

Mediant™ 5000 and Mediant™ 8000 Command Line Interface (CLI) Reference Guide

Version 6.6

Document #: LTRT-89210



Table of Contents

1	CLI Overview	9
1.1	Introducing the CLI.....	9
1.2	Starting a CLI Management Session.....	10
1.3	Configuration and Navigation Concept	12
1.4	Command Editing and Auto-Completion.....	16
1.5	MO Aliases and Shortcuts	17
1.6	Command Aliases	17
1.7	Managing CLI Users	18
1.8	Activity Logging	18
1.9	Customizing the CLI Environment	19
1.10	Using Standard UNIX Utilities.....	19
1.10.1	Redirecting Command Output to File	19
1.10.2	Viewing the File	20
1.10.3	Viewing the Long CLI Command Output	20
1.10.4	Filtering Command Output.....	21
1.10.5	Editing the File.....	22
1.11	CLI Usage Tips.....	23
1.12	Parameters Documentation	23
2	Commands.....	25
2.1	Help Command	27
2.1.1	help.....	27
2.2	Navigation Commands.....	30
2.2.1	go.....	30
2.2.2	Up.....	31
2.2.3	moTree	32
2.3	Show Command	34
2.3.2	show	35
2.3.3	show conf.....	38
2.3.4	show param	40
2.3.5	show status.....	41
2.3.6	show pm	42
2.3.7	show alarms.....	43
2.3.8	show hw.....	46
2.3.9	show inventory.....	47
2.3.10	show ip.....	51
2.3.11	show calls	54
2.3.12	show mgcp.....	56
2.3.13	show megaco.....	58
2.3.14	show ams.....	60
2.3.15	show tdm	61
2.3.16	show dsp.....	62
2.3.17	show voiceprompt.....	63
2.3.18	show tones.....	64
2.3.19	show children.....	66
2.3.20	searchParam	67

2.4	Modify Commands	68
2.4.1	modify	68
2.4.2	lock	70
2.4.3	unlock	71
2.4.4	add	72
2.4.5	remove	73
2.4.6	addBoard	74
2.4.7	removeBoard	75
2.4.8	Action	76
2.5	Scripting Commands	77
2.5.1	execFromFile	77
2.5.2	waitParam	78
2.6	Debug Commands	79
2.6.1	tpPing	79
2.6.2	tpCmd	81
2.6.3	Log	83
2.7	Administration Commands	85
2.7.1	Tools	85
3	Performing Commonly Used Tasks	87
3.1	Checking the Status of the Media Gateway	87
3.2	Checking Definitions Related to the IP Network	87
3.3	Checking the Media Gateway Board's Configuration and Status	88
3.4	Adding and Removing the Media Gateway Board	88
3.5	Navigating through the MO Tree	88
4	Scripting Capabilities	89
4.1	Simple CLI Script	89
4.2	Variables	89
4.2.1	Special Symbols in String Variables	90
4.2.2	String Operations	90
4.2.3	Arithmetic Operations	90
4.2.4	Assigning Command Output to Variable	90
4.2.5	Parameters of the CLI Script	91
4.3	Test Conditions	91
4.4	Branches	92
4.5	Loops	93
4.6	Functions	93
4.7	Getting Input from User	93
4.8	Testing the Result of a CLI Command	94
4.9	Waiting for a CLI Command to Complete	94
4.10	Using CLI Commands from PERL Scripts	95
A	Appendix A – Supported MO Actions	97
B	Appendix B – Supported Media Gateway Board's Debug Command	103
B.1	General Commands	103
B.2	Call Detail Reports (CDR) Commands	104

List of Figures

Figure 1-1: Starting a CLI Management Session	10
Figure 1-2: Current MO Displayed in the CLI Command Prompt.....	15
Figure 1-3: Auto-Completion of the Command <i>show</i>	16

List of Tables

Table 2-1: CLI Commands and their Options.....	25
Table 4-1: MOs, Actions and Action Syntax.....	97
Table 4-2: General Commands	103
Table 4-3: Subcommands of Call Detail Reports (CDR) Command	104

Notice

This CLI Reference Guide describes the commands supported by the Mediant 5000/8000 Media Gateways and the IPmedia 8000 available from AudioCodes.

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 can be viewed by registered customers at <http://www.audiocodes.com/downloads>.

© 2012 AudioCodes Ltd. All rights reserved.
This document is subject to change without notice.

Date Published: October-14-2012



Note: This Reference Guide is relevant to the following products:

- Mediant 5000 Media Gateway
- Mediant 8000 Media Gateway

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 service are generally provided by AudioCodes' Distributors, Partners, and Resellers from whom the product was purchased. For technical support for products purchased directly from AudioCodes, or for customers subscribed to AudioCodes Customer Technical Support (ACTS), contact support@audiocodes.com.

Documentation Conventions

References to 'gateway' and 'media gateway' include the media server, even though it's not explicitly stated. Command examples and script examples are **Courier New**. Utilities, keys and files are in **bold Arial**.

Related Documentation

Manual Name
EMS Parameter Guide for the Mediant 5000, 8000
Mediant 5000 Product description
Mediant 8000 Product Description
Mediant 5000 Media Gateway Installation, Operation & Maintenance Manual
Mediant 8000 Media Gateway Installation, Operation & Maintenance Manual
Media Gateway Programmer's User Manual
Mediant 5000 8000 Release Notes
Element Management System (EMS) User's Manual
Element Management System (EMS) Server Installation & Maintenance Manual
Element Management System (EMS) Product Description
Element Management System (EMS) Release Notes
Element Management System (EMS) Online Help

1 CLI Overview

AudioCodes provides two interfaces for the operation, configuration, maintenance and support tasks of its carrier class media gateways and media servers:

- Command Line Interface (CLI)
- Element Management System (EMS) GUI

This Reference Guide documents the CLI for the Mediant 5000 Media Gateway, Mediant 8000 Media Gateway and IPmedia 8000 Media Server.

1.1 Introducing the CLI

The CLI is available via the following management interfaces:

- SSH (Secure Shell)
- RS-232 console

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
- Scripting capabilities
- Debugging

All CLI commands are fully documented and are provided with a versatile auto-completion system, activated by pressing the **Tab** key at any stage while typing the command.

In addition to gateway-specific commands, complete UNIX shell functionality and standard UNIX command-line utilities are available to users. These utilities can be combined with gateway-specific commands to accomplish complicated maintenance tasks.

1.2 Starting a CLI Management Session

➤ To start a CLI management session, take these steps:

1. Connect to the Media Gateway's CLI interface via SSH (Secure Shell) or RS-232 serial console. Use the global OAM IP address when connecting via the SSH protocol.



Note: Use of the Telnet protocol for connecting to the Media Gateway's CLI interface may be supported in certain non-secure Media Gateway configurations; however it is strongly discouraged due to the severe security flaws in the Telnet protocol.

2. Login as CLI user, e.g.:
 - Username: acladmin
 - Password: pass_1234



Note: The CLI username and password can be altered by the media gateway administrator. Multiple CLI users can be defined. Refer to Section 0, “

Managing CLI Users”, on page 18 for additional information.

3. Wait until the CLI prompt is displayed (refer to the figure below).

Figure 1-1: Starting a CLI Management Session

```

C:\WINNT\system32\cmd.exe - telnet 10.7.5.234

SunOS 5.9
login: cli
Password:
Last login: Wed Jan 18 10:17:48 from 10.13.2.46

+-----+
| Welcome to Media Gateway's CLI |
+-----+

Short help:
  go      - go to another MO
  show    - show status & configuration
  setParam - alter configuration

Type 'help' for more information.
client177 -- ~ -- /moGW
=> _
    
```



Note: A standard Windows™ XP Telnet client has limited support of VT-100 control characters. For a better CLI experience, it's advisable to use one of the following Telnet/SSH clients instead:

- TeraTerm v2.3 (<http://hp.vector.co.jp/authors/VA002416/teraterm.html>)
- TeraTerm Pro v4.x (<http://sourceforge.jp/projects/ttssh2>)
- PuTTY (<http://www.putty.nl>)

1.3 Configuration and Navigation Concept

Media gateway configuration is represented by a set of Managed Objects (MOs) which combine physical and logical entities. Each MO contains multiple attributes that represent different properties of a specific entity. Each attribute has access permissions (read-write or read-only) that indicate whether a specific property can be user-modified or not.

All MOs are grouped into an MO Tree that represents the hierarchical relationship between different MOs. moGW is located at the top of the MO Tree and represents a Gateway Entity.

The schema represents the MOs hierarchy (also available under the command `moTree`) (refer to Section 0,

moTree, on page 32):

```
moGW
----| moFile
----| moSubnet
----| moNetworkProfile
----|----| moNetworkSubnet
----|----| moTPStaticRoute
----| moSCStaticRoute
----| moMGC
----|----| moMGCPatternLevel
----|----| moCAAddress
----| moTpRG
----| moFirewallProfile
----|----| moFirewallRule
----| moSCIPSECRule
----|----| moSCIKEProposal
----| moTPIPSECProfile
----|----| moTPIPSECRule
----|----| moTPIKEProposal
----| moSS7Mtp3Group
----|----| moSS7Node
----|----|----| moSS7Linkset
----|----|----|----| moSS7LinksetLink
----|----|----| moSS7Routeset
----|----|----|----| moSS7Route
----|----|----| moSS7NodeRdcy
----|----|----| moSS7AliasPointCode
----|----| moSS7NodeTimers
----|----| moSS7LinksetTimers
----| moNfsProfile
----|----| moNfsServer
----| moAMRCodecPoliciesProfile
----|----| moAMRCodecPolicy
----| moNtpServer
----| moFanTray
----|----| moFan
----| moTPConfigurationBackdoor
----| moBoard
----|----| moBoardRealm
----|----| moFiberGroup
----|----| moTrunk
----|----| moDS3
----|----| moSS7DataLink
----|----| moMTP2Profile
----|----| moM3UARoutingGroup
----|----|----| moM3UARoutingContext
----|----| moUALGroup
```

```

----|----| moUALInterface
----|----| moV52LELink
----|----| moV52LEInterface
----|----| moSIPCoderGroup
----|----|----| moSIPCoderType
----|----| moSIPProfileSettingsIp
----|----| moSIPAccount
----|----| moSIPDNSTable
----|----| moSIPSRV2IP
----|----| moSIPMessagePolicy
----|----| moSIPIPGroup
----|----| moSIPSRD
----|----| moSIPProxySet
----|----|----| moSIPProxy
----|----| moSIPInterface
----|----| moSIPNATTranslation
----|----| moSIPRoutingRuleGroups
----|----| moSIPCostGroupTable
----|----|----| moSIPCostGroupTimeBands
----|----| moSIPTrunkGroup
----|----| moSIPTrunkGroupSettings
----|----| moSIPProfileSettingsTel
----|----| moSIPCauseMapISDN2SIP
----|----| moSIPCauseMapSIP2ISDN
----|----| moSIPRoutingTel2Ip
----|----| moSIPRoutingIp2Tel
----|----| moSIPRedundantRoutingTel2Ip
----|----| moSIPRedundantRoutingIp2Tel
----|----| moSIPManipulationDstIp2Tel
----|----| moSIPManipulationDstTel2Ip
----|----| moSIPManipulationSrcTel2Ip
----|----| moSIPManipulationSrcIp2Tel
----|----| moSIPCallingNameManipulationsIP2Tel
----|----| moSIPCallingNameManipulationsTel2IP
----|----| moSIPRedirectNumberManipulationIp2Tel
----|----| moSIPRedirectNumberManipulationTel2Ip
----|----| moSIPSBCClassification
----|----| moSIPSBCA AdmissionControl
----|----| moSIPRoutingIp2Ip
----|----| moSIPSBCA AlternativeRouteReasons
----|----| moSIPIPtoIPInboundManipulation
----|----| moSIPIPtoIPOutboundManipulation
----|----| moSIPConditionTable
----|----| moSIPASRegistrationManipulation
----|----| moVirtualGW
----|----| moDSPTemplates
----|----| moTPDebugRecRule
----|----| moBoardIPAddress
    
```

```

----| moSwitchBoard
----|----| moSwitchBoardPort
----|----| moESMirrorFilter
----| moESBoard
----|----| moESBoardPort
----|----| moMirrorFilter
----| moCPUBoard
----| moTimingModules
----|----| moLineReference
----| moSatHW
----|----| moExternalInterface
----| moAaaServerInfo
----| moVlanTag
----| moPMThreshold

```

Media Gateway configuration commands - e.g., **show** or **modify** - are typically performed on the 'current MO' (unless the MO name is explicitly specified). On initial login, moGW is considered to be the 'current MO'. Users can navigate across the MO Tree using the command **go** (refer to Section 2.2.1 on page 30) - the syntax mimics the syntax of command **cd** used for navigation across the file system.

'Current MO' is displayed as part of the CLI user's command prompt, together with SC hostname and status, active user name and current directory:

Figure 1-2: Current MO Displayed in the CLI Command Prompt

The diagram shows a CLI command prompt: `acladmin@sc1 (active) -- /data -- /moG`. Four callout boxes point to specific parts of the prompt:

- 'Active User' points to `acladmin`.
- 'SC Hostname and Status' points to `@sc1 (active)`.
- 'Current directory' points to `-- /data --`.
- 'Current MO' points to `/moG`.

The full MO name consists of 2 parts - MO Type and MO Index - delimited by a # sign. E.g., **moBoard#4** or **moSS7DataLink#1**. Certain MOs, that have only a single instance, miss the Index part - e.g., **moGW**.

MOs that represent the SC, ES and Media Gateway boards - moCPUBoard, moESBoard and moBoard - have indexes that correspond to the occupied slot number. E.g., **moCPUBoard#2** or **moBoard#10**.

1.4 Command Editing and Auto-Completion

The following keys and shortcuts can be used to type a CLI command:

- **Left/Right arrows** – position the cursor while editing the command
- **Ctrl-A** – position the cursor at the beginning of the command
- **Ctrl-E** – position the cursor at the end of the command

Previously entered commands are stored in a 'history list'. It is possible to view commands from the 'history list' and repeat and/or modify them:

- **Up/Down arrows** – navigate through the 'history list'
- **Esc-P** – find the command in the 'history list' that matches the first typed-in characters

Auto-completion is available for all CLI commands and greatly simplifies command typing. Start typing the command and then press the **Tab** key to see the list of all possible completions. Press **Tab** key more than once for advanced completion options or to cycle across possible completions.

Figure 1-3: Auto-Completion of the Command `show`

```

Telnet 10.7.13.91
  modify - alter configuration
Type 'help' for more information.
client238 -- ~ -- /moGW
=> show
moBoard          moFirewallProfile    moSS7Mtp3Group
moClock          moMGC                 moStaticRoute
moCPUBoard       moNfsProfile          moSwitchBoard
moFile           moOamSecurityProfile  moTpRG
-----
board            esboard              firewall             mtp3                oamsecurity         route
clock           file                 mgc                 nfs                  rdcy                 scboard
-----
-nd              -- show only the parameters whose value differs from default
-r              -- recursively show all MO's children
-----
children        -- show all children of current MO
hw              -- show hardware summary
ip              -- show summary of all IP addresses utilized by the media gateway
-----
conf            -- show configuration of current/specific MO
status          -- show status of current/specific MO
param           -- show value of specific parameter
pm              -- show performance measurements
  
```



Note: On most SSH/Telnet clients, arrow keys can be used as an alternative mechanism for cycling across the possible completions.

CLI commands and key words are case sensitive. However, auto-completion (**Tab** key) automatically converts the partially typed command into the correct case.

1.5 MO Aliases and Shortcuts

All MO Names start with **mo** characters and are typically relatively long. Therefore, typing an MO name, even with the help of auto-completion, may not be very convenient, e.g.:

```
client177 -- /data -- /moGW
=> go moBoard#6/moSS7DataLink#2_
client177 -- /data -- /moGW/moBoard#6/moSS7DataLink#2
```

It is possible to use a shorter form of an MO name when typing a command – ‘MO aliases’. The full list of MO aliases can be seen via the command `help aliases`. The auto-completion system fully supports MO aliases. However, ‘Current MO’ is always displayed in a ‘full’ format.

```
client177 -- /data -- /moGW
=> go board#6/link#2_
client177 -- /data -- /moGW/moBoard#6/moSS7DataLink#2
```

MO shortcuts are an alternative form of the MO name. They typically consist of only two characters and are not supported by the auto-completion system. Use the command `help aliases` to see a full list of supported MO shortcuts.

```
client177 -- /data -- /moGW
=> go br#6/ln#2_
client177 -- /data -- /moGW/moBoard#6/moSS7DataLink#2
```

1.6 Command Aliases

The most commonly used CLI commands have shorter *command aliases* that can be used for faster and more convenient typing.

The following aliases are supported:

Command Name	Alias
show	sh
modify	m

1.7 Managing CLI Users

The following types of users are supported on the Media Gateway CLI interface:

- **root** – administrative super-user. Used for the initial system installation, basic network configuration and CLI user administration.
- **CLI administrators** – a group of users who are allowed to perform configuration tasks on the Media Gateway. After SC software installation, a sample CLI administrator user is created – its username is **acladmin**, and its default password is **pass_1234**. Administrative super-user (**root**) may create new CLI administrator users and delete existing users (including the user named **acladmin**).
- **CLI monitors** – a group of users who are allowed to monitor the status of the Media Gateway, but not allowed to perform configuration tasks. Administrative super-user (**root**) may create new CLI monitor users and delete existing users.

Media Gateway supports several operation modes, where the CLI users database is created locally on the SC boards or is synchronized with a Centralized Authentication server (EMS or 3rd party). CLI user management procedures differ according to the selected operation mode. Refer to the chapter “Administering Media Gateway’s CLI Users” in the “Installation and Operation Manual” for detailed information.



Note: For improved security, it is strongly recommended to create separate **CLI administrator** or **CLI monitor** account for each person who operates the Media Server. Shared accounts (such as **acladmin** account) should be deleted.

1.8 Activity Logging

Activity of all CLI users is logged in the centralized activity log. The following data is recorded:

- CLI user's login
 - Time & date
 - IP that the user logs in from
- All CLI commands (including UNIX commands)
 - Time & date
 - User's name
 - Executed command (security-sensitive data is masked)
 - Execution status (success/failure)
- CLI user's logout

Use command `log activity` to view the current activity logs. Refer to Section 2.6.3 on page 83 for additional options.

1.9 Customizing the CLI Environment

CLI users may customize their working environment by defining custom aliases or adjusting ZSH settings to suit their requirements. This can be performed by creating file `.zshrc` in the user's home directory. User-specific settings are applied to each newly created CLI session and are preserved across software upgrades and re-installations.

```
client195 -- ~ -- /moGW
=> more ~/.zshrc

# disable inactivity auto-logout
TMOUT=0

# add some aliases
alias l="ls -aF"
alias bang="lock /moGW y"

client195 -- ~ -- /moGW
=>
```

Use command `edit .zshrc` to create and edit the `.zshrc` file. For detailed information on command `edit`, refer to Section 1.10.5 on page 22.

System administrators may deploy a “system-wide” customized CLI environment for all CLI users by creating and/or editing the file `/etc/zshrc.cli`. Root permissions are required for this task. The Custom configuration is not applied to `root` and `ems` users.

1.10 Using Standard UNIX Utilities

Use the ZSH shell's functionality and standard UNIX command-line utilities to complement the available gateway-specific CLI commands and perform complicated tasks.

The following sections describe most typically used commands and operations. Note that this description is incomplete and you should refer to the *main page* of each command for additional information (e.g., `man grep`).

1.10.1 Redirecting Command Output to File

Output of any CLI command may be redirected to file as follows:

```
client195 -- ~ -- /moGW
=> show moBoard#6 > board6.txt
```

The file is created in the current directory (the user's home directory, by default), unless the full pathname is specified. The created file may be viewed or processed via standard UNIX commands – e.g., `less`, `more` or `grep`.

The regular redirect command described above deletes the destination file, if the latter exists. To append command output to the destination file (e.g., to store in it multiple command's output), use the *append* redirect command:

```
client195 -- ~ -- /moGW
=> show moBoard#6 >> board6.txt
```

1.10.2 Viewing the File

The following standard UNIX utilities can be used to view the text file:

- **cat** – shows the content of the file; the whole file is displayed at once
- **more** – shows the content of the file page by page; the user is prompted to press **SPACE** after each displayed page; pressing **q** – aborts the utility
- **less** – similar to **more** but it has a better search and position functionality; press **h** to view the detailed help on available commands
- **tail** – shows last X lines of the file – e.g., **tail -10 board6.txt** shows the last 10 lines of the file **board6.txt**
- **head** – shows first X lines of the file – e.g., **head -10 board6.txt** shows the first 10 lines of the file **board6.txt**

1.10.3 Viewing the Long CLI Command Output

Some CLI commands produce very long output that may be hard to see on the standard terminal. The *redirect output* ZSH shell functionality can be used to redirect output of such commands to one of the standard UNIX file viewing utilities, e.g.,:

```
client195 -- ~ -- /moGW  
=> show moBoard#6 | more
```

The output of command **show moBoard#6** is redirected to the utility **more** and displayed page by page.

1.10.4 Filtering Command Output

The output of CLI commands can be filtered – e.g., certain lines can be searched for or omitted. Standard UNIX `grep` or `egrep` utilities are typically used for performing these tasks.

For example, to see only those lines that contain the word 'IPAddress' in the `show moBoard#6` command output, type the following command (note that *command output redirection* functionality, similar to that described in the previous section, is used):

```
client195 -- ~ -- /moGW
=> show moBoard#6 | grep NoOp
#modify /moGW/moBoard#8 NoOpMode(rw,instant,none)=Disable(0)
#modify /moGW/moBoard#8 NoOpIntervalmsec(rw,instant,none)=10000
#modify /moGW/moBoard#8 NoOpPayloadType(rw,instant,none)=120
```

To perform a case-insensitive search, use the flag `-i`:

```
client195 -- ~ -- /moGW
=> show moBoard#6 | grep -i noop
#modify /moGW/moBoard#8 NoOpMode(rw,instant,none)=Disable(0)
#modify /moGW/moBoard#8 NoOpIntervalmsec(rw,instant,none)=10000
#modify /moGW/moBoard#8 NoOpPayloadType(rw,instant,none)=120
```

Finally, to see lines that contain either word SCTP or SRTP, you can use command `egrep`:

```
client195 -- ~ -- /moGW
=> show moBoard#6 | egrep "(SCTP|SRTP)"
#modify /moGW/moBoard#6 SCTPChecksumMethod(rw,online)=Adler32(0)
#modify /moGW/moBoard#6 SRTPMediaSecurity(rw,online)=Disable(0)
#modify /moGW/moBoard#8 UnencryptedSRTPOffer(rw,instant,all)=Disable(0)
```

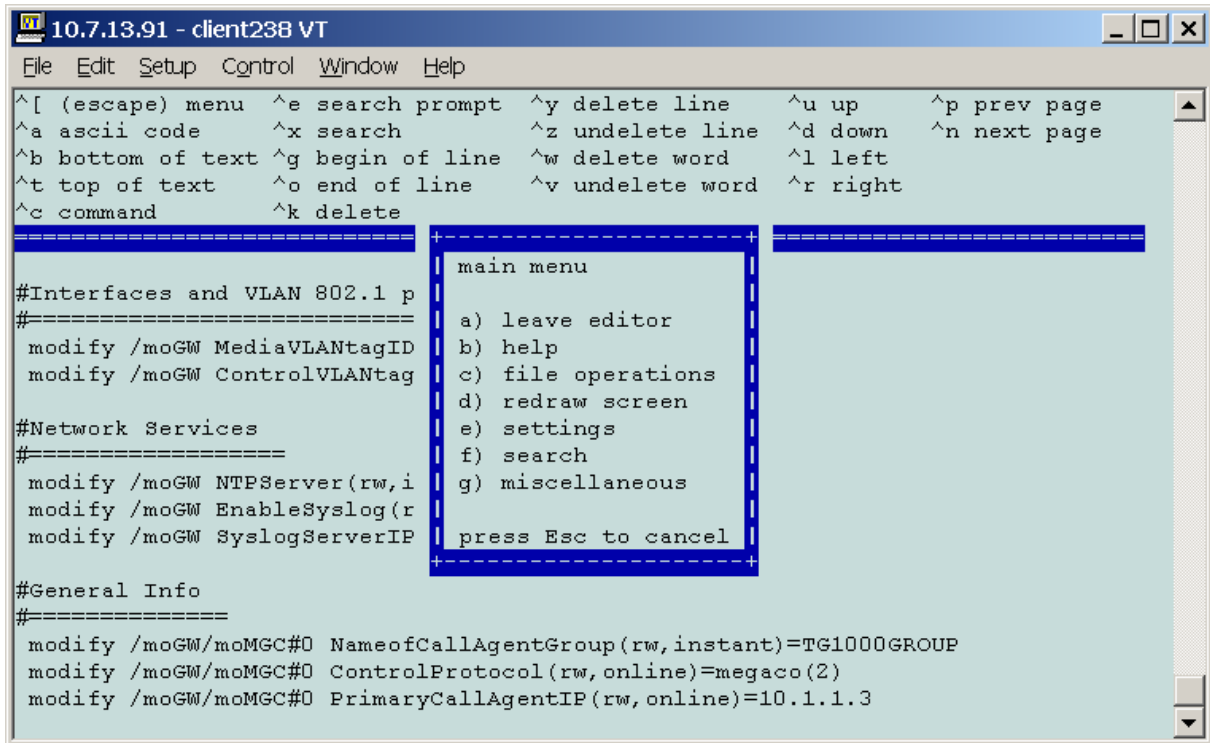
Other UNIX utilities – e.g., `sed` and `awk` – can be used to perform advanced command filtering. Description of their use is outside the scope of this brief overview. Refer to the relevant *man pages* or publicly available tutorials on the Internet.

1.10.5 Editing the File

Use the following command to edit the text file (e.g., CLI script):

```
edit <filename>
```

You're presented with a simple menu-driven textual editor:



The uppermost section of the screen contains a brief list of all available commands. The symbol ^ corresponds to the Ctrl key. The screen is automatically updated to match the current context.

When you finish editing the file, press ESC to get to the Main Menu and choose the option "a) leave editor". You are prompted to save the file if its content was changed.

Note that other standard UNIX editors – e.g., `vi` or `ed` – can be used to edit the files as well. However, their use may be less straightforward to new users.

1.11 CLI Usage Tips

Use the command `help` to view a detailed description and syntax of any CLI command, e.g.:

```
help show - shows syntax and options of the command show
help commands - shows a list of all available commands
help - shows the complete help for all available commands
```

Use the command `man` to view manual pages of standard UNIX utilities and commands, e.g.:

```
man grep - shows the manual page of grep utility
```

1.12 Parameters Documentation

Users can refer to the following related AudioCodes documentation:

Documentation
Mediant 5000 / 8000 Media Gateway Installation and Operation Manual
Mediant 5000 / 8000 Media Gateway Programmer's User Manual
EMS Parameter Guide for the Mediant 5000 and Mediant 8000 Gateways
EMS Online Help

Reader's Notes

2 Commands

Table 2-1 summarizes the CLI commands and their options.

Table 2-1: CLI Commands and their Options

Purpose	Command	Description
Help	help	Show the help for a specific MO, command, action or parameter.
Navigation	go	Go to another MO
	up	Go up one level.
	moTree	Show the complete MO tree.
Show	show	Show all parameters of the specific/current MO. Both configuration and status parameters are displayed.
	show conf	Show configuration parameters only
	show param	Show value of specific parameter
	show status	Show status summary
	show pm	Show performance monitoring statistics
	show alarms	Show notification traps (alarms and events)
	show hw	Show hardware summary
	show inventory	Show inventory
	show ip	Show IP configuration of VoP board or the media gateway
	show children	Show all possible children MOs with currently available instances.
	show calls	Show list of active calls on VoP board
	show mgcp	Show MGCP parameters of VoP board
	show megaco	Show MEGACO parameters of VoP board
	show ams	Show media server data of VoP board
	show tdm	Show PSTN-related statistics on VoP board
	show dsp	Show DSP statistics on VoP board
show voiceprompt	Show voice prompts table of VoP board	
show tones	Show information about special tones of VoP board	

Purpose	Command	Description
	searchParam	Search a parameter by its full or partial name.
	version	Show software version
Modify	modify	Modify a value of the specific parameter.
	lock	Lock MO
	unlock	Unlock MO
	add	Add dynamic MO
	remove	Remove dynamic MO
	addBoard	Add media gateway board
	removeBoard	Remove media gateway board
	action	Perform MO-specific action
Scripting	execFromFile	Execute commands from a script file.
	waitParam	Wait until the parameter gets a specific value.
Debug	tpPing	Send ICMP (Internet Control Message Protocol) echo request packets from a specific media gateway board to a defined IP address.
	tpCmd	Perform a command on the VoP board.
	log	View log file
Administration	tools	Perform media gateway administration tasks – stop/start the media gateway software and create a bug report file

The following conventions are used in the sections below to describe the syntax of CLI commands :

<param> - mandatory parameter

[<param>] - optional parameter

{ <param1> | <param2> | <param3> } - choice between one of the parameters

2.1 Help Command

2.1.1 help

2.1.1.1 Description

Shows online help

```
help
help commands
help <command>
help params [<Mo Name>] [<Param Name>]
help actions [<Mo Name>] [<Action Name>]
help aliases
```

2.1.1.2 Syntax Description

```
help
```

Shows the complete description of all available CLI commands.

```
help commands
```

Shows a brief list of all available CLI commands.

```
help <command>
```

Shows the description of a specific CLI command.

Syntax	Description
<command>	The name of the command on which help is required. Output shows the command purpose and syntax. Use help command to get a list of all available commands.

```
help params [<Mo Name>] [<Param Name>]
```

Lists all parameters of the current/specific MO. If <Param Name> is specified, the description of the specific parameter is shown.

Syntax	Description
<MO Name>	The name of the MO as it appears in the command moTree output. If not specified, the parameters of the current MO are displayed.
<Param Name>	Parameter name

```
help actions [<Mo Name>] [<Action Name>]
```

Lists all actions of current/specific MO. If <Action Name> is specified – shows description of a specific action.

Syntax	Description
<MO Name>	The name of the MO as it appears in the command moTree output. If not specified, the actions of the current MO are displayed.
<Action Name>	Action name

```
help aliases
```

Shows a list of supported MO aliases.

2.1.1.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.
5.0	Command syntax was modified.

2.1.1.4 Examples

```
client195 -- ~ -- /moGW
```

```
=> help go
go
--
Go to another MO
Syntax: go <Mo Path>
Hint: use "show children" command to discover all available MOs
```

```
client195 -- ~ -- /moGW
```

```
=> help help
help
----
Show online help
Syntax:
  help
  show description of all available CLI commands
  help commands
  brief list of all commands
  help <command>
  show description of specific command
```

```

help params [<Mo Name>] [<Param Name>]
    list all parameters of current/specific MO;
    if <Param Name> is specified - show description
    of specific parameter
help actions [<Mo Name>] [<Action Name>]
    list all actions of current/specific MO;
    if <Action Name> is specified - show description
    of specific action
help aliases
    list MO aliases and shortcuts

```

```
client195 -- ~ -- /moGW
```

```
=> help params moSS7Linkset
```

```

-----
moSS7Linkset
-----
    SS7LinkSetID (ro)
    SS7LinkSetName (rw)
    AdministrativeState (ro)
    OperationalState (ro)
    AlarmSeverityLevel (ro)
    ActionsID (rw)
    ActionsParameters (rw)
    ActionsResults (ro)
    ConfigurationStatus (ro)
    ActivityStatus (ro)
    DestinationPC (rw)
    LinkSetProfileID (rw)
    LinkMask (rw)
    AlternateLinkMask (rw)
    ReferenceCount (ro)
    ReadyStatus (ro)
    ReferenceDescription (ro)
    UserRequestAdminState (rw)

```

```
client195 -- ~ -- /moGW
```

```
=> help params moMGC ControlProtocol
```

```

-----
ControlProtocolType
-----

```

Use this parameter to determine the Control Protocol Type to be used.

```

ENUM
    0 -- none
    1 -- mgcp
    2 -- megaco
rw, online

```

```
client195 -- ~ -- /moGW
```

```
=> help actions moBoard
```

```

-----
moBoard actions
-----
    makeBoardRedundant
    makeBoardNonRedundant
    switchback
    switchover
    updateFk
    resetPm

```

```

client195 -- ~ -- /moGW
=> help actions moBoard updateFk
  action updateFk
  -----
  Update board's feature key
  Syntax: action [<moBoard Path>] updateFk <Feature Key>
    
```

2.2 Navigation Commands

2.2.1 go

2.2.1.1 Description

Navigates to another MO

```
go <MO Path>
```

2.2.1.2 Syntax Description

Syntax	Description
MO Path	The full or relative 'path' of the destination MO in the moTree. Path syntax is similar to the one used for navigation across a UNIX file system – e.g., moBoard#6, /moGW/moBoard#6 or ../moBoard#6/moTrunk#7. When the command is successfully executed, the CLI prompt is updated with the current MO's path.

2.2.1.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.
5.8	"go /" command should be used to go to the top of the MO Tree (moGW) instead of the obsolete "top" command.

2.2.1.4 Usage Guidelines

Use the command **show children** to display a list of all possible MO instances. To obtain the complete hierarchical list of MO types, run command **moTree**.

2.2.1.5 Examples

```

client195 -- ~ -- /moGW
=> go moBoard#6

client195 -- ~ -- /moGW/moBoard#6
=> show children
Available children Mo's for moBoard:
  moATMGW - No instances.
  moATMPort - Instances: 1, 2, 3
  moAAL2PVC - No instances.
  moALCAPProfile - Instances: 1
  moALCAPInstance - No instances.
    
```

```

moSAALProfile - Instances: 1
moMTP2Profile - Instances: 1
moSS7DataLink - No instances.
moUALGroup - No instances.
moUALInterface - No instances.
moFiberGroup - Instances: 1
moTrunk - Instances: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72,
73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84

```

```
client195 -- ~ -- /moGW/moBoard#6
```

```
=> go moBoard#5/moTrunk#4
```

```
client195 -- ~ -- /moGW/moBoard#6/moTrunk#4
```

```
=> go ../moTrunk#6
```

```
client195 -- ~ -- /moGW/moBoard#6/moTrunk#6
```

```
=>
```

2.2.2 Up

2.2.2.1 Description

Navigates to the upper level in the MO tree (to the 'parent' MO).

```
Up
```

2.2.2.2 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.

2.2.2.3 Examples

```
client195 -- ~ -- /moGW
```

```
=> go moBoard#6/moTrunk#4
```

```
client195 -- ~ -- /moGW/moBoard#6/moTrunk#4
```

```
=> up
```

```
client195 -- ~ -- /moGW/moBoard#6
```

```
=>
```

2.2.3 moTree

2.2.3.1 Description

Shows the complete MO tree.

Only MO types are shown. To view all MO instances that reside under the current/specific MO, use command **show children**.

```
moTree
```

2.2.3.2 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.qq

2.2.3.3 Examples

```
client195 -- ~ -- /moGW
moGW
----| moFile
----| moSubnet
----| moNetworkProfile
----|----| moNetworkSubnet
----|----| moTPStaticRoute
----| moSCStaticRoute
----| moMGC
----|----| moMGCPatternLevel
----|----| moCAAddress
----| moTpRG
----| moFirewallProfile
----|----| moFirewallRule
----| moSCIPSECRule
----|----| moSCIKEProposal
----| moTPIPSECProfile
----|----| moTPIPSECRule
----|----| moTPIKEProposal
----| moSS7Mtp3Group
----|----| moSS7Node
----|----|----| moSS7Linkset
----|----|----|----| moSS7LinksetLink
----|----|----| moSS7Routeset
----|----|----|----| moSS7Route
----|----|----| moSS7NodeRdcy
----|----|----| moSS7AliasPointCode
----|----| moSS7NodeTimers
----|----| moSS7LinksetTimers
----| moNfsProfile
----|----| moNfsServer
----| moAMRCodecPoliciesProfile
----|----| moAMRCodecPolicy
----| moNtpServer
----| moFanTray
```



```

-----|-----| moFan
-----| moTPConfigurationBackdoor
-----| moBoard
-----|-----| moBoardRealm
-----|-----| moFiberGroup
-----|-----| moTrunk
-----|-----| moDS3
-----|-----| moSS7DataLink
-----|-----| moMTP2Profile
-----|-----| moM3UARoutingGroup
-----|-----|----| moM3UARoutingContext
-----|-----| moUALGroup
-----|-----| moUALInterface
-----|-----| moV52LELink
-----|-----| moV52LEInterface
-----|-----| moSIPCoderGroup
-----|-----|----| moSIPCoderType
-----|-----| moSIPProfileSettingsIp
-----|-----| moSIPAccount
-----|-----| moSIPDNSTable
-----|-----| moSIPSRV2IP
-----|-----| moSIPMessagePolicy
-----|-----| moSIPIPGroup
-----|-----| moSIPSRD
-----|-----| moSIPProxySet
-----|-----|----| moSIPProxy
-----|-----| moSIPInterface
-----|-----| moSIPNATTranslation
-----|-----| moSIPRoutingRuleGroups
-----|-----| moSIPCostGroupTable
-----|-----|----| moSIPCostGroupTimeBands
-----|-----| moSIPTrunkGroup
-----|-----| moSIPTrunkGroupSettings
-----|-----| moSIPProfileSettingsTel
-----|-----| moSIPCauseMapISDN2SIP
-----|-----| moSIPCauseMapSIP2ISDN
-----|-----| moSIPRoutingTel2Ip
-----|-----| moSIPRoutingIp2Tel
-----|-----| moSIPRedundantRoutingTel2Ip
-----|-----| moSIPRedundantRoutingIp2Tel
-----|-----| moSIPManipulationDstIp2Tel
-----|-----| moSIPManipulationDstTel2Ip
-----|-----| moSIPManipulationSrcTel2Ip
-----|-----| moSIPManipulationSrcIp2Tel
-----|-----| moSIPCallingNameManipulationsIP2Tel
-----|-----| moSIPCallingNameManipulationsTel2IP
-----|-----| moSIPRedirectNumberManipulationIp2Tel
-----|-----| moSIPRedirectNumberManipulationTel2Ip
-----|-----| moSIPSBCClassification
-----|-----| moSIPSBCA AdmissionControl
-----|-----| moSIPRoutingIp2Ip
-----|-----| moSIPSBCA AlternativeRouteReasons
-----|-----| moSIPIPtoIPInboundManipulation
-----|-----| moSIPIPtoIPOutboundManipulation
-----|-----| moSIPConditionTable

```

```

----|----| moSIPASRegistrationManipulation
----|----| moVirtualGW
----|----| moDSPTemplates
----|----| moTPDebugRecRule
----|----| moBoardIPAddress
----|    | moSwitchBoard
----|----| moSwitchBoardPort
----|----| moESMirrorFilter
----|    | moESBoard
----|----| moESBoardPort
----|----| moMirrorFilter
----|    | moCPUBoard
----|    | moTimingModules
----|----| moLineReference
----|    | moSatHW
----|----| moExternalInterface
----|    | moAaaServerInfo
----|    | moVlanTag
----|    | moPMThreshold

```

```
client195 -- ~ -- /moGW
```

```
=>
```

2.3 Show Command

2.3.1.1 Description

Shows the configuration or status summary

```

show [<Mo Path>] [-nd] [-r]
show [<Mo Path>] conf [-nd] [-r]
show [<Mo Path>] param <Param Name> [-v]
show [<Mo Path>] status
show [<Mo Path>] pm [<Option>]
show alarms [-h] [-f] [<alarm#>]
show hw [<Option>]
show inventory
show ip
show [<moBoard Path>] calls {[<trunk#>] | detail
{<endpoint>|<call_id>} | dur}
show [<moBoard Path>] ip {conf [<if#>] | perf | route |
firewall | tls_version | cert}
show [<moBoard Path>] mgcp {conf | perf | ner | calls
[<trunk#>] | detail {<endpoint>|<call_id>} | dur | rsip | err |
cs }
show [<moBoard Path>] megaco {conf [<vgw#>] | perf | ner |
calls [<trunk#>] [<vgw#>] | detail {<endpoint>|<call_id>} | dur
| err | cs }
show [<moBoard Path>] ams {perf}
show [<moBoard Path>] tdm {status | perf | summary}
show [<moBoard Path>] dsp {status | perf}
show [<moBoard Path>] voiceprompt {numofentries | entries
<start#> [<num>]}
show [<moBoard Path>] tones {cpt | udt | rngt | prt} [<tone#>]
show children

```

2.3.1.2 Alias

The alias `sh` can be used for faster typing.

2.3.2 show

2.3.2.1 Description

Shows all parameters of the current/specific MO.

Configurable (writeable) parameters are prefixed with the word ***modify*** to simplify conversion of the output into the executable script. All lines are prefixed with a comment sign (#) for safety reasons.

```
show [<Mo Path>] [-nd] [-r]
```

2.3.2.2 Syntax Description

Syntax	Description
MO Path	The full or relative 'path' of the MO that will be shown. If unspecified, the parameters of the current MO are displayed.
-nd	Show only the parameters whose value differs from the default.
-r	Recursively show parameters of the specified/current MO and all children MOs. Note: the output of option '-r' can be long so it is recommended to combine options '-r' and '-nd' together.

2.3.2.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.
5.0	Command syntax was modified.

2.3.2.4 Examples

```

client195 -- ~ -- /moGW
=> show moMGC#1

# Parameters for /moGW/moMGC#1:
# =====
# MGC Parameters Provisioning
# =====

#General Info
#=====
#      /moGW/moMGC#1 NumberOfCallAgentGroup(ro,online)=1
#      /moGW/moMGC#1 NumAgentsinGroup(ro,online)=0
#      /moGW/moMGC#1 ActiveCallAgent(ro,online)=0
#      /moGW/moMGC#1 NumTrunksunderGroup(ro,online)=0
#modify /moGW/moMGC#1 NameofCallAgentGroup(rw,instant)=MGC#
#modify /moGW/moMGC#1 ControlProtocol(rw,online)=none(0)
#modify /moGW/moMGC#1 UDPPort(rw,online)=2944
#      /moGW/moMGC#1 ReferenceDescription(ro,offline)=
#modify /moGW/moMGC#1 MegacoCheckMGCLegality(rw,instant)=Enable(1)
#modify /moGW/moMGC#1 MGCPVersion(rw,online)="MGCP 1.0"
#      /moGW/moMGC#1 AdministrativeState(ro,online)=Locked(0)
#      /moGW/moMGC#1 OperativeState(ro,online)=Enabled(1)
#modify /moGW/moMGC#1 UserRequestAdminState(rw,instant)=Locked(0)
#      /moGW/moMGC#1 UsedUnusedinCAGList(ro,online)=unused(0)

#MGC Addressing
#=====
#modify /moGW/moMGC#1 AddressingType(rw,online)=IPAddress(0)
#modify /moGW/moMGC#1 FQDN(rw,online)=none
#modify /moGW/moMGC#1 MGCP-RedundantCAFQDN(rw,online)=
#modify /moGW/moMGC#1 PrimaryCallAgentIP(rw,online)=0.0.0.0
#modify /moGW/moMGC#1 PrimaryAgentSecurityProfile(rw,online)=0
    
```

```
#modify /moGW/moMGC#1 1stRedundantAgentIP(rw,online)=0.0.0.0
#modify /moGW/moMGC#1 1stRedundantAgentSecurityProfile(rw,online)=0
#modify /moGW/moMGC#1 2ndRedundantAgentIP(rw,online)=0.0.0.0
#modify /moGW/moMGC#1 2ndRedundantAgentSecurityProfile(rw,online)=0
#modify /moGW/moMGC#1 3rdRedundantAgentIP(rw,online)=0.0.0.0
#modify /moGW/moMGC#1 3rdRedundantAgentSecurityProfile(rw,online)=0
#modify /moGW/moMGC#1 4thRedundantAgentIP(rw,online)=0.0.0.0
#modify /moGW/moMGC#1 4thRedundantAgentSecurityProfile(rw,online)=0
#modify /moGW/moMGC#1 5thRedundantAgentIP(rw,online)=0.0.0.0
#modify /moGW/moMGC#1 5thRedundantAgentSecurityProfile(rw,online)=0
#modify /moGW/moMGC#1 6thRedundantAgentIP(rw,online)=0.0.0.0
#modify /moGW/moMGC#1 6thRedundantAgentSecurityProfile(rw,online)=0
#modify /moGW/moMGC#1 7thRedundantAgentIP(rw,online)=0.0.0.0
#modify /moGW/moMGC#1 7thRedundantAgentSecurityProfile(rw,online)=0
```

#External Endpoints

```
#=====
```

```
#modify /moGW/moMGC#1 1stEndpoint(rw,online)=ds
#modify /moGW/moMGC#1 2ndEndpoint(rw,online)=
#modify /moGW/moMGC#1 3rdEndpoint(rw,online)=e1-
#modify /moGW/moMGC#1 4thEndpoint(rw,online)=
#modify /moGW/moMGC#1 PhysTermNamePattern(rw,online)=
#modify /moGW/moMGC#1 PhysTermOffset1stLevel(rw,online)=1
#modify /moGW/moMGC#1 PhysTermOffset2ndLevel(rw,online)=1
#modify /moGW/moMGC#1 RTPTerminationConvention(rw,online)=
#modify /moGW/moMGC#1 RTPTerminationsOffset(rw,online)=0
```

#IP Media Settings

```
#=====
```

```
#modify /moGW/moMGC#1 MegacoAMSPackage(rw,online)=2
#modify /moGW/moMGC#1
MegacoAnnouncementTerminationConvention(rw,online)=audio/audiopool0/
*
#modify /moGW/moMGC#1
MegacoBCTTerminationConvention(rw,online)=bct/bctpool0/*
#modify /moGW/moMGC#1
MegacoConfTerminationConvention(rw,online)=conf/confpool0/*
#modify /moGW/moMGC#1 MegacoTrunk-
TestTerminationConvention(rw,online)=tlt/tltpool0/*
```

```
client195 -- ~ -- /moGW
```

```
=>
```

2.3.3 show conf

2.3.3.1 Description

Shows all the configuration parameters of the current/specific MO.

Parameters are prefixed with the word **modify** to simplify conversion of the output into the executable script. All lines are prefixed with a comment sign (#) for safety reasons, except for the command **show conf -r** at the **moGW** level. Parameter access permissions (read-only or read-write) and provisioning type (instant, online or offline) are also displayed.

```
show [<Mo Path>] conf [-nd] [-r]
```

2.3.3.2 Syntax Description

Syntax	Description
MO Path	The full or relative 'path' of the MO that will be shown. If unspecified, the parameters of the current MO are displayed.
-nd	Show only the parameters whose value differs from the default.
-r	Recursively show parameters of the specified/current MO and all children MOs. Note: the output of option '-r' can be long so it is recommended to combine options '-r' and '-nd' together.

2.3.3.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.
5.0	Command syntax was modified.

2.3.3.4 Examples

```
client195 -- ~ -- /moGW
=> show moBoard#8 conf -nd

# Parameters for /moGW/moBoard#8:
#add tp1610 8 general

# =====
# Board1610 Parameters Provisioning
# =====

#General Settings
#=====
#modify /moGW/moBoard#8 IPAddress1(rw,online)=10.7.9.232
#modify /moGW/moBoard#8 IPAddress2(rw,online)=10.7.9.233

#Board Debug Tools
#=====
#modify /moGW/moBoard#8 WebConfigurartion(rw,online)=Enabled(0)
```

```
client195 -- ~ -- /moGW  
=>
```

2.3.4 show param

2.3.4.1 Description

Shows value of specific parameter of current/specific MO.

```
show [<Mo Path>] param <Param Name> [-v]
```

2.3.4.2 Syntax Description

Syntax	Description
MO Path	The full or relative 'path' of the MO that will be shown. If unspecified, the parameters of the current MO are displayed.
Param Name	The name of the parameter to be shown. Use command help params to view all available parameter names.
-v	Optional flag that limits output to the parameter value only. Typically used in CLI scripts when a specific parameter value must be tested.

2.3.4.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.
5.0	Command syntax was modified.

2.3.4.4 Examples

```
client195 -- ~ -- /moGW
=> show moBoard#8 param WebAccess
WebAccess=Enabled(0)
```

```
client195 -- ~ -- /moGW
=>
```


2.3.5 show status

2.3.5.1 Description

Shows status summary of the current/specific MO.

Index wildcard (*) can be used to show the status of multiple MOs of the same type.

```
show [<Mo Path>] status
```

2.3.5.2 Syntax Description

Syntax	Description
MO Path	The full or relative 'path' of the MO that will be shown. If unspecified, parameters of the current MO are displayed.

2.3.5.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.
5.0	Command syntax was modified.

2.3.5.4 Examples

```
client195 -- ~ -- /moGW
=> show moMGC#1 status

Status of /moGW/moMGC#1
=====
Number of Call Agent Group: 1
Name of Call Agent Group: MGC#
Control Protocol: none(0)
Primary Call Agent IP: 0.0.0.0
FQDN: none
Administrative State: Locked(0)
Operative State: Enabled(1)
Reference Description:
Used/Unused in CAG List: unused(0)

client195 -- ~ -- /moGW
=> show moBoard#8/moTrunk#* status

Status Table for moTrunk under /moGW/moBoard#8
# | Name          | Protocol          | Admin State | Op State | DS1 Path |
-----|-----|-----|-----|-----|-----|
1 | Trunk#1      | E1Transparent30  | Locked      | Disabled | none     |
2 | Trunk#2      | E1Transparent30  | Locked      | Disabled | none     |
3 | Trunk#3      | E1Transparent30  | Locked      | Disabled | none     |
4 | Trunk#4      | E1Transparent30  | Locked      | Disabled | none     |
5 | Trunk#5      | E1Transparent30  | Locked      | Disabled | none     |
6 | Trunk#6      | E1Transparent30  | Locked      | Disabled | none     |
7 | Trunk#7      | E1Transparent30  | Locked      | Disabled | none     |
```

8	Trunk#8	E1Transparent30	Locked	Disabled	none	
9	Trunk#9	E1Transparent30	Locked	Disabled	none	
10	Trunk#10	E1Transparent30	Locked	Disabled	none	
11	Trunk#11	E1Transparent30	Locked	Disabled	none	
12	Trunk#12	E1Transparent30	Locked	Disabled	none	
13	Trunk#13	E1Transparent30	Locked	Disabled	none	
14	Trunk#14	E1Transparent30	Locked	Disabled	none	
15	Trunk#15	E1Transparent30	Locked	Disabled	none	
16	Trunk#16	E1Transparent30	Locked	Disabled	none	

```
client195 -- ~ -- /moGW
=>
```

2.3.6 show pm

2.3.6.1 Description

Shows performance measurement (PM) data for the current/specific MO.

```
show [<Mo Path>] pm [<Option>]
```

2.3.6.2 Syntax Description

Syntax	Description
MO Path	The full or relative 'path' of the MO that will be shown. If unspecified, parameters of the current MO are displayed.
Option	The following options are supported for moBoard context: <ul style="list-style-type: none"> • calls – show call-related PMs • conf – show conference PMs • dsp – show DSP PMs • ip – show IP-related PMs • ivr – show voice recording PMs • megaco – show PMs of MEGACO call control protocol • mgcp – show PMs of MGCP call control protocol • trunks – show trunks PMs • control – show call control PMs • security – show security PMs

2.3.6.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.
5.0	Command syntax was modified.
5.2	New options control and security were added.

2.3.6.4 Examples

```

client195 -- ~ -- /moGW
=> show moBoard#8 pm

TPM DSP
=====
DSP Utilization: TPM1=112 , TPM2=0

TPM Megaco Control
=====
### Bad SNMP column table was given for dynamic table.
Num of Active Contexts: TPM1=105 , TPM2=0
Context Duration: TPM1=13 , TPM2=0
Num of Command Received: TPM1=31920 , TPM2=0
Num of Command Transmit: TPM1=0 , TPM2=0
[H.248] Service Change Cmd - Disconnected: TPM1=0 , TPM2=0
[H.248] Service Change Cmd - Restart: TPM1=0 , TPM2=0
[H.248] Service Change Cmd - Forced: TPM1=0 , TPM2=0
[H.248] Service Change Cmd - Graceful: TPM1=0 , TPM2=0
[H.248] Service Change Cmd - FailOver: TPM1=0 , TPM2=0
[H.248] Service Change Cmd - Handoff: TPM1=0 , TPM2=0
Success [H.248] Add: TPM1=@SNMP_NULL@ , TPM2=@SNMP_NULL@
Success [H.248] Move: TPM1=@SNMP_NULL@ , TPM2=@SNMP_NULL@
Success [H.248] Modify: TPM1=@SNMP_NULL@ , TPM2=@SNMP_NULL@
Success [H.248] Subtract: TPM1=@SNMP_NULL@ , TPM2=@SNMP_NULL@
Success [H.248] Service Change: TPM1=@SNMP_NULL@ , TPM2=@SNMP_NULL@
Success [H.248] Audit-Value: TPM1=@SNMP_NULL@ , TPM2=@SNMP_NULL@
Success [H.248] Audit-Capabilities: TPM1=@SNMP_NULL@ , TPM2=@SNMP_NULL@
...

client195 -- ~ -- /moGW
=>

```

2.3.7 show alarms

2.3.7.1 Description

Shows notification traps (alarms and events).

```
show alarms [-h] [-f] [<alarm#>]
```

2.3.7.2 Syntax Description

Syntax	Description
<default>	Show active alarms.
-h	Show alarms history.
-f	Show full alarm information.
<alarm#>	Show specific alarm (either from active or history table – according to -h switch).

2.3.7.3 History

Version	Description
5.4	This command was introduced for the carrier class media gateways and media servers.

2.3.7.4 Examples

```

client195 -- ~ -- /moGW
=> show alarms

Active Alarms Summary
=====
# | Severity          | Date                | Source                |
-----|-----|-----|-----|
0 | minor             | Oct 31 09:52:42 2007 | ES#19                 |
1 | major             | Oct 31 09:52:42 2007 | ES#19                 |
2 | minor             | Oct 31 09:52:49 2007 | SAT/PS/Top-Left      |
3 | minor             | Oct 31 09:52:49 2007 | SAT/PS/Bottom-Left   |
4 | major             | Oct 31 11:27:43 2007 | Board#13              |
5 | major             | Oct 31 11:27:43 2007 | Board#13/PSTN FbrGrp#1 |
6 | indeterminate     | Oct 31 11:27:44 2007 | Board#13              |
7 | critical           | Nov 07 08:07:18 2007 | /tg                    |
8 | critical           | Nov 07 08:07:19 2007 | /tg                    |

Table Continued...
-----
Name                |
-----|-----
Admin State Change  |
Operative State Change |
Power Supply Alarm   |
Power Supply Alarm   |
Operative State Change |
Operative State Change |
Board Restart Alarm  |
LineSync Interface Alarm |
1.3.6.1.4.1.5003.9.1.2.2.0.52 |

client195 -- ~ -- /moGW
=> show alarms -f 3

Active Alarms Summary:
=====

***** Alarm *****
1) ActiveAlarmIndex: 3
2) Severity: minor
3) Date: Oct 31 09:52:49 2007
4) Source: SAT/PS/Bottom-Left
5) NotificationOID: 1.3.6.1.4.1.5003.9.1.2.2.0.13
6) Name: Power Supply Alarm
7) Description: Power Supply cassette 3 failure
8) AlarmUniqueID: 10
9) Type: equipmentAlarm
10) AdditionalInfo1: Current status of PS cassette: Not Exist. Status of PS
cassettes: Top-Left=Not Exist, Top-Right=OK, Bottom-Left=Not Exist, Bottom-Right=OK
11) AdditionalInfo2:
12) AdditionalInfo3: 10.7.19.100
13) ProbableCause: powerProblem
***** END *****
    
```

```
client195 -- ~ -- /moGW  
=>
```

2.3.8 show hw

2.3.8.1 Description

Shows media gateway hardware summary.

```
show hw [<Option>]
```

Syntax Description

Syntax	Description
Option	The following options are supported: <ul style="list-style-type: none"> • hw-info – show hardware status • mg-info – show media gateway information • slots-table – show slots table • tp-table – show Media Gateway boards table

2.3.8.2 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.
5.0	Command syntax was modified.

2.3.8.3 Examples

```
client195 -- ~ -- /moGW
=> show hw

MGW Information
=====
MG FQDN (sysName): none
Administrative State: Unlocked(2)
Operative State: Enabled(1)
MG Alarm Severity Level: clear(0)
Media Gateway Type: mediant8000(0)
Chassis Type: M8K(2)
Software Version: 5.0.31
System Up Duration String: 0 days 0h:36m:43s
System Enabled Time String: 08/23/2006 03:08:22
System Last Disable Reason: undetermined(0)

MGW HW Status
=====
Sat1 Admin State: Unlocked(2)
Sat1 Oper State: Enabled(1)
Sat1 Severity: clear(0)
Sat2 Admin State: Locked(0)
Sat2 Oper State: Enabled(1)
Sat2 Severity: clear(0)
Top Left Power Supply: Enabled(1)
Top Right Power Supply: Enabled(1)
Bottom Left Power Supply: Enabled(1)
Bottom Right Power Supply: Enabled(1)
Voltage +12: 12062
Voltage -12: -12184
```

```

Voltage +5: 4990
Voltage +3: 3416

Slots Table
=====

Slot# | Board Type | Admin State | Oper State | Severity | Activity | Up Time
| Enabled Time | Last disable reason |
-----|-----|-----|-----|-----|-----|-----
1 | Sc | Unlocked | Enabled | clear | normal | 0 days
0h:29m:33s | 08/23/2006 00:38:56 | switchover
2 | Sc | Unlocked | Enabled | clear | redundant | 0 days
0h:30m:55s | 08/23/2006 03:41:29 | switchover
7 | Tp1610 | Unlocked | Enabled | clear | redundant | 0 days
0h:17m:34s | 08/23/2006 03:46:41 | undetermined
8 | Tp1610 | Unlocked | Enabled | clear | normal | 0 days
0h:1m:3s | 08/23/2006 03:46:39 | undetermined
9 | Switch4411 | Unlocked | Enabled | clear | normal | 0 days
0h:14m:52s | 08/23/2006 03:32:31 | lock
19 | Switch4411 | Unlocked | Enabled | clear | redundant | 0 days
0h:15m:41s | 08/23/2006 03:37:42 | lock

TP Boards Table
=====

Slot# | Board Type | Software Version | Temperature 1 |
-----|-----|-----|-----|
7 | Tp1610 | 5.00.007.009 | 29 |
8 | Tp1610 | 5.00.007.009 | 29 |

client195 -- ~ -- /moGW
=>

```

2.3.9 show inventory

2.3.9.1 Description

Shows media gateway inventory.

```
show inventory
```

2.3.9.2 History

Version	Description
5.2	This command was introduced for the carrier class media gateways and media servers.

2.3.9.3 Examples

```

client195 -- ~ -- /moGW
=> show inventory

# =====
# MG Inventory
# =====

#Chassis Info

```

```

#=====
# /moGW MediaGatewayType(ro,online,none)=mediant5000(2)
# /moGW ChassisType(ro,online,none)=Unknown(0)
# /moGW ChassisSerialNumber(ro,offline,none)=AC00000
# /moGW EPROMVersion(ro,offline,none)=
# /moGW HardwareVersion(ro,offline,none)=

#Board Table
#=====
SlotNum | BoardType | BoardActiveRedundant | SlotsAdminState |
-----|-----|-----|-----|
0 | Alarm | normal | Unlocked |
1 | Sc | redundant | Unlocked |
2 | Sc | normal | Unlocked |
3 | Switch4411 | normal | Unlocked |
4 | Switch4411 | redundant | Locked |
5 | None | normal | Locked |
6 | Tp1610 | normal | Unlocked |
7 | None | normal | Locked |
8 | Tp1610 | normal | Unlocked |
9 | None | normal | Locked |
10 | Tp1610 | normal | Unlocked |

Table Continued...
-----
SlotOperationalState | AlarmSeverityLevel |
-----|-----|
Enabled | clear |
Disabled | clear |
Enabled | clear |
Enabled | clear |
Disabled | warning |
Disabled | clear |
Enabled | clear |
Disabled | clear |
Enabled | clear |
Disabled | clear |
Enabled | clear |

#TP Boards Table
#=====
IPAddress1 | SlotNum | IPAddress2 | SerialNumber1 | SerialNumber2 |
BoardType |
-----|-----|-----|-----|-----|
-----
10.7.9.224 | 6 | 10.7.9.225 | 391815 | 391801 | Tp1610
|
10.7.9.222 | 8 | 10.7.9.223 | 391699 | 391806 | Tp1610
|
10.7.9.220 | 10 | 10.7.9.221 | 231424 | 231417 | Tp1610
|

Table Continued...
-----
BoardApplicationType | MACAddress1 | MACAddress2 | TPSoftwareVersion |
-----|-----|-----|-----|
Gateway | 00908f05fa87 | 00908f05fa79 | 5.20.010.005 |
Gateway | 00908f05fa13 | 00908f05fa7e | 5.20.010.005 |
Gateway | 00908f038800 | 00908f0387f9 | 5.20.010.005 |

Table Continued...
-----
FlashVersion | BoardHardwareVersion | BaseBoardFirmwareVersion |

```



```

-----
192          | 0                      | 0                      |
192          | 0                      | 0                      |
192          | 0                      | 0                      |

Table Continued...
-----
ModuleFirmwareVersion | TrunksInBoard |
-----
50                    | 16             |
50                    | 16             |
17                    | 16             |

Table Continued...
-----
LicenseKeyString1
|
-----
tkR6r5to0n4x9lF8g3xvf3wk80gOf2R4f0grch8thz0LaRh6q3Vba3wk8y4OcyF3dwQrch0ugj
4JalBdY4Ql00c9i8232hddgQvfnMjgzbtSlNeoPRba2Ue5hoZcZFacxs8 |
jQR6r5toi74x9lF8g3xvf3wk32c?c2lc30Qrch8p13QB8lh6jjlaaisfc1UYcyhd2wUheNIk3z
7hRicNhjR0a00c5PAZcOFd0gcteOgH2P8xbRdlhjSXS20c304ZcZFa21s8 |
okRTr5topD4PbBF85zxf3Qhm0lIf25e8wUzelMtazQRb5t56Pla820cnilxcyR4bwUzelcsaj
oJc5NeWYl2a2keY8QAt2pe8MYyehwv9j8U9lRfX3x7czE0 |

Table Continued...
-----
LicenseKeydetailedstring1
|
-----
Key features: Board Type: TrunkPack 1610 SS7 Links: MTP2=8 MTP3=8 M2UA=8
M3UA=1 PSTN Protocols: IUA=8 DSP Voice features: RTCP-XR
AMRPolicyManagement Security: IPSEC MediaEncryption StrongEncryption
EncryptControlProtocol E1Trunks=8 T1Trunks=8 Coders: G723 G729 G728
NETCODER GSM-FR GSM-EFR AMR EVRC-QCELP G727 ILBC EVRC-B AMR-WB G722
Control Protocols: MGCP MEGACO H323 SIP PCI TPNCP Channel Type: RTP ATM
PCI DspCh=240 IP Media: Conf VXML VoicePromptAnnounc(H248.9) CALEA
TrunkTesting Default features: Coders: G711 G726 |
Key features: Board Type: TrunkPack 1610 E1Trunks=8 T1Trunks=8 SS7 Links:
MTP2=8 MTP3=8 M2UA=8 M3UA=1 Control Protocols: MGCP MEGACO H323 SIP PCI
TPNCP PSTN Protocols: IUA=8 Coders: G723 G729 G728 NETCODER GSM-FR GSM-
EFR AMR EVRC-QCELP G727 ILBC EVRC-B AMR-WB G722 DSP Voice features: RTCP-
XR AMRPolicyManagement Security: IPSEC MediaEncryption StrongEncryption
EncryptControlProtocol IP Media: Conf VXML VoicePromptAnnounc(H248.9)
CALEA TrunkTesting Channel Type: RTP ATM PCI DspCh=240 Default features:
Coders: G711 G726 |
Key features: Board Type: TrunkPack 1610 SB Control Protocols: MGCP
MEGACO H323 SIP PSTN Protocols: IUA=1 Security: IPSEC MediaEncryption
StrongEncryption EncryptControlProtocol SS7 Links: MTP2=8 MTP3=8 M3UA=1
IP Media: Conf Channel Type: RTP DspCh=248 M3K HA Coders: G723 G729
NETCODER GSM-EFR AMR EVRC-QCELP G727 G722 H264 MPEG4 E1Trunks=8
T1Trunks=8 DSP Voice features: IpmDetector RTCP-XR Default features:
Coders: G711 G726 |

Table Continued...
-----
LicenseKeyString2
|
-----
jQR6r5toi74x9lF8g3xvf3wk228232hd3MQiflMj0PEB8lh6h3Nfay8e1OASc2lc5xgjeMt13
QEaG20eA9ba28d10A?cipf3wMvfnMj2PbtSlNehjRba2Uee1oZcZFa21s8 |
okR6r5topD4x9lF85zxf3wknigOf2R4a0grch8tazcy9B18739bad3Yn24Zc0lc9MYI1x8t93
cCaBNe20lba2genycRe0lftf1Ihxcs9PcB8l9c90Fba2cen08ZcZFa9xs8 |
tlNTr5to0lQPbBF8pxhmf3Qh8PJEe2R4f1kyeh8tjhYX9BZcol92a28dazpEc2lcel4ze1QugN
YQalJfrlVeb3Al8zWr00lcch?eBlhrhxMSaRRFDlh7czE0 |

```

```

Table Continued...
-----
LicenseKeydetailedstring2
|
-----
-----
Key features: Board Type: TrunkPack 1610 Control Protocols: MGCP MEGACO
H323 SIP PCI TPNC P SS7 Links: MTP2=8 MTP3=8 M2UA=8 M3UA=1 PSTN
Protocols: IUA=8 DSP Voice features: RTPC-XR AMRPolicyManagement
Security: IPSEC MediaEncryption StrongEncryption EncryptControlProtocol
Coders: G723 G729 G728 NETCODER GSM-FR GSM-EFR AMR EVRC-QCELP G727 ILBC
EVRC-B AMR-WB G722 E1Trunks=8 T1Trunks=8 Channel Type: RTP ATM PCI
DspCh=240 IP Media: Conf VXML VoicePromptAnnounc(H248.9) CALEA
TrunkTesting Default features: Coders: G711 G726 |
Key features: Board Type: TrunkPack 1610 SS7 Links: MTP2=8 MTP3=8 M2UA=8
M3UA=1 Security: IPSEC MediaEncryption StrongEncryption
EncryptControlProtocol Channel Type: RTP ATM PCI DspCh=240 Control
Protocols: MGCP MEGACO H323 SIP PCI TPNC P DSP Voice features: RTPC-XR
AMRPolicyManagement PSTN Protocols: IUA=8 Coders: G723 G729 G728
NETCODER GSM-FR GSM-EFR AMR EVRC-QCELP G727 ILBC EVRC-B AMR-WB G722 IP
Media: Conf VXML VoicePromptAnnounc(H248.9) CALEA TrunkTesting E1Trunks=8
T1Trunks=8 Default features: Coders: G711 G726 |
Key features: Board Type: TrunkPack 1610 SB SS7 Links: MTP2=8 MTP3=8
M3UA=1 Security: IPSEC MediaEncryption StrongEncryption
EncryptControlProtocol E1Trunks=8 T1Trunks=8 DSP Voice features:
IpmDetector RTPC-XR IP Media: Conf Control Protocols: MGCP MEGACO H323
SIP Channel Type: RTP DspCh=248 M3K HA Coders: G723 G729 NETCODER GSM-
EFR AMR EVRC-QCELP G727 G722 PSTN Protocols: IUA=1 Default features:
Coders: G711 G726 |

#CPU Boards Table
#=====
IPAddress1 | SlotNum | SerialNumber | MacAddress | OSName |
OSVersion |
-----
-----
10.7.13.104 | 1 | | | |
|
10.7.13.91 | 2 | 83789aeb | 0:3:ba:78:9a:eb | SunOS | 5.9
|

Table Continued...
-----
OSServicePackpatchversion | CPUtype | CPUSpeed | HardDiskCapacity |
-----
-----
118558-39 Dec 2006 | sparcv9 | 650 MHz | 0 | 37430300 |

Table Continued...
-----
MemorySize | HardwareVersion | FirmwareVersion |
-----
-----
0 | | |
524288 | | 1.0.23 |

#Switch Boards Table
#=====
SwitchBoardIPAddr | SlotNum | SerialNumber | SwitchBoardType |
-----
-----
0.0.0.0 | 3 | (Not Available) | cpc4411 |
0.0.0.0 | 4 | 0 | cpc4411 |

Table Continued...
-----
MACAddress | HardwareVersion | MIBVersion | BootSoftwareVersion |
-----
-----

```

```

00c08c000000 | (Not Available) | 5.5.16 | 4.0.0 |
00c08c000000 | 0 | 5.5.16 | 4.0.0 |

Table Continued...
-----
ApplicationSoftwareVersion | BootLoaderVersion | BootScriptVersion |
-----
5.5.16 | 5.3.156 | 1.0.0 |
5.5.16 | 5.3.156 | 1.0.0 |

client195 -- ~ -- /moGW
=>
    
```

2.3.10 show ip

2.3.10.1 Description

Shows a IP configuration of the media gateway or of the current/specific VoP board.

```

show ip
show [<moBoard Path>] ip {conf [<if#>] | perf | route |
firewall | tls_version | cert}
    
```

2.3.10.2 Syntax Description

Syntax	Description
moBoard Path	The full or relative 'path' of the moBoard that will be shown. If unspecified and the current MO is moBoard , IP configuration of the current moBoard are displayed.
conf	When specified, shows the IP configuration of the VoP board
if#	Number of the network interface
perf	When specified, shows the IP performance measurements of the VoP board
route	When specified, shows the IP routing table of the VoP board
firewall	When specified, shows the IP firewall rules of the VoP board
tls_version	When specified, shows the TLS version of the VoP board
cert	When specified, shows the X.509 certificates of the VoP board

2.3.10.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.
5.0	Command syntax was modified.
6.2	Added an option to show IP configuration of current/specific VoP board.

2.3.10.4 Examples

```

client195 -- ~ -- /moGW
=> show ip

MGW IP
=====
Global IP Address: 10.7.19.20
Default Gateway Address: 10.7.0.1
Subnet Mask: 255.255.0.0

Subnet Separation & Static Routes
=====
Enable multiple subnets: Disabled

Network Services
=====
Main DNS Server IP Address: 0.0.0.0
DNS Server 2 IP Address: 0.0.0.0
DNS Server 3 IP Address: 0.0.0.0
NTP Server: 10.7.6.21
EMS IP Address: 10.7.6.21
NMS IP Address: 0.0.0.0
OSS IP Address: 0.0.0.0
APS IP Address: 0.0.0.0
Enable Syslog: enable(1)
Syslog Server IP: 10.7.19.20

SC IPs
=====

SC# | SC IP Address |
-----
1 | 10.7.13.58 |
2 | 10.7.13.59 |

Switch Boards IPs
=====

# | Local IP | Local Subnet |
-----
9 | 0.0.0.0 | 255.255.0.0 |
19 | 0.0.0.0 | 255.255.0.0 |

Switch Boards IPs - Cont.
=====

# | Virt Int IP | Virt Int Subnet | Virt Ext IP | Virt Ext Subnet |
-----
9 | 127.0.0.9 | 255.255.255.0 | 0.0.0.0 | 0.0.0.0 |
19 | 127.0.0.19 | 255.255.255.0 | 0.0.0.0 | 0.0.0.0 |

TP Boards IPs
=====

Index | IP Address 1 | Control IP 1 | Media IP 1 | SCTP IP 1 |
-----
7 | 10.7.19.7 | 0.0.0.0 | 0.0.0.0 | 0.0.0.0 |
8 | 10.7.19.3 | 0.0.0.0 | 0.0.0.0 | 0.0.0.0 |

TP Boards IPs - Cont.
=====

Index | IP Address 2 | Control IP 2 | Media IP 2 | SCTP IP 2 |
-----
    
```

```

7      | 10.7.19.8      | 0.0.0.0      | 0.0.0.0      | 0.0.0.0      |
8      | 10.7.19.4      | 0.0.0.0      | 0.0.0.0      | 0.0.0.0      |

MGC's IPs
=====

# | Name      | Protocol | Port  | Addressing Type | IP          | FQDN |
-----
0 | MGC#1    | mgcp    | 2427 | IPAddress      | 10.7.2.38  | none |

MGC's & TP Association.
=====

# | Name      | Voip Board  |
-----
0 | MGC#1    | Board#7,Board#8 |

NFS IPs
=====
No NFS profiles.

client195 -- ~ -- /moGW
=> show moBoard#8 ip conf

/sh ip conf

ACTIVE INTERFACE TABLE :
=====

Multiple IPs Disabled, VLANs Disabled, 1 interfaces active;

No IP Address      Pfx  Name
-----
0 10.7.19.154      16   O+M+C

* MAC address: 00-90-8f-09-2a-33

Note: Use "SHow IP CONF <index>" for details regarding a specific interface.

client195 -- ~ -- /moGW
=>

```

2.3.11 show calls

2.3.11.1 Description

Shows a list of all active calls on the current/specific media gateway board.

Note: The command is supported for MGCP or MEGACO call control protocols only.

```
show [<moBoard Path>] calls { [<trunk#>] | detail
{<endpoint>|<call_id>} | dur }
```

2.3.11.2 Syntax Description

Syntax	Description
moBoard Path	The full or relative 'path' of the moBoard that will be shown. If unspecified and the current MO is moBoard , calls of the current moBoard are displayed.
trunk #	When specified, shows calls on the specific trunk. The Trunk # should be specified in the 'internal board's format'. Use '-1' offset – e.g. 10 for moTrunk#11 . The same format is used in the command's output.
detail	When specified, shows detailed information on a specific call identified by endpoint name or CallID .
endpoint	Endpoint name as used by the Call Agent, e.g.. 'ds/Tr2/5'.
call_id	Call ID – e.g., 'C=1a2ff01'
dur	When specified, shows the call duration average for the past 48 hours.

2.3.11.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.
5.0	Command syntax was modified.

2.3.11.4 Examples

```
client195 -- ~ -- /moGW
```

```
=> show moBoard#5 calls
```

```
/sh mgcp calls
```

Endpoint	CallID(C)	ConnID(I)	Time(T)	Port(P)	Mode(M)	DSP
ds/e1-1/1	C=0f3e57	I=25221	T=5	P=4310,5860	M=sendrecv	1
ds/e1-1/2	C=0c92aa	I=25223	T=5	P=4320,5870	M=sendrecv	2
ds/e1-1/3	C=077175	I=25225	T=5	P=4330,5880	M=sendrecv	3
ds/e1-1/4	C=0f9abb	I=25227	T=5	P=4340,5890	M=sendrecv	4
ds/e1-1/5	C=040366	I=25229	T=5	P=4350,5900	M=sendrecv	5
ds/e1-1/6	C=083d0d	I=25231	T=5	P=4360,5910	M=sendrecv	12
ds/e1-1/7	C=047ce4	I=25233	T=5	P=4370,5920	M=sendrecv	13
ds/e1-1/8	C=0a00dc	I=25235	T=5	P=4380,5930	M=sendrecv	14
ds/e1-1/9	C=00d97b	I=25237	T=5	P=4390,5940	M=sendrecv	15
ds/e1-1/10	C=027b30	I=25239	T=5	P=4400,5950	M=sendrecv	16
ds/e1-1/11	C=0c7cb6	I=25241	T=5	P=4410,5960	M=sendrecv	17
ds/e1-1/12	C=05bb1d	I=25243	T=5	P=4420,5970	M=sendrecv	24
ds/e1-1/13	C=02b5c5	I=25245	T=5	P=4430,5980	M=sendrecv	25

```
client195 -- ~ -- /moGW
```

```
=> show moBoard#5 calls detail C=0c92aa
```

```
/sh mgcp calls
```

Endpoint	CallID(C)	ConnID(I)	Time(T)	Port(P)	Mode(M)	DSP
ds/e1-1/2	C=0c92aa	I=25223	T=5	P=4320,5870	M=sendrecv	2

```
client195 -- ~ -- /moGW
```

```
=>
```

2.3.12 show mgcp

2.3.12.1 Description

Shows MGCP parameters of the current/specific media gateway board.

```
show [<moBoard Path>] mgcp {conf | perf | ner | calls
[<trunk#>] | detail {<endpoint>|<call_id>} | dur | rsip | err |
cs }
```

2.3.12.2 Syntax Description

Syntax	Description
moBoard Path	The full or relative 'path' of the moBoard that will be shown. If unspecified and the current MO is moBoard , parameters of the current moBoard are displayed.
conf	When specified, shows the MGCP configuration.
perf	When specified, shows the MGCP performance measurements.
ner	When specified, shows the MGCP network efficiency rate.
calls	When specified, shows the currently active calls.
trunk#	The trunk number. Should be specified in the 'internal board's format', using -1 offset. E.g. trunk number 10 corresponds to the moTrunk#11 . The same format is used in the command's output
detail	When specified, shows detailed information on a specific call identified by endpoint name or CallID .
endpoint	Endpoint name as used by the Call Agent, e.g.. 'ds/Tr2/5'.
call_id	Call ID – e.g., 'C=1a2ff01'
dur	When specified, shows the calls duration statistics.
rsip	When specified, shows the MGCP RSIP statistics.
err	When specified, shows the MGCP failed responses statistics.
cs	When specified, shows the MGCP calls per second statistics.

2.3.12.3 History

Version	Description
6.2	This command was introduced for the carrier class media gateways and media servers.

2.3.12.4 Examples

```
client195 -- ~ -- /moGW
```

```
=> show moBoard#5 mgcp calls
```

```
/sh mgcp calls
```

Endpoint	CallID(C)	ConnID(I)	Time(T)	Port(P)	Mode(M)	DSP
ds/e1-1/1	C=0f3e57	I=25221	T=5	P=4310,5860	M=sendrecv	1
ds/e1-1/2	C=0c92aa	I=25223	T=5	P=4320,5870	M=sendrecv	2
ds/e1-1/3	C=077175	I=25225	T=5	P=4330,5880	M=sendrecv	3
ds/e1-1/4	C=0f9abb	I=25227	T=5	P=4340,5890	M=sendrecv	4
ds/e1-1/5	C=040366	I=25229	T=5	P=4350,5900	M=sendrecv	5
ds/e1-1/6	C=083d0d	I=25231	T=5	P=4360,5910	M=sendrecv	12
ds/e1-1/7	C=047ce4	I=25233	T=5	P=4370,5920	M=sendrecv	13
ds/e1-1/8	C=0a00dc	I=25235	T=5	P=4380,5930	M=sendrecv	14
ds/e1-1/9	C=00d97b	I=25237	T=5	P=4390,5940	M=sendrecv	15
ds/e1-1/10	C=027b30	I=25239	T=5	P=4400,5950	M=sendrecv	16
ds/e1-1/11	C=0c7cb6	I=25241	T=5	P=4410,5960	M=sendrecv	17
ds/e1-1/12	C=05bb1d	I=25243	T=5	P=4420,5970	M=sendrecv	24
ds/e1-1/13	C=02b5c5	I=25245	T=5	P=4430,5980	M=sendrecv	25

```
client195 -- ~ -- /moGW
```

```
=> show moBoard#5 mgcp calls detail C=0c92aa
```

```
/sh mgcp calls
```

Endpoint	CallID(C)	ConnID(I)	Time(T)	Port(P)	Mode(M)	DSP
ds/e1-1/2	C=0c92aa	I=25223	T=5	P=4320,5870	M=sendrecv	2

```
client195 -- ~ -- /moGW
```

```
=>
```

2.3.13 show megaco

2.3.13.1 Description

Shows MEGACO parameters of the current/specific media gateway board.

```
show [<moBoard Path>] megaco {conf | perf | ner | calls
[<trunk#>] | detail {<endpoint>|<call_id>} | dur | err | cs }
```

2.3.13.2 Syntax Description

Syntax	Description
moBoard Path	The full or relative 'path' of the moBoard that will be shown. If unspecified and the current MO is moBoard , parameters of the current moBoard are displayed.
conf	When specified, shows the MEGACO configuration.
perf	When specified, shows the MEGACO performance measurements.
ner	When specified, shows the MEGACO network efficiency rate.
calls	When specified, shows the currently active calls.
trunk#	The trunk number. Should be specified in the 'internal board's format', using -1 offset. E.g. trunk number 10 corresponds to the moTrunk#11 . The same format is used in the command's output
detail	When specified, shows detailed information on a specific call identified by endpoint name or CallID .
endpoint	Endpoint name as used by the Call Agent, e.g.. 'ds/Tr2/5'.
call_id	Call ID – e.g., 'C=1a2ff01'
dur	When specified, shows the calls duration statistics.
err	When specified, shows the MEGACO failed responses statistics.
cs	When specified, shows the MEGACO calls per second statistics.

2.3.13.3 History

Version	Description
6.2	This command was introduced for the carrier class media gateways and media servers.

2.3.13.4 Examples

```
client195 -- ~ -- /moGW
```

```
=> show moBoard#5 megaco cs
```

```
/sh megaco cs
```

```
Call Attempts Per Second
```

```
-----  
Last second:      4                      Total call attempts count: 37
```

```
History:           Minimum    Maximum    Average    Interval Duration  
                   Attempts     Attempts                (minutes:seconds)  
-----  
Current Interval T      3         4         4         0:10  
Prev Interval T-1       1         3         2         15:00  
Prev Interval T-2       0         0         0         15:00
```

```
client195 -- ~ -- /moGW
```

```
=>
```

2.3.14 show ams

2.3.14.1 Description

Shows media server data of the current/specific media gateway board.

```
show [<moBoard Path>] ams {perf}
```

2.3.14.2 Syntax Description

Syntax	Description
moBoard Path	The full or relative 'path' of the moBoard that will be shown. If unspecified and the current MO is moBoard , parameters of the current moBoard are displayed.
perf	When specified, shows the media server performance measurements.

2.3.14.3 History

Version	Description
6.2	This command was introduced for the carrier class media gateways and media servers.

2.3.14.4 Examples

```
client195 -- ~ -- /moGW
```

```
=> show moBoard#5 ams perf
```

```
/sh ams perf
```

Supported only for boards with an IPMedia Feature key

```
client195 -- ~ -- /moGW
```

```
=>
```

2.3.15 show tdm

2.3.15.1 Description

Shows PSTN-related statistics of the current/specific media gateway board.

```
show [<moBoard Path>] tdm {status | perf | summary}
```

2.3.15.2 Syntax Description

Syntax	Description
moBoard Path	The full or relative 'path' of the moBoard that will be shown. If unspecified and the current MO is moBoard , parameters of the current moBoard are displayed.
status	When specified, shows the alarm status of the E1/T1 trunks.
perf	When specified, shows the performance measurements of the E1/T1 trunks.
summary	When specified, shows the PSTN summary statistics.

2.3.15.3 History

Version	Description
6.2	This command was introduced for the carrier class media gateways and media servers.

2.3.15.4 Examples

```
client195 -- ~ -- /moGW
=> show moBoard#5 tdm status
/sh tdm status
Trunk 1: Active
Trunk 2: Active
Trunk 3: Active
Trunk 4: Active
...
client195 -- ~ -- /moGW
=>
```

2.3.16 show dsp

2.3.16.1 Description

Shows DSP statistics of the current/specific media gateway board.

```
show [<moBoard Path>] dsp {status | perf}
```

2.3.16.2 Syntax Description

Syntax	Description
moBoard Path	The full or relative 'path' of the moBoard that will be shown. If unspecified and the current MO is moBoard , parameters of the current moBoard are displayed.
status	When specified, shows the DSP version and status.
perf	When specified, shows the DSP performance measurements.

2.3.16.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.
5.0	Command syntax was modified.
6.2	Command syntax was modified.

2.3.16.4 Examples

```
client195 -- ~ -- /moGW
=> show moBoard#5 dsp status
/sh dsp status
DSP firmware:491096AE8 Version:0620.16 - Used=3 Free=2013 Total=2016
  DSP device  0:  Active   Used= 1  Free=15  Total=16
  DSP device  1:  Active   Used= 1  Free=15  Total=16
  ...
client195 -- ~ -- /moGW
=>
```

2.3.17 show voiceprompt

2.3.17.1 Description

Shows voice prompts table of the current/specific media gateway board.

```
show [<moBoard Path>] voiceprompt {numofentries | entries
<start#> [<num>]}
```

2.3.17.2 Syntax Description

Syntax	Description
moBoard Path	The full or relative 'path' of the moBoard that will be shown. If unspecified and the current MO is moBoard , parameters of the current moBoard are displayed.
numofentries	When specified, shows the number of entries in voice prompts table.
entries	When specified, shows the voice prompts table.
start#	Start entry number.
num	Number of entries to display.

2.3.17.3 History

Version	Description
6.2	This command was introduced for the carrier class media gateways and media servers.

2.3.17.4 Examples

```
client195 -- ~ -- /moGW
```

```
=> show moBoard#5 voiceprompt numofentries
```

```
/sh voiceprompt numofentries
```

```
First Used VoicePrompt Index: 0 First Free VoicePrompt Index: 0
```

```
client195 -- ~ -- /moGW
```

```
=>
```

2.3.18 show tones

2.3.18.1 Description

Shows information about special tones of the current/specific media gateway board.

```
show [<moBoard Path>] tones {cpt | udt | rngt | prt} [<tone#>]
```

2.3.18.2 Syntax Description

Syntax	Description
moBoard Path	The full or relative 'path' of the moBoard that will be shown. If unspecified and the current MO is moBoard , parameters of the current moBoard are displayed.
cpt	When specified, shows the call progress tones table.
udt	When specified, shows the user-defined tones table.
rngt	When specified, shows the ring tones table.
prt	When specified, shows the pre-recorded tones table.
tone#	Tone index. If absent, generic information is displayed. If negative, next tone is displayed.

2.3.18.3 History

Version	Description
6.2	This command was introduced for the carrier class media gateways and media servers.

2.3.18.4 Examples

```
client195 -- ~ -- /moGW
```

```
=> show moBoard#5 tones cpt
```

```
/sh tones cpt
```

```
Call Progress Tone - General information:
```

```
-----
```

```
Num of Tones: 11, (11 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
```

```
client195 -- ~ -- /moGW
```

```
=>
```

2.3.19 show children

2.3.19.1 Description

Shows all children of the current/specific MO. Typically used to determine possible inputs for the command `go`.

```
show [<Mo Path>] children
```

2.3.19.2 Syntax Description

Syntax	Description
MO Path	The full or relative 'path' of the MO whose children will be shown. If not specified, children of current MO are displayed.

2.3.19.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.
5.0	Command syntax was modified.

2.3.19.4 Examples

```
client195 -- ~ -- /moGW
=> show children
Available children Mo's for moGW:
  moFile - Instances: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23
  moSubnet - Instances: 1
  moNetworkProfile - Instances: 1
  moSCStaticRoute - No instances.
  moMGC - Instances: 0
  moTpRG - Instances: 0
  moFirewallProfile - No instances.
  moSCIPSECRule - No instances.
  moTPIPSECCProfile - No instances.
  moSS7Mtp3Group - No instances.
  moNfsProfile - No instances.
  moAMRCodecPoliciesProfile - No instances.
  moNtpServer - Instances: 1
  moTPConfigurationBackdoor - No instances.
  moBoard - Instances: 7, 8
  moSwitchBoard - No instances.
  moESBoard - Instances: 3, 4
  moCPUBoard - Instances: 1, 2
  moTimingModules
  moSatHW - Instances: 1, 2
  moAaaServerInfo - No instances.
  moVlanTag - No instances.
  moPMThreshold - No instances. client195 -- ~ -- /moGW
=>
```

2.3.20 searchParam

2.3.20.1 Description

Search parameter by partial name.

```
searchParam <partial name>
```

2.3.20.2 Syntax Description

Syntax	Description
partial name	Partial name of the parameter to be searched.

2.3.20.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers

2.3.20.4 Usage Guidelines

Use the command 'searchParam' to find all references of a partial parameter name string. The command returns a list of all matching parameters with the following information per parameter: MO, Frame Name (usually the same as the MO name), Tab Name (configuration rubric title).

After the result for command 'searchParam' is returned, users can perform **show param** for the appropriate MO to view the parameter's value.

2.3.20.5 Examples

```

client195 -- ~ -- /moGW
=> searchParam dns
DNSQueryType - MO: moBoard, Frame: SIP Protocol Definitions, Tab: General Settings
DNSQueryType - MO: moBoard, Frame: SIP Protocol Definitions, Tab: General Settings
MainDNSServerIPAddress - MO: moGW, Frame: Media Gateway Parameters Provisioning,
Tab: Network Services
DNSServer2IPAddress - MO: moGW, Frame: Media Gateway Parameters Provisioning, Tab:
Network Services
DNSServer3IPAddress - MO: moGW, Frame: Media Gateway Parameters Provisioning, Tab:
Network Services
MGFQDNsysName - MO: moGW, Frame: Media Gateway Parameters Provisioning, Tab: MG
General Settings
ProxyDNSQueryType - MO: moBoard, Frame: SIP Proxy Properties, Tab: Proxy Options
client195 -- ~ -- /moGW
=> show /moGW param MainDNSServerIPAddress
MainDNSServerIPAddress=0.0.0.0
client195 -- ~ -- /moGW
=>
    
```

2.4 Modify Commands

2.4.1 modify

2.4.1.1 Description

Modifies the value of a specific parameter in the current/specific MO.

```
modify [<Mo Path>] <Param Name>=<Value>
```

2.4.1.2 Syntax Description

Syntax	Description
Mo Path	Specifies the path of the MO. If unspecified, the current MO is used.
Param Name	Name of the required parameter
Value	Required parameter value

2.4.1.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers
5.0	Command is renamed

2.4.1.4 Usage Guidelines

Use the auto-completion functionality (by pressing **Tab** while typing the command) to view a list of all available MOs, the parameters of a specific MO, and a specific parameter's description and values.

In addition to the above, use the command **help params** to check which parameters can be provisioned in each MO and their possible values. Use the command **show param** to check current parameter values.

2.4.1.5 Examples

```
client195 -- ~ -- /moGW
=> modify moBoard#6 BoardLoggingonExternalServer=0
client195 -- ~ -- /moGW
=>
```

2.4.2 lock

2.4.2.1 Description

Takes the current/specific MO out of service by changing its attribute AdministrativeState to LOCKED. Users must explicitly acknowledge the command unless flag **-f** is specified.

Graceful lock is supported on moGW and moBoard. It allows gradually dropping the service within the specified graceful timeout. Media gateway boards indicate the beginning of the graceful timeout to the Media Gateway Controllers via appropriate call control messages and the latter stops sending new calls to the media gateway. The media gateway boards are locked when either all active calls are terminated or the graceful timeout expires. In the latter case, the remaining calls are dropped.

```
lock [<Mo Path>] [<timeout>] [-f]
```

2.4.2.2 Syntax Description

Syntax	Description
Mo Path	Specifies the path of the MO. If unspecified, the current MO is used.
timeout	Graceful lock timeout (in seconds) on moGW or moBoard.
-f	Force lock – don't ask for the user's acknowledge.

2.4.2.3 History

Version	Description
5.0	This command was introduced for the carrier class media gateways and media servers

2.4.2.4 Examples

```
client195 -- ~ -- /moGW
=> lock moBoard#6
-----
Do you really want to lock /moGW/moBoard#6? [y/n] y
-----
Performing lock action on /moGW/moBoard#6 - OK!
client195 -- ~ -- /moGW
=>
```

2.4.3 unlock

2.4.3.1 Description

Restores service on the current/specific MO by changing its attribute AdministrativeState to UNLOCKED.

It may take some time until the service on a specific MO is actually restored. When the latter happens, the MO's attribute OperationalState is updated to ENABLED.

```
unlock [<Mo Path>]
```

2.4.3.2 Syntax Description

Syntax	Description
Mo Path	Specifies the path of the MO. If not specified, the current MO is used.

2.4.3.3 History

Version	Description
5.0	This command was introduced for the carrier class media gateways and media servers.

2.4.3.4 Examples

```
client195 -- ~ -- /moGW
=> unlock moBoard#6
Performing unlock action on /moGW/moBoard#6 - OK!
client195 -- ~ -- /moGW
=>
```

2.4.4 add

2.4.4.1 Description

Adds a new MO.

```
add <Mo Path> [<Options>]
```

2.4.4.2 Syntax Description

Syntax	Description
Mo Path	Specifies the path of the MO to be added.
Options	When adding moBoard, the following options must be specified: <ul style="list-style-type: none"> HW Type: tp-6310-t3, tp-6310-stm1, ipm-6310-t3, ipm-6310-stm1, tp-8410, ipm-8410 Application Type: gateway, media-server, sip-gateway, sip-media-server

2.4.4.3 History

Version	Description
5.0	This command was introduced for the carrier class media gateways and media servers
5.2	Syntax of add command for VoP boards was modified.
5.4	Syntax of add command for VoP boards was extended to support 8410 boards.

2.4.4.4 Examples

```
client195 -- ~ -- /moGW
=> add moBoard#6/moSS7DataLink#1
# moSS7DataLink instance 1 was added to /moGW/moBoard#6 (table range is 1-32).
client195 -- ~ -- /moGW
=>
```


2.4.5 remove

2.4.5.1 Description

Removes an existing MO.

The MO must be locked prior to removing it. Note that some MOs are not dynamic and cannot be removed (e.g., moCPUBoard).

```
remove <Mo Path>
```

2.4.5.2 Syntax Description

Syntax	Description
Mo Path	Specifies the path of the MO to be removed.

2.4.5.3 History

Version	Description
5.0	This command was introduced for the carrier class media gateways and media servers

2.4.5.4 Examples

```
client195 -- ~ -- /moGW
=> remove moBoard#6/moSS7DataLink#1
# moSS7DataLink instance 1 was removed from /moGW/moBoard#6
client195 -- ~ -- /moGW
=>
```

2.4.6 addBoard

2.4.6.1 Description

Adds a new media gateway board.

```
addBoard <HW Type> <Slot Number> <App Type>
```

2.4.6.2 Syntax Description

Syntax	Description
HW Type	Hardware type of the media gateway board. The following types are supported: <ul style="list-style-type: none"> • tp-6310-t3 • tp-6310-stm1 • ipm-6310-t3 • ipm-6310-stm1 • tp-8410 • ipm-8410
Slot Number	Number of the slot in which the media gateway board resides.
App Type	Application type of the media gateway board. The following types are supported: <ul style="list-style-type: none"> • gateway • media-server • sip-gateway • sip-media-server

2.4.6.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers
5.0	Command syntax was modified
5.2	HW Type and App Type syntax was modified.
5.4	Added support for 8410 boards.

2.4.6.4 Usage Guidelines

Note that the command **add** can be used to add media gateway boards as well. The command **addBoard** is preserved mainly for backward compatibility.

2.4.6.5 Examples

```
client195 -- ~ -- /moGW
```

```
=> addBoard tp-6310-stm1 9 gateway
add Tp6310 GeneralMg
# Board #9 was added successfully.
```

```
client195 -- ~ -- /moGW
```

```
=> add moBoard#10 tp-6310-stm1 gateway
add Tp6310 GeneralMg
# Board #10 was added successfully.
```

```
client195 -- ~ -- /moGW
```

```
=>
```

2.4.7 removeBoard

2.4.7.1 Description

Removes an existing media gateway board. The media gateway board must be locked prior to removing it.

```
removeBoard <Slot Number>
```

2.4.7.2 Syntax Description

Syntax	Description
Slot Number	Number of the slot where a media gateway board resides.

2.4.7.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.

2.4.7.4 Usage Guidelines

Note that the command `remove` can also be used to remove media gateway boards. The command `removeBoard` is preserved mainly for backward compatibility.

2.4.7.5 Examples

```
client195 -- ~ -- /moGW
=> removeBoard 9
# Board #9 was removed successfully.
client195 -- ~ -- /moGW
=> remove moBoard#10
# Board #10 was removed successfully.
client195 -- ~ -- /moGW
=>
```

2.4.8 Action

2.4.8.1 Description

Performs an action on the current/specific MO.

```
action [<Mo Path>] <Action> [<Action Params>]
```

2.4.8.2 Syntax Description

Syntax	Description
Mo Path	The path of the MO. If unspecified, the current MO is used.
Action	Each MO supports a set of different actions. Refer to Appendix A – Supported MO Actions for a full list of all supported actions. To view a list of the actions supported on the specific MO, enter: <code>help actions <MO Path></code>
Action Params	According to the command <code>help actions <MO Path> <action></code> .

2.4.8.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.
5.0	Command syntax was modified. Separate CLI commands were introduced for common actions – lock, unlock, add and remove.

2.4.8.4 Examples

```
client195 -- ~ -- /moGW
=> action moBoard#8/moSS7DataLink#3 activate
client195 -- ~ -- /moGW
=>
```

2.5 Scripting Commands

2.5.1 execFromFile

2.5.1.1 Description

Executes commands from a CLI script file

```
execFromFile <File Name> [-v]
```

2.5.1.2 Syntax Description

Syntax	Description
File Name	The name of the file containing the CLI script to be executed. A full or relative path name can be specified.
-v	Hidden flag enabling verbose output of all script lines before they are executed. Used mainly to debug CLI scripts.

2.5.1.3 Usage Guidelines

Refer to Scripting Capabilities on page 89 for a detailed description of CLI scripting capabilities and a brief script language reference.

2.5.1.4 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.
5.0	Command syntax is modified.

2.5.2 waitParam

2.5.2.1 Description

Waits until a parameter gets a specific value.

```
waitParam [<Mo Path>] <Param Name> <Value> <Timeout>
```

2.5.2.2 Syntax Description

Syntax	Description
Mo Path	Specifies the path of the Managed Object. If not specified, the current MO is used.
Param Name	Parameter name
Value	The parameter value that the user is waiting for.
Timeout	If the timeout expires and the parameter specified by the user in <Param Name> still hasn't gotten the value that the user is waiting for, the command waitParam exits with a non-zero return code, returned to the shell (and stored in variable \$?).

2.5.2.3 Usage Guidelines

The command **waitParam** is used mostly in automatic or semi-automatic scripts. For example, if you need to wait for an ES board to complete an unlock process after you perform an unlock action, write the following in your shell script:

```
unlock es#3
waitParam es#3 OperationalState 1 600
if [[ $? -ne 0 ]]; then
print "ERROR: failed to unlock ES board"
fi
```

2.5.2.4 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.

2.6 Debug Commands

2.6.1 tpPing

2.6.1.1 Description

Sends ICMP echo request packets from a specific media gateway board

```
tpPing [<moBoard Path>] [<options>] <IP address>
```

The following options are supported:

```
-n count
-l size
-w timeout
-I interface address
-p class-of-service
```

2.6.1.2 Syntax Description

Syntax	Description
-n count	Number of echo requests to send. Default = 5.
-l size	Send the buffer size.
-w timeout	Timeout, in seconds, to wait for each reply.
-I interface_address	Set source address to specified interface address
-p class-of-service	Class of service (802.1p) to assign with the ping packet. <ul style="list-style-type: none"> • 0 = NTWRK • 1 = PRM_MEDIA • 2 = PRM_CTRL • 3 = GOLD • 4=BRNZE

2.6.1.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.
6.2	"-I interface_address" option is added; "-v traffic_type" option is removed.

2.6.1.4 Examples

```
client195 -- ~ -- /moGW
```

```
=> tpPing moBoard#5 10.7.5.234
```

```
tpPing moBoard#5 10.7.5.234
```

```
Reply from 10.7.5.234: bytes=32 time=0ms
```

```
Reply from 10.7.5.234: bytes=32 time=0ms
```

```
Reply from 10.7.5.234: bytes=32 time=0ms
```

```
Reply from 10.7.5.234: bytes=32 time=0ms
```

```
Ping statistics for 10.7.5.234:
```

```
    Packets: Sent = 4, Received = 4, Lost 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
client195 -- ~ -- /moGW
```

```
=>
```


2.6.2 tpCmd

2.6.2.1 Description

Performs the media gateway board's debug command. Provides a direct interface to the media gateway board's command shell.

```
tpCmd [<moBoard Path>] <Command> [<Command Params>]
```

2.6.2.2 Syntax Description

Syntax	Description
moBoard Path	Path of the moBoard. Cannot be specified if the current MO is moBoard.
Command	Media gateway board's debug command as specified in Appendix B – Supported Media Gateway Board's Debug Command .
Command Params	Parameters of the media gateway board debug command as specified in Appendix B.

2.6.2.3 Usage Guidelines

Refer to Appendix B – Supported Media Gateway Board's Debug Command on page 103 for a detailed description of supported **tpCmd** commands.

2.6.2.4 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.

2.6.2.5 Examples

```
client195 -- ~ -- /moGW
```

```
=> tpCmd moBoard#5 sh info
sh info
Board type: TrunkPack 6310 SDH, firmware version 4.80.017.004
Uptime: 0 days, 0 hours, 23 minutes, 48 seconds
Memory size: 256MB, usage 65%
Temperature reading: 28 C
Last reset reason:
Board was restarted due to a hardware reset
```

```
client195 -- ~ -- /moGW
```

```
=> tpCmd moBoard#5 /cp/cdr
/cp/cdr
CallDetailReport - CDR
CallDetailReport (CDR) - Controls collection of CDR records for completed voice
calls.
Usage:
  CDR START [SYSLOG|FILE|BOTH] Starts collecting CDR records
  CDR SHOW           Displays CDR buffer
  CDR SEND <nfs_URI> Sends collected CDR data to specified NFS location,
                    e.g. file://10.31.2.10/tmp/cdrfile.txt
                    NOTE: the NFS filesystem must be mounted.
  CDR STOP           Stops collecting CDR records and clears the buffers.
```

```
client195 -- ~ -- /moGW
```

```
=> tpCmd moBoard#5 /cp/cdr start both
/cp/cdr start both
Starting CDR collection.
```

```
client195 -- ~ -- /moGW
```

```
=> tpCmd moBoard#5 /cp/cdr show
/cp/cdr show
15:59:52.575 10.7.12.20 MGCP_CALL_STATISTICS: Endpoint Name: ds/e1-10/26, Connec
tion deleted by Call Agent, Coder: UNKNOWN, ConnectionID: 6610, CallId: 07fcc4,
Call duration: 63 seconds, Local RTP port: 7350, Remote RTP address: 10.7.12.20
port 5800, Tx/Rx bytes 504960/504960, Tx/Rx packets 3156/3156
15:59:52.600 10.7.12.20 MGCP_CALL_STATISTICS: Endpoint Name: ds/e1-5/27, Connect
ion deleted by Call Agent, Coder: UNKNOWN, ConnectionID: 6611, CallId: 0ee343, C
all duration: 63 seconds, Local RTP port: 5810, Remote RTP address: 10.7.12.20 p
ort 7360, Tx/Rx bytes 504960/504960, Tx/Rx packets 3156/3156
15:59:52.600 10.7.12.20 MGCP_CALL_STATISTICS: Endpoint Name: ds/e1-10/27, Connec
tion deleted by Call Agent, Coder: UNKNOWN, ConnectionID: 6612, CallId: 08bdb5,
Call duration: 63 seconds, Local RTP port: 7360, Remote RTP address: 10.7.12.20
port 5810, Tx/Rx bytes 505120/505120, Tx/Rx packets 3157/3157
(end of CDR buffer)
```

```
client195 -- ~ -- /moGW
```

```
=> tpCmd moBoard#5 /cp/cdr send file://10.7.5.234/Project/bin/log/cdr/tp5.txt
/cp/cdr send file://10.7.5.234/Project/bin/log/cdr/tp5.txt
Sending CDR data to NFS location file://10.7.5.234/Project/bin/log/cdr/tp5.txt
```

```
client195 -- ~ -- /moGW
```

```
=> tpCmd moBoard#5 /cp/cdr stop
/cp/cdr stop
Stopping CDR collection.
```

```
client195 -- ~ -- /moGW
```

```
=>
```

2.6.3 Log

2.6.3.1 Description

Shows the log file.

```
log [<option>] <Log File>
```

2.6.3.2 Syntax Description

Syntax	Description
Log File	<p>The media gateway logs different software and hardware activities of all system components.</p> <p>The following log files are intended for customer use:</p> <ul style="list-style-type: none"> • trap – SNMP traps produced by the gateway (this is typically the only log that is interfaced by customers; the other logs contain debug information intended mainly for FAEs/developers) <p>The following log files are intended for FAE/developers use and are not typically used by customers:</p> <ul style="list-style-type: none"> • core – main management process logs • bootp – BOOTP server logs • cli – CLI server logs • hbg – heart beat generator logs • hbm – heart beat monitor logs • tpncp – TPNC interface logs • sat – SA-1 alarm card logs • syslog – SYSLOG server logs • upgrade – Online Software Upgrade logs • watchdog – software watchdog logs • activity – CLI activity logs • radius – RADIUS server logs • xfer – auxiliary files transfer logs • tftp – TFTP server logs • configchange – configuration change logs • aaad – centralized authentication and authorization daemon logs <p>The media gateway board's logs can also be viewed via command log. Use the following syntax:</p> <p>board#<slot_number></p>
options	<p>When no options are specified, the command log displays the new logs as they are generated.</p> <p>The following options are also supported:</p> <ul style="list-style-type: none"> → -<number> – displays the last <number> of lines from the logfile (e.g., log -100 trap displays the last 100 logs from the trap log file). → -f – displays the complete log file (with syntax highlighting). → -F – displays the complete log file (without syntax highlighting). → -h – allows viewing the history of the log files (10 last log files are stored in the history).

2.6.3.3 History

Version	Description
5.0	This command was introduced for the carrier class media gateways and media servers.
5.2	Syntax of board's log was modified. New logs radius and xfer were added.
5.4	New logs tftp and configchange were added.

2.6.3.4 Examples

```

client195 -- ~ -- /moGW
=> log -10 trap
- TextualDescription: VoP board Health Diagnostics test passed successfully. Test
Initiative: Periodic Diagnostics
- Type: equipmentAlarm, Probable Cause: equipmentMalfunction, Id: 496
- AdditionalInfo1: Fault element (from last diagnostics test): Master TPM:
CONFERENCE_CHANNEL(NotChecked)
- AdditionalInfo2: Test Initiative: Periodic
09/21 08:27:00.170 moBoard 4_2 (Board#7) sent alarm:
- Name: Board Diagnostics Alarm, Severity: cleared, Source: Board#7
- TextualDescription: VoP board Health Diagnostics test passed successfully. Test
Initiative: Periodic Diagnostics
- Type: equipmentAlarm, Probable Cause: equipmentMalfunction, Id: 497
- AdditionalInfo1: Fault element (from last diagnostics test): Slave TPM:
CONFERENCE_CHANNEL(NotChecked)
- AdditionalInfo2: Test Initiative: Periodic

client195 -- ~ -- /moGW
=> log board#8 tpm=1
09/21 08:49:57.597 [Notice] Loading default data for LOGO-BKG file
[File:CfgFileHndlr.cpp Line:3937] [Time: 0:0:2]
09/21 08:49:58.949 [Warning] *** Table SS7Mtp2Params - Old MTP2 parameters
Configuration has been used. Table has been generated. [Trunk:-1 Bchannel:-1
ConnID:-1] [Code:777 File:SS7Mtp2ParmsTable.cpp Line:411] [Time: 0:0:3]
09/21 08:49:59.948 [Notice] Board was restarted due to a hardware reset
[File:Exception.cpp Line:1143] [Time: 0:0:4]
09/21 08:50:00.645 [Warning] DSP Version Info: acNumberOfDSPs=40,
acNumberOfDSPChannels=240 [File:Main.cpp Line:843] [Time: 0:0:5]
^C
client195 -- ~ -- /moGW
=>
    
```

2.7 Administration Commands

2.7.1 Tools

2.7.1.1 Description

Perform Media Gateway administration tasks.

```
tools [<options>]
```

`tools` is an interactive script that performs different media gateway administration tasks. When executed without any option, it displays an interactive menu. However, it's also possible to specify required menu options in the string `<options>` (e.g., `tools sc p`).

The following menus can be accessed by CLI users who have administrator privileges:

- **sc** – stop/start media gateway software
- **report** – create bug report summary file
- **bk** – create backup file

All other script menus and options should be run by **root** user only. Refer to Section 0 on page 18 for a description of the **user** menu and to the Mediant 5000 IOM Manual for descriptions of the other menus.

2.7.1.2 Syntax Description

Syntax	Description
<code>tools sc p</code>	Displays the summary of media gateway software processes on a specific SC board.
<code>tools sc dn</code>	Stops media gateway software on a specific SC board. Note that to completely cease media gateway operation, media gateway software on <i>both</i> SC boards must be stopped. Also, for normal media gateway operation, there is no need to stop media gateway software and the media gateway's LOCK/UNLOCK commands should be used instead.
<code>tools sc up</code>	Starts media gateway software on a specific SC board.
<code>tools report</code>	Generates a bug report file that should be submitted to Technical Support as part of the problem description.
<code>tools bk</code>	Generates a backup file. Refer to Mediant 5000/8000 Installation and Operation Manual for more details.

2.7.1.3 History

Version	Description
3.2	This command was introduced for the carrier class media gateways and media servers.

2.7.1.4 Examples

```

client195 -- ~ -- /moGW
=> tools sc p

LIST OF SC SOFTWARE PROCESSES
-----
20751 tg_syslo tg_syslog
20748 tg_core tg_core
20747 tg_bootp tg_bootp
20746 tg_tpncp tg_tpncpif
20745 tg_sat tg_sat
20744 tg_hbg tg_hbg
20743 tg_hbm tg_hbm
20709 tg_watch /Project/bin/exe/tg_watchdog
    
```

```

client195 -- ~ -- /moGW
=> tools sc dn

STOP SC SOFTWARE
-----
Stop SC software? ([y]/n) : y

>>> Stop watchdog process...
>>> Wait for application to stop...
    
```

```

client195 -- ~ -- /moGW
=>
    
```

3 Performing Commonly Used Tasks

3.1 Checking the Status of the Media Gateway

Use the following command:

```
show hw
```

Three subsections are displayed:

4. Info
5. Hardware
6. Boards Information.

The subsection 'Info' includes basic information such as the gateway name, version, chassis type.

The subsection 'Hardware' displays the status of all the FRUs.

The subsection 'Boards Information' displays a list of all boards with respect to type, status, redundancy, board's uptime and the last reload reason. For media gateway boards, the software version and the temperature are also displayed.

3.2 Checking Definitions Related to the IP Network

Use the following command:

```
show ip.
```

The command displays all information relating to network definitions and all IPs defined when provisioning the gateway, including the following subsections:

1. MGW IP
2. Subnet Separation and Static Route - shown if the subnet separation feature is enabled.
3. Network Services - DNS, NTP, EMS, NMS, OSS, APS and Syslog server IPs.
4. SC, Ethernet Switch and media gateway boards IPs
5. MGCs IPs and basic info
6. NFS IPs

3.3 Checking the Media Gateway Board's Configuration and Status

Use the command `show moBoard#X` to view all parameters of the Media Gateway board. The output will be long because each Media Gateway board features hundreds of parameters. Use the following commands to produce shorter output:

- `show moBoard#X conf -nd` – to view configuration parameters whose value differs from the default

For additional status information, use the following commands:

- `show moBoard#X status` – to view the status of the media gateway board
- `show moBoard#X calls` – to view active MGCP calls (with a sub-option for details)
- `show moBoard#X pm` – to view performance measurements

Each Media Gateway board MO has many 'children' MOs that represent different properties and applications of the board – e.g., Trunks, SS7 Data Links, etc.

Use the command `show moBoard#X children` to view all 'children' MOs of the Media Gateway board. Use the command `show moBoard#X conf -r -nd` to view the non-default configuration of the Media Gateway board and all 'children' MOs.

3.4 Adding and Removing the Media Gateway Board

Use the command `add` or `addBoard` to add a new Media Gateway board. Refer to section 'addBoard' on page 74 for a detailed description of the command's syntax.

Use the command `remove` or `removeBoard` to remove the Media Gateway board. Refer to Section 2.4.7 on page 75 for a detailed description of the command's syntax.

3.5 Navigating through the MO Tree

Use the command `go` to go to a specific MO. The MO name can be entered in one of the following ways:

- MO name (for children MOs) – e.g., `go moBoard#5`
- Absolute MO path – e.g., `go /moGW/moBoard#5/moTrunk#7`
- Relative MO path – e.g., `go moBoard#5/moTrunk#7` or `go ../moTrunk#7`

Use the commands `show children` and `moTree` to view all possible navigation options at the current level.

4 Scripting Capabilities

The CLI provides advanced scripting capabilities based on ZSH scripting language. The latter is feature-rich so that even the most complicated maintenance tasks, including multiple configuration actions on different MOs and tasks that require user input, can easily be coded in it.

This section provides a brief overview of the basic and most widely used functionalities available in ZSH scripting language. For additional information, refer to the ZSH manual (<http://zsh.sunsite.dk/Doc>) or ZSH user guide (<http://zsh.sunsite.dk/Guide>).

Note that it's also possible to utilize CLI functionality from scripts or programs written in other programming language. Refer to 'Using CLI Commands from PERL Scripts' on page 95 for a detailed explanation on how CLI functionality can be utilized from PERL scripts.

4.1 Simple CLI Script

Simple CLI script contains one or more regular CLI commands typed by the user at the command prompt. Each command is placed on a separate line. Comments can be added to the script to simplify maintenance and should be prefixed with symbol #.

```
# add new TP-1610 Media Gateway board in slot 8
addBoard tp1610 8 general

# set it's IP addresses
go /moGW/moBoard#8
modify IPAddress1=10.7.9.232
modify IPAddress1=10.7.9.233
```

Use the command `edit <script_name>` to edit the script (other text editors, e.g., `vi`, can be used as well). It is recommended to use extension `.zsh` for CLI scripts for consistency. When finished, type `execFromFile <script_name>` to run the script.

4.2 Variables

Variables are a widely-used concept of a symbolic name for a chunk of memory which users can assign values to, read and manipulate contents.

Variables are used as follows:

```
BOARD_NUM=8
IPADDR1="10.7.9.232"
modify /moGW/moBoard#{BOARD_NUM} IPAddress1=${IPADDR1}
```

Surround string values with double quotation marks to prevent misinterpretation of complex values (e.g., those that contain spaces). Note too that there should be no spaces around the `=` (equal) sign in the assignment operation – i.e., line `BOARD_NUM = 8` will not work.

To use a variable's value, put its name inside the construct `${ }`. In simple commands, prefixing the variable name with symbol `$` can also work (e.g., `BOARD_NUM`) – but for complex commands, it's advantageous to use full syntax.

4.2.1 Special Symbols in String Variables

Certain symbols – e.g., " (double quote), ' (single quote) and \ (backslash) – have special meaning within the string variable and therefore cannot be used as part of the latter. To use them anyway, prefix these symbols with a \ (backslash).

For example:

```
MESSAGE="This is a double quote \" and this is a  
backslash \\  
print $MESSAGE
```

4.2.2 String Operations

String operations are typically performed by simply appending the variable's value (either string or numeric) to some other string.

For example:

```
BOARD_NUM=8  
BOARD_NAME="moBoard#{BOARD_NUM} "  
TRUNK_NAME="$ {BOARD_NAME} /moTrunk#1 "
```

4.2.3 Arithmetic Operations

To perform arithmetic operations, surround the expression with double brackets.

Note that arithmetic operations are less sensitive to the exact command syntax and the = (equal) sign can be surrounded with spaces, contrary to the normal variable declaration/assignment.

For example:

```
BOARD_NUM=8  
print $BOARD_NUM  
( ( BOARD_NUM = $ {BOARD_NUM} + 1 ) )
```

4.2.4 Assigning Command Output to Variable

The output of a CLI command – e.g., 'show param' – can be assigned to the variable for future testing and manipulations. This can be done by surrounding the specific command in `` (backquotes) and putting it on the right side of the variable declaration/assignment structure.

For example:

```
ADMIN_STATE=`show param moTrunk#1 AdministrativeState -v`
```

Note that option **-v** of command **show param** is usually used to assign the parameter value only.

4.2.5 Parameters of the CLI Script

Special variables – **\$1**, **\$2** etc. – are assigned with the values of command line parameters of the CLI script. For example:

Sample script **lock_board.zsh**:

```
if [[ $1 == "" ]]; then
    print "You must specify board number"
else
    print "Lock board#$1"
    lock moBoard#$1
fi
```

When executing this script, the number of the board to be locked should be provided as a script parameter, e.g.:

```
client195 -- ~ -- /moGW
=> execFromFile lock_board.zsh 5
Lock board#5
```

4.3 Test Conditions

Test conditions are typically used as part of branch or loop commands and provide Boolean evaluation of a specific expression (i.e., having a value of yes/no). The following examples use the simplest form of IF branch (as described in the next section) to demonstrate the use of test conditions.

The simplest test condition looks as follows:

```
if [[ ${BOARD_NAME} == "moBoard#5" ]]; then
    lock ${BOARD_NAME}
fi

if [[ ${BOARD_NUM} -eq 5 ]]; then
    lock ${BOARD_NAME}
fi
```

The test operators differ for string and numeric variables/values.

For string comparison, use the following operators:

```
==      !=      <      >      <=     >=
```

For numeric comparison, use the following operators (corresponding to the above):

```
-eq     -ne     -lt     -gt     -le     -ge
```

Test conditions may be combined together with the help of **&&** (AND) or **||** (OR) operators to produce more complicated tests, e.g.:

```
if [[ ${BOARD_NUM} -eq 5 || ${BOARD_NUM} -eq 6 ]]; then
    lock ${BOARD_NAME}
fi
```

For string comparison, it can also be useful to search for the presence of a specific substring. To do this, use the expression **\${VAR[(I)STR]}**. The expression returns the first symbol in string variable **\$VAR** that matches the specific substring **STR**. If a match is not found, 0 (zero) is returned.

The expression result can be tested with regular numeric comparison operators as follows:

```
MONAME="moTrunk#1"
if [[ ${MONAME[(I)moMGC]} -ne 0 ]]; then
print "MO is moMGC"
elif [[ ${MONAME[(I)moTrunk]} -ne 0 ]]; then
print "MO is moTrunk"
fi
```

4.4 Branches

Branch commands allow altering the flow of CLI script based on certain conditions. They are a vital part of any complicated script and can be used to configure different system configurations via a single CLI script, to behave differently based on user input, or to verify the result of the previous command.

The simplest form of the IF branch looks as follows:

```
if [[ ${BOARD_NAME} == "moBoard#5" ]]; then
lock ${BOARD_NAME}
fi
```

Note the double square brackets around the test condition and the `;` (semicolon) symbol after the closing bracket.

More complicated IF branches can contain additional elements:

```
if [[ ${BOARD_NUM} -eq 5 || ${BOARD_NUM} -eq 6 ]]; then
lock moBoard#${BOARD_NUM}
elif [[ ${BOARD_NUM} -eq 7 ]]; then
unlock moBoard#${BOARD_NUM}
else
print "Invalid board number ${BOARD_NUM}"
fi
```

When a single variable must be tested for multiple values, a CASE branch can be used:

```
case ${BOARD_NUM} in
5|6)
lock moBoard#${BOARD_NUM}
;;
7)
unlock moBoard#${BOARD_NUM}
;;
*)
print "Invalid board number ${BOARD_NUM}"
;;
esac
```

Note the `;;` symbol that delimits between different CASE branches. Multiple values can be combined within a single branch by using delimiter `|`. The `*)` branch matches 'all other values'.

4.5 Loops

WHILE loop is the simplest way to iterate across multiple MOs. Use it as follows:

```
#
# unlock all boards in slots from 5 to 10
#
BOARD_NUM=5
while [[ ${BOARD_NUM} -le 10 ]]; do
  unlock moBoard#${BOARD_NUM}
  (( BOARD_NUM = ${BOARD_NUM} + 1 ))
done
```

To perform actions on a non-continuous values set, use the FOR loop as follows:

```
#
# unlock all trunks on all boards in slots 7, 11 and 14
#
for SLOT in 7 11 14; do
  TRUNK=1
  while [[ $TRUNK -le 16 ]];do
    unlock /moGW/moBoard#$SLOT/moTrunk#$TRUNK
    (( TRUNK=$TRUNK+1 ))
  done
done
```

4.6 Functions

ZSH scripting language supports function definitions. The syntax is simple:

```
unlock_board()
{
  unlock /moGW/moBoard#$1 y
}

unlock_board 6
unlock_board 7
```

Function arguments are stored in the positional variables – **\$1**, **\$2**, etc – and can be used inside the function body.

4.7 Getting Input from User

Interactive ZSH scripts typically need to get the values of certain variables from the user. This can be done in the following way:

```
print "Enter 1st IP address of Board#8"
read IPADDR1
modify /moGW/moBoard#8 IPAddress1=$IPADDR1
```

Use flag **-n** of the command **print** to prevent printing of the EOF (end of line) symbol.

4.8 Testing the Result of a CLI Command

Each CLI command can succeed or fail. Instead of parsing the command output, you can test the 'return code' of each CLI command that is stored in a special numeric ZSH variable `$?`. On successful completion of the CLI command, the 'return code' variable is set to 0 (zero) value. A value other than zero indicates failure.

```
unlock moBoard#6
if [[ $? -ne 0 ]]; then
    print "Failed to unlock board#6"
fi
```

4.9 Waiting for a CLI Command to Complete

Some CLI commands can take considerable time to complete. For example, when you unlock the media gateway board, the command **unlock** itself completes as soon as the **AdministrativeState** of the media gateway board is updated. However, it can take up to 3 minutes for the media gateway to actually return to service and correctly update its **OperationalState**.

Two alternative commands that can be used to wait for the completion of a CLI command.

```
sleep <timeout>
```

Waits for a predefined timeout; after completing the **sleep** command, it's recommended to test the state of the MO to verify that the expected state was in fact reached

```
waitParam [<MO Path>] <Param Name> <Value> <Timeout>
```

This is a more versatile and thus highly recommended alternative of a wait command; the software monitors the specified parameter and either waits until its value changes to the expected one or until the timeout expires; 'return code' of the command **waitParam** indicates whether the current MO state matches the expected one or not.

Use the **waitParam** command as follows:

```
unlock moBoard#6
waitParam moBoard#6 OperationalState 1 300
if [[ $? -ne 0 ]]; then
    print "ERROR: failed to unlock board#6"
fi
```

Note that the command **waitParam** returns immediately as the specified parameter's value changes to the expected one – therefore it's much more efficient than the command **sleep**.

4.10 Using CLI Commands from PERL Scripts

Users accustomed to other scripting or programming languages can use the programming language of their choice to write CLI scripts and only call gateway-specific CLI commands when required.

This section describes how CLI commands can be used from PERL script. However, any other programming language can use CLI commands in a similar way.

To execute a CLI command from the PERL script, use the following construct:

```
system("zsh -c '<CLI command>'")
```

For example:

```
my $BoardNum=8;
system("zsh -c 'show moBoard#$BoardNum conf -nd'");
```

Note that CLI commands are only valid in a ZSH environment so to run them, run executable **zsh** and pass the CLI command as a parameter to it. If the CLI command contains multiple words, it must be enclosed in quotation marks – either single or double.

Use of CLI commands from the PERL script has certain limitations, namely:

The command **go** can't practically be used because the 'current MO context' is lost after each **system()** call terminates; use the parameter <Mo Path> of the CLI commands to overcome this limitation

Performance is significantly worse than performance of the native ZSH scripts, because a separate instance of ZSH must be created and initialized for each command; put multiple CLI commands in a single line separated by semicolon (;) to overcome this limitation; consider using native ZSH scripts for massive CLI maintenance tasks.

Reader's Notes

A Appendix A – Supported MO Actions

Table 4-1: MOs, Actions and Action Syntax

MO Name	Action Name	Action Syntax
moGW	addFile	Add auxiliary file Syntax: action [<moGW Path>] addFile <File Name> <File Type> <Description> File Type: 2 -- call progress tone 3 -- pre-recorded tone 4 -- voice prompt 5 -- CAS file 6 -- XML or voice XML file 8 -- external coders 9 -- user information 10 -- alarm properties 11 -- X.509 private key 12 -- X.509 certificate 13 -- X.509 trusted root certificate 14 -- dial plan 16 -- alarm propagation rules 17 -- V5.2 user ports configuration file Note: the file must be placed in the /ftp directory prior to running the action.
	removeFile	Remove auxiliary file Syntax: action [<moGW Path>] removeFile <File Name>
moBoard	makeBoardRedundant	Configure VoP board to be a redundant board Syntax: action [<moBoard Path>] makeBoardRedundant <Board Number> <Board Number> - number of VoP board that configuration will be copied from
	makeBoardNonRedundant	Configure VoP board to be a normal (non-redundant) board Syntax: action [<moBoard Path>] makeBoardNonRedundant
	switchback	Perform activity switchback from redundant board back to normal mode Syntax: action [<moBoard Path>] switchback
	switchover	Perform activity switchover to redundant board Syntax: action [<moBoard Path>] switchover
	updateFk	Update board's feature key Syntax: action [<moBoard Path>] updateFk <Feature Key>

Table 4-1: MOs, Actions and Action Syntax

MO Name	Action Name	Action Syntax
	resetPm	Reset performance monitoring statistics Syntax: action [<moBoard Path>] resetPm
	startRecord	Start board debug recording Syntax: action [<moBoard Path>] startRecord [<TPM ID (1/2)>]
	stopRecord	Stop board debug recording Syntax: action [<moBoard Path>] stopRecord [<TPM ID (1/2)>]
	savelniFile	Get ini file from board. File is stored in the /ftp directory and is named CURRENT_SlotX_MASTER.ini. Syntax: action [<moBoard Path>] savelniFile
moCPUBoard	switchover	Perform activity switchover to redundant board Syntax: action [<moCPUBoard Path>] switchover
	cleanErrors	Clean errors on the SC board Syntax: action [<moCPUBoard Path>] cleanErrors <errorType> <errorType> : hardDiskError
moSwitchBoard	clearSeverity	Clear severity Syntax: action [<moSwitchBoard Path>] clearSeverity
	aline2me	Align all boards to this SwitchBoard Syntax: action [<moSwitchBoard Path>] aline2me
	changeMirrorPort	Enable/disable port mirroring Syntax: action [<moSwitchBoard Path>] changeMirrorPort {disable enable}
	mirroredToPort23	Send mirrored traffic to port 23 (mirror port) Syntax: action [<moSwitchBoard Path>] mirroredToPort23
	mirroredToRedSc	Send mirrored traffic to redundant SC board Syntax: action [<moSwitchBoard Path>] mirroredToRedSc
moSwitchBoardPort	changeMirrorMode	Change port mirror mode Syntax: action [<moSwitchBoardPort Path>] changeMirrorMode {nomirror ingress egress both}
moESMirrorFilter	activateInEs1	Activate the filter in 1 st ES board Syntax: action [<moESMirrorFilter Path>] activateInEs1
	activateInEs2	Activate the filter in 2 nd ES board Syntax: action [<moESMirrorFilter Path>] activateInEs2

Table 4-1: MOs, Actions and Action Syntax

MO Name	Action Name	Action Syntax
	deactivateInEs1	Deactivate the filter in 1 st ES board Syntax: action [<moESMirrorFilter Path>] deactivateInEs1
	deactivateInEs2	Deactivate the filter in 2 nd ES board Syntax: action [<moESMirrorFilter Path>] deactivateInEs2
moESBoard	aline2me	Align all boards to this SwitchBoard Syntax: action [<moESBoard Path>] aline2me
	changeMirrorPort	Enable/disable port mirroring Syntax: action [<moESBoard Path>] changeMirrorPort {disable enable}
	mirroredToPort10	Send mirrored traffic to port 10 (mirror port) Syntax: action [<moESBoard Path>] mirroredToPort10
	mirroredToRedSc	Send mirrored traffic to redundant SC board Syntax: action [<moESBoard Path>] mirroredToRedSc
moESBoardPort	changeMirrorMode	Change port mirror mode Syntax: action [<moESBoardPort Path>] changeMirrorMode {nomirror ingress egress both}
moMirrorFilter	activateInEs1	Activate the filter in 1 st ES board Syntax: action [<moMirrorFilter Path>] activateInEs1
	activateInEs2	Activate the filter in 2 nd ES board Syntax: action [<moMirrorFilter Path>] activateInEs2
	deactivateInEs1	Deactivate the filter in 1 st ES board Syntax: action [<moMirrorFilter Path>] deactivateInEs1
	deactivateInEs2	Deactivate the filter in 2 nd ES board Syntax: action [<moMirrorFilter Path>] deactivateInEs2
moSS7DataLink	deactivate	Deactivate MTP3 data link Syntax: action [<moSS7DataLink Path>] deactivate
	activate	Activate MTP3 data link Syntax: action [<moSS7DataLink Path>] activate
	inhibit	Inhibit MTP3 data link Syntax: action [<moSS7DataLink Path>] inhibit
	uninhibit	Uninhibit MTP3 data link Syntax: action [<moSS7DataLink Path>] uninhibit

Table 4-1: MOs, Actions and Action Syntax

MO Name	Action Name	Action Syntax
	block	Block MTP3 data link Syntax: action [<moSS7DataLink Path>] block
	unlock	Unlock MTP3 data link Syntax: action [<moSS7DataLink Path>] unlock
moSS7Linkset	deactivate	Deactivate SS7 Linkset Syntax: action [<moSS7Linkset Path>] deactivate
	activate	Activate SS7 Linkset Syntax: action [<moSS7Linkset Path>] activate
moSS7LinksetLink	dILock	Lock corresponding SS7 Data Link Syntax: action [<moSS7LinksetLink Path>] dILock
	dIUnlock	Unlock corresponding SS7 Data Link Syntax: action [<moSS7LinksetLink Path>] dIUnlock
	dIDeactivate	Deactivate corresponding SS7 Data Link Syntax: action [<moSS7LinksetLink Path>] dIDeactivate
	dIActivate	Activate corresponding SS7 Data Link Syntax: action [<moSS7LinksetLink Path>] dIActivate
	dIIInhibit	Inhibit corresponding SS7 Data Link Syntax: action [<moSS7LinksetLink Path>] dIIInhibit
	dIUninhibit	Uninhibit corresponding SS7 Data Link Syntax: action [<moSS7LinksetLink Path>] dIUninhibit
	dIBlock	Block corresponding SS7 Data Link Syntax: action [<moSS7LinksetLink Path>] dIBlock
	dIUnlock	Unlock corresponding SS7 Data Link Syntax: action [<moSS7LinksetLink Path>] dIUnlock
moSS7Node	deactivate	Deactivate SS7 Node Syntax: action [<moSS7Node Path>] deactivate
	activate	Activate SS7 Node Syntax: action [<moSS7Node Path>] activate
	recursiveUnlock	Recursively unlock all MTP3-related MOs associated with this SS7 Node If board is locked - mark these MOs to be unlocked during next board's unlock Syntax: action [<moSS7Node Path>] recursiveUnlock
moTrunk	startLoopback	Creates Loopback on Trunk Syntax: action [<moTrunk Path>] startLoopback
	stopLoopback	Removes Loopback from Trunk Syntax: action [<moTrunk Path>] stopLoopback

Table 4-1: MOs, Actions and Action Syntax

MO Name	Action Name	Action Syntax
	deactivate	Stop service on trunk (send AIS alarm) Syntax: action [<moTrunk Path>] deactivate
	activate	Restore service on trunk (clear AIS alarm) Syntax: action [<moTrunk Path>] activate
moV52LEInterface	offline	Set V5.2 interface to Offline state Syntax: action [<moV52LEInterface Path>] offline
	inService	Set V5.2 interface to InService state Syntax: action [<moV52LEInterface Path>] inService
	protectionSwitchOver	V5.2 interface protection switch over Syntax: action [<moV52LEInterface Path>] protectionSwitchOver
moV52LELink	block	Send BLOCK to Link Syntax: action [<moV52LELink Path>] block
	unlock	Send UNBLOCK to Link Syntax: action [<moV52LELink Path>] unlock
	linkIdCheck	Send Link ID Check Syntax: action [<moV52LELink Path>] linkIdCheck

Reader's Notes

B Appendix B – Supported Media Gateway Board's Debug Command

B.1 General Commands

Table 4-2: General Commands

Command	Short Format	Arguments	Description
/SHow INFO	/sh info	-	Displays device hardware information, versions, uptime, temperature reading and the last reset reason.
/MGmt/PERFormance	/mg/perf	basic control dsp net ds1 ss7 reset	Displays performance statistics. '/mg/perf reset' clears all statistics to zero.
/ControlProtocol/CallDetailReport	/cp/cdr	start show send stop	Generates CDR records when a voice call ends. Refer to subsection 'Call Detail Reports' for additional details.

B.2 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 subcommands are available:

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

Subcommand	Description
start [syslog file both]	<p>Starts generating CDR records.</p> <p>If 'syslog' is specified, the records are sent to the Syslog.</p> <p>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 command <code>/cp/cdr send</code>.</p> <p>If 'both' is specified, the records are sent to both the Syslog and the file. If no argument is specified, 'file' is assumed.</p>
show	<p>Displays current CDR file (history of last calls).</p> <p>Note that in a high-load system, the file is overwritten relatively quickly as it can hold approximately 1000 CDRs (possibly less than a minute of activity). Using the command <code>/cp/cdr show</code> can yield unpredictable results.</p>
send <nfs_location>	<p>Sends the CDR file to an NFS host. The remote NFS filesystem must be predefined and mounted (for detailed information on NFS support, refer to the IOM Manual).</p> <p>The argument to this command must be a URI in the form: <code>file://server-ip-address/path/filename</code></p> <p>Note that the URI is case-sensitive.</p>
stop	<p>Stops generation of CDR records and clears the CDR file.</p>

Reader's Notes

**Mediant 5000 and Mediant 8000
Command Line Interface (CLI) Reference Guide**

Version 6.6

Document #: LTRT-89210



www.audiocodes.com