Reference Guide

# TR-069 & TR-104

## CPE WAN Management Protocol (CWMP)

Version 6.8

Sounds Better

# Table of Contents

# List of Figures

## Notice

This document explains how to configure the AudioCodes gateway using TR-069 and TR-104 protocols.

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. Before consulting this document, check the corresponding Release Notes regarding feature preconditions and/or specific support in this release. In cases where there are discrepancies between this document and the Release Notes, the information in the Release Notes supersedes that in this document. Updates to this document and other documents as well as software files can be downloaded by registered customers at http://www.audiocodes.com/downloads.

Date Published: April-29-2015

## Trademarks

AudioCodes, AC, AudioCoded, Ardito, CTI2, CTI², CTI Squared, HD VoIP, HD VoIP Sounds Better, InTouch, IPmedia, Mediant, MediaPack, NetCoder, Netrake, Nuera, Open Solutions Network, OSN, Stretto, TrunkPack, VMAS, VoicePacketizer, VoIPerfect, VoIPerfectHD, What's Inside Matters, Your Gateway To VoIP and 3GX 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.

## Abbreviations and Terminology

Each abbreviation, unless widely used, is spelled out in full when first used.

## Document Revision Record

| LTRT | Description |
|---|---|
| 52331 | Initial document release for Version 6.8. |
| 52333 | MP-118 added. |

## 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 Documentation Feedback form on our Web site at http://www.audiocodes.com/downloads.

# 1 Introduction

This document describes the CPE WAN Management Protocol (TR-069), intended for communication between a CPE and Auto-Configuration Server (ACS). The CPE WAN Management Protocol defines a mechanism that encompasses secure auto-configuration of a CPE, and also incorporates other CPE management functions into a common framework.

The CPE WAN Management Protocol is intended to support a variety of functionalities to manage a collection of CPE, including the following primary capabilities:

- Auto-configuration and dynamic service provisioning
- Software/firmware image management
- Software module management
- Status and performance monitoring
- Diagnostics

**Note:** CWMP is supported by the Multi-Service Business Router (MSBR) product line as well as the MediaPack MP-1xx product line.

This page is intentionally left blank.

# 2 Overview

The following describes the TR-069 and TR-104 models.

## 2.1 TR-069 Data Model

TR-069 is a specification published by Broadband Forum (www.broadband-forum.org) entitled CPE WAN management protocol (CWMP). It defines an application layer protocol for remote management of end-user devices.

TR-069 uses a bi-directional SOAP/HTTP protocol for communication between the customer premises equipment (CPE) and the Auto Configuration Servers (ACS). For MSBR devices, the TR-069 connection to the ACS can be done on the LAN or WAN interface.

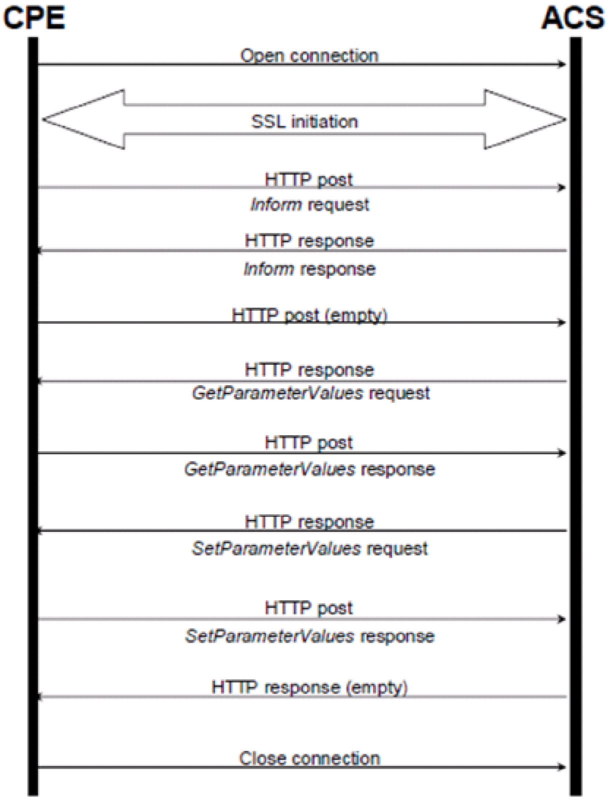
The protocol stack looks as follows:

**Figure 2-1: TR-069 Protocol Stack**

| **CPE/ACS Management Application** |
|---|
| RPC Methods |
| SOAP |
| HTTP |
| SSL/TLS |
| TCP/IP |

Communication is typically established by the CPE; hence, messages from CPE to ACS are typically carried in HTTP requests, and messages from ACS to CPE in HTTP responses.

**Figure 2-2: TR-069 Session Example**

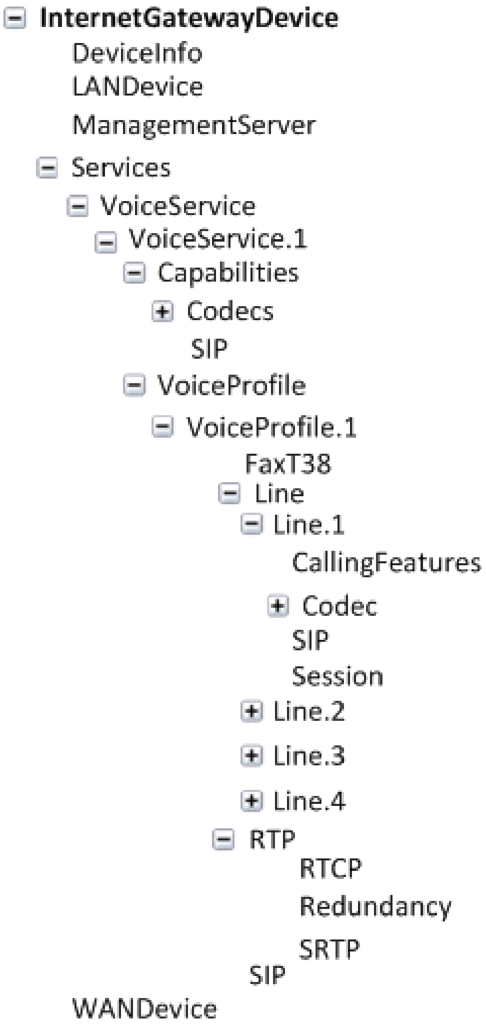


Communication between ACS and CPE is defined via Remote Procedure Call (RPC) methods. TR-069 defines a generic mechanism by which an ACS can read or write parameters to configure a CPE and monitor CPE status and statistics. It also defines the mechanism for file transfer and firmware/software management. However, it does not define individual parameters; these are defined in separate documents, as described below.

Some of the RPC methods are Configuration File Download, Firmware upgrade, Get Parameter Value, Set Parameter Value, Reboot, and the upload and download files.

TR-106 defines the “data model” template for TR-069 enabled devices. The Data Model consists of objects and parameters hierarchically organized in a tree with a single Root Object, typically named *Device*. Arrays of objects are supported by appending a numeric index to the object name (e.g. ABCService.1 in the example below); such objects are called “multi-instance objects”.

**Figure 2-3: TR-069 Model Data Example**



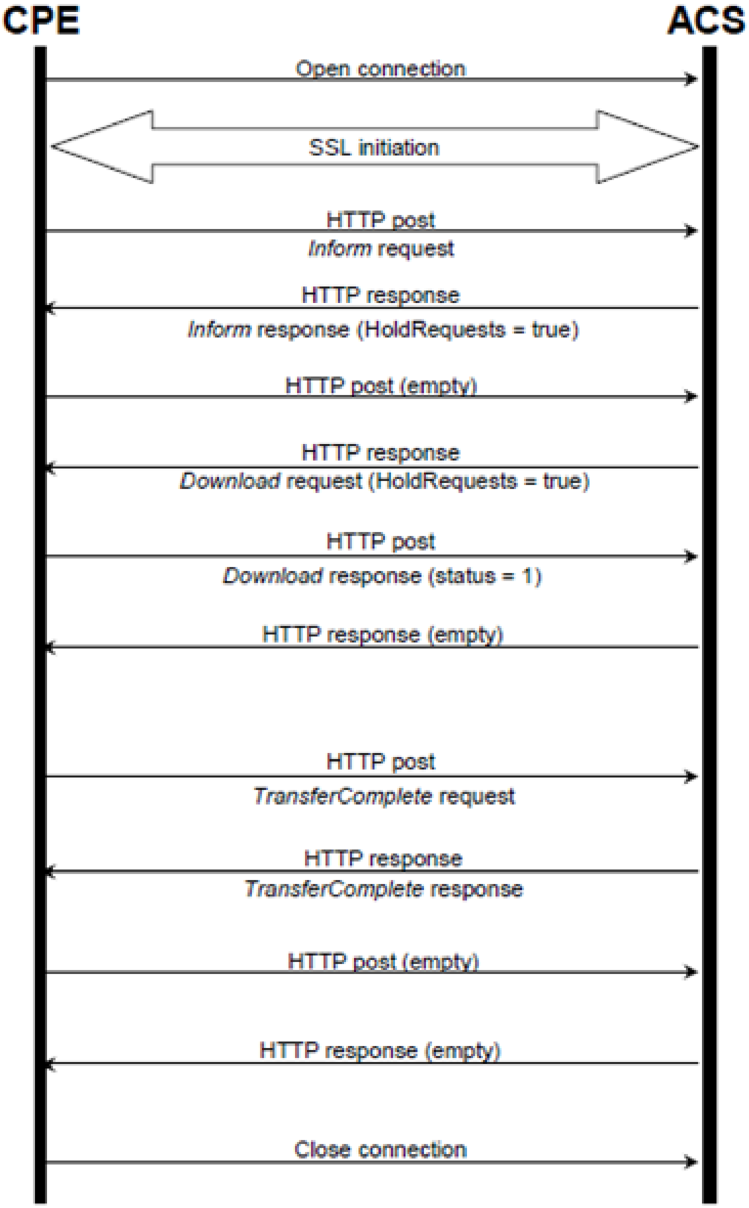
The following is a list of some of the TR-069 methods:

- **CPE Methods:**
  - GetRPCMethods: Used by the CPE or ACS to discover the set of methods supported by the Server or CPE it is in communication with.
  - SetParameterValues: Used by the ACS to modify the value of CPE parameter(s).
  - GetParameterValues: Used by the ACS to obtain the value of CPE parameter(s).
  - GetParameterNames: Used by the ACS to discover the parameters accessible on a particular CPE.
  - SetParameterAttributes: Used by the ACS to modify attributes associated with CPE parameter(s).
  - GetParameterAttributes: Used by the ACS to read the attributes associated with CPE parameter(s).
  - AddObject: Used by the ACS to create a new instance of a multi-instance object—a collection of parameters and/or other objects for which multiple instances are defined.
  - DeleteObject: Removes a particular instance of an object.

- Download: Used by the ACS to cause the CPE to download the following file(s) from a designated location:
  - Firmware Upgrade Image (File Type = 1) - cmp file.
  - Vendor Configuration File (File Type = 3) - output of `show running-config` CLI command, which includes Data (MSBR devices only) and Voice configuration.

  The CPE responds to the Download method, indicating successful or unsuccessful completion via one of the following:
  - A DownloadResponse with the Status argument set to zero (indicating success), or a fault response to the Download request (indicating failure).
  - A TransferComplete message sent later in the same session as the Download request (indicating either success or failure). In this case, the Status argument in the corresponding DownloadResponse has a value of one.
  - A TransferComplete message sent in a subsequent session (indicating success or failure). In this case, the Status argument in the corresponding DownloadResponse has a value of one.

**Figure 2-4: Download Method Execution Example**



- Upload: Used by the ACS to cause the CPE to upload (to the ACS) the following files to a designated location:
    - Vendor Configuration File (File Type = 1 or 3): Output of `show running-config` CLI command, which includes Data (MSBR devices only) and Voice configuration. For File Type 3 (where index is included – see below) only one instance of the file is supported.
    - Vendor Log File (File Type = 2 or 4): “Aggregated” log file. For File Type 2, the last file is supported. For File Type 4 (where index is included – see below), multiple files is supported.

  The CPE responds to the Upload method, indicating successful or unsuccessful completion via the UploadResponse or TransferComplete method.

  For a complete description of the Upload method, refer to TR-069 Amendment 3 section A.4.1.5.
- Reboot: Reboots the CPE. The CPE sends the method response and completes the remainder of the session prior to rebooting.
- X_0090F8_CommandResponse: Runs CLI commands.

### Proprietary RPC Commands:

The following is an example of our proprietary RPC commands.

A number of CLI commands can be sent together from the ACS to the CPE.

This number must be the value of the *<number_of_cli_commands>* parameter, in the third line of this example.

The CPE replies with a status for each CLI command indicating whether or not it completed successfully.

```
<cwmp:X_00908F_RunCommand>
    <Timeout>100000</Timeout>
      <Commands soap-enc:arrayType="xsd:string[<number_of_cli_commands>]">
        <string>cli_command_1</string>
        <string>cli_command_2</string>
        …
        <string>cli_command_n</string>
    </Commands>
</cwmp:X_00908F_RunCommand>
```

The CPE replies with the following:

```
<xmp ><cwmp:X_00908F_RunCommandResponse xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
<CommandList xmlns:soap-enc="http://schemas.xmlsoap.org/soap/encoding/" soap-
enc:arrayType="cwmp:CommandStruct[5]">
  <CommandStruct>
    <Command> cli_command_1</Command>
    <Status>command_succesful_state</Status>
   <Output>cli_reply_1<Output/>
  </CommandStruct>
  <CommandStruct>
….
<CommandStruct>
    <Command> cli_command_n</Command>
    <Status>command_succesful_state</Status>
   <Output>cli_reply_n<Output/>
  </CommandStruct>
  <CommandStruct>
</CommandList>
</cwmp:X_0090F8_RunCommandResponse>
```

- **ACS Methods:**
  - Inform: A CPE must call this method to initiate a transaction sequence whenever a connection to an ACS is established.
  - TransferComplete: Informs the ACS of the completion (either successful or unsuccessful) of a file transfer initiated by an earlier Download or Upload method call.

## 2.1.1 ManagementServer

**Table 2-1: ManagementServer Parameters**

| Parameter Name | Description |
|---|---|
| URL | Defines the URL for the CPE to connect to the ACS using the WAN Management Protocol (e.g., http://10.37.5.5:8080/dps-basic/TR069). |
| Username | Defines the username used to authenticate the CPE when connecting to the ACS (e.g., tr069). |
| Password | Defines the password used to authenticate the CPE when connecting to the ACS (e.g., *****). |
| PeriodicInformEnable | Determines whether or not the CPE must periodically send CPE information to the ACS using the Inform method call. |
| PeriodicInformInterval | Defines the duration in seconds of the interval for which the CPE must attempt to connect with the ACS and call the Inform method if the PeriodicInformEnable parameter is set to true (e.g., 123). |
| PeriodicInformTime | Defines the absolute time reference in UTC to determine when the CPE should initiate the Inform method calls (e.g., 1970-01-01T00:00:00Z). |
| ParameterKey | Provides the ACS with a reliable and extensible means to track changes made by the ACS (e.g., AUDC). |
| ConnectionRequestURL | Defines the HTTP URL for an ACS to make a Connection Request notification to the CPE in the format of: http://host:port/path (e.g., http://10.31.3.11:82/tr069/). |
| ConnectionRequestUsername | Defines the username used to authenticate an ACS making a Connection Request to the CPE (e.g., ftacs). |
| ConnectionRequestPassword | Defines the password used to authenticate an ACS making a Connection Request to the CPE (e.g., *****). |
| UpgradesManaged | Indicates whether or not the ACS will manage upgrades for the CPE:<br>▪ If set to "1", the CPE should not use other means other than the ACS to seek out available upgrades.<br>▪ If set to "2", the CPE may use other means for this purpose. |

### 2.1.2 DeviceInfo

**Table 2-2: DeviceInfo Parameters**

| Parameter Name | Description |
|---|---|
| Manufacturer | Defines the manufacturer of the CPE (e.g., AudioCodes). |
| ManufacturerOUI | Defines an Organizationally Unique Identifier (OUI) of the associated Internet Gateway Device. An empty string indicates that no associated Internet Gateway Device has been detected (e.g., 00908F). |
| ModelName | Defines the model name of the CPE (e.g., M800 MSBR). |
| Description | Defines the description of the CPE device (e.g., M800 MSBR). |
| ProductClass | Defines the identifier of the class of product for which the serial number applies (e.g., M800 MSBR). |
| SerialNumber | Defines the serial number of the CPE (e.g., 2261520). |
| HardwareVersion | Defines CPE model and version (e.g., CPU Speed [MHz]: 300). |
| SoftwareVersion | Defines the software version currently installed in the CPE (e.g., 6.60Da.147.005). |
| ModemFirmwareVersion | Defines the version of the modem firmware currently installed in the CPE (e.g., 11.234). |
| EnabledOptions | Defines a list of options currently enabled in the CPE (e.g., Default-String-Value). |
| AdditionalHardwareVersion | Defines additional hardware version information the vendor may wish to supply (e.g., DSP Software Name: 5014AE3_R_LD). |
| AdditionalSoftwareVersion | Defines additional software version information the vendor may wish to supply (e.g., Flash Version: 710). |
| SpecVersion | Defines the version of the specification implemented by the device. Currently 1.0 is the only available version  (e.g., 1.0). |
| ProvisioningCode | Defines the identifier of the primary service provider and other provisioning information, which may be used by the ACS to determine service provider-specific customization and provisioning parameters (e.g., 1.0). |
| UpTime, sec | Defines the time in seconds since the CPE was last restarted (e.g., 98). |
| FirstUseDate | Defines the date and time in UTC that the CPE successfully established an IP-layer network connection and acquired an absolute time reference using NTP or equivalent over that network connection.(e.g., Unknown Time value). |
| DeviceSummary | Provides an explicit summary of the top-level data model of the device, including version and profile information (e.g., “Device:1.0[](Baseline:1), ABCService:1.0[1](Baseline:1)”). |

## 2.2 TR-104 Data Model

TR-104 defines the data model for the provisioning of a voice-over-IP (VoIP) CPE device by an Auto-Configuration Server (ACS) using the mechanism defined in TR-069.

The following profiles are implemented by MSBR products:

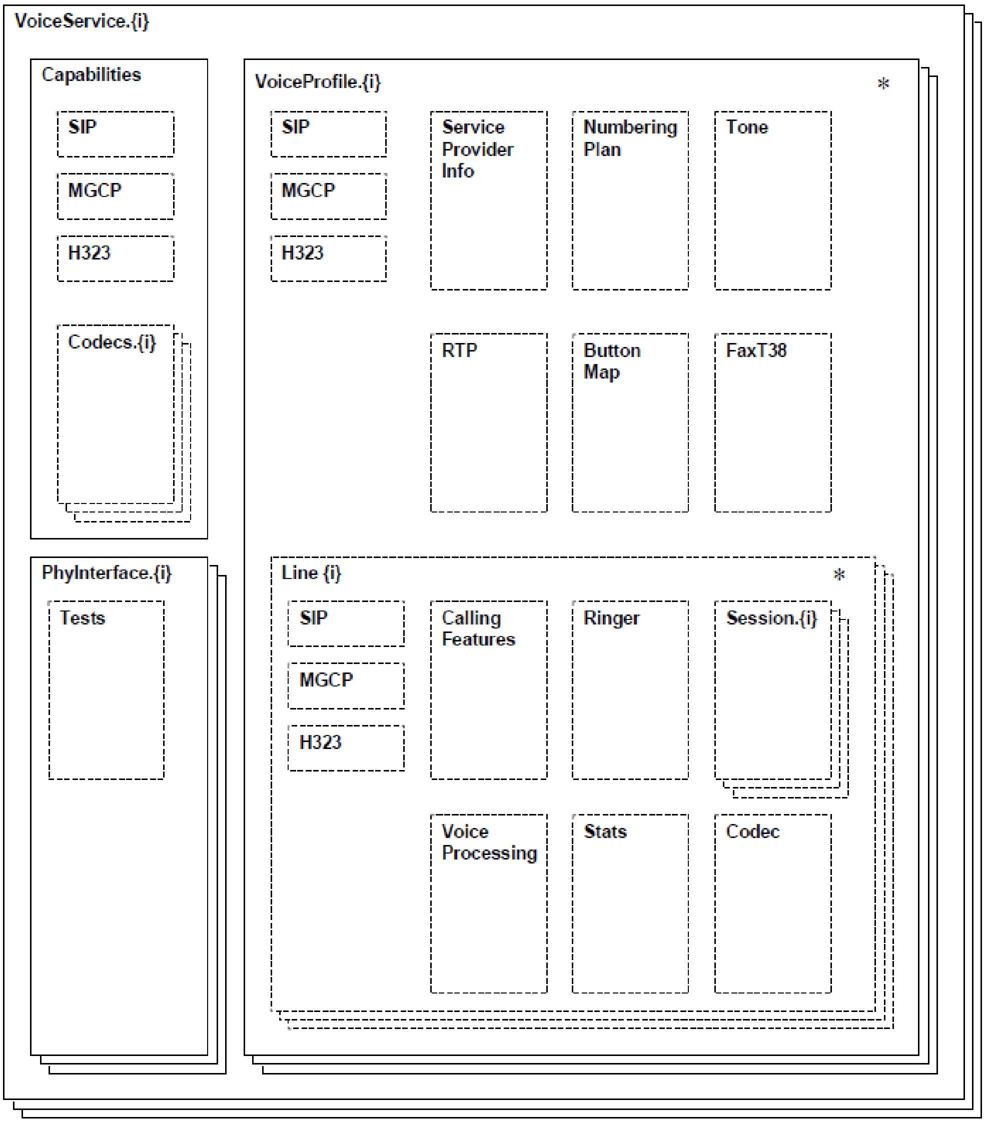
- Endpoint Profile
- SIP Endpoint Profile

**Note:** TR-104 is supported only for FXS interfaces.

### 2.2.1 Major Elements

The following diagram provides an overview of all objects defined in TR-104:

**Figure 2-5: Objects Defined in TR-104**



TR-104 describes functionality of a VoIP device that has one or more FXS lines.

The MSBR configuration model differs considerably from the model described in TR-104. For example, instead of the *VoiceProfile* object that corresponds to a group of phone lines, a separate *TelProfile* and/or IP Profile objects are used. TR-104 uses an implicit routing model while MSBR supports explicit routing tables that may be used to implement much more complicated call routings.

To “bridge” between these two very different configuration models, the following is assumed regarding the way MSBR is configured:

- Default Tel Profile (0) is used for all Hunt Groups (FXS Lines).
- “IP to Tel” Routing for phone lines (FXS ports) is implemented via the *PSTNPrefix* table. For each phone line (FXS port), a separate and unique entry in this table is created.
- “Tel to IP” routing is performed by the default *ProxySet (0)* and when the *IsProxySet* parameter is set to "1" (enabled). The first IP address entry in the Proxy Set configuration is used.
- Default IP Profile (0) is used for all routing rules.
- Outbound Proxy (if needed) is implemented via a single line in the *PSTNPrefix* table with ‘*’ wildcard for source and destination prefix.

## 2.2.2 VoiceService

VoiceService is a container "service" object as defined in TR-106.

MSBR implements a single instance of this object – VoiceService.1 only if it is equipped with the FXS ports.

### 2.2.2.1 VoiceService.{i}.Capabilities.Codecs

Codecs table describes the set of supported codecs. The table is read-only. Line.{i}.Codec.List table is used instead to customize list and parameters of coders assigned to the specific phone line.

MSBR will populate this table with *all supported* coders (as per CoderName_Type and CoderName_Rate ENUMs, taking into account the current DSP template). Each entry will contain "default" parameters (e.g. p-time) that correspond to the specific coder.

### 2.2.2.2 VoiceService.{i}.VoiceProfile

VoiceProfile corresponds to one or more phone lines (FXS ports) that share the same basic configuration.

MSBR implements a single instance of the VoiceProfile.1 object.

### 2.2.2.3 VoiceService.{i}.VoiceProfile.{i}.Line

The Line object corresponds to a single phone line (FXS port).

MSBR implements an instance of this object for each phone line (FXS port) configured in Hunt Group (TrunkGroup) table. Add/remove operations will be supported to allow configuration/removal of specific FXS port (See details in the table below).

### 2.2.2.4 VoiceService.{i}.VoiceProfile.{i}.Line.{i}.Codec

The Codec object describes voice coder currently used by the specific phone line. In addition to that it provides a list of coders (Codec.List.{i}) enabled on the specific line and the ability to modify this configuration.

MSBR implements a global Coder table that is shared by all phone lines (FXS ports). It allows "per-line" customization of coders via Coder Group table. However, use of this functionality requires use of different Tel Profiles for different Hunt Groups (FXS port) and number of coder groups that may be configured is limited to 4.

Map Codec.List.{i} table directly to Coders (CoderName) table. This essentially means that all lines share the same configuration and configuration change for one line immediately affects all other lines.

### 2.2.2.5 VoiceService.{i}.VoiceProfile.{i}.Line.{i}.SIP

The Line.{i}.SIP object contains username/password assigned to a specific phone line (FXS port).

MSBR will map this object to corresponding entry in Authentication Table.

## 2.2.3 Endpoint Profile

MSBR complies with Version 1 of Endpoint profile (Endpoint:1) as defined in TR-104 Section 4.2.

**Table 2-3: Endpoint Profile Table**

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| **InternetGatewayDevice. Services.VoiceService.{i}.** | object | – | The top-level object. | A single instance of VoiceService object will be created. |
| VoiceProfileNumberOfEntries | unsignedInt | – | Defines the number of instances of VoiceProfile. | 1 |
| **.VoiceService.{i}.Capabilities.** | object | – | Defines the overall capabilities of the VoIP CPE. | |
| MaxProfileCount | unsignedInt | – | Defines the maximum total number of distinct voice profiles supported. | 1 |
| MaxLineCount | unsignedInt | – | Defines the maximum total number of lines supported across all profiles.<br>This parameter is applicable only for a VoIP endpoint. | Total number of FXS interfaces (e.g. 4). |
| MaxSessionsPerLine | unsignedInt | – | Defines the maximum number of voice sessions supported for any given line across all profiles. A value greater than one indicates support for CPE provided conference calling.<br>This parameter is applicable only for a VoIP endpoint. | 1 |
| MaxSessionCount | unsignedInt | – | Defines the maximum total number of voice sessions supported across all lines and profiles. (This might differ from MaxLineCount if each line can support more than one session for CPE provided conference calling. This value MAY be less than the product of MaxLineCount and MaxSessionsPerLine.) | same value as MaxLineCount |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| SignalingProtocols | string(256) | – | Defines the comma-separated list of signaling protocols supported. Each item is an enumeration of:<br>“SIP”<br>“MGCP”<br>“MGCP-NCS”<br>“H.248”<br>“H.323”<br>Each entry MAY be appended with a version indicator in the form “/X.Y”. For example: “SIP/2.0”<br>The list MAY include vendor-specific protocols, which MUST be in the format defined in [3]. For example: “X_EXAMPLE-COM_MyProt” | “SIP” |
| Regions | string(256) | – | Defines the comma-separated list of geographic regions supported by the CPE. Each item is the list MUST be an alpha-2 (two-character alphabetic) country code as specified by ISO 3166.<br>An empty list indicates that the CPE does not support region-based customization via the Region parameter in the VoiceService.{i}.VoiceProfile.{i} object. | <empty> |
| RTCP | boolean | – | Defines support for RTCP. A true value indicates support for the object VoiceService.{i}.VoiceProfile.{i}.RTP.RTCP.<br>This parameter is applicable only for a VoIP endpoint. | True |
| SRTP | boolean | – | Defines support for SRTP. A true value indicates support for the object VoiceService.{i}.VoiceProfile.{i}.RTP.SRTP. A true value also indicates that the SRTPKeyingMethods and SRTPEncryptionKeySizes parameters in this object are present.<br>This parameter is applicable only for a VoIP endpoint. | True |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| SRTPKeyingMethods | string(256) | – | Defines comma-separated list of keying protocols supported by this endpoint for SRTP. Each item is an enumeration of:<br>“Null”<br>“Static”<br>“SDP”<br>“IKE”<br>This list MAY include vendor-specific keying methods, which MUST use the format defined in [3].<br>This parameter is applicable only if the SRTP parameter in this object is equal to true.<br>**Note:** This parameter is NOT part of *EndpointProfile*, but it’s mandatory when SRTP is supported. | “Static” |
| SRTPEncryptionKeySizes | string(256) | – | Defines comma-separated list of unsigned integers, each represented a supported SRTP encryption key size.<br>This parameter is applicable only if the SRTP parameter in this object is equal to true. | according to SRTPofferedSuites parameter possible values, the only supported value is “128” |
| RTPRedundancy | boolean | – | Defines support for RTP payload redundancy as defined in RFC 2198. A true value indicates support for VoiceService.{i}.VoiceProfile.{i}.RTP.Redundancy.<br>This parameter is applicable only for a VoIP endpoint. | True |
| DSCPCoupled | boolean | – | Defines a true value that indicates that the CPE is constrained such that transmitted call control packets use the same DSCP marking as transmitted RTP packets.<br>If the value is true, the CPE MUST NOT support the DSCPMark parameter for call control.<br>This parameter is applicable only for a VoIP endpoint. | False |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| EthernetTaggingCoupled | boolean | – | Defines a true value that indicates that the CPE is constrained such that transmitted call control packets use the same Ethernet tagging (VLAN ID Ethernet Priority) as transmitted RTP packets.<br>If the value is true, the CPE MUST NOT support the VLANIDMark or EthernetPriorityMark parameters within a call control object (e.g., SIP, MGCP, or H323).<br>This parameter is applicable only for a VoIP endpoint. | False |
| PSTNSoftSwitchOver | boolean | – | Defines a true value that indicates the CPE is capable of supporting the PSO_Activate Facility Action, which allows a call to be switched to a PSTN FXO (Foreign eXchange Office) line.<br>This parameter is applicable only for a VoIP endpoint. | False |
| FaxT38 | boolean | – | Defines support for T.38 fax. A true value indicates support for the object VoiceService.{i}.VoiceProfile.{i}.FaxT38.<br>This parameter is applicable only for a VoIP endpoint. | True |
| FaxPassThrough | boolean | – | Defines support for fax pass-through. A true value indicates support for the parameter VoiceService.{i}.VoiceProfile.{i}.FaxPassThrough.<br>This parameter is applicable only for a VoIP endpoint. | True |
| ModemPassThrough | boolean | – | Defines support for modem pass-through. A true value indicates support for the parameter VoiceService.{i}.VoiceProfile.{i}.ModemPassThrough.<br>This parameter is applicable only for a VoIP endpoint. | True |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| ToneGeneration | boolean | – | Defines support for tone generation. A true value indicates support for the object VoiceService.{i}.VoiceProfile.{i}.Tone.<br>A true value also indicates that the ToneDescriptionsEditable, PatternBasedToneGeneration, and FileBasedToneGeneration parameters in this object are present.<br>This parameter is applicable only for a VoIP endpoint. | False |
| RingGeneration | boolean | – | Defines support for ring generation. A true value indicates support for control of ring generation via the VoiceService.{i}.VoiceProfile.{i}.Line.{i}.Ringer object.<br>A true value also indicates that the RingDescriptionsEditable, PatternBasedRingGeneration, and FileBasedRingGeneration parameters in this object are present.<br>This parameter is applicable only for a VoIP endpoint. | False |
| NumberingPlan | boolean | – | Defines support for a configurable numbering plan. A true value indicates support for a configurable numbering plan via the VoiceService.{i}.VoiceProfile.{i}.NumberingPlan object.<br>This parameter is applicable only for a VoIP endpoint. | False |
| ButtonMap | boolean | – | Defines support for a configurable button map. A true value indicates support for a configurable button map via the VoiceService.{i}.VoiceProfile.{i}.ButtonMap object.<br>This parameter is applicable only for a VoIP endpoint. | False |
| VoicePortTests | boolean | – | Defines support for remotely accessible voice-port tests.<br>A true value indicates support for the VoiceService.{i}.PhyInterface.{i}.Tests object.<br>This parameter is applicable only for a VoIP endpoint. | False |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| **.VoiceService.{i}.Capabilities.Codecs.{i}.** | object | – | Table to describe the set of supported codecs.<br>Each entry refers to a distinct combination of codec and bit rate. The table MUST include a distinct entry for each supported combination of these parameters.<br>Applicable only for a VoIP endpoint. | This table will be populated with entries for each supported coder (as per CoderName_Type enum and applicable bit-rates). If possible we should limit list of supported coders to those supported in current DSP template only.<br>For each entry, default coder parameters (e.g. p-time) will be specified.<br>Note: This table is NOT mapped to Coders (CoderName) table. The latter is mapped to Line.{i}.Codec.List table. |
| EntryID | unsignedInt [1:] | – | Defines a unique identifier for each entry in this table. | |
| Codec | string(64) | – | Defines an identifier of the type of codec. Enumeration of:<br>“G.711MuLaw”<br>“G.711ALaw”<br>“G.726”<br>“G.729”<br>“G.729a”<br>“G.729e”<br>“G.728”<br>“G.723.1”<br>“G.722”<br>“G.722.1”<br>“G.722.2”<br>“GSM-FR”<br>“GSM-HR”<br>“GSM-EFR”<br>“iLBC”<br>The parameter MAY instead be a vendor-specific codec, which MUST be in the format defined in [3]. For example: “X_EXAMPLE-COM_MyCodec” | |
| BitRate | unsignedInt | – | Defines a Bit Rate in bits per second. The value MUST be among the values appropriate for the specified codec. | |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| PacketizationPeriod | string(64) | – | Defines a comma-separated list of supported packetization periods, in milliseconds, or continuous ranges of packetization periods. Ranges are indicated as a hyphen-separated pair of unsigned integers.<br>Examples:<br>“20” indicates a single discrete value.<br>“10, 20, 30” indicates a set of discrete<br>values.<br>“5-40” indicates a continuous inclusive range.<br>“5-10, 20, 30” indicates a continuous range in addition to a set of discrete values<br>A range MUST only be indicated if all values within the range are supported. | |
| SilenceSuppression | boolean | – | Indicates support for silence suppression for this codec. | |
| **.VoiceService.{i}.VoiceProfile.{i}.** | object | C | Support for creation and deletion of Profiles is REQUIRED only if more than one Profile is supported as indicated by VoiceService.{i}.Capabilities.MaxProfileCount. | A single instance of VoiceProfile object will be created.<br>Add/remove operations will NOT be supported. |
| Enable | string | W | Enables or disables all lines in this profile, or places it into a quiescent state.<br>Enumeration of:<br>“Disabled”<br>“Quiescent”<br>“Enabled”<br>On creation, a profile MUST be in the Disabled state.<br>In the Quiescent state, in-progress sessions remain intact, but no new sessions are allowed.<br>Support for the Quiescent state in a CPE is optional. If this parameter is set to “Quiescent” in a CPE that does not support the Quiescent state, it MUST treat it the same as the Disabled state. | Only ‘Enabled’ state will be supported. |
| NumberOfLines | unsignedInt | – | Defines  the number of instances of Line within this VoiceProfile.<br>Applicable only for a VoIP endpoint. | Number of entries in Hunt Group (TrunkGroups) table of type FXS. |
| Name | string(64) | W | Defines a human-readable string to identify the profile instance. | ‘Default Profile’<br>Write operation will not be supported. |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| SignalingProtocol | string(64) | W | Defines the protocol to be used for this profile. A single protocol selected from among the available protocols indicated in VoiceService.{i}.Capabilities .SignalingProtocols. | “SIP” |
| MaxSessions | unsignedInt | W | Defines the limit on the number of simultaneous voice sessions across all lines in this profile. Must be less than or equal to VoiceService.{i}.Capabilities .-<br>MaxSessionCount. (This MAY be greater than the number of lines if each line can support more than one session, for example for CPE provided conference calling.) | We will assume that a single session is supported per line – hence we will use the same value as for NumberOfLines. |
| DTMFMethod | string(64) | W | Defines the method by which DTMF digits MUST be passed.<br>Enumeration of:<br>“InBand”<br>“RFC2833”<br>“SIPInfo”<br>If the parameter DTMFMethodG711 is non-empty, then this parameter applies only when the current codec is not G.711.<br>The value “SIPInfo” is applicable only if the SignalingProtocol is SIP.<br>This parameter is applicable only for a VoIP endpoint. | RxDTMFOption==0 && TxDTMFOption==1 → “SIPInfoNortel”<br>RxDTMFOption==0 && TxDTMFOption==3 → “SIPInfo”<br>RxDTMFOption==0 && TxDTMFOption==2 → “Notify”<br>RxDTMFOption==3 && TxDTMFOption==4 → “RFC2833”<br>RxDTMFOption==0 && TxDTMFOption==0 && DTMFTransportType==2 → “InBand”<br>Any other combination → “Other” (read-only)<br>**Note:** These are _global_ parameters. They may be overwritten by setting corresponding parameter in IPProfile. If we manage to “link” TelProfile and IPProfile together (see …) , we’ll use parameters from IPProfile; otherwise global parameters will be used. |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| DTMFMethodG711 | string(64) | W | Defines the method by which DTMF digits MUST be passed if the current codec is G.711.<br>Enumeration of:<br>“InBand”<br>“RFC2833”<br>“SIPInfo”<br>An empty value for this parameter indicates that the value of the DTMFMethod parameter is to apply whether or not the current codec is G.711.<br>The value “SIPInfo” is applicable only if the SignalingProtocol is SIP.<br>This parameter is applicable only for a VoIP endpoint. | We support “empty” values only. |
| **.VoiceService.{i}.VoiceProfile.{i}.RTP.** | object | – | Voice profile parameters related to the voice stream sent via RTP.<br>Applicable only for a VoIP endpoint. | |
| LocalPortMin | unsignedInt [0:65535] | W | Defines the base of port range to be used for incoming RTP streams for this profile. | BaseUDPport<br>(or use Media Realm’s Port Range Start, if Media Realm is defined) |
| DSCPMark | unsignedInt [0:63] | W | | |
| TelephoneEventPayloadType | unsignedInt [0:128] | W | Defines the payload type to be used for RTP telephone events.<br>This parameter indicates the payload type to be used for DTMF events if RFC 2833 transmission of DTMF information is in use. | According to RFC 2833 PayloadType. |
| **.VoiceService.{i}.VoiceProfile.{i}.Line.{i}.** | object | C | Defines  the support for creation and deletion of Lines is REQUIRED only if more than one Line is supported as indicated by VoiceService.{i}.Capabilities .MaxLineCount. | An instance of this object is created for each Hunt Group / TrunkGroup entry of type FXS.<br>Add operation will be supported only if some “unused” FXS line exists. It will create a new entry in Hunt Group / TrunkGroup table.<br>Remove operation will delete corresponding entry from Hunt Group table. |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| Enable | string | W | Enables or disables this line, or places it into a quiescent state. Enumeration of:<br>"Disabled"<br>"Quiescent"<br>"Enabled"<br>On creation, a line MUST be in the Disabled state.<br>In the Quiescent state, in-progress sessions remain intact, but no new sessions are allowed.<br>Support for the Quiescent state in a CPE is optional. If this parameter is set to "Quiescent" in a CPE that does not support the Quiescent state, it MUST treat it the same as the Disabled state (and indicate Disabled in the Status parameter). | Quiescent state is not supported.<br>Disabled state is implemented "at TR069 level only" (the entry will NOT exist in Hunt Group table). |
| Status | string | – | Indicates the status of this line. Enumeration of:<br>"Up"<br>"Initializing"<br>"Registering"<br>"Unregistering"<br>"Error"<br>"Testing"<br>"Quiescent"<br>"Disabled" | The following statuses are supported:<br>- Up<br>- Disabled |
| CallState | string | – | Indicates the call state for this line. Enumeration of:<br>"Idle"<br>"Calling"<br>"Ringing"<br>"Connecting"<br>"InCall"<br>"Hold"<br>"Disconnecting" | Need to query SIP database to get this info. |
| **.VoiceService.{i}.VoiceProfile.{i}.Line.{i}.Codec.** | object | – | This object indicates the state of the transmit and receive codec for this voice line instance. | **Note:** This object shows the data about currently established media session (i.e. the data should be taken from MediaEngine). |
| TransmitCodec | string(64) | – | Defines the codec currently in use for the outgoing voice stream. Enumeration from the list of available codecs as given in the VoiceService.{i}.Capabilities.Codecs table. | |
| ReceiveCodec | string(64) | – | Defines the codec currently in use for the incoming voice stream. Enumeration from the list of available codecs as given in the VoiceService.{i}.Capabilities.Codecs table. | |
| TransmitBitRate | unsignedInt | – | Defines the codec bit rate in bits per second for the codec currently in use for the outgoing voice stream. | |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| ReceiveBitRate | unsignedInt | – | Defines the codec bit rate in bits per second for the codec currently in use for the incoming voice stream. | |
| TransmitSilenceSuppression | boolean | – | Defines whether or not silence suppression is in use for the outgoing voice stream. | |
| ReceiveSilenceSuppression | boolean | – | Defines whether or not silence suppression is in use for the incoming voice stream. | |
| TransmitPacketizationPeriod | unsignedInt | – | Defines the current outgoing packetization period in milliseconds. | |
| **.VoiceService.{i}.VoiceProfile.{i}.Line.{i}.Codec.List.{i}.** | object | – | Table to describe the set of codecs enabled for use with this line. Each entry in this table refers to a distinct combination of codec and bit rate.<br>When a Line is created, this object MUST be populated with the set of supported codecs matching the VoiceService.{i}.Capabilities.Codecs table. The ACS MAY restrict and/or prioritize the codec support for this profile using this object.<br>Applicable only for a VoIP endpoint. | The table will be populated with all supported coders (similar to Capabilities.Codecs table).<br>Each “Enabled” object will be mapped to the entry in Coders (CoderName) table. Index in Coders table will be determined according to the object’s Priority parameter.<br>“Disabled” objects will exist “at TR-069 level only”. |
| EntryID | unsignedInt [1:] | – | Defines  the unique identifier for each entry in this table. The value MUST match that of the corresponding entry in the VoiceService.{i}.Capabilities.Codecs table. | Similar to Capabilities.Codecs |
| Codec | string(64) | – | Defines  the Identifier of the codec type. The value MUST match that of the corresponding entry in the VoiceService.{i}.Capabilities.Codecs table. | Initial value - similar to Capabilities.Codecs; when “Enabled” – CoderName_Type |
| BitRate | unsignedInt | – | Defines the Bit rate in bits per second. The value MUST match that of the corresponding entry in the VoiceService.{i}.Capabilities.Codecs table. | Initial value - similar to Capabilities.Codecs; ; when “Enabled” – CoderName_Rate |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| PacketizationPeriod | string(64) | W | Defines the comma-separated list of supported packetization periods, in milliseconds, or continuous ranges of packetization periods as defined in VoiceService.{i}. Capabilities.Codecs.PacketizationPeriod.<br>The set of packetization periods may be restricted by modifying the value of this parameter to a more restricted set of values than is listed in VoiceService.{i}.Capabilities. Codecs.PacketizationPeriod. The CPE MUST ignore any values or portions of ranges outside of those specified in VoiceService.{i}.Capabilities. Codecs.Packetization.Period. | Initial value – similar to Capabilities.Codecs; when “Enabled” – CoderName_PacketInterval<br>Can be modified by user. |
| SilenceSuppression | boolean | W | Indicates support for silence suppression for this codec. If silence suppression is supported, it can be disabled for this codec/bit-rate by setting this parameter to false | Initial value – similar to Capabilities.Codecs; when “Enabled” – CoderName_SCE<br>Can be modified by user. |
| Enable | boolean | W | This parameter is REQUIRED to be writable only if there is more than one entry in this table. | When set to ‘true’ corresponding entry in Coders (CoderName) table is created (with index that corresponds to Priority parameter).<br>***When set to ‘false’, corresponding entry is removed from the Coders table***.<br>**Note:** in DR add 2 options: Either not able option to put DISABLE to this param, or add new field of Admin-State to Coder Tables |
| Priority | unsignedInt [1:] | W | This parameter is REQUIRED to be writable only if there is more than one entry in this table. | Will be used to determine index of entry in Coders (CoderName) table. When value is changed and Enable==’true’ existing Coders entry will be removed and a new entry will be created instead (with a new index). |
| **.VoiceService.{i}.VoiceProfile.{i}.Line.{i}.Session.{i}.** | object | – | Information on each active session associated with this voice line instance. | Information about currently active call – should be taken from SIP DB. |
| SessionStartTime | date time | – | Defines the time that the session started, in UTC. | |
| SessionDuration | Unsignedint | – | Defines the duration time of the current session, in seconds. | |
| FarEndIPAddress | string | – | Defines the IP address of far end VoIP device. | |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| FarEndUDPPort | unsignedInt [0:65535] | – | Defines the UDP port used for current RTP session in the far end device. | |
| LocalUDPPort | unsignedInt [0:65535] | – | Defines the local UDP port used for current RTP session. | |
| **.VoiceService.{i}.VoiceProfile.{i}.Line.{i}.Stats.** | object | – | Statistics for this voice line instance. | Will be mapped to corresponding PMs (if such exist), SIP or VE counters. We may need to create new PMs to support all needed counters. |
| ResetStatistics | boolean | W | Defines when set to one, resets the statistics for this voiceline. Always False when read. | |
| PacketsSent | unsignedInt | – | Defines the total number of RTP packets sent for this line. | |
| PacketsReceived | unsignedInt | – | Defines the total number of RTP packets received for this line. | |
| BytesSent | unsignedInt | – | Defines the total number of RTP payload bytes sent for this line. | |
| BytesReceived | unsignedInt | – | Defines the total number of RTP payload bytes received for this line. | |
| PacketsLost | unsignedInt | – | Defines the total number of RTP packets that have been lost for this line. | |
| Overruns | unsignedInt | – | Defines the total number of times the receive jitter buffer has overrun for this line. | |
| Underruns | unsignedInt | – | Defines the total number of times the receive jitter buffer has underrun for this line. | |
| IncomingCallsReceived | unsignedInt | – | Defines the total incoming calls received. | |
| IncomingCallsAnswered | unsignedInt | – | Defines the total incoming calls answered by the local user. | |
| IncomingCallsConnected | unsignedInt | – | Defines the total incoming calls that successfully completed call setup signaling. | |
| IncomingCallsFailed | unsignedInt | – | Defines the total incoming calls that failed to successfully complete call setup signaling. | |
| OutgoingCallsAttempted | unsignedInt | – | Defines the total outgoing calls attempted. | |
| OutgoingCallsAnswered | unsignedInt | – | Defines the total outgoing calls answered by the called party. | |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| OutgoingCallsConnected | unsignedInt | – | Defines the total outgoing calls that successfully completed call setup signaling. | |
| OutgoingCallsFailed | unsignedInt | – | Defines the total outgoing calls that failed to successfully complete call setup signaling. | |
| CallsDropped | unsignedInt | – | Defines the total number calls that were successfully connected (incoming or outgoing), but dropped unexpectedly while in progress without explicit user termination. | |
| TotalCallTime | unsignedInt | – | Defines the cumulative call duration in seconds. | |

### 2.2.3.1 Endpoint Profile Extensions

The following parts of TR-104 are not part of Endpoint profile (Endpoint:1) however are applicable to MSBR due to one of the following:

- are implied from the way we implement VoiceService.1.Capabilities object.
- were explicitly requested by potential customer

**Table 2-4: Endpoint Profile Extensions Table**

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| **.VoiceService.{i}.VoiceProfile.{i}.** | object | – | | |
| FaxPassThrough | string | W | Specifies the behavior of the CPE for pass-through of fax data. Enumeration of:<br>"Disable"<br>"Auto"<br>"Force"<br>The value "Disable" prevents the CPE from switching to a fax pass-through mode.<br>The value "Auto" allows the CPE to automatically detect fax data to determine whether or not to switch to a fax pass-through mode.<br>The value "Force" forces the CPE to switch to a fax pass-through mode regardless of whether fax signaling is detected.<br>If this parameter is supported, the capability VoiceService.{i}.Capabilities.FaxPassThrough MUST be equal to true.<br>This parameter is appropriate only for a VoIP endpoint. | According to IsFaxUsed and FaxTransportMode parameters:<br>**Read:**<br>(IsFaxUsed==2) \|\| ((IsFaxUsed==0) && (FaxTransportMode==2)) →"Auto"<br>Otherwise → "Disabled"<br>**Write:**<br>"Auto" → IsFaxUsed=2<br>"Disable"→ IsFaxUsed=0, FaxTransportMode=0 (unless FaxT38.Enable == true)<br>**Note**: the same values may be taken from "default" IPProfile 0. |
| ModemPassThrough | string | W | Specifies the behavior of the CPE for passthrough of modem data. Enumeration of:<br>"Disable"<br>"Auto"<br>"Force"<br>The value "Disable" prevents the CPE from switching to a modem pass-through mode.<br>The value "Auto" allows the CPE to automatically detect modem data to determine whether or not to switch to a modem pass-through mode.<br>The value "Force" forces the CPE to switch to a modem pass-through mode regardless of whether modem signaling is detected.<br>If this parameter is supported, the capability VoiceService.{i}.Capabilities.ModemPassThrough MUST be equal to true.<br>This parameter is appropriate only for a VoIP endpoint. | According to V21ModemTransportType, V22ModemTransportType, V23ModemTransportType, V32ModemTransportType, V34ModemTransportType and BellModemTransportType parameters:<br>**Read:**<br>If all VxxModemTransportType == 2 → "Auto"<br>otherwise → "False"<br>**Write:**<br>"Auto" → all VxxModemTransportType = 2<br>"Disable" → all VxxModemTransportType = 0 |
| **.VoiceService.{i}.VoiceProfile.{i}.RTP.RTCP.** | object | – | Voice profile parameters related to RTCP.<br>If this object is supported, the | Neither of the below parameters are configurable via |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| | | | capability VoiceService.{i}.Capabilities.RTCP MUST be equal to true.<br>Applicable only for a VoIP endpoint. | WEB/CLI/INI file. |
| Enable | boolean | W | Enables or disables RTCP. | Internal parameter: RtcpInterval==0 |
| TxRepeatInterval | unsignedInt [1:] | W | Defines the Transmission repeat interval, in milliseconds | Internal parameter: RtcpInterval |
| **VoiceService.{i}.VoiceProfile.{i}.RTP.SRTP.** | object | – | Defines the Voice profile parameters for secure voice transmission via SRTP.<br>If this object is supported, the capability VoiceService.{i}.Capabilities.SRTP MUST be equal to true.<br>Applicable only for a VoIP endpoint. | |
| Enable | boolean | W | Enables or disables the use of SRTP.<br>If RTCP is enabled, a true value of this parameter also implies the use of SRTCP. | EnableMediaSecurity |
| KeyingMethods | string(256) | W | Defines a comma-separated list of keying methods that may be used. The value MUST be a subset of those listed in the parameter VoiceService.{i}.Capabilities.SRTP KeyingMethods. | “Static”<br>Write operation will not be supported. |
| EncryptionKeySizes | string(256) | W | Defines a comma-separated list of encryption key sizes that may be used. The value MUST be a subset of those listed in the parameter VoiceService.{i}.Capabilities.SRTP EncryptionKeySizes. | According to SRTPofferedSuites parameter. But since we support only “128” – this is the only possible value. |
| **VoiceService.{i}.VoiceProfile.{i}.RTP.Redundancy** | object | – | Defines Voice profile parameters for RTP payload redundancy as defined by RFC 2198.<br>If this object is supported, the capability VoiceService.{i}.Capabilities.Redundancy MUST be equal to true.<br>Applicable only for a VoIP endpoint. | |
| Enable | boolean | W | Enables or disables the use of RTP payload redundancy as defined by RFC 2198. | RTPRedundancyDepth |
| PayloadType | unsignedInt [0:127] | W | Defines the Payload Type of RTP packet using RFC 2198. Values should be within the range of dynamic Payload Types (96-127). | RFC2198PayloadType |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| FaxAndModemRedundancy | int[-1:5] | W | Specifies the redundancy number for fax and modem pass-through data transmissions.<br>A non-negative value indicates that RFC 2198 is to be used for fax and modem pass-through data. The value indicates the number of redundant copies to be transmitted (the total number transmitted is one plus this value).<br>A value of -1 indicates RFC 2198 is not to be used for fax and modem pass-through data. If the optional parameter ModemRedudancy is present, then FaxAndModemRedudancy applies only to fax transmissions, but not to modem transmissions. | FaxRelayRedundancyDepth |
| **VoiceService.{i}.VoiceProfile.{i}.FaxT38** | object | – | Defines T.38 Fax information for devices that support T.38 relay.<br>If this object is supported, the capability VoiceService.{i}.Capabilities.FaxT38 MUST be equal to true.<br>Applicable only to a VoIP endpoint. | |
| Enable | boolean | W | Enables or disables the use of T.38. | According to IsFaxUsed and FaxTransportMode parameters:<br>**Read**:<br>((IsFaxUsed==1) \|\| (IsFaxUsed==3) \|\| ((IsFaxUsed==0) && (FaxTransportMode==1)) → true<br>otherwise → false<br>**Write:**<br>true → IsFaxUsed=1<br>false → IsFaxUsed=0, FaxTransportMode=0 (unless FaxPassThrough == “Auto”)<br>**Note:** The same values may be taken from “default” IPProfile 0. |
| BitRate | unsignedInt | W | Defines the maximum data rate for fax. Enumeration of the following values:<br>▪ 2400<br>▪ 4800<br>▪ 7200<br>▪ 9600<br>▪ 12000<br>▪ 14400<br>▪ 33600 | mapped to FaxRelayMaxRate |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| HighSpeedPacketRate | unsignedInt | W | Defines the rate at which high speed data will be sent across the network, in milliseconds. Enumeration of the following values:<br>▪ 10<br>▪ 20<br>▪ 30<br>▪ 40 | Mapped to FaxModemBypassBasicRTPPacketInterval |
| HighSpeedRedundancy | unsignedInt [0:3] | W | Specifies the packet-level redundancy for high-speed data transmissions (i.e., T.4 image data). The value MUST be in the range 0 through 3. | FaxRelayRedundancyDepth |
| LowSpeedRedundancy | unsignedInt [0:5] | W | Specifies the packet-level redundancy for low-speed data transmissions (i.e., T.30 handshaking information). The value MUST be in the range 0 through 5. | FaxRelayEnhancedRedundancyDepth |
| **.VoiceService.{i}.VoiceProfile.{i}.Line.{i}.** | object | C | Defines support for creation and deletion of Lines is REQUIRED only if more than one Line is supported as indicated by VoiceService.{i}.Capabilities.MaxLineCount. | |
| DirectoryNumber | string(32) | W | Defines the directory number associated with this line. May be used to identify the line to the user.<br>In case of H.323 signaling, this MUST be an E.164 number. | TrunkGroup_FirstPhoneNumber<br>(from corresponding entry in Hun Group table) |
| **.VoiceService.{i}.VoiceProfile.{i}.Line.{i}.CallingFeatures.** | object | – | Defines Voice line parameters related to optional endpoint based calling features. | |
| CallerIDNameEnable | boolean | W | Enables or disables the transmission of caller ID information on outgoing calls. | This and the next parameters will be mapped to Caller Display Information table (CallerDisplayInfo)<br>CallerDisplayInfo_IsCidRestricted |
| CallerIDName | string(256) | W | Defines a string used to identify the caller. | CallerDisplayInfo_DisplayString |

### 2.2.4 SIP Endpoint Profile

MSBR complies with Version 1 of SIP Endpoint profile (SIPEndpoint:1) as defined in TR-104 section 4.3.

Table 2-5: SIP Endpoint Profile Table

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| **.VoiceService.{i}.Capabilities.SIP.** | object | – | Defines SIP-specific capabilities. Applicable only if SIP is among the list of supported protocols. | |
| Role | string | – | Defines the role of this VoIP CPE. Enumeration of:<br>“UserAgent”<br>“BackToBackUserAgents”<br>“OutboundProxy”<br>A single VoiceService instance MUST have only one role. If a device includes the capabilities for more than one role, each role MUST be represented as separate VoiceService instances. | Always set to “UserAgent” |
| Extensions | string(256) | – | Defines a comma-separated list of SIP extension methods supported. SIP extension methods MUST be in the form of the method name in upper case.<br>The list MAY include vendor-specific extensions, which MUST use the format defined in [3].<br>Examples:<br>▪ “REFER”<br>▪ “INFO”<br>▪ “X_EXAMPLE-COM_MyExt” | “REFER, INFO” SUBSCRIBE , etc.. |
| Transports | string(256) | – | Defines a comma-separated list of SIP transport protocols supported. Each entry is an enumeration of:<br>▪ “UDP”<br>▪ “TCP”<br>▪ “TLS”<br>▪ “SCTP”<br>The list MAY include vendor-specific transports, which MUST use the format defined in [3]. | According to SIPTransportType<br>UDP/TCP/TLS |
| URISchemes | string(256) | – | Defines a comma-separated list of URI schemes supported beyond the URI schemes required by the SIP specification. Each URI scheme is given by the URI prefix, without the colon separator. Example:<br>▪ “tel, fax” | “” (empty) |
| EventSubscription | boolean | – | Defines support for SIP event subscription. A true value indicates support for the VoiceService.{i}.VoiceProfile.{i}.SIP.EventSubscribe and VoiceService.{i}.VoiceProfile.{i}.Line.{i}.SIP.EventSubscribe.{i} objects. | False |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| ResponseMap | boolean | – | Defines support for SIP response map. A true value indicates support for the VoiceService.{i}.VoiceProfile.{i}.SIP.ResponseMap object.<br>This parameter is applicable only for a VoIP endpoint. | False |
| **.VoiceService.{i}.VoiceProfile.{i}.SIP.** | object | – | Defines Voice profile parameters that are specific to SIP user agents. | |
| ProxyServer | string(256) | W | Defines a Host name or IP address of the SIP proxy server.<br>All SIP signaling traffic MUST be sent to the host indicated by this parameter and the port indicated by the ProxyServerPort parameter unless OutboundProxy parameter is non-empty or a different route was discovered during normal operations SIP routing operation.<br>Regardless of which host the traffic gets sent to (the ProxyServer or the OutboundProxy), the value of this parameter MUST be used to derive the URI placed into the SIP Route header field of all requests originated by this end-point unless a different proxy host was discovered dynamically during normal SIP routing operations. | ProxyIp_IpAddress from the 1st entry in “default" Proxy Set (0); note that ProxyIp_IpAddress contains _both_ hostname and port – hence some manipulation is needed.<br>During “write” operation ensure that IsProxySet parameter is set to 1).<br>During “read” operation, if IsProxySet parameter is set to 0, return empty string. |
| ProxyServerPort | unsignedInt [0:65535] | W | Defines the Destination Port to be used in connecting to the SIP server. | ProxyIp_IpAddress from the 1st entry in “default" Proxy Set (0); note that ProxyIp_IpAddress contains _both_ hostname and port – hence some manipulation is needed. |
| ProxyServerTransport | string | W | Defines the Transport protocol to be used in connecting to the SIP server. Must be chosen from among the transports supported, as indicated by VoiceService.{i}.Capabilities.SIP.Transports.<br>Enumeration of:<br>▪ “UDP”<br>▪ “TCP”<br>▪ “TLS”<br>▪ “SCTP” | According to ProxyIp_TransportType from the 1st entry in “default" Proxy Set (0 |
| RegistrarServer | string(256) | W | Defines a Host name or IP address of the SIP registrar server.<br>If this parameter is empty, the CPE MUST obtain all of the registrar server configuration information, including host name or IP address, port, and transport protocol, from the corresponding ProxyServer parameters (ProxyServer, ProxyServerPort, and ProxyServerTransport), ignoring all of the registrar server parameters (RegistrarServer, RegistrarServerPort and RegistrarServerTransport). | RegistrarIP<br>**Note:** RegistrarIP may contain both hostbname and port – hence some manipulation is needed. |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| RegistrarServerPort | unsignedInt [0:65535] | W | Defines the Destination port to be used in connecting to the SIP registrar server.<br>If the RegistrarServer parameter is empty, the CPE MUST obtain all of the registrar server configuration information, including host name or IP address, port, and transport protocol, from the corresponding ProxyServer parameters (ProxyServer, ProxyServerPort, and ProxyServerTransport), ignoring all of the registrar server parameters (RegistrarServer, RegistrarServerPort and RegistrarServerTransport). | RegistrarIP<br>**Note:** RegistrarIP may contain both hostbname and port – hence some manipulation is needed |
| RegistrarServerTransport | string | W | Defines the Transport protocol to be used in connecting to the registrar server. Must be chosen from among the transports supported, as indicated by VoiceService.{i}.Capabilities.SIP.Transports.<br>Enumeration of:<br>“UDP”<br>“TCP”<br>“TLS”<br>“SCTP”<br>If the RegistrarServer parameter is empty, the CPE MUST obtain all of the registrar server configuration information, including host name or IP address, port, and transport protocol, from the corresponding ProxyServer parameters (ProxyServer, ProxyServerPort, and ProxyServerTransport), ignoring all of the registrar server parameters (RegistrarServer, RegistrarServerPort and RegistrarServerTransport). | According to RegistrarTransportType |
| UserAgentDomain | string(256) | W | Defines a CPE domain string. If empty, the CPE SHOULD use its IP address as the domain. | ProxyName |
| UserAgentPort | unsignedInt [0:65535] | W | Defines a port used for incoming call control signaling. | LocalSIPPort or TCPLocalSIPPort or TLSLocalSIPPort – according to SIPTransportType |
| UserAgentTransport | string | W | Defines the Transport protocol to be used for incoming call control signaling. Must be chosen from among the transports supported, as indicated by VoiceService.{i}.Capabilities.SIP.Transports.<br>Enumeration of:<br>▪ “UDP”<br>▪ “TCP”<br>▪ “TLS”<br>▪ “SCTP” | SIPTransportType |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| OutboundProxy | string(256) | W | Defines the Host name or IP address of the outbound proxy.<br>If a non-empty value is specified, the SIP endpoint MUST send all SIP traffic (requests and responses) to the host indicated by this parameter and the port indicated by the OutboundProxyPort parameter. This MUST be done regardless of the routes discovered using normal SIP operations, including use of Route headers initialized from Service-Route and Record-Route headers previously received. The OutboundProxy value is NOT used to generate the URI placed into the Route header of any requests. | Outbound Proxy will be mapped to the 1st entry in PREFIX table that (if exists) must look as follows:<br>[ PREFIX ]<br>FORMAT PREFIX_Index = PREFIX_DestinationPrefix, PREFIX_DestAddress, PREFIX_SourcePrefix, PREFIX_ProfileId, PREFIX_MeteringCode, PREFIX_DestPort, PREFIX_SrcIPGroupID, PREFIX_DestHostPrefix, PREFIX_DestIPGroupID, PREFIX_SrcHostPrefix, PREFIX_TransportType, PREFIX_SrcTrunkGroupID;<br>PREFIX 0 = *, 10.8.211.180, *, 0, 255, 0, -1, , -1, , -1, -1;<br>[ \PREFIX ] |
| OutboundProxyPort | unsignedInt [0:65535] | W | Defines the Destination port to be used in connecting to the outbound proxy. This parameter MUST be ignored unless the value of the OutboundProxy parameter in this object is non-empty. | PREFIX_DestPort (see above) |
| RegistrationPeriod | unsignedInt [1:] | W | Defines the Period over which the user agent must periodically register, in seconds. | RegistrationTime |
| RegisterExpires | unsignedInt [1:] | W | Defines the Register Request Expires header value, in seconds. | RegistrationTime (the same value as RegistrationPeriod) |
| **.VoiceService.{i}.VoiceProfile.{i}.Line.{i}.SIP.** | object | – | Defines the Voice line parameters that are specific to SIP call signaling. | |
| AuthUserName | string(128) | W | Defines the Username used to authenticate the connection to the server. | This and the next parameter will be mapped to the corresponding entry in Authentication table.<br>Authentication_UserId<br>On write, we will ensure that AuthenticationMode is set to 3 (PerFXS).<br>On read, if AuthenticationMode is not set to 3 (PerFXS), empty string will be returned. |

| Name | Type | Write | Description | Comment |
|---|---|---|---|---|
| AuthPassword | string(128) | W | Defines the Password used to authenticate the connection to the server.<br>When read, this parameter returns an empty string, regardless of the actual value. | Authentication_UserPassword<br>On write, we will ensure that AuthenticationMode is set to 3 (PerFXS).<br>On read, if AuthenticationMode is not set to 3 (PerFXS), empty string will be returned. |

# 3 Configuring TR-069

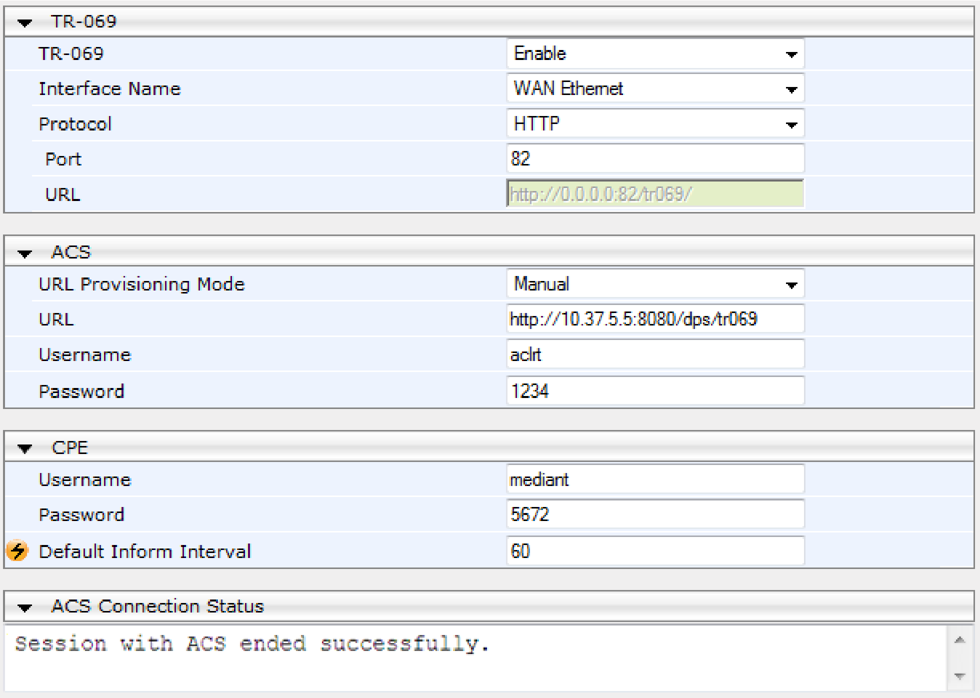
## 3.1 General Configuration

The CWMP/TR-069 Settings page is used to enable and configure TR-069.

➢ **To configure TR-069:**

1. Open the CWMP/TR-069 Settings page (**Configuration** tab > **System** menu > **Management** > **TR-069 (CWMP) Settings**).

**Figure 3-1: CWMP/TR-069 Settings Page**



TR-069
- TR-069: Enable
- Interface Name: WAN Ethernet
- Protocol: HTTP
- Port: 82
- URL: http://0.0.0.0:82/tr069/

ACS
- URL Provisioning Mode: Manual
- URL: http://10.37.5.5:8080/dps/tr069
- Username: aclrt
- Password: 1234

CPE
- Username: mediant
- Password: 5672
- Default Inform Interval: 60

ACS Connection Status

```
Session with ACS ended successfully.
```

2. Configure the parameters as required.
3. Click **Submit** to apply your changes.
4. Save the changes to flash memory

The TR-069 parameters are described in the table below.

**Table 3-1: TR-069 Parameters**

| Parameter | Description |
|---|---|
| Web: TR069<br>CLI: service<br>**[TR069ServiceEnable]** | Enables TR-069 management.<br>▪ **[0]** Disable (Default)<br>▪ **[1]** Enable<br>**Note:** For this parameter to take effect, a device reset is required. |
| Web: Interface Name<br>CLI: interface-type<br>**[TR069NetworkSource]** | Defines the device's network interface for the TR-069 connection.<br>▪ **[0]** LAN<br>▪ **[1]** WAN (default)<br>**Note:** The parameter is applicable only to the MSBR product line. |

| Parameter | Description |
|---|---|
| Web: Protocol<br>CLI: protocol<br>**[TR069Protocol]** | Defines the protocol used for the TR-069 connection.<br>▪ [0] HTTP (default)<br>▪ [1] HTTPS |
| Web: Port<br>CLI: port<br>**[TR069HTTPPort]** | Defines the local HTTP/S port used for TR-069.<br>The valid range is 0 to 65535. The default is 82.<br>**Note:** For this parameter to take effect, a device reset is required. |
| Web: URL<br>CLI: acl-url<br>**[TR069AcsUrl]** | Defines the URL address of the Auto Configuration Servers (ACS) to which the device connects. For example, http://10.4.2.1:10301/acs/.<br>By default, no URL is defined. |
| Web: Username<br>CLI: acs-user-name<br>**[TR069AcsUsername]** | Defines the login username that the device uses for authenticated access to the ACS.<br>The valid value is a string of up to 256 characters. By default, no username is defined. |
| Web: Password<br>CLI: acs-password<br>**[TR069AcsPassword]** | Defines the login password that the device uses for authenticated access to the ACS.<br>The valid value is a string of up to 256 characters. By default, no password is defined. |
| Web: URL Provisioning Mode<br>CLI: acs-url-provisioning-mode<br>[Tr069AcsUrlProvisioningMode] | Defines the method for configuring the URL of the TR-069 ACS.<br>▪ [0] Manual (default) = URL must be configured manually on the device. The URL is configured using the TR069ConnectionRequestUrl parameter.<br>▪ [1] Automatic = Device uses DHCP Option 43 to obtain URL address of ACS. |
| Web: URL<br>CLI: connection-request-url<br>**[TR069ConnectionRequestUrl** | Defines the URL for the ACS connection request. For example, http://10.31.4.115:82/tr069/. |
| Web: Username<br>CLI: connection-request-user-name<br>**[TR069ConnectionRequestUsername]** | Defines the connection request username used by the ACS to connect to the device.<br>The valid value is a string of up to 256 characters. By default, no username is defined. |
| Web: Password<br>CLI: connection-request-password<br>**[TR069ConnectionRequestPassword]** | Defines the connection request password used by the ACS to connect to the device.<br>The valid value is a string of up to 256 characters. By default, no password is defined. |
| Web: Default Inform Interval<br>CLI: inform-interval<br>**[TR069PeriodicInformInterval]** | Defines the inform interval (in seconds) at which the device periodically communicates with the ACS. Each time the device communicates with the ACS, the ACS sends a response indicating whether or not the ACS has an action to execute on the device.<br>The valid value is 0 to 4294967295. The default is 60. |
| **[TR069RetryinimumWaitInterval]** | Defines the minimum interval (in seconds) that the device waits before attempting again to communicate with the ACS after the previous communication attempt failure.<br>The valid value is 1 to 65535. The default is 5. |

| Parameter | Description |
|---|---|
| CLI: debug-mode<br>**[TR069DebugMode]** | Defines the debug mode level, which is the type of messages sent to the Syslog server.<br>The valid value is between 0 and 3, where 0 (default) means no debug messages are sent and 3 is all message types are sent. |

## 3.2 Remotely Triggering Device to Request TR-069 Connection with ACS

The device can be remotely triggered to request TR-069 connection with the Auto-Configuration Server (ACS). This is done using SIP NOTIFY messages. The device forwards the NOTIFY with a TR-069 Connection Request, received from the PBX, to the ACS. This feature is typically required when the ACS is unable to "reach" the device's management IP address (e.g., due to NAT) and thus, unable to send a "standard" TR-069 Connection Request. In such a scenario, the ACS requests the PBX to send a special SIP NOTIFY message to the device. Upon receipt, the device proceeds as if it received a "standard" TR-069 Connection Request. In other words, it establishes a TR-069 connection with the ACS and sends an Inform TR-069 RPC method containing the "6 CONNECTION REQUEST" event code.

The SIP NOTIFY message contains an Event header set to the following proprietary value to activate connection request for TR-069:

```
Event: cwmp-connect
```

An example of a NOTIFY message with this Event header value is shown below:

```
NOTIFY sip:<user>@<dsthost> SIP/2.0
To: sip:<user>@<dsthost>
From: sip:sipsak@<srchost>
CSeq: 10 NOTIFY
Call-ID: 1234@<srchost>
Event: cwmp-connect
```

To enable this feature, set the 'SIP Remote Reset' parameter (EnableSIPRemoteReset) to **Enable**.

**International Headquarters**
1 Hayarden Street,
Airport City
Lod 7019900, Israel
Tel: +972-3-976-4000
Fax: +972-3-976-4040

**AudioCodes Inc.**
27 World's Fair Drive,
Somerset, NJ 08873
Tel: +1-732-469-0880
Fax: +1-732-469-2298

**Contact us:** www.audiocodes.com/info
**Website:** www.audiocodes.com

Document #: LTRT-52333

AudioCodes