to 'CALEA Package' on page 675. For more information refer to 'Electronic Surveillance (CALEA)' on page 513.
CCI Package	cci	H.248.60	All	6.2	This package is used to implement CALEA functionality. For more information refer to 'Electronic Surveillance (CALEA)' on page 513.
IP Domain Connection	ipdc	H.248.41	All	5.8	For more

General Packages

Package					information, refer to 'Media IP Address Allocations' on page 585.
Application Data Inactivity Detection Package	adid	H.248.50	All	6.0	For more information refer to 'Reporting Media Failure' on page 583.

7.2.6.1.2 Trunking Gateway Packages

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	For more information and call flow examples refer to 'CAS Calls Support' on page 617.
Basic CAS addressing package	bcasaddr	H.248.25	All	4.8	For more information and call flow examples refer to 'CAS Calls Support' on page 617.
Robbed bit signalling package	rbs	H.248.25	All	4.8	Signals generation only from 5.2. For more information and call flow examples refer to 'CAS Calls Support' on page 617.
Operator services and emergency services package	oses	H.248.25	All	4.8	For more information and call flow examples refer to 'CAS Calls Support' on page 617.

Operator services extension package	osext	H.248.25	All	5.2	For more information and call flow examples refer to 'CAS Calls Support' on page 617.
International CAS Package	icas	H.248.28	All	4.6	For more information and call flow examples refer to 'CAS Calls Support' on page 617.
Quality Alert Ceasing Package	qac	H.248.13	All	6.2	For more information refer to 'Reporting Media Quality Alert' on page 584.
CAS Blocking Package	casblk	H.248.28	All	4.6	For more information and call flow examples refer to 'CAS Calls Support' on page 617.
International CAS Compelled Package	icasc	H.248.29	All	4.6	For more information and call flow examples refer to 'CAS Calls Support' on page 617.

Note: The following Media Server Packages Table is only applicable to **IPmedia.**

7.2.6.1.3 Media Server Packages

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	aass, aasb	H.248.9	All IPM	4.2	For more information, refer to 'Interactive Voice Response (IVR)' on page 626.
AAS Digit Collection Package	aasdc	H.248.9	All IPM	4.2	For more information, refer to 'Interactive Voice Response (IVR)' on page 626.
AAS Recording Package	aasrec	H.248.9	All IPM	4.2	For more information, refer to 'Interactive Voice Response (IVR)' on page 626.
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	

Context Reservation Package	ctxr	AudioCodes Proprietary Package	All IPM	4.2	For package definition, refer to 'Proprietary Packages' on page 675.
ACL Proprietary PoC Package	ACPoC	AudioCodes Proprietary Package	All IPM	5.4	For package definition, refer to 'Proprietary Packages' on page 675.
Test Trunks Package	nnttrk	AudioCodes Proprietary Package	All IPM	4.2	For package definition, refer to 'Proprietary Packages' on page 675. For implementation details refer to Test Trunks Package.

7.2.6.1.4 Analog and Access Gateway Packages

Analog/Access Gateway Packages

Package Name	Pkg Id	Standard	Supporting Devices	First Supported Version	Note
Analog Line Supervision Package	al		Analog and Access GW	5.8	For implementation details refer to 'V5' on page 449.
Enhanced Alerting	alert	H.248.23	Analog and Access GW	5.8	For implementation details refer to 'V5' on page 449.
Analogue Display Signalling	andisp	H.248.23	Analog and Access GW	5.8	For implementation details refer to 'V5' on page 449.
Extended Analog Line Supervision	Xal	H.248.26	Access GW	5.8	For implementation details refer to 'V5' on page 449.
Automatic Metering	amet	H.248.26	Access GW	5.8	For implementation details refer to 'V5' on page 449.

7.2.6.1.5 IMS Gateway Packages

IMS Gateway Packages

Package Name	Pkg Id	Standard	Supporting Devices	First Supported Version	Note
Bearer Characteristics	bcp	Q.1950	IMS GW	**6.6**	
Generic Bearer Connection	gb	Q.1950	IMS GW	**6.6**	
Bearer Network Connection Cut Through	bnct	Q.1950	IMS GW	**6.6**	
Bearer Control Tunneling	bt	Q.1950	IMS GW	**6.6**	
3GUP	threegup	3GPP TS 29.232	IMS GW	**6.6**	

7.2.6.2 Compliance Matrix

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.

MEGACO Compliance Matrix

Reference (in RFC 3015)	Item	Support	Comments
7	**Commands supported:**		
	Add	Yes	
	Modify	Yes	
	Subtract	Yes	
	Move	Yes	
	AuditValue	Yes	
	AuditCapabilities	Yes	
	Notify	Yes	
	ServiceChange	Yes	
7.1	**Descriptors**		
7.1.1	**Specifying Parameters:**		
	Fully specified	Yes	
	Under specified	Yes	
	Over specified	Yes	

MEGACO Compliance Matrix

	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	
	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	

MEGACO Compliance Matrix

	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		
	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	

MEGACO Compliance Matrix

	DigitMap	Yes	
	Statistics	Yes	
	Packages	Yes	
	EventBuffer	Yes	
	Empty descriptor	Yes	
7.1.13	**Service Change Descriptor**		
	ServiceChangeMethod	Yes	
	ServiceChangeReason	Yes	
	ServiceChangeAddress	Yes	
	ServiceChangeDelay	Yes	
	ServiceChangeProfile	Yes	
	ServiceChangeVersion	Yes	
	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	

MEGACO Compliance Matrix

	Oneway	Yes	
	Bothway	Yes	
	CHOOSE wildcard	Yes	
	ALL wildcard	Yes	
7.2	**Command API**		
7.2.1	**Add**		
	Termination ID	Yes	
	MediaDescriptor	Yes	
	ModemDescriptor	No	
	MuxDescriptor	No	
	EventsDescriptor	Yes	
	SignalsDescriptor	Yes	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	

MEGACO Compliance Matrix

	MuxDescriptor	No	
	EventsDescriptor	Yes	
	SignalsDescriptor	Yes	Up to 2 signals per channel Up to 30 signals in a signal list
	DigitMapDescriptor	Yes	
	AuditDescriptor	Yes	
7.2.5	**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	
	Statistics	Yes	
	Packages	Yes	

MEGACO Compliance Matrix

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	
	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	

MEGACO Compliance Matrix

	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	
	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	

MEGACO Compliance Matrix

	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	
	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	

MEGACO Compliance Matrix

	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**		
	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	

MEGACO Compliance Matrix

	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**		
		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	

7.2.6.3 Proprietary Packages

7.2.6.3.1 CALEA Package

1. Scope

This recommendation defines a package that provides a way to support the electronic surveillance capability (CALEA) as defined in PKT-SP-ESP-I01-991230 (See below).

According to the above specification, when a call is subject to electronic surveillance, all valid media packets sent and received on it should be replicated and forwarded to the Electronic Surveillance Delivery Function after inclusion of a Call Content Connection Identifier. This replication is done on the network level, in which the original packet is wrapped in a new envelope, including in it the CCCI Identifier.

2. References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- ITU-T Recommendation H.248.1 (9/2005), Gateway Control Protocol: Version 3
- Packet Cable Interface Specification, Packet Cable PSTN Gateway Call Signaling Protocol Specification PKT-SP-TGCP-I02-011221
- Packet Cable Interface Specification, Packet Cable Electronic Surveillance Specification, PKT-SP-ESP-I01-991230

3. Terms and Definitions

None

4. Abbreviations

This Recommendation uses the following abbreviations:

- **CALEA** - Communications Assistance for Law Enforcement Act
- **MGC** - Media Gateway Controller
- **RTP** - Real-Time Transport Protocol

5. Conventions

None

6. Electronic Surveillance Support Package

Package Name:	CALEA package
PackageID:	calea, 0x???
Description:	This package defines parameters and mechanism for supporting the electronic surveillance capability (CALEA).
Version:	1
Designed to be extended only:	no
Extends:	None

6.1 Properties

6.1.1 Call Content Connection Identifier

Property Name:	Electronic surveillance call content connection identifier
PropertyID:	cccid, 0x0001

Description: This property is used to identify the call by the receiving surveillance authority.

Type: Octet string of 8 hexa digits

Possible values: 0x00000000 to 0xFFFFFFFF

Default: 0

Defined in: LocalControl

Characteristics: Read/Write

6.2 Events

None

6.3 Signals

None

6.4 Statistics

None

6.5 Error Codes

None

6.6 Procedures

In order to add electronic surveillance to the call, the MGC should add new RTP terminations to the existing call (context) with the *cccid* property and connect it to the already existing RTP terminations by using the *"OneWayBoth"* topology. This way, all incoming and outgoing packets will be duplicated and sent to the surveillance destination, defined in the remote descriptor. The media parameters MUST be defined as agnostic data (where the coder is not defined). If an attempt is made to connect two terminations with media-aware streams, a 447 "Descriptor illegal for command" error will be returned.

Below is a call flow example which demonstrates how to create a call with CALEA surveillance.

Notes:

- The package is applicable only on the RTP termination, as this is the packet's side.
- For gateways which do not support version 3 of the protocol and therefore can't use the "onewayboth" topology, it is suggested to use the "Oneway" topology instead, and internally handle the knowledge that as this is a CALEA surveillance, The exterior side should be replicated.
- Surveillance termination MUST be added with calea/cccid property.

Call Flow Example:

Setting a termination under surveillance starts by setting the original call:

```
MEGACO/2 [10.2.1.228]:2944
   Transaction = 8974 {
       Context = $ {
          Add = Phys/1,
          Add = $ {Media {LocalControl {
                    Mode = Receiveonly},
                    Local {
v=0
c=IN IP4 $
m=audio $ RTP/AVP 0
 }}}}}
!/2 [10.4.4.47]:2944
P=8974{
C=1{
A = Phys/1,A = RTP/0{
M{
L{

v=0
c=IN IP4 10.4.4.47
m=audio 4000 RTP/AVP 0
a=ptime:20
a=silencesupp:off - - - -
}}}}}
```

Now, the MGC would like to put the call originator under surveillance. In order to do that, the MGC will add a new termination to the call, with the *cccid* property, and connects it with *OneWayBoth* to the RTP termination. (Note: The package is applicable only on the RTP termination, as this is the packets side):

```
MEGACO/2 [10.2.1.228]:2944
   Transaction = 8975 {
       Context = 1 { Topology{$, Phys/0, isolate, RTP/0,
$,OneWayBoth},
          Add = $ {Media {
LocalControl { Mode = Sendonly, calea/cccid=0xb564a37d},
                       Local {
v=0
c=IN IP4 $
m=audio $ RTP/AVP -
 },
                     Remote{
v=0
c=IN IP4 192.155.40.10
m=audio 5443 RTP/AVP -
}}}}}

!/2 [10.4.4.47]:2944
P=8975{
C=1{ Topology{ RTP/1, Phys/0, isolate, RTP/0, RTP/1, OneWayBoth},
A = RTP/1{
```

```
M{
L{

v=0
c=IN IP4 10.4.4.47
m= audio 4010 RTP/AVP -
}}}}}
```

Once this has been done, you can now modify the original RTP termination with the remote information, and the surveillance is on the way.

7.2.6.3.2 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.

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

2. Events

None

3. Signals

None

4. Statistics

None

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.

7.2.6.3.3 PoC Proprietary Package

Package Name: ACL proprietary PoC Package

PackageID: ACPoC

Description: This package defines the interaction 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

1. Events

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

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

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

7.2.6.3.4 Test Trunk Package

Package Name: Test Trunks Package

PackageID: nnttrk (0x0095)

Description: This package provides support for trunk tests used to verify connections to adjacent switching offices.

Version: 2

Extends: none

Properties

None

Events

Test Complete

Event Name: Test Complete

EventID: tc (0x0001)

Description: Indicates the completion of a trunk test.

EventsDescriptor parameters: None

ObservedEventsDescriptor parameters:

Loss

Parameter Name: Loss

ParameterID: loss (0x0001)

Description: Milliwatt tone loss measurement.

Type: Double

Optional: Yes

Possible values: A four-byte whole number and 4 byte fraction representing the loss in dB

Default: 0.0

Noise

Parameter Name: Noise

ParameterID: noise (0x0002)

Description: Noise (quiet termination) measurement.

Type: Double

Optional: Yes

Possible values: A four-byte whole number and 4 byte fraction representing the noise in dB

Default: 0.0

Far End to Near End Loss

Parameter Name: Far End to Near End Loss

ParameterID: fnloss (0x0003)

Description: Far end to near end milliwatt tone loss measurement.

Type: Double

Optional: Yes

Possible values: A four-byte whole number and 4 byte fraction representing the loss in dB

Default: 0.0

Near End to Far End Loss

Parameter Name: Near End to Far End Loss

ParameterID: nfloss (0x0004)

Description: Near end to far end milliwatt tone loss measurement.

Type: Double

Optional: Yes

Possible values: A four-byte whole number and 4 byte fraction representing the loss in dB

Default: 0.0

Far End to Near End Noise

Parameter Name: Far End to Near End Noise

ParameterID: fnnoise (0x0005)

Description: Far end to near end noise measurement.

Type: Double

Optional: Yes

Possible values: A four-byte whole number and 4 byte fraction representing the noise in dB

Default: 0.0

Near End to Far End Noise

Parameter Name: Near End to Far End Noise

ParameterID: nfnoise (0x0006)

Description: Near end to far end noise measurement.

Type: Double

Optional: Yes

Possible values: A four-byte whole number and 4 byte fraction representing the noise in dB

Default: 0.0

Return Code

Parameter Name: Return Code

ParameterID: rc (0x0007)

Description: Summary of the test result.

Type: Enumeration

Optional: Yes

Possible values:		
"OK"	(0x0001) Successful completion	
"TO"	(0x0002) Test tone timed out	
"UT"	(0x0003) Test interrupted by user	

Default: OK

Test Fail

Event Name: Test Fail

EventID: tf (0x0002)

Description: Indicates the failure of a trunk test.

EventsDescriptor parameters: None

ObservedEventsDescriptor parameters:
Reason
Parameter Name: Reason
ParameterID: reason (0x0001)
Description: Integral reason code for the failure of the test.
Type: Integer
Optional: No
Possible values: 0 to 255
Default: None
Signals
Test Start
Signal Name: Test Start
SignalID: ts (0x0001)
Description: Directs the termination to start the specified test.
Signal Type: TimeOut
Duration: 180000 milliseconds (3 minutes)
Additional parameters:
Test Name
Parameter Name: Test Name
ParameterID: name (0x0001)
Description: Indicates the name of the trunk test to start
Type: Enumeration
Optional: No
Possible values: "T100" (0x0001) Test Trunk Test 100
"T102" (0x0002) Test Trunk Test 102
"T105" (0x0003) Test Trunk Test 105
"TWSP"(0x0004) Test Trunk Tone Swap Test
"T904" (0x0005) Test Trunk Test 904
Default: None
Direction
Parameter Name: Direction
ParameterID: dir (0x0002)
Description: Indicates which direction the test is to be run
Type: Enumeration
Optional: No
Possible values: "TERM" (0x0001) Terminating Test
"ORIG" (0x0002) Originating Test
Default: None
Frequency
Parameter Name: Frequency
ParameterID: freq (0x0003)
Description: Indicates the frequency of the test tone
Type: Integer
Optional: Yes
Possible values: Positive number of Hertz
Default: Provisioned
Power

Parameter Name: Power

ParameterID: pwr

Description: Indicates the power level of the test tone

Type: Double

Optional: Yes

Possible values: A four-byte whole number and 4 byte fraction representing the power level in dB

Default: Provisioned

Statistics

None

Procedures

H248 Signaling for Tone Swap Test

1. GWC reserves AMS resources by sending a request to add a physical termination from the test pool:

```
MEGACO/1 [47.142.112.32]:2944 Transaction=3001{
     Context=${
          Add=tlt/${
               Signals{}
          }
     }
}
```

2. AMS replies with context id and termination id:

```
MEGACO/1 [47.172.112.16]:2944 Reply=3001{
     Context=3{
          A=tlt/tltpool0/2
     }
}
```

3. GWC requests that the ephemeral be added to the context and sends partial SDP:

```
MEGACO/1 [47.142.112.32]:2944 Transaction=3002{
   Context=3{
      Add=${
         Media{
            LocalControl{
               Mode= SendReceive,
               ReservedGroup=ON,
               ReservedValue=ON
            },
            Local{SDP},
            Remote{SDP}
         }
      }
   }
}
```

4. AMS replies with ephemeral name and completed SDP:

```
MEGACO/1 [47.172.112.16]:2944 Reply=3002{
     Context=3{
          Add=te/tepool10/5{
               Media{
                    Local{SDP}
               }
          }
     }
}
```

5. GWC requests that the tone swap test begin with a modify:

```
MEGACO/1 [47.142.112.32]:2944 Transaction=3003{
     Context=3{
          Modify=tlt/tltpool0/2{
               Media{
                    LocalControl{
                         Mode=SendReceive
               },
               Events=123{
                    nnttrk/tc{KeepActive},
                    nnttrk/tf,
               },
               Signals{
                    nnttrk/ts{
                         name=TSWP,
                         dir=ORIG,
                         freq=1004,
                         pwr=0.0,
                         SignalType=TimeOut,
                         Duration=xxxx
                    }
               }
          }
     }
}
```

6. AMS acknowledges (and starts the tone swap test):

```
MEGACO/1 [47.172.112.16]:2944 Reply=3003{
     Context=3{
          Modify=tlt/tltpool0/2
     }
}
```

The 1004 Hz tone is played periodically to the far end.

7. When a 1004 Hz tone from the far end is successfully detected, the AMS reports the test complete event:

```
MEGACO/1 [47.172.112.16]:2944 Transaction=1400{
     Context=3{
```

```
          Notify=tlt/tltpool0/2{
               ObservedEvents=123{
                    19700101T00045775 : nnttrk/tc{
                         fnloss=6.0
                    }
               }
          }
     }
}
```

If an error is detected, the AMS reports “test failed”:

```
MEGACO/1 [47.172.112.16]:2944 Transaction=1400{
     Context=3{
          Notify=tlt/tltpool0/2{
               ObservedEvents=123{
                    19700101T00045775 : nnttrk/tf{
                         reason=xxxx
                    }
               }
          }
     }
}
```

8. In either case, the GWC acknowledges:

```
MEGACO/1 [47.142.112.32]:2944 Reply=1400{
   Context=3{
      Notify=tlt/tltpool0/2
   }
}
```

Messages 7 and 8 are repeated periodically while the test is active.

9. If the test is still active when the timeout timer expires, the AMS reports the “timeout” event:

```
MEGACO/1 [47.172.112.16]:2944 Transaction=1401{
     Context=3{
          Notify=tlt/tltpool0/2{
               ObservedEvents=123{
                    19700101T00045775 : nnttrk/tc{
                        rc=TO
                    }
                       }
                  }
             }
         }
```

10. After the timeout timer expires or when the core requests that the test be halted, the GWC tears down the test, beginning with the ephemeral:

```
MEGACO/1 [47.142.112.32]:2944 Transaction=3004{
     Context=3{
```

```
          Subtract=te/tepool0/5{
               Audit{}
          }
     }
}
```

11. AMS acknowledges:

```
MEGACO/1 [47.172.112.16]:2944 Reply=3004{
     Context=3{
          Subtract=te/tepool0/5
     }
}
```

12. GWC subtracts the physical termination:

```
MEGACO/1 [47.142.112.32]:2944 Transaction=3005{
     Context=3{
          Modify=tlt/tltpool0/2{
               Media{
                    LocalControl{
                         Mode=Inactive
                    }
               }
          },
          Subtract=tlt/tltpool0/2{
               Audit{}
          }
     }
}
```

13. AMS acknowledges:

```
MEGACO/1 [47.172.112.16]:2944 Reply=3005{
     Context=3{
          Modify=tlt/tltpool0/2,
          S=tlt/tltpool0/2
     }
}
```

H.248 Signaling for T904 Test

The first four messages are the same as in the tone swap case.

5. Near end—GWC requests that the tone swap test begin with a modify specifying originator:

```
MEGACO/1 [47.142.112.32]:2944 Transaction=3300{
     Context=3{
          Modify=tlt/tltpool0/2{
               Media{
                    LocalControl{
                         Mode=SendReceive
               },
               Events=123{
```

```
                    nnttrk/tc,
                    nnttrk/tf
               },
               Signals{
                    nnttrk/ts{
                         name=T904,
                         dir=ORIG,
               }
          }
     }
}
```

Far end—GWC requests that the tone swap test begin with a modify specifying terminator:

```
MEGACO/1 [47.142.112.32]:2944 Transaction=3300{
     Context=3{
          Modify=tlt/tltpool0/2{
               Media{
                    LocalControl{
                         Mode=SendReceive
               },
               Events,
               Signals{
                    nnttrk/ts{
                         name=T904,
                         dir=TERM
               }
          }
     }
}
```

Acknowledgement is the same as with the tone swap test. The test results are reported as follows:

```
Test successful:

MEGACO/1 [47.172.112.16]:2944 Transaction=1500{
     Context=3{
          Notify=tlt/tltpool0/2{
               ObservedEvents=123{
                    19700101T00045775 : nnttrk/tc{
                         fnloss=6.0
                    }
               }
          }
     }
}
```

```
Test unsuccessful:
```

```
MEGACO/1 [47.172.112.16]:2944 Transaction=1500{
     Context=3{
          Notify=tlt/tltpool0/2{
               ObservedEvents=123{
                    19700101T00045775 : nnttrk/tf{
                         reason=xxxx
                    }
               }
          }
     }
}
```

7.2.7 Solutions

Note: This Solutions section provides only a partial set of the solution for where our Gateway can be used.

7.2.7.1 VoIP Trunk Gateway (TGW)

A trunk media gateway provides interworking between a packet network and the PSTN at the trunk level. It converts from PSTN signaling to H.248 and from TDM payload to VoIP RTP (and visa versa). It can be used for backhauling TDM over IP, part of class 4 & 5 TDM switch replacements It enables the following capabilities:

- Wide Range of the Voice Coders
- Fax and Modem Support (T.38, ByPass)
- PSTN Signaling (CAS, MFCR2, MFCR1, transparent)
- Media and control Security

7.2.7.2 Advanced Media Server (AMS)

Note: The AMS is only applicable to IPmedia.

Advanced Media Server operates as the MRFP (Media Resource Function Processor) and provides the following capabilities:

- Interactive Voice Response (IVR)
- Bearer Channel Tandeming (BCT)
- Conferencing
- Test Trunk Support
- Video
- Transcoding

7.2.7.3 Border Gateway Function (BGF)

BGF (Border Gateway Function) is a packet-to-packet gateway for the media traffic. It provides the interface between two IP-transport domains. BGF may reside at the boundary between an access network and a core network or between two core networks.

The AudioCodes current BGF solution supports the control of the following functionality:

- NAPT (Network Address and Port Translation)
- NAPT-PT (NAPT and protocol translation) – Interworking between IPv4 and IPv6 networks
- Allocation of the IP address and port according to the IP realm/domain indication
- Hosted NAT traversal (Listen for incoming media and latch to the remote address information of that media)
- QoS marking for outgoing traffic
- Transcoding (Translation from one type of encoded media format to another different media format)

For a detailed description of individual functionality and a list of the supported packages, refer to the Feature Operation, IP-to-IP Interworking and Supported Packages sections.

Notes:

- BGF solutions can be configured as pure IP-to-IP gateway or co-exist with the Trunking Gateway and can run on the same blade as a combined PSTN-to-IP/IP-to-IP gateway.
- Pure IP-to-IP GW configuration is achieved by deactivating the PSTN via the Feature Key.
- BGF packet-to-packet functionality is supported under the SBC feature key only.

7.2.7.4 Access Gateway

V5.2 Access Core Gateway functionality provides cost–effective V5.2 LE replacement.It converts from V5.2 signaling to H.248 ITU standard analog packages and from TDM payload to VoIP RTP (and visa versa).

The AudioCodes V5 Access application aggregates legacy circuit-switched voice from the subscriber side, converts V5.2 protocol messages to H.248 standard analog packages and then hands them off to a softswitch and vice versa. The softswitch replaces the traditional Class 5 switchFor more information about this functionality refer to 'V5 Protocol' on page 449.

The following is a basic and partial list of the supported features:

- Offhook, onhook, flashhook - ‘al’ – Analog line supervision package
- Caller ID/FSK – Using “andisp” package
- Cadenced Ringing: Using Enhanced Alerting “alert” package (H.248.23)
- Call progress Tones and announcements: Dial tone , ringing, busy, special tones ,service announcements (“unrecognized number”, “All lines are busy” …)
- Digit collection packages: ‘dd’,’xdd’,mfd’
- T.38 Fax and RFC2833 DTMF Relay support
- 3xWay & call waiting toggling – via FlashHook reporting (Notify al/fl)

- CFW – All, No answer, Busy – “transparent to the GW”
- Optional Reverse Polarity, Normal Polarity - metering

7.2.7.5 High Definition Gateway

AudioCodes’ HD platform provides rich HD services. The platform provides the following capabilities:

- Wide Band coders
- True HD Transcoding between different wideband coders while retaining a wideband quality
- HD RTP Streaming - enables playback of high quality audio files

8 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 164.

Note:. The following sub-section on Voice Streaming Features is applicable to **ALL** devices.

8.1 Voice Streaming Features

The following summarizes the Voice Streaming features which are supported both on HTTP servers and NFS servers unless stated differently:

8.1.1 Supported File Formats

The voice streaming layer provides support for *.wav, *.au, *.raw file formats. The maximum header size of the file should be 150 bytes.

In *.wav format, only Mono mode and supported/known coders are supported. The maximum number of
non-data, non-fmt chunks can be up to 5.

8.1.2 Basic Streaming Play

A user may play a **.wav, *.au or *.raw* file from a remote server using G.711 coders.

8.1.2.1 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.

8.1.3 Working with Remote File Systems

A user may configure up to 16 remote file systems working with the device through NFS mounting.

8.1.4 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.

8.1.5 Combining HTTP and NFS Play / Record

A client may use any combination of HTTP/NFS play and HTTP/NFS record on the same channel.

8.1.6 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 164.

Notes:

- The following features are only applicable to **TP-260/UNI, TP-1610, IPM-260/UNI, IPM-1610 and IPmedia 2000**.
- The following features are relevant for both NFS and HTTP.

8.1.7 Play LBR Audio File

A user may play a file using low bit rate coders for **.wav* and **.raw* files.

8.1.8 Basic Record

A user may record a *.wav, *.au or *.raw files to a remote server using G.711 coders.

8.1.9 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.

8.1.10 Record Files Using LBR

A user may record a file using low bit rate coders for **.wav* and **.raw* files.

Notes:

- The following features are only applicable to **TP-6310, TP-8410, IPM-6310, IPM-8410 and IPmedia 3000**.
- The following features are relevant only for working with NFS servers.

8.1.11 Basic Record

A user can record a **.wav, *.au* or **.raw* files to a remote server using G.711 coders.

8.1.12 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.

8.2 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?'.

Notes:

- The following "stale cache" problem exists when playing a recorded file when stream caching is enabled:
  - Record to file x.
  - After the record is complete, play file x.
  - Device caches file x.
  - Record to file x again.
  - 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.

8.3 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
- **WB** - Linear PCM 16HhZ Wide Band Coder
- **IP** - The direction is to the network
- **TDM** - The direction is to the TDM

Notes:

- 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.
- The following three tables are only applicable to **TP-260/UNI, TP-1610, IPM-260/UNI, IPM-1610 and IPmedia 2000**.

8.3.1 Playing a File to TDM/IP

Note: For IPM devices, a file may also be played towards IP.

Coder Combinations - Playing a file to TDM/IP

File Coder	File Type											
	*.wav				*.au				*.raw			
	Channel Coder				Channel Coder				Channel Coder			
	PCMA	PCMU	LBR	WB	PCMA	PCMU	LBR	WB	PCMA	PCMU	LBR	WB
PCMA	V	V	V	V	V	V	V	x	V	V	V	V
PCMU	V	V	V	V	V	V	V	x	V	V	V	V
LBR	x	x	x	V	x	x	x	x	x	x	x	V
WB	V	V	V	V	x	x	x	x	V	V	V	V

Note: For IPM devices, a file may also be played towards IP.

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**.

8.3.2 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.

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	V	V	V	V	V	V	V	V	V
PCMU	V	V	V	V	V	V	V	V	V
LBR	x	x	V	x	x	x	x	x	V

Coder Combinations - Recording a file from IP/TDM

File Coder	File Type											
	*.wav				*.au				*.raw			
	Channel Coder				Channel Coder				Channel Coder			
	PCMA	PCMU	LBR	WB	PCMA	PCMU	LBR	WB	PCMA	PCMU	LBR	WB
PCMA	V	V	V	x	V	V	V	x	V	V	V	x
PCMU	V	V	V	x	V	V	V	x	V	V	V	x
LBR	x	x	V	x	x	x	x	x	x	x	V	x
WB	x	x	x	x	x	x	x	x	x	x	x	x

8.4 Maximum Concurrent Playing and Recording

For more details, refer to the relevant Release Notes relating to the device.

8.5 Supporting LBR Coders

The following table describes the different low bit rate coders and their support for **.wav*, **.au* and **.raw* files:

LBR Coders and Their File Support

Coder	*.wav file	*.raw file	*.au file
G726_16	V	V	x
G726_24	V	V	x
G726_32	V	V	x
G726_40	V	V	x
G727_16	x	V	x
G727_24_16	x	V	x
G727_24	x	V	x
G727_32_16	x	V	x
G727_32_24	x	V	x
G727_32	x	V	x
G727_40_16	x	V	x
G727_40_24	x	V	x
G727_40_32	x	V	x
G723_LOW	V	V	x
G723_HIGH	V	V	x
G729	V	V	x
GSM610	V	V	x
GSM610MS (see note below)	V	V	x
GSM_EFR	V	V	x
G728	V	V	x
NET_CODER_6_4	x	V	x
NET_CODER_7_2	x	V	x
NET_CODER_8	x	V	x
NET_CODER_8_8	x	V	x
NET_CODER_9_6	x	V	x
VOX_ADPCM	V	V	x
G729E	x	V	x

LBR Coders and Their File Support

Coder	*.wav file	*.raw file	*.au file
LINEAR_PCM	x	V	x
AMR_4_75	x	V	x
AMR_5_15	x	V	x
AMR_5_9	x	V	x
AMR_6_7	x	V	x
AMR_7_4	x	V	x
AMR_7_95	x	V	x
AMR_10_2	x	V	x
AMR_12_2	x	V	x
QCELP_8	x	V	x
QCELP_13	x	V	x

Notes:

- GSM610MS may only be used to record from TDM. Recording from IP is currently **not** supported.
- Actual coder support depends on the specific DSP template version set in teh device.

8.6 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
```

8.7 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.

8.8 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 243 and 'NFS Servers Table Parameters' on page 270.

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.

8.9 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.

8.9.1 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>
```

8.10 Supporting NFS Servers

The following is a list of NFS servers that are known to be compatible with AudioCodes Voice Streaming functionality.

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
Windows™ 2000	Cygwin nfsd	2 (Note 2)

Note: Cygwin and winnfsd support only NFSv2.

8.10.1 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

8.10.2 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)
```

8.11 Common Problems and Solutions

Be sure to inspect the Syslog for any problem you encounter; in many cases the cause will appear there.

8.11.1 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
```

8.11.2 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.

8.11.3 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.

9 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

Notes:

- Some Security features are optional and can be ordered or upgraded at a future time.
- 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.

9.1 IKE and IPSec

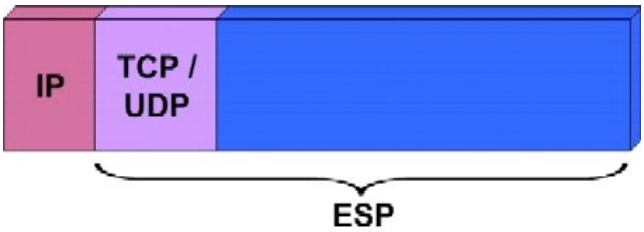
Internet Key Exchange (IKE) and IP Security (IPSec) protocols are part of the IETF standards for establishing a secured IP connection between two applications (also referred to as peers). Providing security services at the IP layer, IKE and IPSec are transparent to IP applications.

IKE and IPSec are used together 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 the IPSec Security Association or 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 58: IPSec Encryption using Encapsulation Security Payload (ESP) Protocol



9.1.1 IKE (ISAKMP)

IKE is used to obtain the Security Associations (SAS) between peers (the gateway and the application it's trying to contact). The SA contains the encryption keys and profile used by IPSec to encrypt the IP stream.

The IKE negotiation is separated into two phases - Main Mode and Quick Mode. The purpose of Main Mode is to obtain a "master" encryption key (without any prior keys), and authenticate the peers to each other. Once initial security is set up, Quick Mode sets up the encrypted IPSec tunnel.

- Main Mode creates a secured channel for Quick Mode:
  - SA Negotiation – The peers negotiate their capabilities using up to four proposals. Each proposal includes three parameters - Encryption Method, Authentication Algorithm and the DH group to use. The master key's lifetime is also negotiated in this stage. For detailed information on configuring proposals, refer to 'Proposal Configuration' on page 713.
  - Key Exchange (DH) – The DH protocol is used to create the master key. DH requires both peers to agree on certain mathematical parameters, known as the "group".
  - Authentication – The two peers authenticate one another using a pre-shared key (configured in the IPSec association table) or by using certificate-based authentication. For information regarding authentication methods, refer to 'Peer Configuration' on page 710.

- Quick Mode creates IPSec tunnels:
  - SA Negotiation – An IPSec SA is created by negotiating encryption and authentication capabilities, using the same proposal mechanism as in Main Mode.
  - Key Exchange – A symmetrical key is created for encrypting IPSec traffic; the peers communicate it to each other in encrypted form, secured by the previously negotiated "master" key.

IKE Specifications:

- Authentication methods - pre-shared key or certificate-based authentication
- Main mode is supported for IKE Phase 1
- Diffie-Hellman group 1 or group 2
- Supported encryption algorithms - AES, DES and 3DES
- Supported hash algorithms - SHA1 and MD5

9.1.2 IPSec

IPSec is responsible for encrypting and decrypting IP traffic.

The IPSec Security Association table defines up to 20 IP peers to which IPSec security is applied. IPSec can be applied to all traffic to and from a specific IP address; alternatively IPSec can be applied to a specific flow, specified by port (source or destination) and protocol type.

Each outgoing packet is analyzed and compared to the 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 'IKE' on page 708) and an IPSec SA is established. The packet is encrypted and transmitted. If a match is not found, the packet is transmitted without encryption.

Notes:

- An incoming packet whose parameters match one of the entries of the association table but is received without encryption, shall be dropped.
- IPSec does not function properly if the device IP address is changed on-the-fly. Therefore, reset the device after you change its IP address.

IPSec Specifications:

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

9.1.3 Configuring IKE and IPSec

Note: To enable IKE and IPSec processing, set the *EnableIPSec* *ini* file parameter to '1'. Note that on some device models, the channel capacity would be reduced; even if no IPSec peers are configured (refer to the IMPORTANT Note at the beginning of this chapter).

9.1.3.1 Peer Configuration

Up to 20 peers (hosts or networks) can be defined for IPSec/IKE. Each of the entries in the IPSec Security Association table controls both Main Mode and Quick Mode configuration for a single peer. The following table lists the available fields of the configuration table.

IPSec/IKE Table Configuration Parameters

Parameter Name	Description
Operational Mode [IPsecSATable_IPsecMode]	Selects the IPSec mode of operation. 0 = Transport mode (default) 1 = Tunnel mode
Remote Endopint [IPsecSATable_RemoteEndpointAddressOrName]	IP address or DNS host name of the peer. **Note**: This field is applicable only if the *Operational Mode* above is set to *Transport*.
Authentication method [IPsecSATable_AuthenticationMethod]	Selects the method used for peer authentication during IKE main mode. 0 = Pre-shared key (default) 1 = RSA signature (in X.509 certificate) **Note**: For RSA-based authentication, both peers must be provisioned with certificates signed by a common CA, for more information on certificates see 'Server Certificate Replacement' on page 719.

IPSec/IKE Table Configuration Parameters

Parameter Name	Description
Shared key [IPsecSATable_SharedKey]	Defines the pre-shared key (in textual format). Both peers must use the same pre-shared key for the authentication process to succeed. The field is applicable only if the *Authentication Method* parameter above is set to *pre-shared key*. • **Note 1**: The pre-shared key forms the basis of IPSec security and should therefore be handled with care (the same as sensitive passwords). It is not recommended to use the same pre-shared key for several connections. • **Note 2**: Since the INI file is plain text, loading it to the device over a secure network connection is recommended. Use a secure transport such as HTTPS, or a direct crossed-cable connection from a management PC. • **Note 3**: After it is configured, the value of the pre-shared key cannot be retrieved.
Source Port [IPsecSATable_SourcePort]	Defines the source port to which this configuration applies. The default value is 0 (any port).
Destination Port [IPsecSATable_DestPort]	Defines the destination port to which this configuration applies. The default value is 0 (any port).
Protocol [IPsecSATable_Protocol]	Defines the protocol type to which this configuration applies. Standard IP protocol numbers (as defined by IANA, the Internet Assigned Numbers Authority) should be used, e.g.: • 0 = Any protocol (default) • 6 = TCP • 17 = UDP
IKE SA LifeTime (sec) [IPsecSATable_Phase1SaLifetimeInSec]	Determines the duration (in seconds) for which the negotiated IKE SA (main mode) is valid. After the time expires, the SA is re-negotiated. Note that *Main Mode* negotiation is a very processor-intensive operation; for best performance, do not set this parameter to less than 28800 (eight hours). The default value is 0 (unlimited).
IPSec SA LifeTime (sec) [IPsecSATable_Phase2SaLifetimeInSec]	Determines the duration (in seconds) for which the negotiated IPSec SA (quick mode) is valid. After the time expires, the SA is re-negotiated. For best performance, a value of 3600 (one hour) or above is recommended. The default value is 0 (unlimited).

IPSec/IKE Table Configuration Parameters

Parameter Name	Description
IPSec SA LifeTime (KBs) [IPsecSATable_Phase2SaLifetimeInKB]	Determines the maximum volume of traffic (in kilobytes) for which the negotiated IPSec SA (quick mode) is valid. After the specified volume is reached, the SA is re-negotiated. The default value is 0 (the value is ignored).
Dead Peer Detection [IPsecSATable_DPDmode]	Controls dead peer detection (DPD) as per RFC 3706. • 0 = DPD disabled • 1 = DPD enabled, periodic checks • 2 = DPD enabled, on-demand checks
Remote Tunnel Endopint [IPsecSATable_RemoteTunnelAddress]	IP address of the peer router. **Note**: This field is applicable only if the *Operational Mode* above is set to *Tunnel*.
Remote Subnet IP Address [IPsecSATable_RemoteSubnetIPAddress]	IP address of the remote subnetwork. **Note**: This field is applicable only if the *Operational Mode* above is set to *Tunnel*. Together with the following *Prefix Length*, this parameter defines the network with which the IPSec tunnel allows communication.
Remote Subnet Prefix Length [IPsecSATable_RemoteSubnetPrefixLength]	Prefix length of the Remote Subnet IP Address parameter (in bits). **Note**: This field is applicable only if the *Operational Mode* above is set to *Tunnel*. The prefix length defines the subnet class of the remote network. A prefix length of 16 corresponds to a Class B subnet (255.255.0.0); a prefix length of 24 corresponds to a Class C subnet (255.255.255.0).

To configure the IPSec/IKE table using the ini file:

The IPSec/IKE parameters are configured using ini file tables (described in 'IPsec Parameters' on page 242). Each line in the table refers to a different IPSec/IKE peer.

The FORMAT line 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. Note that the FORMAT line must be a single continuous line (without breaks up to the semicolon); for clarity, it is shown below with line breaks.

The following is an example of an IPSec/IKE table.

```
[ IPsecSATable ]
FORMAT IPsecSATable_Index = IPsecSATable_IPsecMode,
IPsecSATable_RemoteEndpointAddressOrName,
IPsecSATable_AuthenticationMethod, IPsecSATable_SharedKey,
IPsecSATable_SourcePort, IPsecSATable_DestPort,
IPsecSATable_Protocol,
IPsecSATable_Phase1SaLifetimeInSec,
IPsecSATable_Phase2SaLifetimeInSec;

IPsecSATable 1 = 0, 10.3.2.73, 0, 123456789, 0, 0, 0, 0, 28800,
3600;
```

```
[ \IPsecSATable ]
```

In the above example, a single IPSec/IKE peer (10.3.2.73) is configured. Pre-shared key authentication is selected, with the pre-shared key set to 123456789. In addition, a lifetime of 28800 seconds is selected for IKE and a lifetime of 3600 seconds is selected for IPSec.

9.1.3.2 Proposal Configuration

IKE can be configured with up to four proposal settings. Each proposal defines an encryption algorithm, an authentication algorithm and a Diffie-Hellman group identifier. The same set of proposals apply to both Main Mode and Quick Mode.

Proposals are set using a configuration table, much like the IPSec peer configuration above. The following table lists the available fields of the configuration table.

IPSec/IKE Proposal Table Configuration Parameters

Parameter Name	Description
EncryptionAlgorithm [IPsecProposalTable_EncryptionAlgorithm]	Selects the encryption (privacy) algorithm: • [0] NONE • [1] DES CBC • [2] 3DES CBC • [3] AES (default)
AuthenticationAlgorithm [IPsecProposalTable_AuthenticationAlgorithm]	Selects the message authentication (integrity) algorithm: • [0] NONE • [2] HMAC SHA1 96 • [4] HMAC MD5 96 (default)
DHGroup [IPsecProposalTable_DHGroup]	Selects the Diffie-Hellman group: • [0] Group 1 (768 Bits) • [1] Group 2 (1024 Bits) - default

If no proposals are defined, the default settings (shown in the table below) are applied.

Default IPSec/IKE Proposals First Phase Proposals

	Encryption	Authentication	DH Group
Proposal 0	3DES	SHA1	Group 2 (1024 bit)
Proposal 1	3DES	MD5	Group 2 (1024 bit)
Proposal 2	DES	SHA1	Group 1 (786 bit)
Proposal 3	DES	MD5	Group 1 (786 bit)

➢ Configuring the Proposal table using the ini file

The IPSec proposal table is configured using a separate ini file table. Each line in the table defines a single proposal; the order of proposals offered by IKE is determined by the order of lines in the table.

Note that the line starting with FORMAT must be a single continuous line (without breaks up to the semicolon); for clarity, it is shown below with line breaks.

The following is an example of a proposal configuration table.

```
[ IPsecProposalTable ]
```

```
FORMAT IPsecProposalTable_Index =
IPsecProposalTable_EncryptionAlgorithm,
IPsecProposalTable_AuthenticationAlgorithm,
IPsecProposalTable_DHGroup;

IPsecProposalTable 0 = 3, 2, 1;
IPsecProposalTable 1 = 2, 2, 1;

[ \IPsecProposalTable ]
```

In the example above, two proposals are defined:

- Proposal 0: AES, SHA1, DH group 2
- Proposal 1: 3DES, SHA1, DH group 2

Note that with this configuration, neither DES nor MD5 can be negotiated.

9.1.3.3 IKE and IPSec Configuration Table's Confidentiality

Since the pre-shared key parameter of the IPSec/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. In SNMP, the pre-shared key parameter is a write-only parameter and cannot be read. In the ini file, an asterisk will be displayed instead of the pre-shared key.

9.1.4 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) remain active until their lifetimes naturally expire, resulting in a "black hole" situation where both peers discard all incoming network traffic.

This situation may be remedied by performing periodic message exchanges between the peers. When no reply is received, the sender assumes the SAs are no longer valid on the remote peer, and attempts to renegotiate.

DPD is automatically negotiated. In order to activate DPD functionality, the DPDMode configuration parameter (in the IPSec security association table) 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)

9.2 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 1024/2048-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.

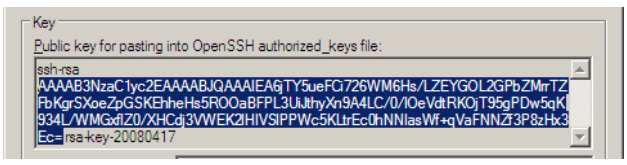
➢ **To configure RSA public keys on Windows (using PuTTY SSH software):**

1. Run **PuTTYgen.exe**. The PuTTY Key Generator screen appears.

Figure 59: PuTTY Key Generator

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 60: PuTTY Example

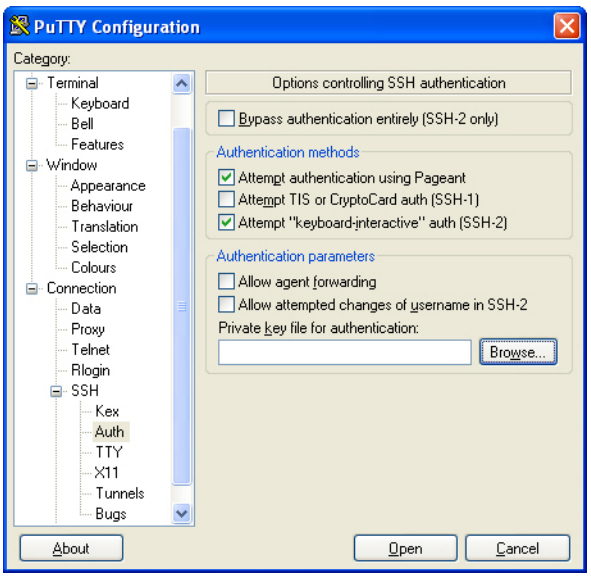


7. Edit the device's *ini* file and set the **SSHAdminKey** to the value copied above, e.g.:

```
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 61: PuTTY Configuration



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):

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.

9.3 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.

9.3.1 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.

9.3.2 Using the Secure Web Server

➢ **To use the secure Web server:**

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
```

9.3.3 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 ]
```

9.3.4 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:**

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 **Configuration > System > 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.

10. To apply loaded certificates for IPsec negotiations refer to the Mediant 3000 User's Manual.

Notes:

- The certificate replacement process may be repeated as necessary, e.g., when the new certificate expires.
- 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.

9.3.5 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:**

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".

9.3.6 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:**

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.

9.3.7 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.

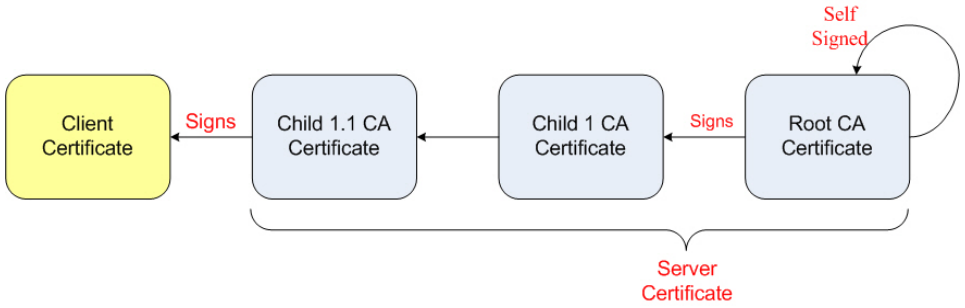
Note: Some PKIs do not support OCSP but generate Certificate Revocation Lists (CRLs). In this case, set up an OCSP server such as OCSPD.

9.3.8 Certificate Chains

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.

In order for the device to trust a whole chain of certificates, combine the certificates into one text file (using a text editor). Upload the file as the "Trusted Root Certificate Store" file as discussed above.

Figure 62: Certificate Chain Hierarchy



Note: When configuring a trusted chain of certificates, the file size is limited to up to 9000 characters (including the certificate's headers).

9.4 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.

9.4.1 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:**

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
```

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.

9.4.2 Configuring RADIUS Support

➢ **To configure RADIUS support on the gateway via the Web interface:**

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:**

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.

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.
   - To set this feature, use Radius with Basic authentication settings:
     - Default settings - You are prompted for your login every time you connect to the device.
     - Enable Radius configuration as described above.
     - Enable Basic HTTP authentication settings.
     - Connect to the device using a URL as in the example.

Note: This feature is restricted to 5 users simultaneously only.

9.5 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:

Internal Firewall Fields

Parameter	Description
Source IP **[AccessList_Source_IP]**	IP address (or DNS name) of source network, or a specific host
Prefix Length **[AccessList_PrefixLen**]	IP network mask. 32 for a single host, or the appropriate value for the source IP addresses. A value of 8 corresponds to IPv4 subnet class A (network mask of 255.0.0.0). A value of 16 corresponds to IPv4 subnet class B (network mask of 255.255.0.0). A value of 24 corresponds to IPv4 subnet class C (network mask of 255.255.255.0). The IP address of the sender of the incoming packet is trimmed in accordance with the prefix length (in bits) 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.

Internal Firewall Fields

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.
Use Specific Interface **[AccessList_Use_Specific_Interface]**	Allows setting a firewall rule on one or all of the device's network interfaces. When set to 1, the next field determines the name of the interface for which this rule is effective.
Interface Identifier **[AccessList_Interface_ID]**	Specifies the name of the network interface for which this rule is effective. This field will only be relevant if the "Use Specific Interface" field is set to 1; otherwise the rule is effective for all interfaces.
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_MaskPrefixLen, 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, 255.255.255.25532, 0, 80, tcp,
0, 0, 0, allow ;
ACCESSLIST 15 = 192.0.0.0, 255.0.0.08, 0, 65535, any, 0, 40000,
50000, block ;
ACCESSLIST 20 = 10.31.4.0, 255.255.255.024, 4000, 9000, any, 0, 0,
0, block ;
ACCESSLIST 22 = 10.4.0.0, 255.255.0.016, 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_MaskPrefixLen, 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, 255.0.0.08, 0, 65535, any, 0, 40000,
50000, allow ;
ACCESSLIST 15 = 10.31.4.0, 255.255.255.024, 4000, 9000, any, 0, 0,
0, allow ;
ACCESSLIST 20 = 0.0.0.0, 0.0.0.00, 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.

9.6 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.

Default TCP/UDP Network Port Numbers

Port number	Peer port	Application	Notes
2	2	Debugging interface	Always ignored
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 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

Default TCP/UDP Network Port Numbers

47000, 47001 and up		HTTP/NFS streaming traffic	Configurable (NFSBasePort) The number of ports used depends on the channel capacity of the device May be disabled (EnableVoiceStreaming).
(random) > 60000	514	Syslog	Configurable (SyslogServerPort) May be disabled (ENABLESYSLOG).
(random) > 60000	162	SNMP Traps	May be disabled (DISABLESNMP)
(random) > 60000	-	DNS client	

9.7 Media Security

Note: The sub-section on Packet Cable Security is not applicable to **MediaPack** and **Mediant 1000**.

9.7.1 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.

9.7.2 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 SIP, MGCP and MECAGO

Note : Using media security reduces the channel capacity of the device.

9.8 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.

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

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

This page is intentionally left blank.

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 22 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. Enumeration begins from 0 and does not exceed the number of Call Progress Tones -1 defined in the first section. The following keys are used:

- **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
- **Tone Modulation Type** – 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:
  - 0 = Default
  - 1 = Continuous
  - 2 = Cadence
  - 3 = 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.

Notes:

- 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.
- The tone frequency should differ by at least 40 Hz from one tone to other defined tones**.**
- 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 787.
- 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. Enumeration begins from 0 and does not exceed the number of User Defined Tones -1 defined in the first section. The following keys are used:

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
- **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. Enumeration begins from 1 and does not exceed 16 using. The following keys are used:
  - **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.

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
#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

Default Call Progress Tones

#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
#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

Default Call Progress Tones

#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”

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
Second Ring Off Time [10msec]=400
#GR-506-CORE Ringing Pattern 3
[Ringing Pattern #3]

Number Of Distinctive Ringing Patterns

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
Freq [Hz]=25
First Ring On Time [10msec]=100
First Ring Off Time [10msec]=300
#EN 300 001 Ring - Finland
[Ringing Pattern #7]

Number Of Distinctive Ringing Patterns

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
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

Number Of Distinctive Ringing Patterns

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:**

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 787.

After making the *dat* file, send it to the device using one of the following:

- The Web interface GUI's Auxiliary Files.
- 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.)
- 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.

Notes:

- The maximum PRT buffer size is 100KB. (**MediaPack** only)
- The maximum PRT buffer size is 1MB (All other products).
  For the AMS configuration, the maximum PRT buffer size is 2MB.
- 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.da*t - 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:**

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 *CallProgressTonesFilename ini* file parameter 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 and which name should be used in the SDP. Additionally, it defines the default payload types and ptime. 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
         MRTA NB          X-MSRTA       114          114
```

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:

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			

Aliases Used for Currently Supported Coders

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-TFO	QCELP_8_TFO	QCELP-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	AMR2	
AMR	AMR_5_15	AMR-5-15	AMR515	AMR2	
AMR	AMR_5_9	AMR-5-9	AMR590	AMR2	
AMR	AMR_6_7	AMR-6-7	AMR670	AMR2	
AMR	AMR_7_4	AMR-7-4	AMR740	AMR2	
AMR	AMR_7_95	AMR-7-95	AMR795	AMR2	
AMR	AMR_10_2	AMR-10-2	AMR1020	AMR2	
AMR	AMR_12_2	AMR-12-2	AMR1220	AMR2	
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				

Aliases Used for Currently Supported Coders

EVRCB					
EVRC1					
EVRCB1					
AMR-WB	AMR_WB				
X-MSRTA	MRTA_NB	MRTA			

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 787. 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

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

Notes:

- This sub-section is only applicable to **TP and Mediant** devices.
- Conferencing functionality is disrupted by CAS. To assure proper conferencing functionality, disable the CAS parameters.

10.6.1 Constructing a CAS Protocol Table

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:**

1. Learn the protocol text file rules the CAS state machine is built from (refer to 'Table Elements' on page 759)
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 787 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 using the downloading methods described in 'CAS Trunk Parameters' on page 350.

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 examples taken from an E&M wink start table protocol file:

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.

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:

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.

Global Parameters

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 (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 the following columns:

- Action/Event
- Function
- Parameters : #1, #2 etc (dependent on the function)
- 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):

10.6.5.1 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).
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

10.6.5.2 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.

10.6.5.3 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.

10.6.5.4 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.

10.6.5.5 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.
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().

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.

10.6.5.6 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**

10.6.6 Function

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

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

10.6.7 Parameters

CAS Parameters

Parameter #1	These columns are used as the function's parameters. The list of global parameters can be found in *CasSetup.h*.
Parameter #2	If a parameter is not essential, it can also be written as NONE.

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.

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

List of Available User Functions and Their Parameters

	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.
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.

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

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).

10.6.8 Next State

The Next State column contains the next state the table moves to after executing the function for that action/event line. When the user selects to stay in the same state, insert NO_STATE or use the current state.

Note the difference between NO_STATE and the current state name in this field. If the user selects to stay in the same current state, the unconditional actions (FUNCTION0) at the beginning of the state are performed. In contrast, NO_STATE skips these functions and waits for another action to come.

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

10.6.9 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.

10.6.9.1 MFC R2 Protocol

- Use the SEND_MF script function to generate the outgoing call destination number. In this case, the first parameter should be ADDRESS (or ANI for source phone number) and the second parameter -3 to 1 (+1), indicating which digit is sent out of the number that the string conveyed by 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.

10.6.10 Changing Default Parameter Values of 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.

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 value is in msec.
CasStateMachineMaxNumOfIncoming AddressDigits	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.
CasStateMachineMaxNumOfIncoming ANIDigits	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	In some cases, when the state machine handles

	Default value is -1, No - 0, Yes - 1.	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.

This page is intentionally left blank.

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

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
9	G.722-64	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

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
90	Linear PCM	20
114	MS/RTA	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

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

Note: The Default RTP/RTCP/T.38 Port Allocation section is applicable to MGCP and TPNCP protocols only. In the H.248 protocol, the UDP port is allocated dynamically from the Media Port pool. The Media Port pool range is [BaseUDPPort, BaseUDPPort + MaxChannelCapacity*10].

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

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.

This page is intentionally left blank.

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 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.

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.

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)

12.6.3.1 Real V.34 Fax Transmission

➢ **To pass V.34 fax (up to 33600 bps) over T.38 fax relay, use the following configuration:**

- DSP template – 10
- T38Version - 3
- Fax & V.34 Fax Transport mode - Relay
- Vxx Modem mode - Disable
- CNG Detectors mode - Disable

12.6.3.2 Fallback from V.34 fax to T.30

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

Configuration:

- Any DSP template
- T38Version - 0
- 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.

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:

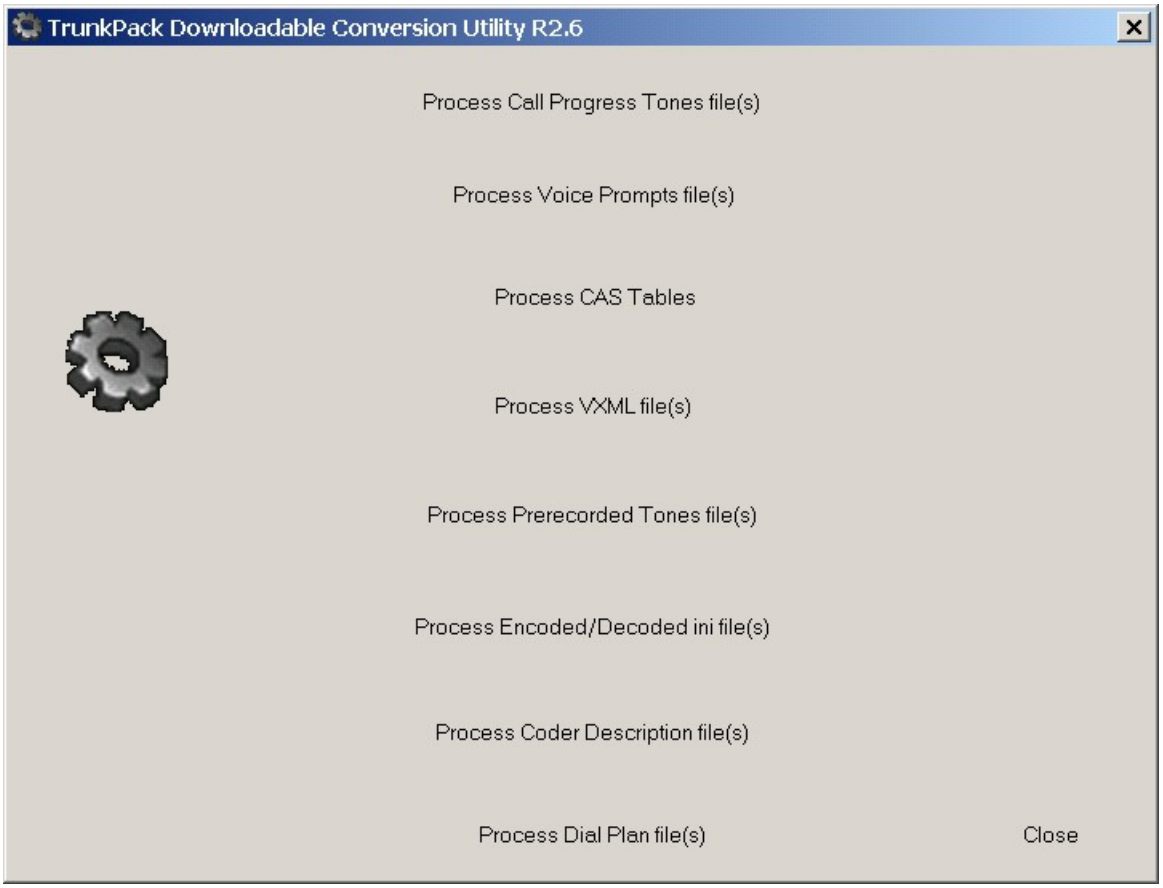
The Apirunce Application - TCL-based demo program available for Linux™, Solaris™ and Windows™ OSs. With this application the user can build scripts online or offline and execute them. All new APIs in this version are supported by Apirunce.

13.2 TrunkPack Downloadable Conversion Utility

LOCATION:

```
.\Utilities\DConvert\DConvert.exe
```

Figure 63: TrunkPack Downloadable Conversion Utility R2.6.2



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.

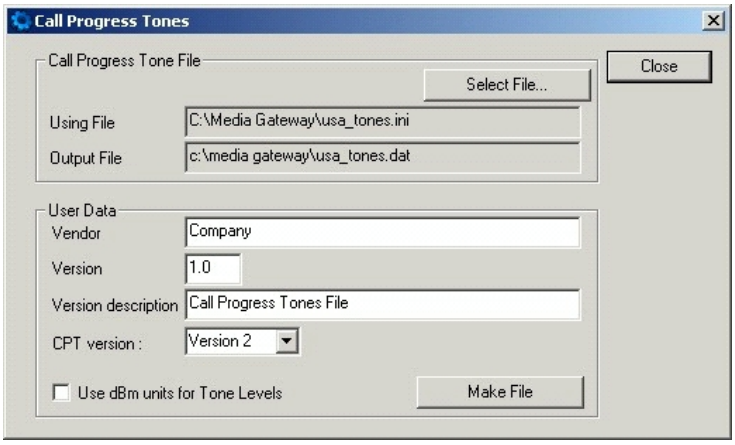
13.2.1 Process Call Progress Tones File(s)

➢ **To convert a CPT ini file to a binary dat file:**

1. For **MediaPack**, create a CPT *ini* file using the direction in 'Modifying the Call Progress Tones File & Distinctive Ringing File' on page 747, 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 746, 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 64: Call Progress Tones Screen



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.

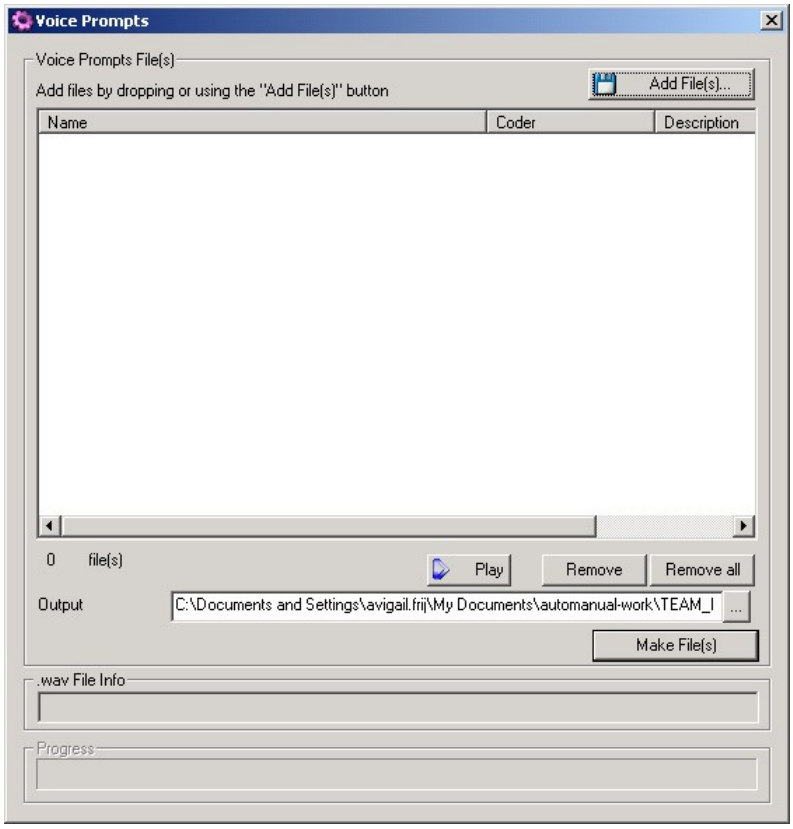
8. Click the **Make File** button. The *dat* file is generated and placed in the same directory as shown in the **Output File** field. A message box informing you that the operation was successful indicates that the process is completed.

13.2.2 Process Voice Prompts File(s)

➢ **To generate a Voice Prompts file:**

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 65: Voice Prompts Screen



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 66: Select Files Window

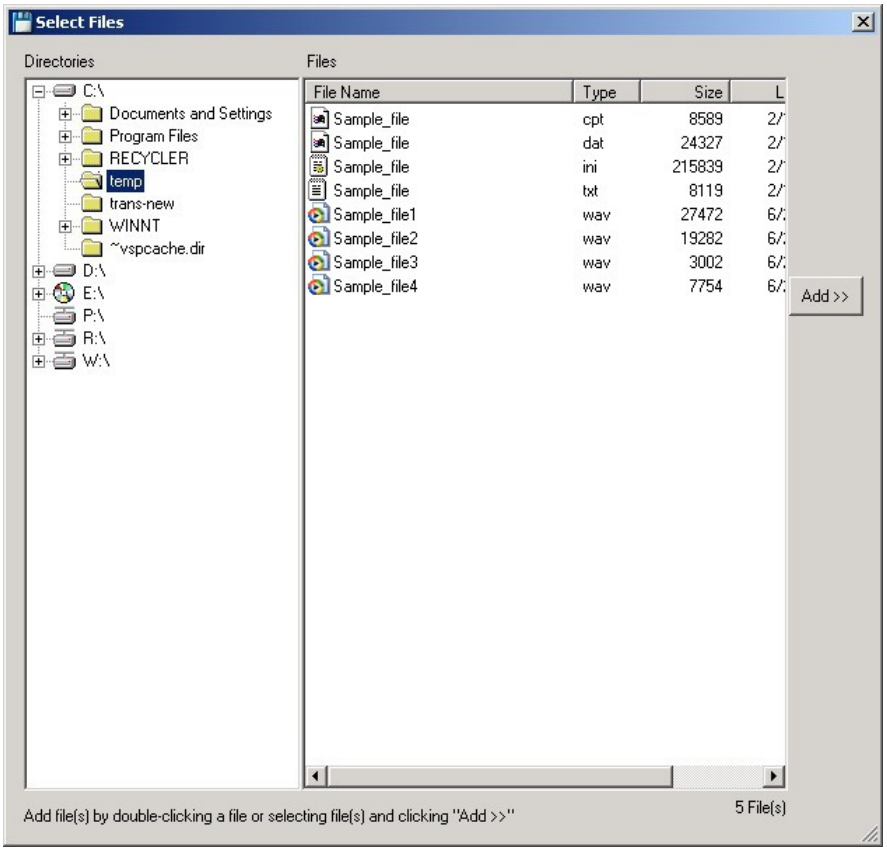
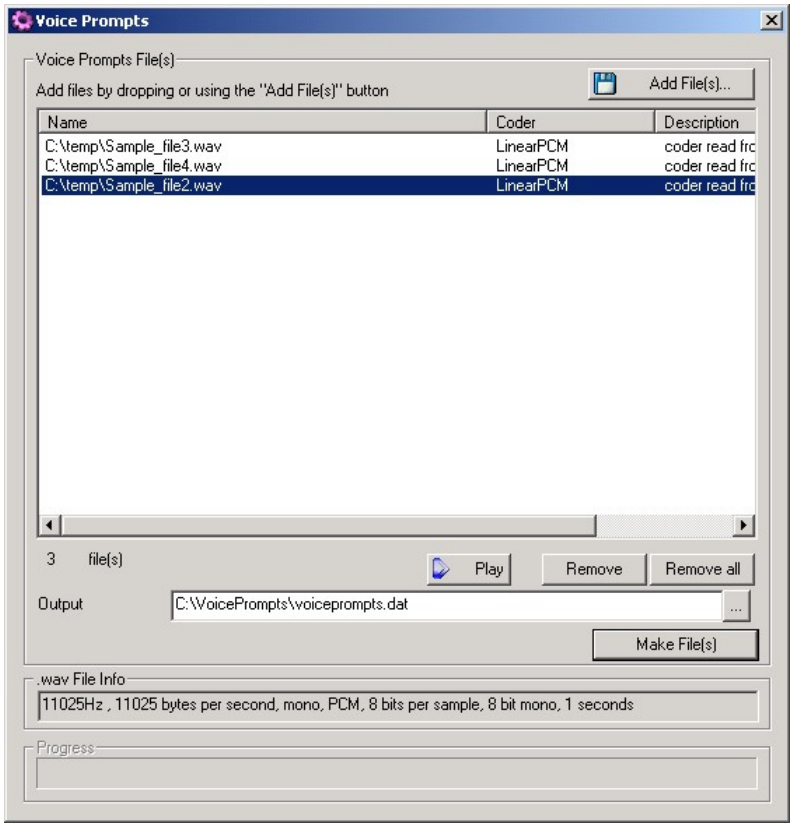


Figure 67: Voice Prompts Window with wav Files



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 68: File Data Window

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,

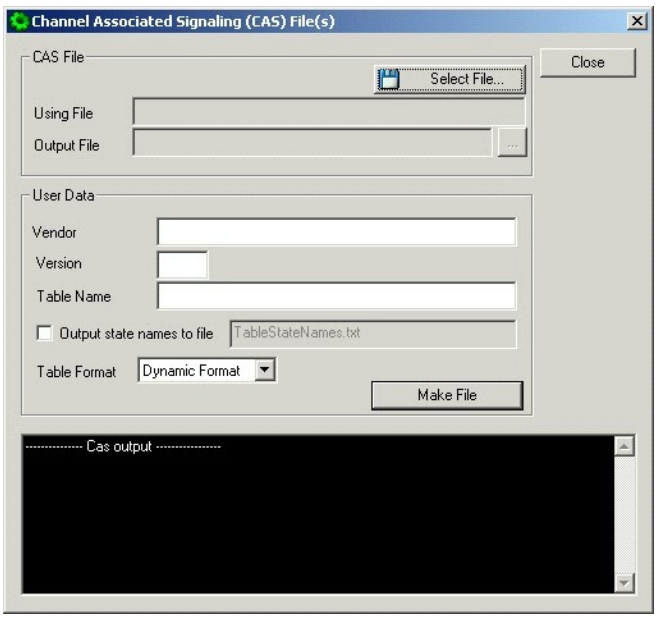
    Use the ... Browse button to select a different Output file. Navigate to the required file and select it. The selected file name and its path appear in the **Output** field.
11. Click the **Make File(s)** button to generate the Voice Prompts file. The Progress bar at the bottom of the window is activated. 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:**

1. Construct the CAS protocol *xxx.txt* and *xxx.h* files according to the instructions in 'Channel Associated Signaling (CAS) Functions' on page 759.
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 69: Call Associated Signaling (CAS) Screen



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.

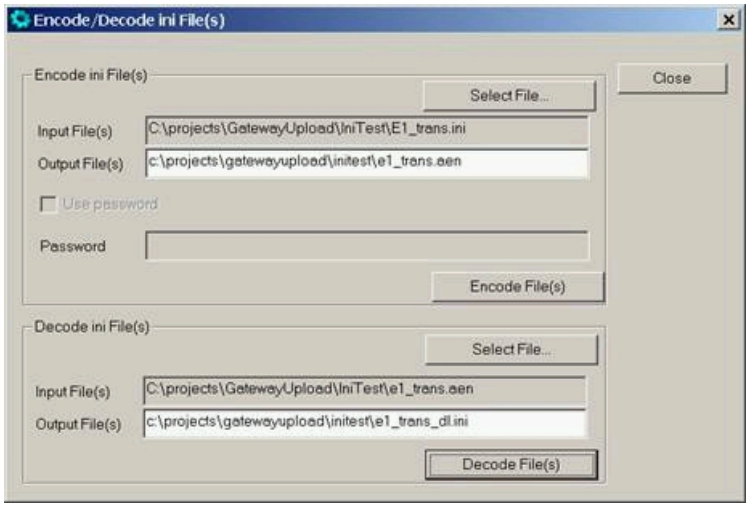
The *ini* file can be both encoded and decoded using **DConvert**. Encoding usually takes place before downloading an *ini* file to the device while decoding usually takes place after uploading an *ini* file from the device.

➢ **To Encode an *ini* file:**

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 70: Encoded ini File(s) Screen



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:

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.

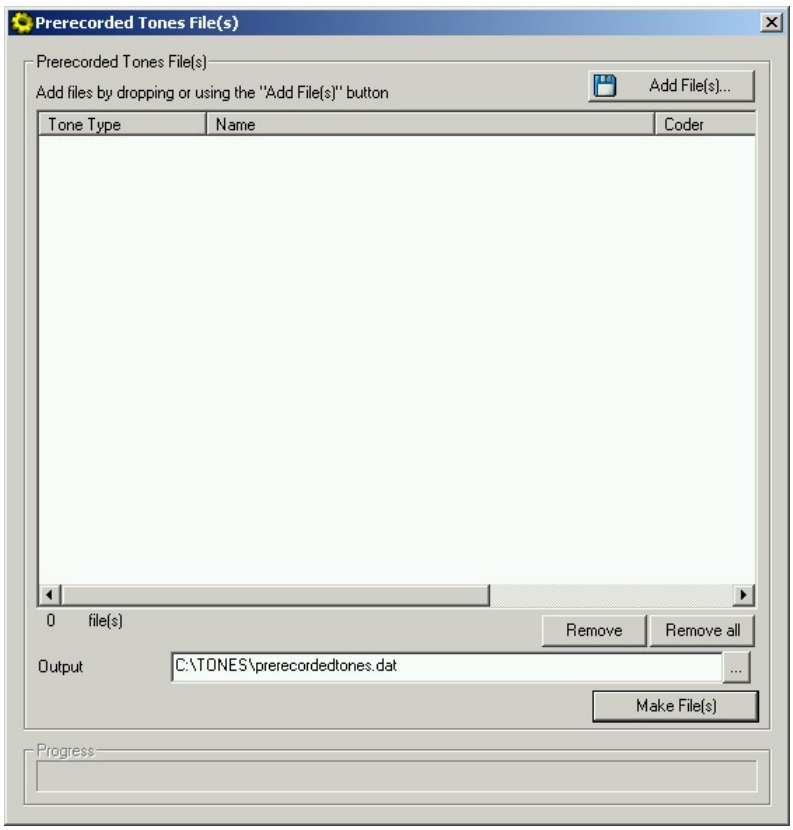
Note: The decoding process verifies the input file for validity. Any change made to the encoded file causes an error and the decoding process is aborted.

13.2.4 Process Prerecorded Tones File(s)

➢ **To generate a Prerecorded Tones file:**

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 71: Prerecorded Tones File(s) Screen

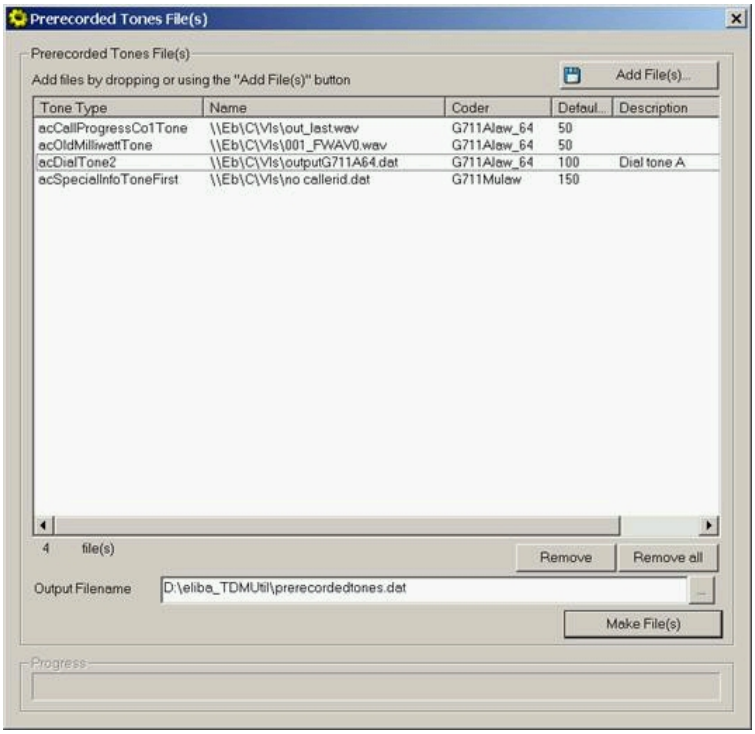


3. Select the raw Prerecorded Tones files (created in Step 1) utilizing one of these actions:
   a. Click the **Add Files** button in the upper right corner. The Add Files window appears. (Refer to the figure, Select Files Window.)

      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 72: Prerecorded Tones File(s) Screen with wav Files



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 73: 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)

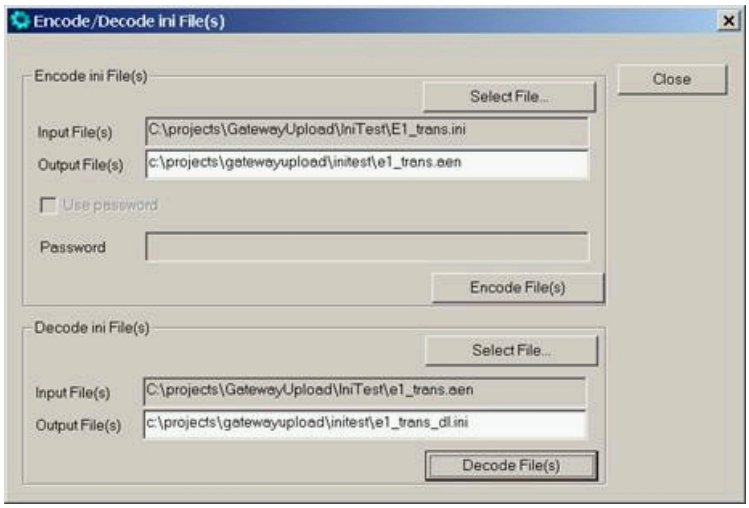
The *ini* file can be both encoded and decoded using **DConvert**. Encoding usually takes place before downloading an *ini* file to the device while decoding usually takes place after uploading an *ini* file from the device.

➢ To Encode an *ini* file:

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 22).

   Execute *DConvert.exe* and click the **Process Encoded *ini* file(s)** button. The Encoded *ini* file(s) window appears.

Figure 74: Encoded ini File(s) Screen



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:

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 exiting *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.

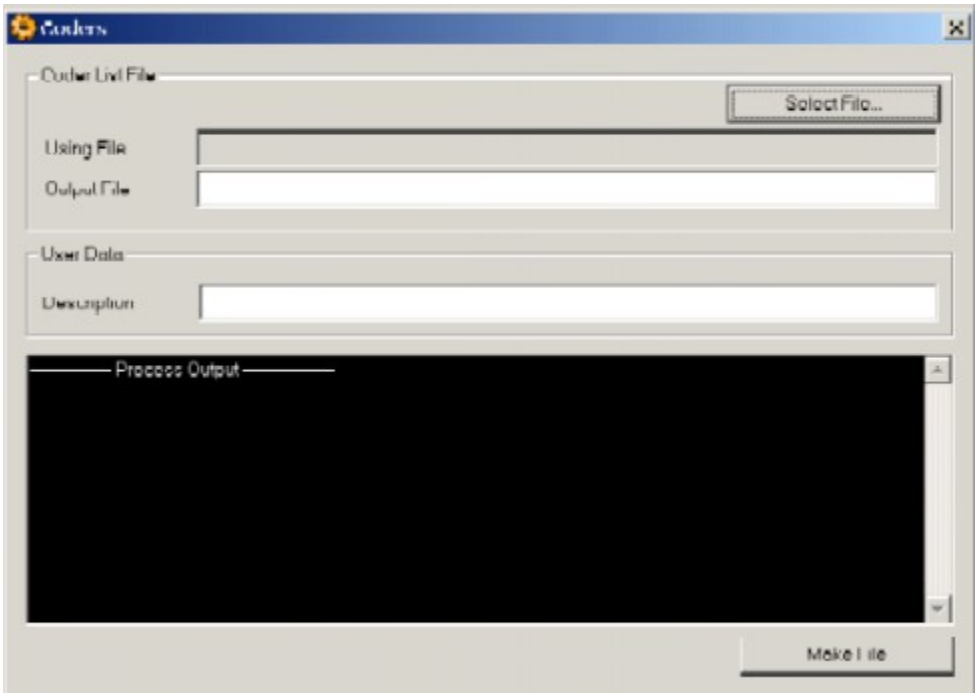
Note: The decoding process verifies the input file for validity. Any change made to the encoded file causes an error and the decoding process is aborted.

13.2.6 Process Coder Description File(s)

➢ **To produce a Coder Description file:**

1. Construct a Coder Description text file according to the instructions in 'Coder Table File' on page 752.
2. Execute DConvert.exe and click the Process Coder Description button. The Coders Window appears.

Figure 75: 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.

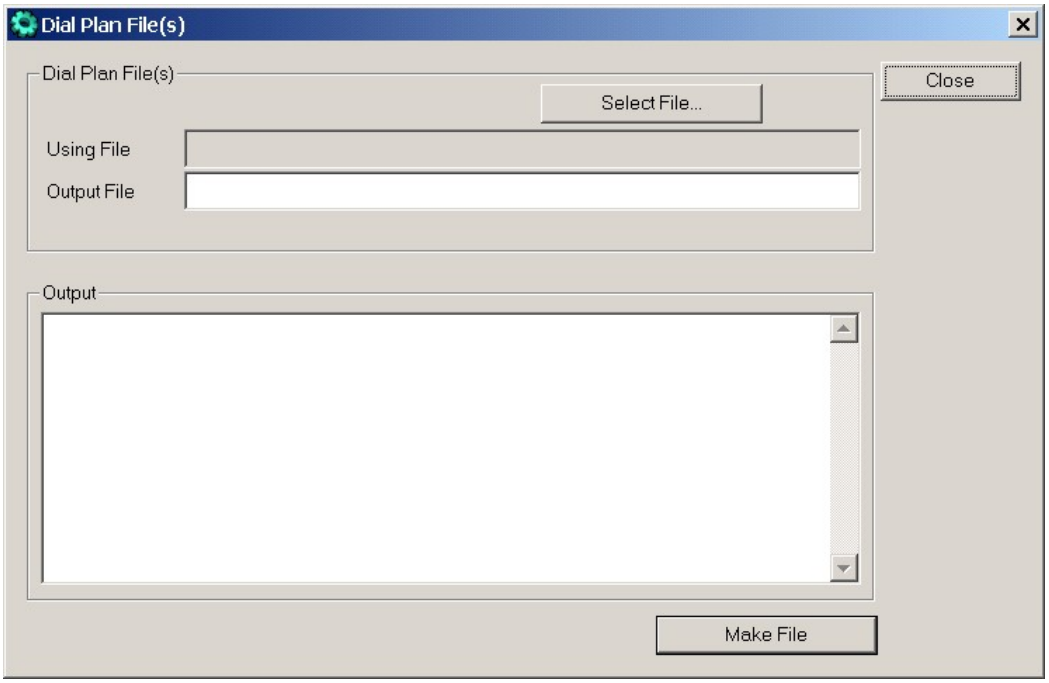
Note: The process verifies the input file for validity. Invalid data will cause an error and abort the process. In this case the log box will contain further information.

13.2.7 Process Dial Plan File(s)

➢ **To produce a Dial Plan file:**

1. Construct a Dial Plan text file according to the instructions in 'Dial Plan File' on page 757.
2. Execute DConvert.exe and click the Process Coder Description button. The Dial Plan window appears.

Figure 76: 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.

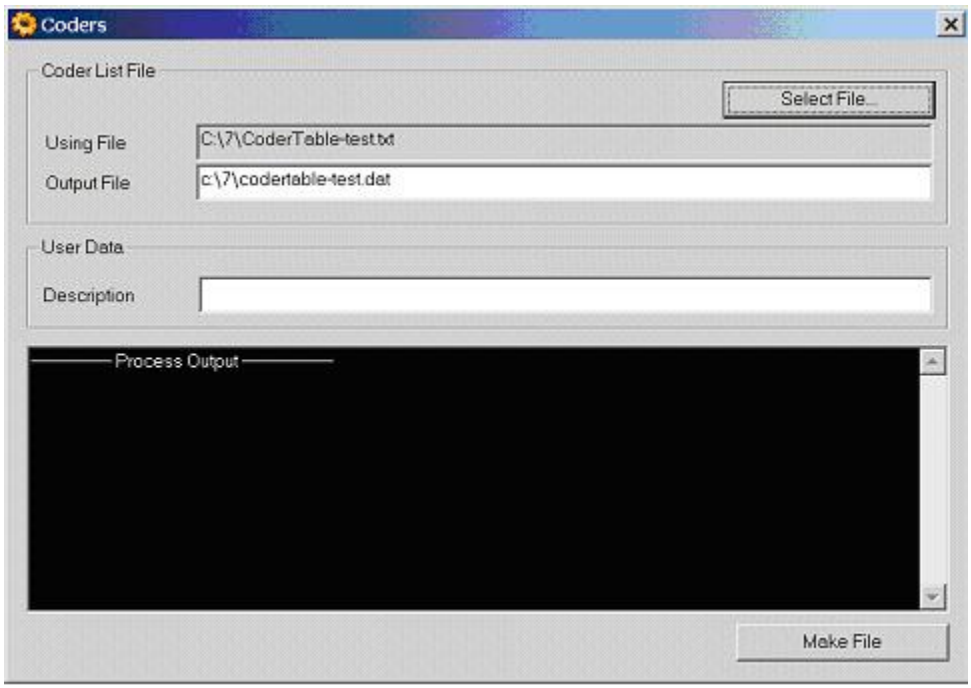
13.2.8 Process Coder Table File(s)

Coder description files are used to define coders properties for the control protocol in use.

➢ **To Encode an ini file:**

1. Create a coder table file. An example is shown in 'Coder Table File' on page 752.
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 77: Process Coder Table Screen



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 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.

13.4 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.

13.4.1 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.

13.4.2 Installation

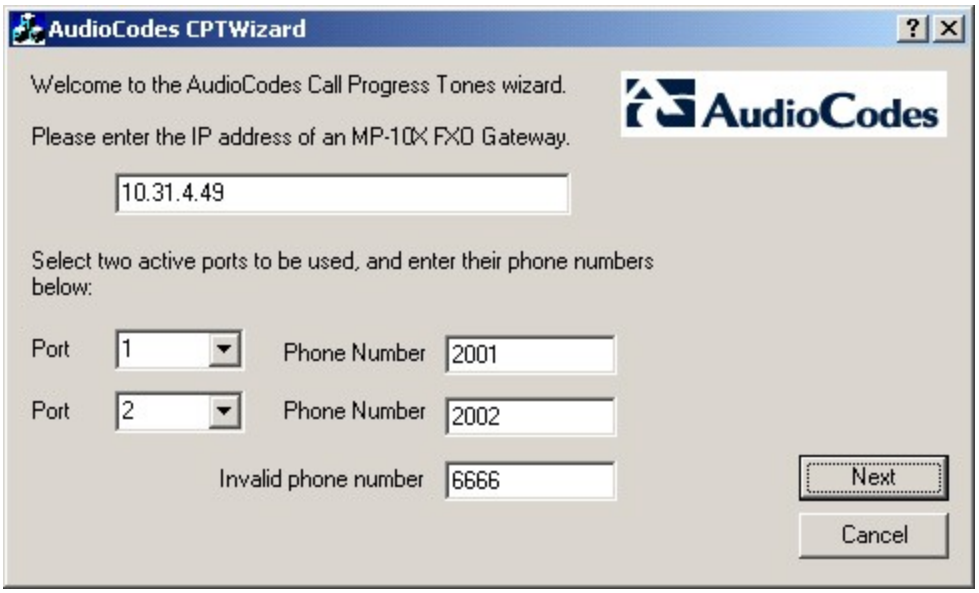
- CPTWizard can be installed on any Windows 2000 or Windows XP based PC. Windows-compliant networking and audio peripherals are required for full functionality.
- To install CPTWizard, copy the files from the installation media to any folder on the PC's hard disk. No further setup is required.
- Approximately 5 MB of hard disk space are required.

13.4.3 Initial Settings

➢ **To start CPTWizard:**

1. Double-click on your copy of the CPTWizard.exe program file. The initial settings dialog is displayed:

Figure 78: Initial Settings Dialog



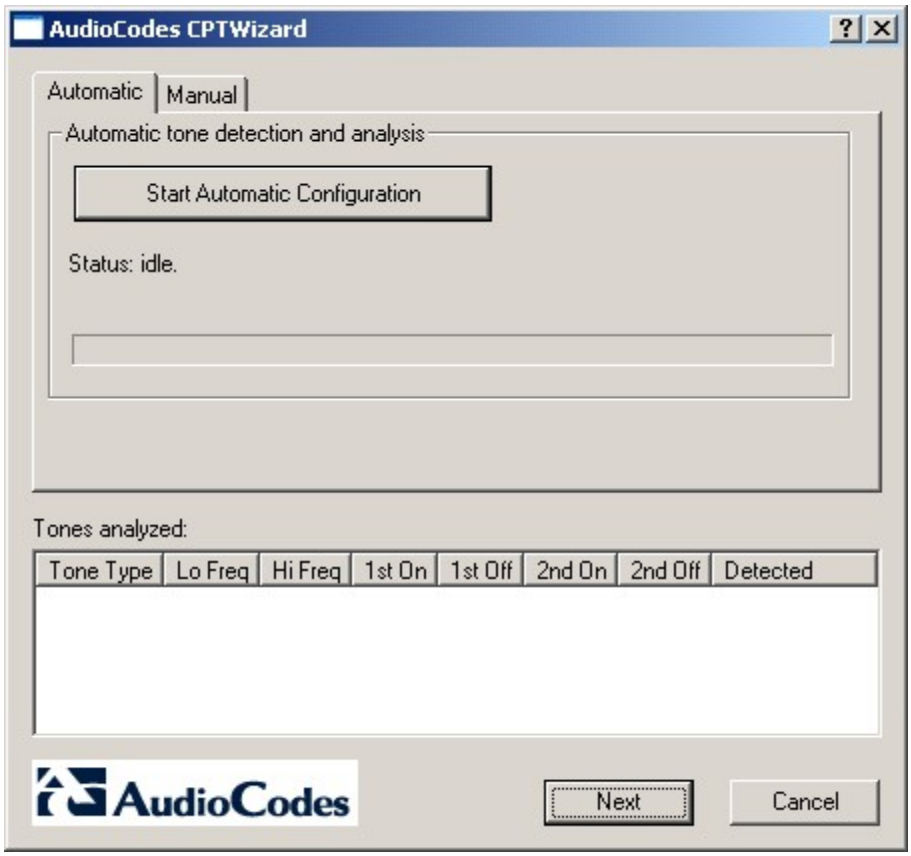
2. In the appropriate fields, fill in the gateway's IP address, select which of the gateway's ports are connected to your PBX, and specify the phone number for each extension.
3. In the "Invalid phone number" box, enter a number which generates a "fast busy" tone when dialed. Usually, any incorrect phone 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.

13.4.4 Recording Dialog – Automatic Mode

Once the connection to the device FXO gateway is established, the recording dialog is displayed:

Figure 79: Recording Dialog

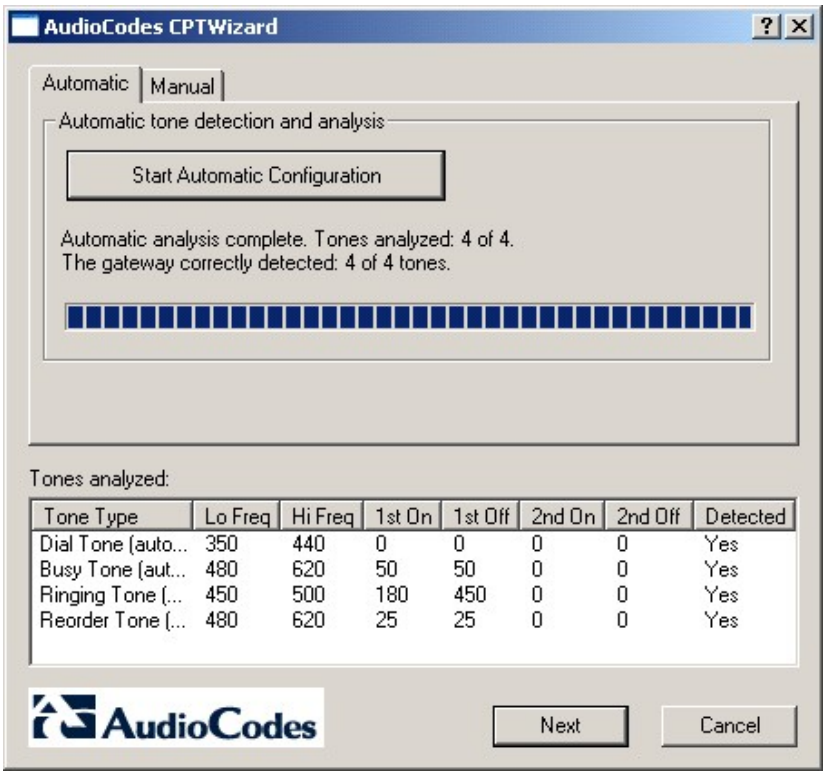


➢ **To start Recording Dialog in Automatic Mode:**

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 80: Recording Dialog after Automatic Detection



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.

Notes:

- 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.
- 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.

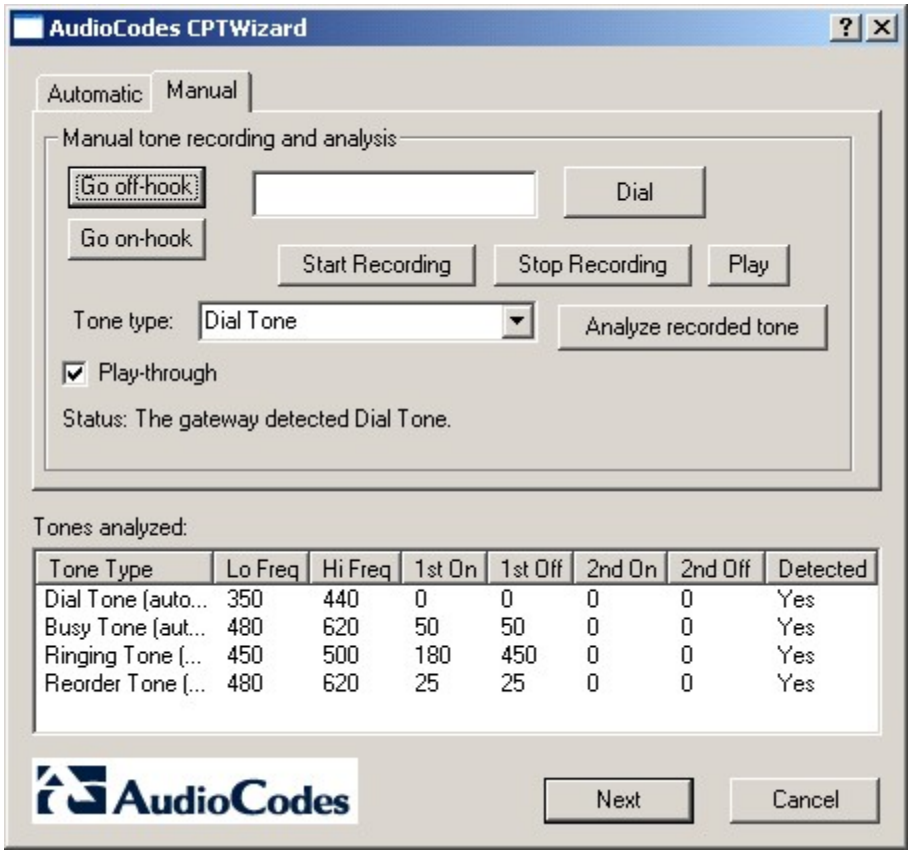
5. At this stage, you can either press NEXT to generate call progress tone *ini* file *dat* files and end the wizard, or continue to manual recording mode.

13.4.5 Recording Dialog – Manual Mode

➢ **To start Recording Dialog in Manual Mode:**

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 81: Recording Dialog in Manual Mode



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.

13.4.6 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 82: 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 83: 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 84: 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 Diagnostics & Troubleshooting

14.1 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
    - MEGACO
  - SNMP
- Solutions to Common Problems.

They are described in the following pages.

14.2 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.ph'p
- 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

14.2.1 Operating the Syslog Server

14.2.1.1 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.

The following error levels are 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
```

Note: Emergency, Warning and Notice level messages are sent to Syslog by default. Info and Debug level messages can be enabled via CLI if necessary, when instructed by AudioCodes support.

14.2.1.2 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, Setting Syslog Server Port, and Enabling Syslog, in an ini File' on page 813 and to the example below.

14.2.1.3 Setting the Syslog Facility Level

Message Facilities are numerically coded with decimal values. each facility may use any of the "local use" facilities. Those Facilities that have been designated are shown in the following table with their appropriate numerical code values:

Numerical Value	Facility Level
16 (default)	local use 0 (local0)
17	local use 1 (local1)
18	local use 2 (local2)
19	local use 3 (local3)
20	local use 4 (local4)
21	local use 5 (local5)
22	local use 6 (local6)
23	local use 7 (local7)

➢ **To set the Facility level of the Syslog messages**

Use the Web interface to configure, or send an ini configuration file containing the *SyslogFacility* facility parameter to the device. This parameter allows values from 16 (local use 0) to 23 (local use 7), where 16 is the default value. For an ini file example showing this parameter, refer to the example below.

14.2.1.4 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
- configuration for the Syslog Facility Level parameter - SyslogFacility

```
[Syslog]
SyslogServerIP=10.2.0.136
EnableSyslog =1
SyslogServerPort =601
SyslogFacility = 17
```

14.3 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 811.

14.4 Control Protocol Reports

The following describes Control Protocol reports.

14.4.1 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.

14.4.2 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.

14.4.3 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 63 for more details on all SNMP traps available on the device.

14.5 Solutions to Possible Problems

14.5.1 Solutions to Possible Common Problems

Solutions to possible common problems are described in the table below.

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
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.

14.5.2 Solutions to Possible Voice Problems

Solutions to possible voice problems are described in the table below.

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.
	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.

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, 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, 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

15 List of Abbreviations

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
ETSI	European Telecommunications Standards Institute
FR	Frame Relay

List of Abbreviations

Abbreviation	Meaning
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)
MGC	Media Gateway Controller
MGCP	Media Gateway Control Protocol

List of Abbreviations

Abbreviation	Meaning
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
Stretto	AudioCodes series of Voice over Wireless Media Gateways
TCP	Transmission Control Protocol.

List of Abbreviations

Abbreviation	Meaning
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

16 Index

A

B

C

D

E

F

S

T

U

V

W

This page is intentionally left blank.

AudioCodes

Product Reference Manual

www.audiocodes.com