



# APPENDIX **A**

## MML User Interface and Command Reference

---

Revised: April, 2010, OL-11616-08

### Introduction

This appendix provides information about Man-Machine Language (MML) command syntax and conventions, batch files, and procedures for starting and stopping MML sessions in the Cisco H.323 Signaling Interface (HSI) application. The appendix contains the following sections:

- [Starting an MML Command Session in the Cisco HSI, page A-1](#)
- [MML Commands, page A-2](#)
- [Starting an MML Session, page A-3](#)
- [Batch Files, page A-4](#)
- [MML Responses, page A-5](#)
- [MML Help, page A-6](#)
- [Quitting an MML Session, page A-6](#)
- [MML Command Reference, page A-6](#)

### Starting an MML Command Session in the Cisco HSI

To start an MML command session within the HSI environment, complete the following steps:

- 
- Step 1** Log in to Cisco HSI as mgcusr.
  - Step 2** Become superuser by typing `su <root password>`.
  - Step 3** To start the software, enter the following command:  
`/etc/init.d/CiscoGW start`
  - Step 4** Exit out of superuser.
  - Step 5** Type `mml` to start the MML command-line interpreter.

**Tip**

---

Press the **Tab** key twice to see a list of MML commands.

---

## MML Commands

To execute MML commands, log in to Cisco HSI and perform one of the following tasks:

- Start the MML session (see the [“Starting an MML Session”](#) section on page A-3) and enter a command.
- Type a batch file command to start an MML session (see the [“Starting a Batch File”](#) section on page A-4).

## MML Command Syntax

MML commands use the following syntax:

```
command_name:target:[Parameter_List][;comments]
```

Parameter\_List consists of a parameter name, an equal sign, and a value for each parameter.

The keywords and the value strings need not be enclosed in quotation marks. Anything you enter after a semicolon (;) is treated as a comment. Use only one MML command on each line.

The [“MML Command Reference”](#) section on page A-6 contains detailed information about the individual MML commands.

**Tip**

- 
- Use the **Up Arrow** key to scroll through all previous MML commands in turn.
  - Use the **Down Arrow** key to move forward in the command buffer.
  - Use the **Left** and **Right Arrow** keys to move along the command line.
  - Use the **Backspace**, **Delete**, and alphanumeric keys to edit an MML command.
- 

## MML Command Conventions

The MML commands use the conventions shown in [Table A-1](#).

**Table A-1 MML Command Conventions**

Convention	Meaning	Comments and Examples
Square brackets ([ ])	Optional elements	<b>command</b> [abc] abc is optional (not required), but you can choose it.
Vertical bars ( )	Separated alternative elements	<b>command</b> [abc   def] You can choose either abc or def, or neither, but not both.
Braces ({   })	Required choice of alternative elements	<b>command</b> {abc   def} You must use either abc or def, but not both.
Angle brackets (< >)	Symbol specifier	—

The MML commands can be interpreted and monitored through a network Transaction Language 1 (TL1) interface. The TL1 symbols shown in [Table A-2](#) are used in MML.

**Table A-2 TL1 Symbols Used in MML**

Symbol	Description
:	A parameter separator.
::	An empty parameter block.
&	Groupss arguments together so that one parameter can convey several arguments.
;	End of command (optional). Anything on the same line after this symbol is treated as a comment.

## Case Sensitivity

Command names and parameter names are not case sensitive. You can enter commands and parameters in either upper- or lowercase. Filenames *are* case sensitive when they are used as arguments in MML commands.

## Starting an MML Session

To start an MML session, complete the following steps:

- 
- Step 1** Log in to Cisco HSI.
- Step 2** Type one of the following commands at the prompt:
- **mml**
  - **mml -b batchfile** (see the [“Starting a Batch File”](#) section on page A-4)
- 

The following example shows the start of an MML session:

```

user@host> mml

Welcome to the Cisco H.323 Signaling Interface.

gw mml>

```

## Batch Files

The Cisco HSI application supports the use of batch files. You can create an ASCII file of MML provisioning commands for use as a batch file. You can also use a script file. When the commands are read, the Cisco HSI executes them sequentially.

The following is an example of an MML provisioning batch file:

```

prov-sta::srcver="new",dstver="first"
prov-add:name="sys_config",nodeid="H323-GW1"
prov-add:name="h323_sys",messages=30000,channels=5000
prov-add:name="ras",responsetimeout=10,allowcallswhennonreg=1
prov-add:name="ras",terminaltype="gateway",timetolive=900
prov-add:name="q931",reponsetimeout=20,connecttimeout=20,maxcalls=5000
prov-cpy

```

The **prov-sta** command establishes a provisioning session. The **prov-cpy** command copies configuration settings from the current provisioning session to the Cisco HSI and activates the configuration. If the command is successful, it also terminates the current provisioning session. If you are not ready to commit a session, use the **prov-stp** command to save and stop the provisioning session.

The application provides a log function (**diaglog** command) that records the MML commands and responses in a log file.

In the MML batch file, you can place a **diaglog** command at the beginning to start logging and a **diaglog** command at the end to stop logging. For more information about the **diaglog** command, see the “[MML Command Reference](#)” section on page A-6

For commands executed in both the process manager and the application, the application logs the user ID, the login date and time, and the name of each command that is executed in batch mode to the `mml_batch_log` file.



### Note

---

Batch files can be defined for complete systems or for the modification of an existing system.

---

## Creating a Batch File

To create a batch file, use an ASCII text editor program. Each command should be on a separate line.

## Starting a Batch File

To start executing a batch file, type **mml -b *batchfilename*** at the UNIX prompt.

After you enter the batch file command, the application displays the result of each MML command as it is executed. Each command and its results are saved in the `mml.log` file. When the batch file is completed, the MML session is ready to accept user commands.

The following example shows the start of a batch file named `nolog.bat` with these contents:

```
prov-sta:srcver=active,dstver=nolog
prov-ed:name=logging,eisup=0x0000
prov-cpy
```

and this output:

```
gp-capetown-16-> mml -b nolog.bat
Starting in batch mode.
Connecting to port 10129 on host gp-capetown

Welcome to the Cisco H.323 Signaling Interface.
gw mml> gw mml> prov-sta:srcver=active,dstver=nolog
H323 Signaling Interface Tue Jan 22 05:57:12 2002
M SUCC
Successfully started provisioning session "nolog" from "active".Note: This provisioning
session has not been verified.
gw mml> prov-ed:name=logging,eisup=0x0000
H323 Signaling Interface Tue Jan 22 05:57:12 2002
M SUCC
Successfully edited provisioning element(s):
MML Name : logging.
Parameter: EISUP.
Value : 0x0000.
gw mml> prov-cpy
H323 Signalling Gateway Tue Jan 22 05:57:13 2002
M SUCC
Successfully activated provisioning session nolog.
gw mml>
```

## MML Responses

The following sections describe the two types of response messages that are displayed by the MML user interface:

- Status messages
- Error messages

## Status Messages

[Table A-3](#) lists the MML status messages and their descriptions.

**Table A-3 MML Status Messages and Descriptions**

Status	Message	Description
RTRV	Retrieve	Retrieve and display the contents of the specified file
SUCC	Successful	Successful completion

## Error Messages

If an MML command does not work properly, an error message is displayed. [Table A-4](#) lists the MML error messages and their descriptions.

Table A-4 MML Error Messages and Descriptions

Error Message	Definition	Description
DENY	Command denied	The system recognizes the command but does not perform the requested function.
ICNV	Input command not valid	The system does not recognize the MML command.
IDNV	Input data not valid	An unknown parameter was entered.
IISP	Input syntax error	Incorrect syntax was entered.
IITA	Invalid target	The system cannot perform the requested operation on the specified component, or the component does not exist.
IPRM	Input parameter missing	An expected parameter was not entered.
SABT	Status abort	The requested operation did not complete within the allotted time.
SNVS	Component not in valid state	The requested operation failed because the component is either not configured to accept the operation or the component is already in the desired state.
SNSP	State not supported	The operation is not supported by the component.
SROF	Status requested operation failed	The requested operation failed.

## MML Help

MML has an online help feature. The MML **help** command displays a list of valid system commands and an explanation of each command's use. To display the online help, start an MML session and type **help** at the command line prompt. See the [“help” section on page A-11](#).

## Quitting an MML Session

To quit an MML session, type **quit** at the prompt.

## MML Command Reference

This section describes the following MML commands:

- [ack-alm](#)
- [clr-alm](#)
- [clr-meas](#)
- [diaglog](#)
- [h](#)

- [help](#)
- [prov-add](#)
- [prov-cpy](#)
- [prov-dlt](#)
- [prov-ed](#)
- [prov-exp](#)
- [prov-rtrv](#)
- [prov-sta](#)
- [prov-stp](#)
- [quit](#)
- [radlog](#)
- [restart-softw](#)
- [rtrv-alms](#)
- [rtrv-calls](#)
- [rrtrv-config](#)
- [rtrv-ctr](#)
- [rtrv-dest](#)
- [rtrv-gapping](#)
- [rtrv-log](#)
- [rtrv-lics](#)
- [rtrv-mml](#)
- [rtrv-ne](#)
- [rtrv-ne-health](#)
- [rtrv-overload](#)
- [rtrv-softw](#)
- [set-dest-state](#)
- [set-gapping](#)
- [set-log](#)
- [set-overload](#)
- [set-tos](#)
- [sta-callproc](#)
- [sta-softw](#)
- [stp-call](#)
- [stp-callproc](#)
- [stp-softw](#)

## ack-alm

The **ack-alm** command acknowledges that an alarm event is recognized but does not clear the alarm.

**ack-alm:event=alarm event**

### Syntax Description

<i>alarm event</i>	The alarm category or the text that appears in the body of the alarm. Alarm event names are defined in <a href="#">Chapter 5, “Troubleshooting Cisco HSI Alarms.”</a>
--------------------	---

### Usage Guidelines

The format of the alarm category name must be the same as the format of the alarm category name that the **rtrv-arms** command displays. It is case sensitive.

### Examples

In this example, the **VSC\_FAILURE** alarm event is recognized, but the alarm is not cleared:

```

gw mml> ack-alm:event=VSC_FAILURE

GW Signaling Interface      2000-12-05 14:19:22
M   SUCC

mml>

```

### Related Commands

Command	Description
<b>clr-alm</b>	Clears an alarm event.
<b>rtrv-arms</b>	Retrieves all active alarms.

## clr-alm

The **clr-alm** command clears an alarm event.

**clr-alm:event=alarm event**

### Syntax Description

<i>alarm event</i>	The alarm event name or the text that appears in the body of the alarm. Alarm names are defined in <a href="#">Chapter 5, “Troubleshooting Cisco HSI Alarms.”</a>
--------------------	---

### Usage Guidelines

The alarm category must match the format shown in the alarm when the **rtrv-arms** command displays the alarm. It is case sensitive.

**Examples**

In this example, the alarm event **VSC\_FAILURE** is cleared.

```
gw mml> clr-alm:event=VSC_FAILURE

GW Signaling Interface    2000-12-05 14:19:22
M   SUCC

mml>
```

**Related Commands**

Command	Description
<b>ack-alm</b>	Acknowledges that an alarm event is recognized but does not clear the alarm.
<b>rtrv-arms</b>	Displays all active alarms.

## clr-meas

The **clr-meas** command resets a measurement counter.

**clr-meas:counter group:name=measurement name**

**clr-meas:counter group**

**Syntax Description**

<i>counter group</i>	Valid counter groups are: <ul style="list-style-type: none"> <li>• RAS</li> <li>• Q931</li> <li>• H245</li> </ul>
<i>measurement name</i>	For a list of measurement names, see <a href="#">Table 4-1</a> , <a href="#">Table 4-2</a> , and <a href="#">Table 4-3</a> .

**Examples**

In this example, a measurement counter, **GK\_DISC\_ATT\_TOT** (Gatekeeper Discovery Attempts), is reset in the counter group **RAS**:

```
gw mml> clr-meas:RAS

GW Signaling Interface    2000-12-05 14:19:22
M   SUCC

mml>

mml> clr-meas:RAS:name=GK_DISC_ATT_TOT

GW Signaling Interface    2000-12-05 14:19:22
M   SUCC
```

**Related Commands**

Command	Description
<b>rtrv-ctr</b>	Displays the measurements for a counter group.

# diaglog

The **diaglog** command starts and stops event logging in a diagnostics log.

**diaglog:***file name*:start | stop

Syntax Description	<i>file name</i>	The user-defined name of the log file. The actual file name has a .log suffix applied. The file is located in the logging directory defined in the configuration data (see <a href="#">Chapter 3, “Provisioning the Cisco HSI”</a> ).
--------------------	------------------	---

## Examples

In this example, event logging is started in a diagnostics log named **test5**:

```
gw mml> diaglog:test5:start
test5_davek15823_20010130053323.log
```

In the preceding example, davek is the user who runs the command, and 15823 is the process ID of the MML process from which the command is run.

## Related Commands

Command	Description
<b>radlog</b>	Starts and stops RADVision logging into a specified log file.
<b>rtrv-log</b>	Displays the logging level of a package or all packages.
<b>set-log</b>	Sets the logging level of a package or all packages.

# h

The **h** command redisplay a command or a series of commands. The selection of items to be displayed depends on a specified number or range. If no number is specified, only the last command is displayed.

**h**[::start=*number*[,end=*number*]]

Syntax Description	start	Entered as a number; specifies the first command to redisplay.
	end	Entered as a number; specifies the end of the range of commands to redisplay.

## Examples

The MML command in the following example displays the last successful command entered:

```
mml> h
VSC H-323 Signaling Interface - H323-GW1 2000-06-20 10:04:28
M RTRV
  "rtrv-log:all"
/* command 1 */
```

The MML command in the following example displays the third from the last successful command entered:

```
mml> h::3
VSC H-323 Signaling Interface - H323-GW1 2000-06-20 10:04:28
M RTRV
  "rtrv-ne"
  /* command 3 */
```

The MML command in the following example displays the last and second to last commands entered:

```
mml> h::start=1,end=2
VSC H-323 Signaling Interface - H323-GW1 2000-06-20 10:04:28
M RTRV
  "rtrv-log:all"
  /* command 1 */
  "rtrv-ne"
  /* command 2 */
```

## help

The **help** command displays a list of valid system commands and an explanation of their use. If you do not enter a command name as a parameter, the **help** command provides a list of MML commands, descriptions, and values. If you enter a command name as a parameter, a description of that command displays.

**help**[:*command name*]

### Syntax Description

<i>command name</i>	The name of the MML command.
---------------------	------------------------------

### Examples

The command shown in the following example displays help for a specific command:

```
mml> help:rtrv-ctr
H323 Signalling Gateway   Tue Feb 12 19:09:58 2002
M SUCC

          RTRV-CTR -- Display the measurements for a counter group
          -----

Purpose:   This MML command displays a measurement counter for a counter group

Format:    rtrv-ctr:<counter group>

Description: * counter group -- The name of an MML counter group (RAS, Q931 or H245)

Example:   The MML command shown in the following example displays measurement
          counters for the counter group RAS.
          mml> rtrv-ctr:ras;
          GW Signalling GateWay 2000-12--5 14:19:32
          M RTRV
          "H323-GW1:GROUP=RAS,NAME=\"GK_DISC_ATT_TOT\",VAL=1000"
          "H323-GW1:GROUP=RAS,NAME=\"GK_REG_ATT_TOT\",VAL=1000"
          "H323-GW1:GROUP=RAS,NAME=\"GK_REG_SUCC_TOT\",VAL=1000"

          mml>
```

If you enter the **help** command without a parameter, the help file displays information about all available commands. The following example shows a portion of the help file that displays if you do not enter a parameter:

```
mml> help

VSC H323 signaling interface - H323-GW1 2000-06-20 10:04:28
M RTRV
Available commands (in alphabetical order):
ack-alm:"<alm cat>" Acknowledges an alarm category on a component
clr-alm:"<alm cat>" Clears an alarm category on a component
clr-meas:"<meas cat>" Resets a measurement category on a component
diaglog:<file name>:START|STOP Starts/stops diagnostics log
h[:<number>[,<number>]] Displays a history of commands for a specified backward number or
range; the last command by default
help[:<command name>] Displays the list of MML commands or the help information on a
specified command
prov-add:name=<MML name>,<param name>=<param value>,... Adds the component
prov-cpy Commits provisioning data
prov-dlt:name=<MML name> Deletes the component
```

## prov-add

The **prov-add** command adds a component to the Cisco HSI.

**prov-add:name=MML name,param name=param value,...**

### Syntax Description

<i>MML name</i>	MML name for the element you are adding. Valid MML names are: <ul style="list-style-type: none"> <li>• sys_config_static</li> <li>• sys_config_dynamic</li> <li>• h323_sys</li> <li>• ras</li> <li>• h245</li> <li>• q931</li> </ul>
<i>param name</i>	The name of a valid configuration parameter for the specified parameter name.
<i>param value</i>	The value you want to assign to the parameter.

### Usage Guidelines

To define more than one parameter, enter additional *param name=param value* descriptions on the command line. See [Chapter 3, "Provisioning the Cisco HSI,"](#) for the list of MML names, parameter names, and their associated values.

**Examples**

The command shown in the following example adds a provisioning element with the MML name `ras`, parameter name `maxFail`, and value `3`:

```
gw mml> prov-add:name=ras,maxfail=3
H323 Signaling Interface Sun Jan 7 15:15:02 2001
M SUCC
Successfully added provisioning element(s):
MML Name : ras.
Parameter: maxFail.
Value : 3.
```

**Related Commands**

Command	Description
<b>prov-cpy</b>	Activates the configuration settings in the current provisioning session.
<b>prov-dlt</b>	Deletes a provisioned component.
<b>prov-ed</b>	Modifies a provisioned component.
<b>prov-exp</b>	Exports the current configuration of the Cisco HSI in MML command form to a file or files.
<b>prov-rtrv</b>	Retrieves information about an existing provisioning session.
<b>prov-sta</b>	Establishes a provisioning session.
<b>prov-stp</b>	Terminates either a specified provisioning session or the current provisioning session.

**prov-cpy**

The **prov-cpy** command activates the current provisioning session. If any client-level parsing fails during the **prov-cpy** command, the system might prompt for confirmation to force the HSI to activate the configuration (**prov-cpy[:confirm]**). However, we recommend that you never use **prov-cpy:confirm** unless you are asked to do so by Cisco.

If client-level parsing fails, there is a severe error within the user configuration.

If the **prov-cpy** command fails, use the failure description and the configuration changes to determine the error. Correct the configuration and reissue the **prov-cpy** command.

Please contact the Cisco TAC if you require assistance with the configuration.

**Syntax Description**

<b>confirm</b>	If any client-level parsing fails during the data session, a <b>confirm</b> is needed for the data to be activated.
----------------	---

**Examples**

The command shown in the following example copies the configuration changes from the current session to the Cisco HSI:

```
gw mml> prov-cpy
H323 Signaling Interface Sun Jan 7 13:53:42 2001
M SUCC
Successfully activated the provisioning session.
```

**Usage Guidelines**

See [Chapter 3, “Provisioning the Cisco HSI,”](#) for a list of MML names, parameter names, and their associated values.

**Related Commands**

Command	Description
<b>prov-add</b>	Adds a component.
<b>prov-dlt</b>	Deletes a provisioned component.
<b>prov-ed</b>	Modifies a provisioned component.
<b>prov-exp</b>	Exports the current configuration of the Cisco HSI in MML command form to a file or files.
<b>prov-rtrv</b>	Retrieves information about an existing provisioning session.
<b>prov-sta</b>	Establishes a provisioning session.
<b>prov-stp</b>	Terminates either a specified provisioning session or the current provisioning session.

## prov-dlt

The **prov-dlt** command deletes a provisioned component. It allows you to delete a parameter rather than deleting the MML group.

**prov-dlt:name=MML name**

**prov-dlt:name=MML name,param=param name**

**prov-dlt:name=MML name param name**

**Syntax Description**

<i>MML name</i>	MML name for the element you are deleting. Valid MML names are: <ul style="list-style-type: none"> <li>• sys_config_static</li> <li>• sys_config_dynamic</li> <li>• h323_sys</li> <li>• ras</li> <li>• h245</li> <li>• q931</li> </ul>
<i>param name</i>	The name of a valid configuration parameter for the specified parameter name.

**Usage Guidelines**

See [Chapter 3, “Provisioning the Cisco HSI,”](#) for a list of MML names, parameter names, and their associated values.

**Examples**

The MML command in the following example deletes the ras element:

```
gw mml> prov-dlt:name=ras
H323 Signaling Interface Sun Jan 7 14:13:05 2001
M SUCC
Successfully deleted provisioning data for ras
```

The MML command in the following example deletes the maxCalls parameter of the ras element:

```
gw mml> prov-dlt:name=ras,param=maxCalls
gw mml> prov-dlt:name=ras,maxCalls
H323 Signaling Interface Sun Jan 7 14:46:01 2001
M SUCC
Successfully deleted provisioning data for ras:maxCalls
```

**Related Commands**

Command	Description
<b>prov-add</b>	Adds a component.
<b>prov-cpy</b>	Activates the configuration settings in the current provisioning session.
<b>prov-ed</b>	Modifies a provisioned component.
<b>prov-exp</b>	Exports the current configuration of the Cisco HSI in MML command form to a file or files.
<b>prov-rtrv</b>	Retrieves information about an existing provisioning session.
<b>prov-sta</b>	Establishes a provisioning session.
<b>prov-stp</b>	Terminates either a specified provisioning session or the current provisioning session.

## prov-ed

The **prov-ed** command modifies a provisioned component.

**prov-ed:name=MML name,param name=param value,...**



**Note** Enter only those parameters that must be modified.

**Syntax Description**

<i>MML name</i>	MML name for the element you are modifying. Valid MML names are: <ul style="list-style-type: none"> <li>• sys_config_static</li> <li>• sys_config_dynamic</li> <li>• h323_sys</li> <li>• ras</li> <li>• h245</li> <li>• q931</li> </ul>
-----------------	---

<i>param name</i>	The name of a valid configuration parameter for the specified parameter name.
<i>param value</i>	The value you want to assign to the parameter.

**Usage Guidelines**

To change more than one parameter, enter additional *param name=value* descriptions on the command line. See [Chapter 3, “Provisioning the Cisco HSI,”](#) for a list of MML names, parameter names, and their associated values.

**Examples**

Use the MML command shown in the following example to edit a provisioning element with the MML name ras, parameter name maxFail, and value 7:

```
gw mml> prov-ed:name=ras,maxfail=7
H323 Signaling Interface Sun Jan 7 15:22:02 2001
M SUCC
Successfully edited provisioning element(s):
MML Name : ras.
Parameter: maxFail.
Value : 7.
```

**Related Commands**

Command	Description
<b>prov-add</b>	Adds a component.
<b>prov-cpy</b>	Activates the configuration settings in the current provisioning session.
<b>prov-dlt</b>	Deletes a provisioned component.
<b>prov-exp</b>	Exports the current configuration of the Cisco HSI in MML command form to a file or files.
<b>prov-rtrv</b>	Retrieves information about an existing provisioning session.
<b>prov-sta</b>	Establishes a provisioning session.
<b>prov-stp</b>	Terminates either a specified provisioning session or the current provisioning session.

## prov-exp

The **prov-exp** command exports the current provisioned configuration of the Cisco HSI in MML command form to a file. With this configuration file, you can prime a system with a cloned configuration from an existing system. It also enables you to restore a baseline configuration to a system. You can use the MML batch feature to import the exported data.

Start a dummy provisioning session with the **prov-sta** command before you use the **prov-exp** command.

```
prov-sta:srcver=active, dstver=dummy1
```

```
prov-exp:dirname=export directory name
```

```
prov-stp
```

<b>Syntax Description</b>	<i>export directory name</i>	Name of the directory to which the data is exported. This directory is a subdirectory within the /opt/GoldWing/export directory established at installation.
---------------------------	------------------------------	--

**Examples**

The MML command shown in the following example saves the active file as config.mml to the export/uk9/ directory:

```
gw mml> prov-exp:dirname=uk9
H323 Signaling Interface Sun Jan 7 14:29:11 2001
M SUCC
Successfully exported "active" to export/uk9/config.mml
```

The UNIX command shown in the following example executes MML in batch mode and imports the configuration file that was exported in the previous example:

```
mml> -b /opt/GoldWing/currentGW/export/uk9/config.mml
```

**Related Commands**

Command	Description
<b>prov-add</b>	Adds a component.
<b>prov-cpy</b>	Activates the configuration settings in the current provisioning session.
<b>prov-dlt</b>	Deletes a provisioned component.
<b>prov-ed</b>	Modifies a provisioned component.
<b>prov-rtrv</b>	Retrieves information about an existing provisioning session.
<b>prov-sta</b>	Establishes a provisioning session.
<b>prov-stp</b>	Terminates either a specified provisioning session or the current provisioning session.

## prov-rtrv

The **prov-rtrv** command retrieves information about an existing provisioning session.

**Note**

This command is retained for backward compatibility. The recommended command is **rtrv-config**, which does not need to be entered within a provisioning session. (See the “[rtrv-config](#)” section on page A-25.)

**prov-rtrv:name=MML name**

**prov-rtrv:all**

**prov-rtrv:session**

**prov-rtrv:list**

**Note**

The **prov-rtrv:list** command is the only **prov-rtrv** command that can be executed outside of a provisioning session. Use the **prov-sta** command to start a provisioning session.

**Syntax Description**

<b>name</b>	The MML name for the elements that you want to display.
<i>MML name</i>	The MML component name for the component you want to display. Valid MML component names are: <ul style="list-style-type: none"> <li>• sys_config_static</li> <li>• sys_config_dynamic</li> <li>• h323_sys</li> <li>• ras</li> <li>• h245</li> <li>• q931</li> </ul>
<b>all</b>	Displays all components that have been provisioned.
<b>session</b>	Displays information about the provisioning session.
<b>list</b>	Provides a list of possible session names that you can use as the srcver parameter to prov-sta:srcver=uk9,dstver=uk10.

**Usage Guidelines**

See [Chapter 3, “Provisioning the Cisco HSI,”](#) for a list of MML names, parameter names, and their associated values.

**Examples**

The **prov-rtrv** command shown in the following example displays information about the MML name ras:

```
gw mml> prov-rtrv:name=ras
H323 Signaling Interface Sun Jan 7 14:46:01 2001
M SUCC
MML Name : ras.
Parameter: maxFail.
Value : 33.
```

The MML command shown in the following example displays information about the MML session:

```
gw mml> prov-rtrv:session
H323 Signaling Interface Sun Jan 7 14:46:01 2001
M RTRV
Session ID = mml 6 | davek
SRCVER = uk9
DSTVER = inter
```

```
gw mml> prov-rtrv:list
H323 Signaling Interface Sun Jan 7 14:46:01 2001
M RTRV
```

```
The following provisioning sessions are available:
uk9 matt inter
gw mml>
```

**Related Commands**

Command	Description
<b>prov-add</b>	Adds a component.
<b>prov-cpy</b>	Activates the configuration settings in the current provisioning session.

Command	Description
<b>prov-dlt</b>	Deletes a provisioned component.
<b>prov-ed</b>	Modifies a provisioned component.
<b>prov-exp</b>	Exports the current configuration of the Cisco HSI in MML command form to a file or files.
<b>prov-sta</b>	Establishes a provisioning session.
<b>prov-stp</b>	Terminates either a specified provisioning session or the current provisioning session.

## prov-sta

The **prov-sta** command establishes a provisioning session. The data files are copied from the source version to the destination version.

**prov-sta::srcver=version,dstver=version**

### Syntax Description

<b>srcver=version</b>	<p>Selects a specific configuration version as the source for configuration changes. The srcver variable represents a directory that exists in \$GWBASE/var/prov/. In place of the configuration version, you can also enter:</p> <ul style="list-style-type: none"> <li>new—Specifies a new default session configuration; no existing source configuration is used.</li> </ul> <p><b>Note</b> Only MML batch files should contain <b>srcver=new</b> because it implies that the entire configuration is being entered (that is, the MML batch file in this case would contain approximately 200 prov-add entries). Typically, <b>srcver=new</b> appears only in MML batch files that are auto-generated through the use of the <b>prov-exp</b> command. If you configure manually using the MML command line, either <b>srcver</b> should be set to active (which implies that the user is going to configure a delta, using <b>prov-add</b>, <b>prov-ed</b>, <b>prov-del</b> in the current configuration) or <b>srcver</b> should be set to an existing configuration name (which implies that you wish to configure a delta on an existing configuration using <b>prov-add</b>, <b>prov-ed</b>, <b>prov-del</b>).</p> <ul style="list-style-type: none"> <li>active—Selects the active configuration as the source for configuration changes.</li> </ul>
<b>dstver=version</b>	<p>Specifies the output version directory for the configuration session results. The dstver variable represents a directory stored in \$GWBASE/var/prov/.</p>

### Usage Guidelines

If the source and destination filenames are the same, the new configuration overwrites the old configuration. It is a good practice to copy an existing configuration instead of overwriting it so that you can return to a known configuration if there are problems with the new one.

If the source provisioning session has not been verified, the message “Note: This provisioning session has not been verified” is displayed, but the session starts normally.

If you try to start with a provisioning session that does not exist, an error message is displayed, along with a list of available sessions.

### Examples

The MML command in the following example starts a provisioning session named **nowt** and creates a new configuration named **blah**:

```
gw mml> prov-sta::srcver=nowt,dstver=blah
H323 Signaling Interface Sun Jan 7 13:32:07 2000
M DENY
The provisioning session called "nowt" does not exist.
The following configurations are available:
sanfran2 uk9 final
telco mgcpvia miki
transit dave matt
```

The MML command in the following example starts a provisioning session and creates a new configuration named **ver1**:

```
gw mml> PROV-STA::SRCVER="new", DSTVER="ver1"
H323 Signaling Interface Sun Jan 7 13:32:07 2001
M SUCC
Successfully started provisioning session "ver1" from "new".
```

The MML command in the following example starts a provisioning session, opens the existing configuration named **ver1**, and overwrites that configuration:

```
gw mml> PROV-STA::SRCVER="ver1", DSTVER="ver1"
```

The MML command shown in the following example starts a provisioning session, opens the existing configuration named **ver1**, and saves the updated configuration as **ver2**:

```
gw mml> PROV-STA::SRCVER="ver1", DSTVER="ver2"
```

### Related Commands

Command	Description
<b>prov-add</b>	Adds a component.
<b>prov-cpy</b>	Activates the configuration settings in the current provisioning session.
<b>prov-dlt</b>	Deletes a provisioned component.
<b>prov-ed</b>	Modifies a provisioned component.
<b>prov-exp</b>	Exports the current configuration of the Cisco HSI in MML command form to a file or files.
<b>prov-rtrv</b>	Retrieves information about an existing provisioning session.
<b>prov-stp</b>	Terminates either a specified provisioning session or the current provisioning session.

## prov-stp

The **prov-stp** command terminates the provisioning session and saves the configuration.

**prov-stp:confirm**

**prov-stp:session name:confirm**

**Syntax Description**

<b>session name</b>	Use the <b>rtrv-mml</b> command to retrieve the MML name given to the MML process that started the provisioning session.
<b>confirm</b>	If no confirm option is entered, the command is rejected and a message notifies you of the potential performance impact of this command.

**Usage Guidelines**

You can use the name given to an MML session to stop a provisioning session. Each MML session (not Telnet) gets an MML name: for example, mml1 or mml2. The maximum number of allowable MML sessions is 12.

If you log in to the Cisco HSI from an MML session and start a provisioning session (for example, gw mml> **prov-sta:srcver=new,dstver=uk9**), you can use the MML name (for example, mml2) to stop the session with **prov-stp** (for example, **prov-stp:mml2:confirm**).

Use the **rtrv-mml** command to display all active MML sessions (see [rtrv-mml, page A-29](#)).

**Examples**

The MML command in the following example terminates the current provisioning session:

```
gw mml> prov-stp:confirm
H323 Signaling Interface Sun Jan 7 14:46:01 2001
M SUCC
Successfully stopped provisioning session "ver1"
```

The MML command in the following example terminates the uk9 provisioning session:

```
gw mml> prov-stp:uk9:confirm
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
Successfully stopped provisioning session "uk9"
gw mml>
```

If the previous session starts from an MML process assigned the name mml2, you can use the following MML command:

```
gw mml> prov-stp:mml2:confirm
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
Successfully stopped provisioning session "uk9"
gw mml>
```

**Related Commands**

Command	Description
<b>prov-add</b>	Adds a component.
<b>prov-cpy</b>	Activates the configuration settings in the current provisioning session.
<b>prov-dlt</b>	Deletes a provisioned component.
<b>prov-ed</b>	Modifies a provisioned component.
<b>prov-exp</b>	Exports the current configuration of the Cisco HSI in MML command form to a file or files.
<b>prov-rtrv</b>	Retrieves information about an existing provisioning session.
<b>prov-sta</b>	Establishes a provisioning session.

# quit

The **quit** command ends an MML session.

**quit**

## Syntax Description

This command has no arguments or keywords.

## Examples

The command in the following example ends an MML session.

```
gw mml> quit
```

# radlog

The **radlog** command starts or stops RADVision logging into a specified log file.

**radlog:[file name]:start | stop**



### Caution

This command is processor intensive and results in very large log files. Use this command only to retrieve information for single test calls, and do not use it on a live network that is processing numerous calls.

## Syntax Description.

<i>file name</i>	The user-defined name of the log file. The actual filename has a <i>.log</i> suffix. The file is located in the logging directory defined in the configuration data (see <a href="#">Chapter 3, “Provisioning the Cisco HSI”</a> ).
------------------	---

## Examples

The command in the following example starts logging into a diagnostics log named file1:

```
gw mml> radlog:file1:start
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
RADLogging requested to start
```

The following command logs RADVision to the standard log file:

```
gw mml> radlog::start
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
RADLogging to standard log file
```

## Related Commands

Command	Description
<b>diaglog</b>	Starts or stops event logging into a diagnostics log.
<b>rtrv-log</b>	Displays the logging level of a package or all packages.
<b>set-log</b>	Sets the logging level of a package or all packages.

# restart-softw

The **restart-softw** command restarts the call processing application. It applies the provisioning data specified in the *configVersion* (if present) that overrides the existing active provisioning data.

```
restart-softw[:configVersion][:confirm]
```

## Syntax Description

<i>configVersion</i>	In <i>configVersion</i> , <b>init</b> is a keyword, and this command restarts the call processing application, applying the <i>etc/GWmain.conf</i> configuration file as the provisioning data. If <i>configVersion</i> is an unverified provisioning session, the command fails.
<i>confirm</i>	If there are active calls, a notification is sent to the craft, and the command must be reentered with the <i>confirm</i> parameter to take effect.

## Examples

In the following example, the call processing application restarts using the *etc/GWmain.conf* configuration files as the provisioning data:

```
gw mml> restart-softw:init
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
```

```
Application is now restarting using the default provisioning session.
There are no active calls.
New call requests are rejected.
Call Processing now stopped.
Application will restart in 60 seconds
```

In the following example, the application would restart using the active provisioning session. There are no active calls, new call requests are rejected, and call processing is now stopped. The application is set to restart in 12 seconds.

```
gw mml> restart-softw
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
```

In the following example, a restart passes an unverified provisioning session. The command fails. You cannot use an unverified provisioning session.

```
gw mml> restart-softw:config2
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M DENY
```

In the following example, a restart uses a specified verified provisioning session. The application restarts by using *original* as the provisioning session. There are no active calls, new call requests are rejected, and call processing is now stopped. The application is set to restart in 12 seconds.

```
gw mml> restart-softw:original
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
```

# rtrv-alm

The **rtrv-alm** command retrieves all active alarms.

**rtrv-alm**

**rtrv-alm:cont**

## Syntax Description

<b>cont</b>	This parameter displays alarm events until you press <b>Ctrl-C</b> . All active alarms are displayed, and then a message appears (for example: “/* Listening for alarm events . . . (Ctrl-C to stop) */”).
-------------	--

## Examples

In the following example, the output contains the standard alarm definition and also a NACK or an ACK for noninformational alarms. NACK and ACK indicate the acknowledgment status of the alarm.

```
gw mml> rtrv-alm
GW Signaling Interface    2000-12-05 14:19:22
M   RTRV
"H323-GW1: 2000-11-27 11:25:12.259, ** ALM=\"VSC FAILURE\",SEV=MJ" NACK
"H323-GW1: 2000-11-27 11:25:13.260, ** ALM=\"CONFIGURATION FAILURE\",SEV=MJ" ACK
"H323-GW1: 2000-11-27 11:25:14.011, A^ ALM=\"ENDPOINT CHANNEL INTERFACE FAILURE\",SEV=IF"
"H323-GW1: 2000-11-27 11:25:14.012, A^ ALM=\"ENDPOINT CHANNEL INTERFACE FAILURE\",SEV=IF"
```

In the following example, the output displays alarm events until you press **Ctrl-C**:

```
gw mml> rtrv-alm:cont
GW Signaling Interface    2000-12-05 14:19:22
M   RTRV
"H323-GW1: 2000-11-27 11:25:12.259, ** ALM=\"VSC FAILURE\",SEV=MJ"
"H323-GW1: 2000-11-27 11:25:13.259,     ALM=\"VSC FAILURE\",SEV=MJ" STATE=CLEARED
"H323-GW1: 2000-11-27 11:25:13.260, ** ALM=\"CONFIGURATION FAILURE\",SEV=MJ"
"H323-GW1: 2000-11-27 11:25:14.011, A^ ALM=\"ENDPOINT CHANNEL INTERFACE FAILURE\",SEV=IF"
"H323-GW1: 2000-11-27 11:25:14.012, A^ ALM=\"ENDPOINT CHANNEL INTERFACE FAILURE\",SEV=IF"

/* Listening for alarm events... (Ctrl-C to stop) */

"H323-GW1: 2000-11-27 11:25:13.259, ** ALM=\"VSC FAILURE\",SEV=MJ"

/* Ctrl-C pressed */
```

## Related Commands

Command	Description
<b>ack-alm</b>	Acknowledges that an alarm event is recognized but does not clear the alarm.
<b>clr-alm</b>	Clears an alarm event.

## rtrv-calls

The **rtrv-calls** command displays all actively connected calls. If the *time elapsed* parameter is provided (in units of minutes), calls display only if they exceed the specified time. The output includes the call direction, time connected, calling and called address, and call reference.

```
rtrv-calls[:time elapsed]
```

<b>Syntax Description</b>	<i>time elapsed</i>	If the time elapsed parameter is provided (in units of minutes), calls display only if they have exceeded the specified time.
---------------------------	---------------------	---

### Examples

In the following example, the command displays all actively connected calls:

```
gw mml> rtrv-calls
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC

CallId SrcAddr DestAddr StartTime
124 04161234567 0299598125 2000-11-27 11:25:13.259
```

## rrtrv-config

The **rtrv-config** command displays the currently provisioned configuration. If the MML name parameter is provided, then a subset of the current configuration is displayed. This is convenient if you are logged on to MML via a Unix terminal with a small scroll buffer.

```
rtrv-config[:MML name]
```

<b>Syntax Description</b>	<i>MML name</i>	This can be set to any of the packages listed when the <b>rtrv-config</b> command is issued. Refer to the <b>prov-add</b> and <b>prov-ed</b> commands for the list. The most useful packages in this context are:  H245, RAS, SYS_CONFIG_DYNAMIC, and SYS_CONFIG_STATIC
---------------------------	-----------------	---

### Examples

```
gw mml> rtrv-config
gw mml> rtrv-config:sys_config_static
```

## rtrv-ctr

The **rtrv-ctr** command displays the measurements for a counter group.

**rtrv-ctr:counter group**

### Syntax Description

<i>counter group</i>	The name of an MML counter group (RAS, Q931, or H245).
----------------------	--

### Examples

In the following example, the command displays the measurements for the RAS counter group:

```

gw mm1> rtrv-ctr:RAS
GW Signaling Interface      2000-12-05 14:19:22
M   RTRV
"H323-GW1:GROUP=\ "RAS\ ",NAME=\ "GK DISC ATT TOT\ ",VAL=10"
"H323-GW1:GROUP=\ "RAS\ ",NAME=\ "GK REG ATT TOT\ ",VAL=0"
"H323-GW1:GROUP=\ "RAS\ ",NAME=\ "GK REG SUCC TOT\ ",VAL=12"
"H323-GW1:GROUP=\ "RAS\ ",NAME=\ "GK RCV UNR ATT TOT\ ",VAL=100"
"H323-GW1:GROUP=\ "RAS\ ",NAME=\ "GK XMIT UNR SUCC TOT \ ",VAL=2000"
"H323-GW1:GROUP=\ "RAS\ ",NAME=\ "GK XMIT UNR ATT TOT\ ",VAL=20"
"H323-GW1:GROUP=\ "RAS\ ",NAME=\ "GK RCV UNR SUCC TOT\ ",VAL=10"
"H323-GW1:GROUP=\ "RAS\ ",NAME=\ " "GK RLS ATT TOT\ ",VAL=20"
"H323-GW1:GROUP=\ "RAS\ ",NAME=\ "GK RLS SUCC TOT\ ",VAL=30"
"H323-GW1:GROUP=\ "RAS\ ",NAME=\ "GK INFO REPORT TOT\ ",VAL=40"

```

### Related Commands

Command	Description
<b>clr-meas</b>	Resets a measurement counter.

## rtrv-dest

The **rtrv-dest** command retrieves status information about the IP links and E-ISUP signaling path to the PGW 2200.

The output produced by this command shows the states of the H323 signalling gateway external interfaces to the Cisco PGW 2200s:

- IS: In Service
- OOS: Out Of Service
- MAN: Manual
- REM: Remote
- FLT: Fault
- NCFG: Not Configured

**Examples**

The MML command in the following example retrieves status information about the IP links and E-ISUP signaling path to the PGW 2200:

```

gw mml> rtrv-dest
H323 Signalling Gateway Thu Aug 23 01:15:32 2007
M SUCC

      VscA:          ACT
      ipLink1toVscA: IS
      ipLink2toVscA: OOS NCFG

      VscB:          OOS NCFG
      ipLink1toVscB: OOS NCFG
      ipLink2toVscB: OOS NCFG

      EisupPath:     IS
gw mml>

```

**Related Commands**

Command	Description
<b>set-dest-state</b>	Changes the service state of an IP link or E-ISUP signaling path to in service (IS) or out of service (OOS).

## rtrv-gapping

The **rtrv-gapping** command retrieves information about overload-triggered call gapping.

The following information displays:

- The active or inactive status of call gapping
- The percentage of calls that are gapped
- The type of calls to which gapping is applied

**rtrv-gapping****Syntax Description**

This command has no arguments or keywords.

**Examples**

The following MML command retrieves the current levels of call gapping for all gapping clients:

```

gw mml> rtrv-gapping

```

Client Name	Direction	Level	Call Type	Active
Overload	Outgoing	10	Normal	No
Overload	Incoming	10	Normal	No
MML	Outgoing	20	All	Yes
MML	Incoming	30	All	Yes

**Related Commands**

Command	Description
<b>set-gapping</b>	Sets the type of calls to be gapped.

## rtrv-lics

The **rtrv-lics** command displays information that verifies whether the license file is installed correctly.

### rtrv-lics

#### Examples

In the following example, the command displays license verification information:

```
gw mml> rtrv-lics
H323 Signalling Gateway Sat Dec 30 22:41:00 2006
M SUCC

HSI licensing current operation status:
  License check not in progress
Current status (Node license):
  Node license feature passed ok, successful
Current status (PGW interworking license):
  HSI is is licensed to interwork with 9.6 PGW or later

New license check invoked.
```

#### Related Commands

Command	Description
<b>rtrv-ne</b>	Retrieves general network element information.

## rtrv-log

The **rtrv-log** command displays the logging level of a package or all packages.

### rtrv-log:package=x

### rtrv-log:all

#### Syntax Description

<b>package=x</b>	Displays the logging level for the various packages that make up the Cisco HSI. For package names, see the <a href="#">“Logging Categories” section on page 4-10</a> .
<b>all</b>	Displays the logging levels of all packages.

#### Examples

In the following example, the command displays the logging levels of all packages:

```
gw mml> rtrv-log:all
H323 Signaling Interface Thu Dec 14 16:28:44 2000
M RTRV

Logging levels:
Application.....0x0000
CallControl.....0xFFFF
Connection.....0x0000
DataManager.....0x0000
Eisup.....0xFFFF
```

```

FaultManager.....0x0000
Gapping.....0x0000
H323.....0xFFFF
Infrastructure....0x0000
OverLoad.....0x0000
ProcessManager....0x0000
Provisioning.....0x0000
Signal.....0x0000
Snmp.....0x0000
SnmpSubagent.....0x0000
Statistics.....0x0000
Trace.....0x0000
UserInterface.....0x0000

```

**Related Commands**

Command	Description
<b>diaglog</b>	Starts and stops event logging into a diagnostics log.
<b>radlog</b>	Starts and stops RADVision logging into a specified log file.
<b>set-log</b>	Sets the logging level of a package or all packages.

## rtrv-mml

The **rtrv-mml** command displays the following information:

- All active MML sessions
- Session numbers of all active MML sessions
- User IDs of the session originators

**rtrv-mml**

**Syntax Description**

This command has no arguments or keywords.

**Examples**

In the following example, the command displays all active MML sessions, their sessions numbers, and the user IDs of the session originators:

```
gw mml> rtrv-mml
```

```

VSC H-323 Signaling Interface - H323-GW1 2000-06-20 10:04:28
M RTRV
mml1:matthewl
mml2:davek

```

## rtrv-ne

The **rtrv-ne** command displays the type, hardware platform, vendor, location, version, and status of the Cisco HSI.

**rtrv-ne**

---

**Syntax Description** This command has no arguments or keywords.

---



---

**Examples** In the following example, the command displays the type, hardware platform, vendor, location, version, and status of the Cisco HSI:

```
gw mml> rtrv-ne
```

```
H323 Signaling Interface Thu Dec 14 16:29:19 2000
M RTRV
```

```
Type: H323 Signaling Interface
Hardware platform: Sun netra t1
Vendor: Cisco Systems, Inc.
Location: H323 - GW1
Version: R1_1_0
Platform Status:
Signaling interface: Active
Call processing: Running
```

## rtrv-ne-health

The **rtrv-ne-health** command displays the following information about the Cisco HSI status:

- CPU load
- Disk space
- Number of currently connected calls
- Number of calls in setup

**rtrv-ne-health**

---

**Syntax Description** This command has no arguments or keywords.

---



---

**Examples** In the following example, the command displays information about the Cisco HSI status:

```
gw mml> rtrv-ne-health
```

```
VSC H-323 Signaling Interface - H323-GW1 2000-06-20 10:04:28
M RTRV
```

```
CPU Load:                23%
Disk space:              123456
Number of connected calls: 23
Number of calls in setup: 12
```

## rtrv-overload

The **rtrv-overload** command displays overload status information and value settings for the three provisionable levels of overload.

### rtrv-overload

#### Syntax Description

This command has no arguments or keywords.

#### Examples

In the following example, the command displays overload status information:

```

gw mml> rtrv-overload
H323 Signaling Interface Tue Jan 30 11:21:45 2001
M SUCC
Overload/Gapping Information
NumCalls : 0 | CPU : 7% | DiskUsage : 27%
Status : Not in Ovld
Overload Configuration
DiskUsageLimit : 29%
OvldSampleRate : 3000ms
OvldLevel1Percent : 65%
OvldLevel1Filter : NORMAL
OvldLevel1ThreshLowerCpu : 30%
OvldLevel1ThreshUpperCpu : 35%
OvldLevel1ThreshLowerCalls : 800
OvldLevel1ThreshUpperCalls : 1000
OvldLevel2Percent : 75%
OvldLevel2Filter : ALL
OvldLevel2ThreshLowerCpu : 45%
OvldLevel2ThreshUpperCpu : 50%
OvldLevel2ThreshLowerCalls : 1100
OvldLevel2ThreshUpperCalls : 1400
OvldLevel3Percent : 90%
OvldLevel3Filter : NORMAL
OvldLevel3ThreshLowerCpu : 55%
OvldLevel3ThreshUpperCpu : 65%
OvldLevel3ThreshLowerCalls : 1400
OvldLevel3ThreshUpperCalls : 1600

```

#### Related Commands

Command	Description
<b>set-overload</b>	Defines the overload handling criteria and behavior.

## rtrv-softw

The **rtrv-softw** command displays the status of the Cisco HSI and call processing activity. The following software states can be displayed for the Cisco HSI:

- Not running
- Starting
- Active
- Restart pending

- Halt pending
- Reboot pending

The following software states can be displayed for call processing:

- Running
- Idle pending
- Idle

**rtrv-softw**



**Note**

When the Cisco HSI is in the restart pending, halt pending, or reboot pending software state, the **sta-callproc** command cancels the pending state.

**Syntax Description**

This command has no arguments or keywords.

**Examples**

In the following example, the command displays the status of the Cisco HSI and call processing activity:

```
gw mml> rtrv-softw
VSC H-323 Signaling Interface - H323-GW1 2000-06-20 10:04:28
M RTRV
Platform Status:
Signaling interface: Active
Call processing:      Running
```

## set-dest-state

The **set-dest-state** command changes the service state of an IP link or E-ISUP signaling path to IS (in service) or OOS (out of service).

**set-dest-state:ipLink1toVscA:IS|OOS**

**set-dest-state:ipLink2toVscA:IS|OOS**

**set-dest-state:ipLink1toVscB:IS|OOS**

**set-dest-state:ipLink2toVscB:IS|OOS**

**set-dest-state:EisupPath:IS|OOS**

**set-dest-state:ipLinkNms:IS|OOS**

**Syntax Description**

IS	In service.
OOS	Out of service.

**Examples**

In the following example, the command changes the service state of an IP link signaling path to IS:

```
gw mml> set-dest-state:ipLink1toVscA:state=IS
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
```

```
Initiating state change of ipLink1toVscA to IS
gw mml> set-dest-state:ipLink1toVscA:state=OOS
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
```

```
Initiating state change of ipLink1toVscA to OOS
```

Related Commands	Command	Description
	<b>rtrv-dest</b>	Retrieves status information about the IP links and E-ISUP signaling path to the PGW 2200.

## set-gapping

The **set-gapping** command sets the type of calls to be gapped.

**set-gapping: inc | otg | both : calltype=normal | all, percent=number**

Syntax Description	Parameter	Description
	<b>inc</b>	Gaps calls from the H.323 network.
	<b>otg</b>	Gaps calls from the PSTN over E-ISUP.
	<b>both</b>	Gaps calls originating from either side.
	<b>normal</b>	Gaps all calls except priority and emergency calls.
	<b>all</b>	Gaps calls of all types.
	<i>number</i>	Specifies the percentage of calls rejected due to call gapping.

Usage Guidelines	Guidelines
	If call gapping is set to 100 percent, all calls are gapped irrespective of the normal or all parameter setting.
	If the overload condition is active and call gapping is active, the higher of the two percentage values determines whether new call attempts are accepted or rejected.

Examples	Example
	In the following example, the command sets all calls to be gapped and specifies that 50 percent of the calls be rejected due to call gapping:

```
gw mml> set-gapping:both:calltype=all,percent=50
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
Successfully set gapping for target 'both', calltype 'all', and percentage 50
```

Related Commands	Command	Description
	<b>rtrv-gapping</b>	Retrieves information about overload-triggered call gapping.

## set-log

The **set-log** command sets the logging level of a package or all packages.

**set-log:package:level=level, [confirm]**

**set-log:all:level=level, [confirm]**

### Syntax Description

<b>package</b>	One of the packages in the Cisco HSI. For a list of package names, see the <a href="#">“Logging Categories” section on page 4-10</a> .
<b>level</b>	Logging levels are set through the use of hexadecimal numbers between 0x0000 and 0xFFFF. The higher the number, the higher the level of debug.
<b>confirm</b>	If any client level parsing fails on the data session, a <b>confirm</b> is needed for the data to be activated.

### Usage Guidelines

Logging at any level implies that upper levels are included. When you are setting logging with the level DEBUG, a confirmation is required because the amount of data logged affects service. For a list of the packages that can log messages, see the [“Logging Categories” section on page 4-10](#).

### Examples

In the following example, the command sets the logging level of the package gapping to 0xFFE0:

```
gw mml> set-log:gapping:0xFFE0
M SUCC
logging level for package gapping set to 0xFFE0
```

### Related Commands

Command	Description
<b>diaglog</b>	Starts and stops event logging into a diagnostics log.
<b>radlog</b>	Starts and stops RADVision logging into a specified log file.
<b>rtrv-log</b>	Displays the logging level of a package or all packages.

## set-overload

The **set-overload** command defines the overload handling criteria and behavior.

**set-overload: level1|level2|level3:cpu,lower=number, upper=number**

**set-overload: level1|level2|level3:calls,lower=number, upper=number**

**set-overload: level1|level2|level3:gap,normal|all : number**

Syntax Description	level1   level2   level3	Overload behavior can be provisioned at three separate levels: 1, 2, and 3 (rising in severity).
	<b>lower=number</b>	The lower threshold for overload detection and restoration of normal call handling service.
	<b>upper=number</b>	The upper threshold for overload detection and restoration of normal call handling service.

### Usage Guidelines

The **set-overload** command defines the upper and lower thresholds for overload detection and restoration of normal call handling service. The percentage of calls to be gapped and the type of calls to be gapped can also be configured. Any changes made become active immediately.

The lower value must always be less than the upper value. If the call gap percentage is set to 0, the system takes no recovery action when overload is encountered, but the appropriate alarm is raised.

Inconsistent threshold settings for different levels can destabilize call processing. For successful execution of this command, ensure that threshold settings are consistent:

- The number of calls gapped at level 2 must be greater than or equal to the number of calls gapped at level 1.
- The number of calls gapped at level 3 must be greater than or equal to the number of calls gapped at levels 1 and 2.
- The lower level value of CPU occupancy must always be less than the upper level value.
- The lower level value of CPU occupancy and the number of calls for level 2 must be greater than or equal to the corresponding values for level 1.
- The lower level value of CPU occupancy and the number of calls for level 3 must be greater than or equal to the corresponding values for levels 1 and 2.

### Examples

In the following example, the command sets the overload handling criteria and behavior at level 1, sets the cpu to the lower threshold of 10, and sets the upper threshold to 14.

```
gw mml> set-overload:level1:cpu, lower =10, upper = 14
H323 Signaling Interface Day Mon 1 11:21:28 2001
M SUCC
Successfully added that configuration item.
```

### Related Commands

Command	Retrieve
<b>rtrv-overload</b>	Displays the overload status and the data values for the three provisionable levels of overload.

## set-tos

The **set-tos** command sets the TOS/Qos field in the IP header of HSI messages.

```
set-tos::tos=<tos_value>
```

Syntax Description	<i>tos_value</i>	[0-184]
--------------------	------------------	---------

### Usage Guidelines

The **set-tos** command takes effect immediately. However, the value set by the command is not saved during a restart of the HSI.

Valid values for *tos\_value* are 0, 32, 40, 64, 72, 96, 104, 128, 136, 184. If you specify any other value in the range 0-184, the HSI sets the value to the default 96.

You can establish a static setting for the static system configuration parameter IpTOS (which remains after an HSI restart) by entering the following MML command:

```
prov-add:name=SYS_CONFIG_STATIC, IpToS=64
```

### Examples

In the following example, the command sets the *tos\_value* to 96 (the default):

```
gw mml> set-tos:tos=96
M SUCC
```

## sta-callproc

The **sta-callproc** command starts call processing.

```
sta-callproc
```

Syntax Description	This command has no arguments or keywords.
--------------------	--

### Examples

In the following example, the command starts call processing:

```
gw mml> sta-softw
gw mml> sta-callproc

H323 Signaling Interface Thu Dec 14 16:31:09 2000
M SUCC
```

Starting call processing.

### Related Commands

Command	Description
<b>sta-softw</b>	Starts the call processing application.

## sta-softw

The **sta-softw** command starts the call processing application.

**sta-softw**

### Syntax Description

This command has no arguments or keywords.

### Examples

In the following example, the command starts the call processing application:

```
gw mml> sta-softw
```

```
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
```

```
The Call Processing Application is starting.
```

### Related Commands

Command	Description
<b>sta-callproc</b>	Starts call processing.

## stp-call

The **stp-call** command terminates a currently active call by forcing a release of the call. Disconnect/release messages are sent in both directions.

**stp-call:callref=x**

**stp-call:all**

### Syntax Description

<b>callref</b>	Refers to a positive integer.
<b>all</b>	Stops all calls.

### Examples

In the following example, the command terminates a currently active call by forcing a release of the call with a callref of 33:

```
gw mml> stp-call:callref=33
```

```
H323 Signaling Interface Thu Dec 14 16:43:54 2000
M SUCC
```

```
Stopped call 33
```

### Related Commands

Command	Description
<b>rtrv-calls</b>	Displays all actively connected calls.

## stp-callproc

The **stp-callproc** command stops further call processing by immediately terminating the handling of new call requests.

**stp-callproc***[[:timeout=T]]*

### Syntax Description

<i>timeout</i>	If no timeout period is provided, existing calls are released immediately. If a timeout period is provided, existing calls are released after the specified amount of time has elapsed. When all calls have been released, a notification message is sent to the craft terminal.
T	T is in seconds.

### Examples

In the following example, the command stops further call processing by immediately terminating the handling of new call requests:

```
gw mml> stp-callproc
```

```
H323 Signaling Interface Thu Dec 14 16:27:07 2000
M SUCC
```

```
Stopped accepting new calls. Existing calls will be released in 5 seconds.
```

```
Stopping Call Processing.
```

## stp-softw

The **stp-softw** command stops the call processing application. This command causes the Cisco HSI to terminate.

**stp-softw***[[:confirm]]*

### Syntax Description

<b>confirm</b>	If there are active calls, a notification is sent to the craft. In order for the command to take effect, it must be reentered with the confirm parameter.
----------------	---

### Examples

In the following example, the command stops the call processing application:

```
gw mml> stp-softw
```

```
H323 Signaling Interface Thu Dec 14 16:27:36 2000
M SUCC
```

```
There are no active calls.
Application is now stopping
```