InnoMedia

ECMM 9500Bx

Administrative Guide

Release: 3.0.1.5

Feb 2018

Table of contents

1. GENERAL 8

1.1. PICTURE OF ECMM WITH EVALUATION BOARD 8

1.2. SETUP THE CONSOLE CONFIGURATION: 8

1.3. COMMAND WINDOW: 9

2. WEB ACCESS 10

2.1. MAIN PAGE – STATUS SOFTWARE 10

2.2. STATUS CONNECTION 11

2.3. STATUS SECURITY 12

2.4. STATUS SNMP EVENT LOG 12

3. BASIC CLI INSTRUCTIONS 14

3.1. ! 14

3.2. ? 14

3.3. REM 14

3.4. CALL 14

3.5. CD 16

3.6. DIR 16

3.7. FIND_COMMAN 17

3.8. HELP 17

3.9. HISTORY 17

3.10. INSTANCES 18

3.11. LS 18

3.12. MAN 18

3.13. PWD 18

3.14. SLEEP 18

3.15. SYNTAX 19

3.16. SYSTEM_TIME 19

3.17. USAGE 19

4. COMMAND LINE DESCRIPTION FOR CM DOCSIS COMMANDS (CM): 20
4.1. CPULOAD 20
4.2. CPUUTILIZATION 20
4.3. EXIT 20
4.4. MEMSHOW 21
4.5. PING 21
4.6. RESET 21
4.7. RUN_APP 21
4.8. TASKINFO 22
4.9. TASKTRACE 22
4.10. USFSSHOW 22
4.11. VERSION 22
4.12. ZONE 23
4.13. TELNET/SSH INTERACTIVE SESSION COMMANDS (CONSOLE) MENU 23
4.13.1. EXIT 24
4.13.2. RESET 24
4.13.3. SET 24
4.13.4. SHOW 24
4.13.5. CM COMMAND TABLE (CM) 25
4.13.5.1. DIAG 25
4.13.5.2. LOG 25
4.13.5.3. PING 25
4.13.5.4. SET 26
4.13.5.5. SHOW 26
4.13.6. ETHERNET COMMAND TABLE (ETHERNET) 27
4.13.6.1. DIAG 27
4.13.6.2. SHOW 27
4.13.7. FTP LITE CLIENT COMMANDS (FTPLITE) 27
4.13.7.1. FTP 28

4.13.8. PING COMMANDS (PINGHELPER) 28
4.13.8.1. ALL_SIZES 29
4.13.8.2. END_SIZE 29
4.13.8.3. HS_NOWAIT 29
4.13.8.4. HS_WAIT 29
4.13.8.5. IP_ADDRESS 29
4.13.8.6. IP_STACK 30
4.13.8.7. IP_SWEEP 30
4.13.8.8. NUMBER_OF_PINGS 30
4.13.8.9. PINGS 31
4.13.8.10. RESTORE_DEFAULTS 31
4.13.8.11. SHOW_*SETTINGS* 31
4.13.8.12. START_SIZE 31
4.13.8.13. STATS 32
4.13.8.14. STATS 32
4.13.8.15. STATS 32
4.13.8.16. TIME_BETWEEN_PINGS 32
4.13.8.17. VERBOSITY 33
4.13.8.18. STATS 33
4.13.8.19. WAIT_ENABLE 33
4.13.8.20. WAIT_TIME 34
4.13.9. POWER MANAGEMENT COMMANDS (POWER) 34
4.13.9.1. CLIENT 34
4.13.9.2. DEBUG 34
4.13.9.3. LEVEL 35
4.13.9.4. NEW_CLIENT 35
4.13.9.5. SHOW 35
4.13.9.6. *STATE* 35
4.13.10. SYSTEM *COMMAND* TABLE (SYSTEM) 36

4.13.10.1. DIAG 36
4.13.10.2. PKTLOG 36
4.13.10.3. SET 37
4.13.10.4. SHOW 37
4.14. HEAPMANAGER COMMANDS (HEAPMANAGER) MENU 38
4.14.1. BCHECK 38
4.14.2. BCHECK_CRASH 38
4.14.3. BCHECK_ENABLE 39
4.14.4. LAST_ERROR 39
4.14.5. MAXALLOC 39
4.14.6. OPTIONS 39
4.14.7. STATS 39
4.14.8. TIME 40
4.14.9. TRACE 40
4.14.10. WALK 40
4.14.11. WALK_ALLOC 40
4.14.12. WALK_ALLOC_TIME 40
4.15. ETHERNET HAL COMMANDS (ENET_HAL) MENU 41
4.15.1. AUTONEG 41
4.15.2. DOWN 41
4.15.3. ETH_POWERDOWN 42
4.15.4. FORCE_LINK 42
4.15.5. FULL_DUPLEX 42
4.15.6. HAL_SHOW 42
4.15.7. READ_MII 43
4.15.8. SHOW 43
4.15.9. SPEED 43
4.15.10. TRANSMIT 43
4.15.11. UP 44

4.15.12. WRITE_MII 44
4.16. EVENT LOG COMMANDS (EVENT_LOG) *MENU* 44
4.16.1. CONTROL 45
4.16.2. FLUSH 45
4.16.3. LOG_EVENT 45
4.16.4. READ 45
4.16.5. SHOW 46
4.16.6. SILENT 46
4.16.7. STOP_STRESS 46
4.16.8. STRESS 46
4.16.9. SYSLOG 47
4.16.10. VERBOSE_SHOW 47
4.17. FTP LITE CLIENT COMMANDS (FTPLITE) MENU 47
4.17.1. CLEAR_COUNTERS 47
4.17.2. COUNTERS 48
4.17.3. DEBUG 48
4.17.4. FTP 48
4.17.5. IP_*STACK* 48
4.17.6. SHOW_SETTINGS 48
4.18. MESSAGE LOG *SETTINGS* COMMANDS (MSGLOG) MENU 49
4.18.1. FIELDS 49
4.18.2. SEVERITIES 49
4.18.3. SHOW_SETTINGS 50
4.18.4. *REMOTE* ACCESS SERVER COMMANDS (REMOTEACCESS) MENU 50
4.18.4.1. READ_DEFAULT_SETTINGS 51
4.18.4.2. RESTART_SERVER 51
4.18.4.3. START_SERVER 51
4.18.4.4. STOP_SERVER 51
4.18.5. ALL_SIZES 52

4.18.6. END_SIZE 52
4.18.7. HS_NOWAIT 52
4.18.8. HS_WAIT 52
4.18.9. IP_ADDRESS 53
4.18.10. IP_STACK 53
4.18.11. IP_SWEEP 53
4.18.12. NUMBER_OF_PINGS 53
4.18.13. PING 54
4.18.14. RESTORE_DEFAULTS 54
4.18.15. SHOW_SETTINGS 54
4.18.16. START_SIZE 55
4.18.17. STATS 55
4.18.18. STEP_AMOUNT 55
4.18.19. STOP 55
4.18.20. TIME_BETWEEN_PINGS 56
4.18.21. VERBOSITY 56
4.18.22. VERIFY_ENABLE 56
4.18.23. WAIT_ENABLE 57
4.18.24. WAIT_TIME 57
4.19. POWER MANAGEMENT COMMANDS (POWER) MENU 57
4.19.1. CLIENT 57
4.19.2. DEBUG 58
4.19.3. LEVEL 58
4.19.4. NEW_CLIENT 58
4.19.5. SHOW 58
4.19.6. STATE 59
4.20. SNMP COMMAND (SNMP) MENU 59
4.20.1. CM_CFGFILE 59
4.20.2. CM_FILTER_TEST 60

4.20.3. DEBUG 60
4.20.4. FLUSH 60
4.20.5. GET 60
4.20.6. GROUP_MEMBERS 61
4.20.7. INSTALL_GROUP 61
4.20.8. KEY_V3 61
4.20.9. LOG 61
4.20.10. N2M 62
4.20.11. NOTIFY_SETUP 62
4.20.12. PARENT_CHAIN 62
4.20.13. RESET 62
4.20.14. SET 63
4.20.15. SEVERITIES 63
4.20.16. SHOW 63
4.20.17. TEST 63
4.20.18. THROTTLE 63
4.20.19. TLV_55 64
4.20.20. TRAP 64
4.20.21. VIEW_V1V2 64

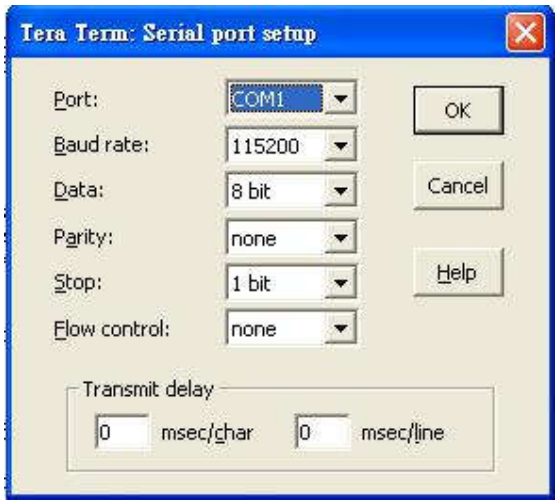
1. GENERAL

An Evaluation Board can aid in the evauation of the ECMM9500-Bx. ECMM9500-Bx can be attached to the ECMM Eval Board as shown in the following picture. A console cable that is provided can then be used to access the ECMM using the setup configuration and use Web GUI or CLI commands to manage or review various configurations within the ECMM.

1.1. Picture of ECMM with Evaluation Board

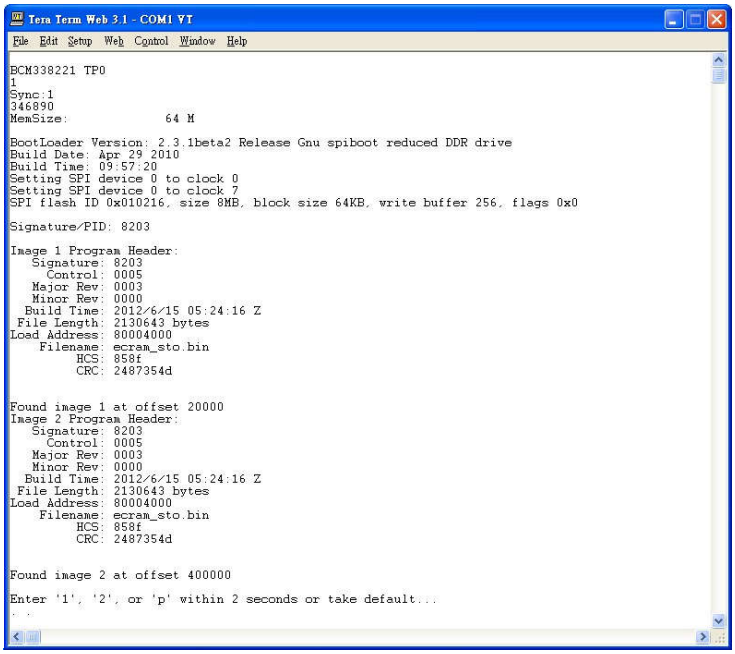
Please ensure that the debug console cable orientation is attached exactly as shown in the picture below.

1.2. Setup the console configuration:



PARAMETERS	VALUE
Baud rate	115200
Data	8bit
Parity	none
Stop	1bit
Flow control	none

1.3. Command Window:



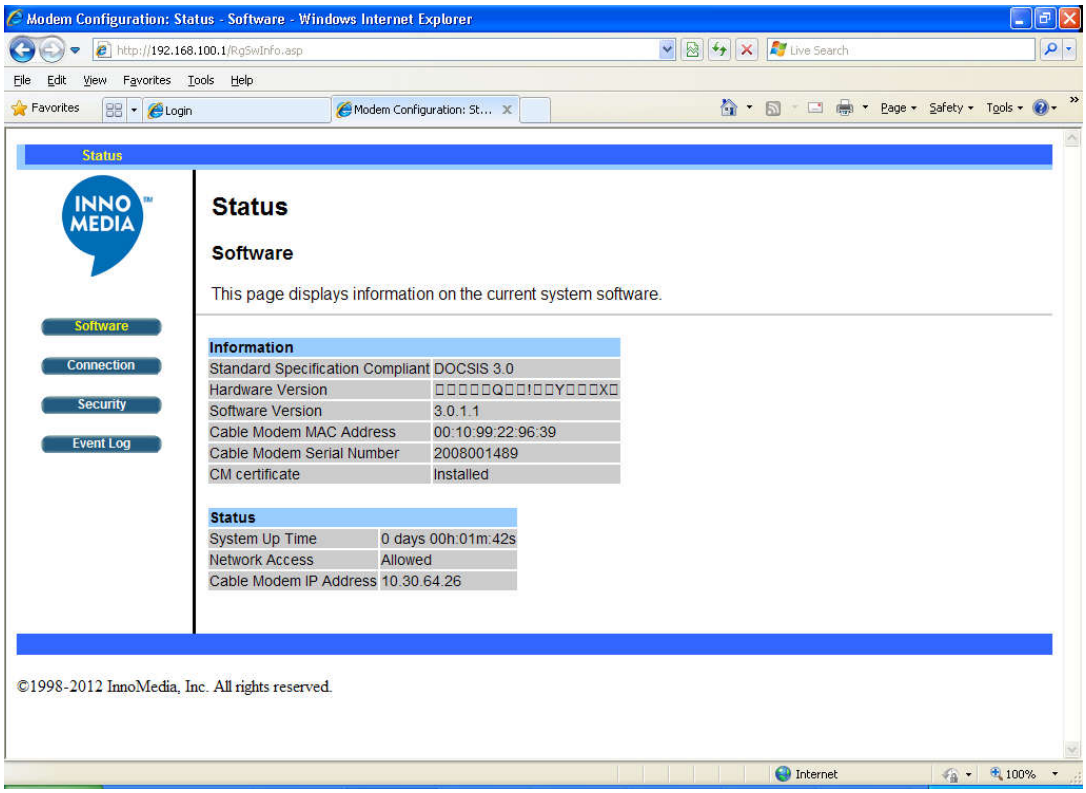
2. WEB ACCESS

Web GUI can also be access to view the ECMM for reviewing its software information, Connection, or Event Log information.

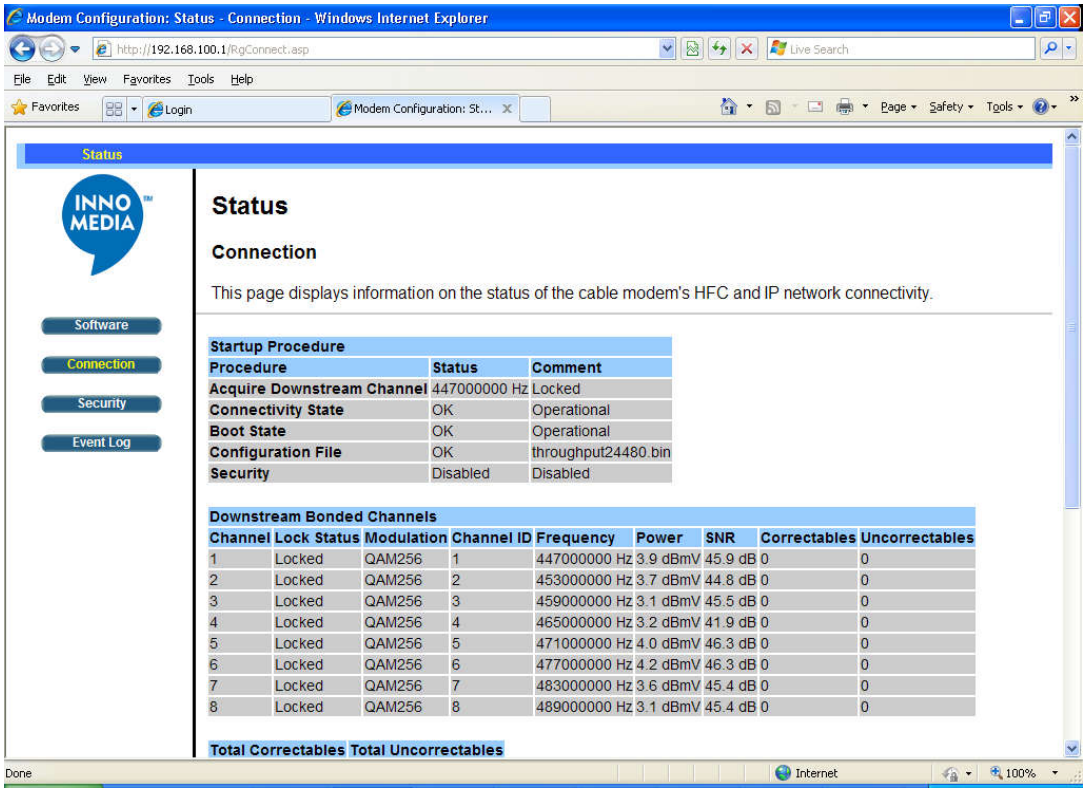
In order to access the Web GUI, connect a PC to the Ethernet port of the Evaluation Board. Ensure that the PC is configured in the 192.168.100.x subnet, and the ECMM GUI can then be accessed via 192.168.100.1 as shown below.

Default ID and Password to access the WEB GUI: **admin** and **password**.

2.1. Main Page – Status Software



2.2. Status Connection



Modem Configuration: Status - Connection - Windows Internet Explorer

http://192.168.100.1/RgConnect.asp

Status

Software
Connection
Security
Event Log

Status

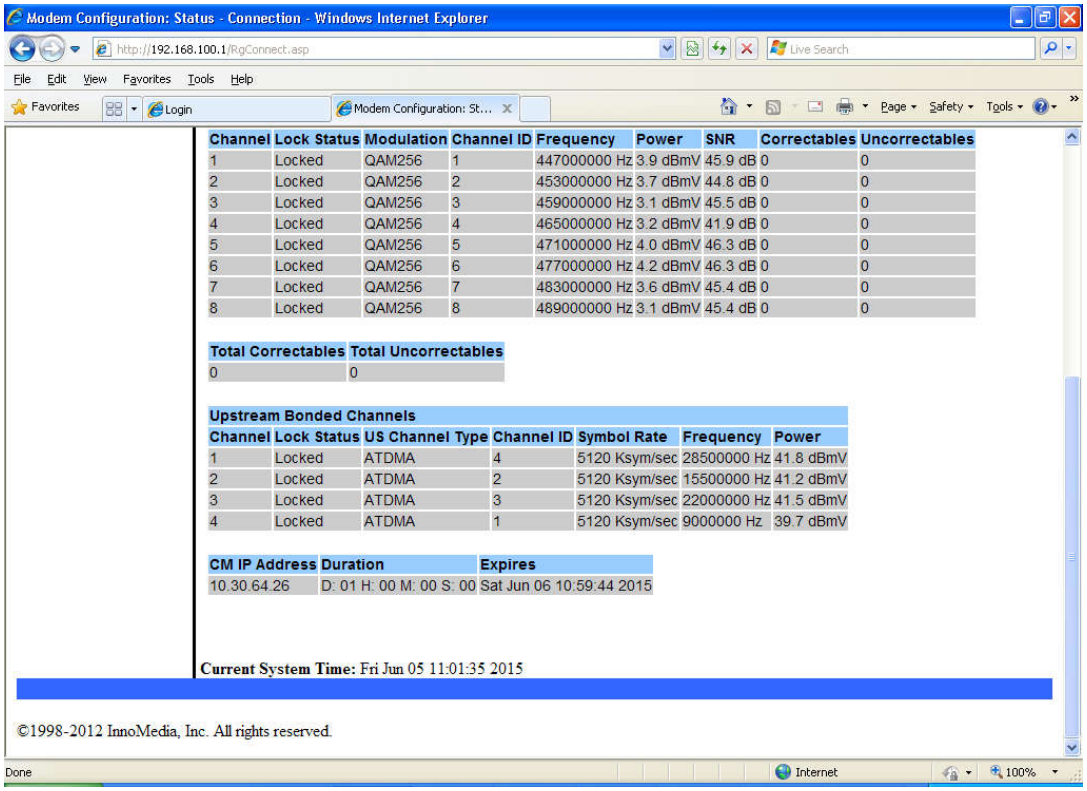
Connection

This page displays information on the status of the cable modem's HFC and IP network connectivity.

Startup Procedure		
Procedure	**Status**	**Comment**
Acquire Downstream Channel	447000000 Hz	Locked
Connectivity State	OK	Operational
Boot State	OK	Operational
Configuration File	OK	throughput24480.bin
Security	Disabled	Disabled

Downstream Bonded Channels								
Channel	**Lock Status**	**Modulation**	**Channel ID**	**Frequency**	**Power**	**SNR**	**Correctables**	**Uncorrectables**
1	Locked	QAM256	1	447000000 Hz	3.9 dBmV	45.9 dB	0	0
2	Locked	QAM256	2	453000000 Hz	3.7 dBmV	44.8 dB	0	0
3	Locked	QAM256	3	459000000 Hz	3.1 dBmV	45.5 dB	0	0
4	Locked	QAM256	4	465000000 Hz	3.2 dBmV	41.9 dB	0	0
5	Locked	QAM256	5	471000000 Hz	4.0 dBmV	46.3 dB	0	0
6	Locked	QAM256	6	477000000 Hz	4.2 dBmV	46.3 dB	0	0
7	Locked	QAM256	7	483000000 Hz	3.6 dBmV	45.4 dB	0	0
8	Locked	QAM256	8	489000000 Hz	3.1 dBmV	45.4 dB	0	0

Total Correctables	Total Uncorrectables



Modem Configuration: Status - Connection - Windows Internet Explorer

http://192.168.100.1/RgConnect.asp

Channel	Lock Status	Modulation	Channel ID	Frequency	Power	SNR	Correctables	Uncorrectables
1	Locked	QAM256	1	447000000 Hz	3.9 dBmV	45.9 dB	0	0
2	Locked	QAM256	2	453000000 Hz	3.7 dBmV	44.8 dB	0	0
3	Locked	QAM256	3	459000000 Hz	3.1 dBmV	45.5 dB	0	0
4	Locked	QAM256	4	465000000 Hz	3.2 dBmV	41.9 dB	0	0
5	Locked	QAM256	5	471000000 Hz	4.0 dBmV	46.3 dB	0	0
6	Locked	QAM256	6	477000000 Hz	4.2 dBmV	46.3 dB	0	0
7	Locked	QAM256	7	483000000 Hz	3.6 dBmV	45.4 dB	0	0
8	Locked	QAM256	8	489000000 Hz	3.1 dBmV	45.4 dB	0	0

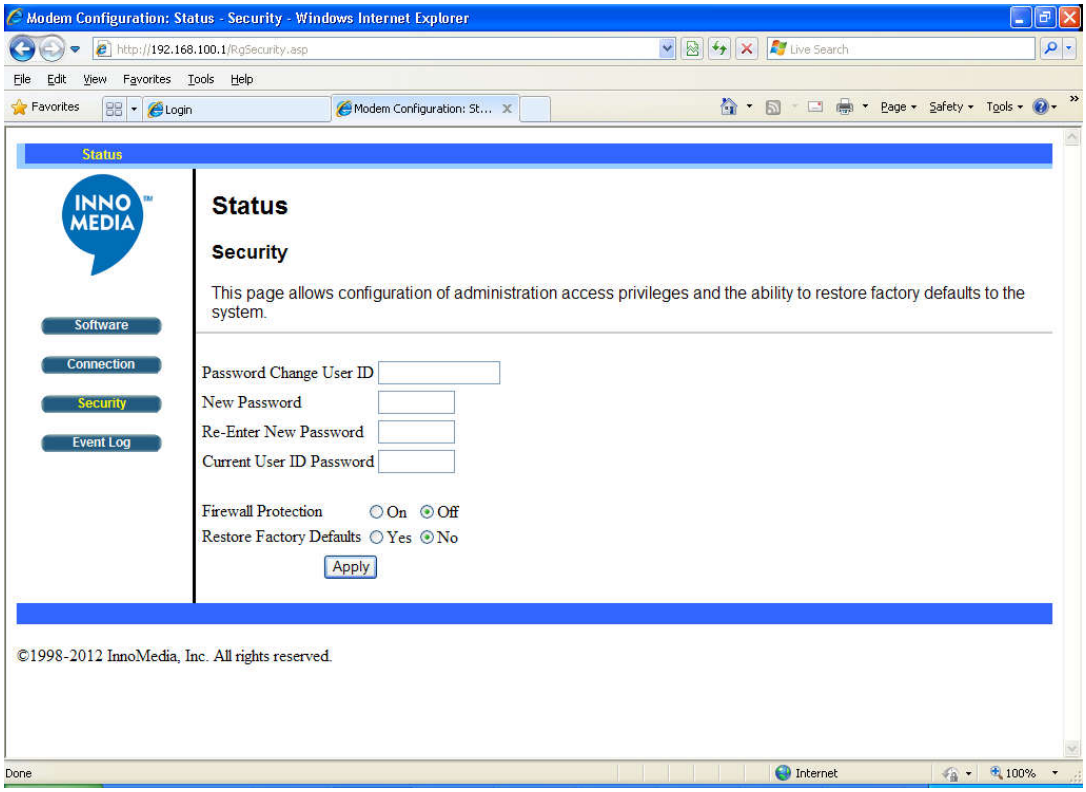
Total Correctables	Total Uncorrectables
0	0

Upstream Bonded Channels						
Channel	**Lock Status**	**US Channel Type**	**Channel ID**	**Symbol Rate**	**Frequency**	**Power**
1	Locked	ATDMA	4	5120 Ksym/sec	28500000 Hz	41.8 dBmV
2	Locked	ATDMA	2	5120 Ksym/sec	15500000 Hz	41.2 dBmV
3	Locked	ATDMA	3	5120 Ksym/sec	22000000 Hz	41.5 dBmV
4	Locked	ATDMA	1	5120 Ksym/sec	9000000 Hz	39.7 dBmV

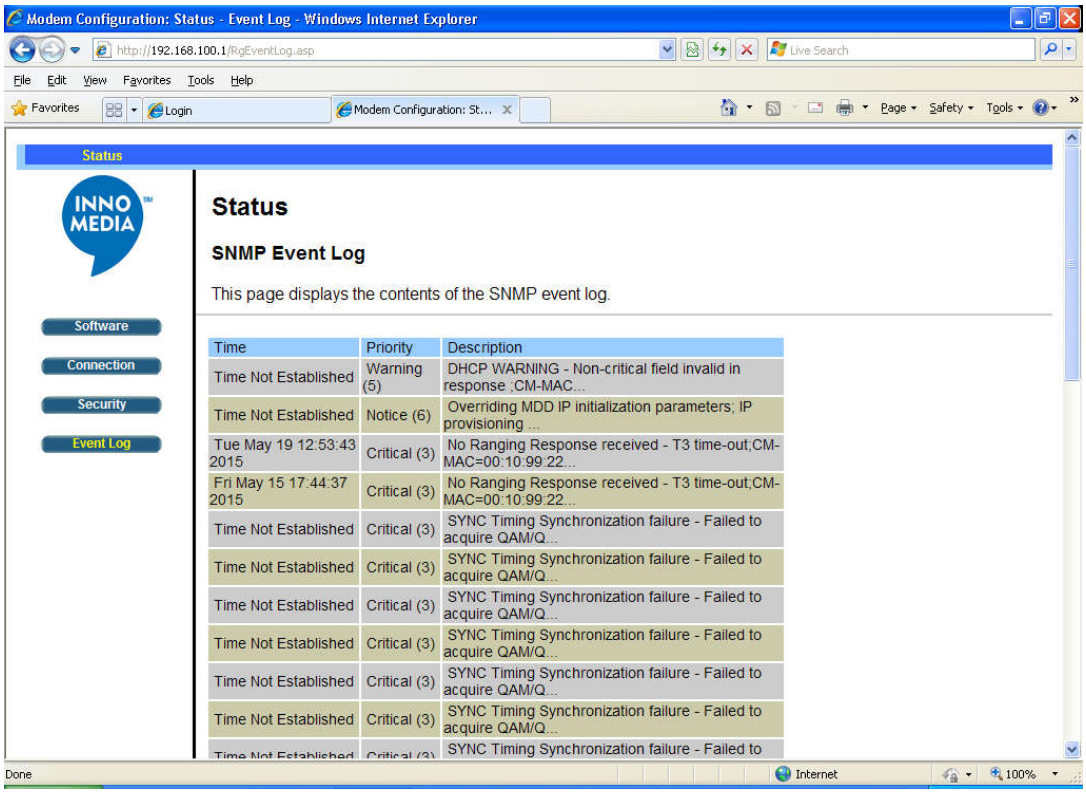
CM IP Address	Duration	Expires
10.30.64.26	D: 01 H: 00 M: 00 S: 00	Sat Jun 06 10:59:44 2015

Current System Time: Fri Jun 05 11:01:35 2015

2.3. Status Security



2.4. Status SNMP Event Log


Modem Configuration: Status - Event Log - Windows Internet Explorer
http://192.168.100.1/RgEventLog.asp
File Edit View Favorites Tools Help
Favorites
Login
Modem Configuration: St...
Page Safety Tools
Status
Software
Connection
Security
Event Log
Status
SNMP Event Log
This page displays the contents of the SNMP event log.
Time | Priority | Description
Time Not Established | Warning (5) | DHCP WARNING - Non-critical field invalid in response ;CM-MAC...
Time Not Established | Notice (6) | Overriding MDD IP initialization parameters; IP provisioning ...
Tue May 19 12:53:43 2015 | Critical (3) | No Ranging Response received - T3 time-out;CM-MAC=00:10:99:22...
Fri May 15 17:44:37 2015 | Critical (3) | No Ranging Response received - T3 time-out;CM-MAC=00:10:99:22...
Time Not Established | Critical (3) | SYNC Timing Synchronization failure - Failed to acquire QAM/Q...
Time Not Established | Critical (3) | SYNC Timing Synchronization failure - Failed to acquire QAM/Q...
Time Not Established | Critical (3) | SYNC Timing Synchronization failure - Failed to acquire QAM/Q...
Time Not Established | Critical (3) | SYNC Timing Synchronization failure - Failed to acquire QAM/Q...
Time Not Established | Critical (3) | SYNC Timing Synchronization failure - Failed to acquire QAM/Q...
Time Not Established | Critical (3) | SYNC Timing Synchronization failure - Failed to acquire QAM/Q...
Done
Internet
100%

3. BASIC CLI INSTRUCTIONS

Default ID and Password to access the CLI management console: **admin** and **password**.

Commands:
!, ?, REM, call, cd, dir, find_command, help, history, instances, ls, man, pwd, sleep, syntax, system_time, usage

3.1. !

Command	!
Description	Executes the last command that was entered. If a command (or history number) is specified, then it executes that command from the history buffer. This works like the Unix '!' command
Usage	! [Number{0..15}] [command{31}]
Example	***CM/>*** ! // This repeats the last command that was entered. ! cd // This repeats the last 'cd' command that was entered.

3.2. ?

Command	?
Description	Same as the “help” command function.
Usage	[command] ?
Example	***CM/>***

3.3. REM

Command	REM
Description	Ignores the text that follows; used for remarks, scripting, etc.
Usage	REM [Remark text{126}]
Example	***CM/>*** REM Started test here

3.4. call

Command	call
Description	This command allows you to call an arbitrary function in the code. It also provides specific support for common functions (such as malloc/free). When calling a function, you specify the address of the function (which must come from the MAP

	file for your image). You can also specify up to 9 parameters to the function. Additionally, you can specify whether or not the function returns anything, allowing the return value to be printed. NOTE: While this command is very useful for tricky debug situations in the field, IT IS SERIOUSLY UNSAFE! You can easily crash the system you are trying to debug if you specify a function address or parameter incorrectly. You could even brick the system if the function happens to write to permanent nonvol at the wrong time, or corrupts the bootloader, or...you get the picture. Make sure you are working with a valid MAP file for the image that is running, as this is your only hope for getting things right! Also, make sure you provide exactly the number of function parameters that are needed for the function. Entering too few or too many parameters will cause registers and stack state to be wrong for the function, and will lead tomayhem. Functions that require more than 9 parameters cannot be called, and they areunlikely candidates for this type of thing anyway. We chose to allow 9because C++ methods use the first parameter as the object pointer (e.g. the 'this' pointer), which allows up to 8 real parameters to the method. The first command parameter must be one of the following: 'function' - causes the function address (specified by -a) to be called with however many function parameters are specified. 'malloc' - causes malloc() to be called with the value specified by the first function parameter. 'free' - causes free() to be called with the value specified by the first function parameter. The remaining command parameters are: -r indicates that the function returns a value, and that the return value should be printed. If left out, the return value (if any) is not printed. The value is not interpreted, it is simply printed as a uint32 value in decimal and hex. -a specifies the address of the function to be called. This is only used for 'call function', since the other named functions know which address they are calling. Param1..9 specify up to 9 parameters to be passed to the function. Make sure you specify the correct number of function parameters (and that they are in the right order). Note that when calling C++ methods, the first parameter is used to hold the object instance (e.g. the 'this' pointer). For functions that take no parameters, don't specify any parameters here.

	One of the interesting aspects of this command is that it allows you to allocate memory, write to it, call a function with it, then free it.
Usage	call function|malloc|free [-r] [-a FuncAddress] [Param1] [Param2] [Param3] [Param4] [Param5] [Param6] [Param7][Param8] [Param9]
Example	Calls malloc(1024); the allocated memory address is printed: call malloc -r 1024 Calls free(0x80001234) to free memory that had previously been allocated: call free 0x80001234 Calls the function at address 0x80010080 with no parameters. Return value not printed: call func -a 0x80010080 Calls the function at address 0x800100a0 with a single parameter (value 0x01) and prints the return value: call func -r -a 0x800100a0 0x01 Calls the function at address 0x80020020 with 4 parameters, and prints the return value: call func -r -a 0x80020020 0x01 0x0202 0x030303 0x04040404 Calls the function at address 0x80020020 with 9 parameters, and prints the return value: call func -r -a 0x80020020 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 0x99

3.5. **cd**

Command	cd
Description	Sets the specified command table as the active table. This works like the DOS or Unix 'cd' command where '..' takes you to the previous table, and '\' or '/' takes you to the root table. If no parameters are specified, then it shows the name of the currently active command table.
Usage	cd [subtable | .. | \ | /{31}]
Example	***CM/>*** cd classifiers // Makes the specified subtable the active command table. cd \ // Makes the main command table active. cd .. // Makes the previous (parent) command table active. cd // Makes the previous (parent) command table active. cd \non\doc // Shows the name of the active command table. // You can specify partial names, and mutliple subdirs.

3.6. **dir**

Command	dir

Description	Alias for 'help'. Type 'help help' for more information.
Usage	dir [-t|-l|-s|-i|-a|-lr] [command [...]{126}]
Example	dir -t

3.7. find_comman

Command	find_comman
Description	Displays the name of all subdirectories which contain the specified command.
Usage	find_command command{31}
Example	***CM/>*** find_command cer // Show the related name of all subdirectories and location.

3.8. help

Command	help
Description	Shows usage information about the specified command(s), or lists the set of commands available in the active table. If no parameters are specified, then an abbreviated list of all commands and subtables is displayed. -t -- Shows the entire tree of command tables and commands (in abbreviated form). -l -- Shows detailed information about all commands and subtables (this can print a LOT of information)! -s -- Shows detailed information on just the subtables. -i -- Shows detailed information on just the registered instances for the active table. -a -- Does everything that the -l, -s, and -i options do. Command is the name (or partial name) of one or more commands and subtables for which you want detailed help to be displayed.
Usage	help [-t|-l|-s|-i|-a|-lr] [command [...]{126}]
Example	***CM/>*** help // This shows an abbreviated list of all commands and subtables. help cd // This shows detailed help on the 'cd' command. help cd ! // This shows detailed help on the 2 commands and subtable diag listed. help -l // Shows detailed help on all available commands and subtables.

3.9. history

Command	history

Description	Shows a list of commands that were previously typed.
Usage	history
Example	*CM/>* history

3.10. **instances**

Command	instances
Description	Shows the set of object instances that have registered with the active table.This is the same as 'help -i', except that it lets you specify a partial instance name in order to limit the list that is displayed (only instances whose names match the partial string are shown). The name is not case sensitive.
Usage	instances [name{31}]
Example	*CM/>* Instances // Shows all instances registered with the command table. instances p // Shows all instances whose name begins with 'p' or 'P'.

3.11. **ls**

Command	ls
Description	Alias for 'help'.
Usage	ls [-t|-l|-s|-i|-a|-lr] [command [...]{126}]

3.12. **man**

Command	man
Description	Alias for 'help'.
Usage	man [-t|-l|-s|-i|-a|-lr] [command [...]{126}]

3.13. **pwd**

Command	pwd
Description	Shows the name of the currently active command table. This is like the Unix 'pwd' command.
Usage	pwd
Example	*CM/>* pwd

3.14. sleep

Command	sleep
Description	Causes the console to sleep for the specified number of milliseconds. This is useful

	for scripting, where you want to delay between commands.
Usage	sleep [Milliseconds]
Example	***CM/>*** sleep 1000 // Makes the console sleep for 1 second.

3.15. syntax

Command	syntax
Description	Displays detailed information on command line syntax and how the parser works.
Usage	syntax
Example	***CM/>*** syntax

3.16. system_time

Command	system_time
Description	Displays the current system millisecond tick counter.
Usage	system_time
Example	***CM/>*** system_time // show as below: The current system millisecond tick counter is 0x16637c (1467260) Time since last query is 1467260 ms.

3.17. usage

Command	usage
Description	Displays information about how the console works, and how to use it.
Usage	usage
Example	***CM/>*** usage

4. COMMAND LINE DESCRIPTION FOR CM DOCSIS COMMANDS (CM):

CM >

The main instruction set:

Commands:
con_high, cpuLoad, cpuUtilization, exit, mbufShow, memShow, mutex_debug, ping, read_memory, reset, routeShow, run_app, shell,socket_debug, stackShow, taskDelete,taskInfo, taskPrioritySet taskResume, taskShow, taskSuspend, taskSuspendAll taskTrace, usfsShow, version, write_memory, zone
Subdirectory:
[Console][HeapManager][HostDqm][cm_hal][docsis_ctl[embedded_target][enet_hal][event_log] [fam] [flash] [forwarder][ftpLite] [ip_hal][msgLog] [non-vol] [pingHelper] [power] [snmp] [vendor]

4.1. cpuLoad

Command	cpuLoad
Description	Displays current CPU load, averaged over the specified number of milliseconds.
Usage	cpuLoad milliseconds{0..10000}
Example	*CM/>* cpuLoad 10000 Over the last 10000 mSec, the CPU load averaged 1%

4.2. cpuUtilization

Command	cpuUtilization
Description	Displays CPU Utilization over the specified number of milliseconds.
Usage	cpuUtilization milliseconds{0..10000}
Example	*CM/>* cpuUtilization 10000

4.3. exit

Command	exit
Description	For Telnet/ssh clients, this lets the user log out cleanly.
Usage	exit
Example	*CM/>* exit

4.4. **memShow**

Command	memShow
Description	Displays current memory allocation.
Usage	memShow
Example	***CM/>*** memShow

4.5. **ping**

Command	ping
Description	Pings the specified target IP address, sending 3 64-byte packets, and waiting up to 5 seconds for a response. This is a basic 'standard' ping. For more options or control over ping parameters and behavior, you will need to go to the Ping Command table ('cd pingHelper'). In order for this to work, the CM must either have successfully completed DHCP, or must otherwise have been configured with a valid IP address. Note that this command causes the ping options to be reset to their default state.This may be disabled if the platform doesn't provide an implementation of ping.
Usage	ping IpAddress
Example	***CM/>*** ping 11.24.4.3 // Ping IP address 11.24.4.3.

4.6. **reset**

Command	reset
Description	Causes the application to exit, shutting everything down and cleaning up resources. On embedded platforms, this usually also triggers the internal CPU reset logic, causing the h/w to reboot.
Usage	reset
Example	***CM/>*** reset

4.7. **run_app**

Command	run_app
Description	If the application was stopped at the console (either via keypress or via non-vol setting that automatically stopped it), then this command will allow it to start running. If the application is already running, this will cause it to start over again.

Usage	run_app
Example	*CM/>* run_app

4.8. **taskInfo**

Command	taskInfo
Description	Displays a specific task information.
Usage	taskInfo
Example	*CM/>* taskInfo Enter Task ID (Use taskShow for Task ID listing) (1..4294967295) [2154341392 (0x8068a410)] 0x8068a410

4.9. **taskTrace**

Command	taskTrace
Description	Displays a stack trace of a task.
Usage	taskTrace
Example	*CM/>* taskTrace

4.10. **usfsShow**

Command	usfsShow
Description	Displays the current USFS Table.
Usage	usfsShow
Example	*CM/>* usfsShow

4.11. **version**

Command	version
Description	Displays the current software version and feature codes by printing the startup banner. This allows the user to view the current version information without having to restart the application.
Usage	version
Example	*CM/>* version

4.12. **zone**

Command	zone
Description	Prints or sets the HAL debug zones; this determines what debug messages will be displayed by HAL drivers. These bits correspond to the HAL debug zones: 0x0001 -- INIT 0x0002 -- TEST1 0x0004 -- TEST2 0x0008 -- TEST3 0x0010 -- TEST4 0x0020 -- TEST5 0x0040 -- TEST6 0x0080 -- BPI 0x0100 -- DOWNSTREAM 0x0200 -- UPSTREAM 0x0400 -- TUNER 0x0800 -- RANGING 0x1000 -- POWERSAVE 0x2000 -- TESTREG 0x4000 -- WARNING 0x8000 -- ERROR
Usage	zone [Bitmask{0xffff}]
Example	***CM/>*** zone 0xc000 // Enables ERROR and WARNING levels.

4.13. Telnet/SSH Interactive Session Commands (Console) Menu

- **Located in:** After the ***CM*** prompt, types "***cd Console***".

```
CM > cd Console
```

```
CM/Console>
```

- **Description**: Telnet/SSH Interactive Session Commands.

The main instruction set:

Commands:
Exit, reset, set, show
Subdirectory:
[cm] [ethernet] [ftpLite] [pingHelper] [power] [system]

4.13.1.**exit**

Command	exit
Description	End the current telnet or ssh interactive session
Usage	exit
Example	***CM/Console>*** exit

4.13.2.**reset**

Command	reset
Description	Resets the system
Usage	reset
Example	***CM/Console>*** reset

4.13.3.**set**

Command	set
Description	Sets the username/password for telnet/ssh logins
Usage	set [username|password]
Example	***CM/Console>*** set username // Sets the username set password // Sets the password

4.13.4.**show**

Command	show
Description	Shows state of the system.
Usage	show [all|tech_support|ip|reset|uptime|version]
Example	***CM/Console>*** show all // All of the following, plus optionally recurse subtables. show tech_support // All of the following; recurse all subtables, no user prompts. show ip // All IP addresses in the system show reset // Reset reason history log. show uptime // Time that system has been up. show version // SW filename, version, and build info.

4.13.5. CM Command Table (cm)

Located in:

CM/Console>cd cm

CM/Console/cm>

The main instruction set:

Commands:
Exit, reset, set, show
Subdirectory:

4.13.5.1.diag

Command	diag
Description	Executes diag commands of the CM.
Usage	Diag [downstream | upstream] [parameter]
Example	CM/Console/cm> diag downstream 525000000 // Changes downstream frequency. diag upstream 125000000 // Changes upstream frequency.

4.13.5.2.log

Command	log
Description	Enable/disable real-time logging of specified message types or events.
Usage	log [dsx | ranging | zone | off] [Log Setting { 0 ~ 65536}]
Example	CM/Console/cm> Log dsx1 // enable DSx message logging. Log dsx0 // disables DSx message logging. Log ranging 1 // enables ranging message logging. Log ranging 0 // disables ranging message logging. Log zone 0xFFFF // enables all zone message logging. Log zone 0x0000 // disables all zone message logging. Log off // disables all CM logging.

4.13.5.3.ping

Command	ping
Description	Pings the specified target IP address from the CM IP stack.
Usage	ping [IP Address]

Example	CM/Console/cm> ping 11.24.4.3 // ping IP address 11.24.4.3

4.13.5.4.set

Command	set
Description	Sets the state of the CM.
Usage	set [firewall | d30version]
Example	CM/Console/cm> set firewall // enables or disables the firewall. set d30version // enables or disables the D30 version capability advertisement in reg-req.

4.13.5.5.show

Command	show
Description	Shows the state of the cablemodem.
Usage	show [all | arp | bpi | certs | cls | cfg | cpe | ds | dscwer | dsx | event | flows | lease | route | status | tod | ucd | us | counters | debug | sockets | errors | us30 | fam]
Example	CM/Console/cm> show all // All options of this command.. show arp // ARP table. show bpi // BPI status. show certs // CM certificates. show cls // Docsis classifiers. show cfg // CM config file name and contents. show cpe // CPE info/table. show ds // Downstream status and signal quality. show dscwer // Downstream Codeword Error Rate. show dsx // DSx message history. show event // CM event log. show flows // Docsis service flows. show lease // CM IP and DHCP options. show route // The routing table. show status // Docsis registration status. show tod // Time of day. show ucd // Upstream descriptors. show us // Upstream status. show counters // HW counters.

	show debug	// Hal debug.
	show sockets	// Bcmsocket debug.
	show errors	// Error counters.
	show us30	// DOCSIS 3.0 upstream counters.
	show fam	// Forward assist manager.

4.13.6. Ethernet Command Table (ethernet)

- Located in:

CM/Console>cd ethernet

CM/Console/ethernet>

- The main instruction set:

Commands:
diag, show
Subdirectory:

4.13.6.1.diag

Command	diag	
Description	Executes diag commands of the Ethernet interface	
Usage	diag [readmii | writemii] [Parm2] [Parm3] [Parm4]	
Example	CM/Console/ethernet>	
	diag readmii 0 0x18	// Reads from ethernet registers.
	diag writemii 0 0x18 0x400	// Writes to ethernet registers.

4.13.6.2.show

Command	show	
Description	Shows debug state of the Ethernet interface.	
Usage	show [all|debug]	
Example	CM/Console/ethernet>	
	show all	// Show all of the below.
	show debug	// Shows debug state of ethernet RX/TX DMA rings.

4.13.7. FTP Lite Client Commands (ftpLite)

- Located in:

```
CM/Console>cd ftpLite
CM/Console/ftpLite>
```

- The main instruction set:

Commands:
ftp
Subdirectory:

4.13.7.1. ftp

Command	ftp
Description	Begins FTP to the specified IP address, using the current settings.
Usage	ftp
Example	CM/Console/ftpLite> ftp // Initiates FTP until a key is pressed. FTP Server IP Address: () [0.0.0.0] Username: () [] Password: () [] TransferType: (get|put) [get] Filename: () [] Begin file transfer? (true|false) [true]

4.13.8. Ping Commands (pingHelper)

- Located in:

```
CM/Console>cd pingHelper
CM/Console/pingHelper>
```

- The main instruction set:

Commands:
all_sizes, end_size, hs_nowait, hs_wait, ip_address,ip_stack,ip_sweep, number_of_pings ping restore_defaults, show_settings, start_size,stats,step_amount, stop, time_between_pings, verbosity verify_enable, wait_enable, wait_time
Subdirectory:

4.13.8.1.All_sizes

Command	all_sizes
Description	Configures the settings for sweeping all packet sizes from 64-1518, with waiting and verification enabled. The time between pings is set to 0 ms, the verbosity is set to full, and the reply wait time is set to 1⁄2 second. The IP address is not changed.
Usage	all_sizes [size{64..1518}]
Example	CM/Console/pinghelper> all_sizes

4.13.8.2.end_size

Command	end_size
Description	Sets or shows the size of the largest ping packet that will be sent (including LLC and IP header overhead). After the packet size is increased by the step amount, if it is larger than this value, then the size is reset to the start size.
Usage	end_size [size{64..1518}]
Example	CM/Console/pinghelper> end_size 1518 // Sets the end size to the maximum allowed.

4.13.8.3.hs_nowait

Command	hs_nowait
Description	Configures the settings for doing high-speed pings (infinite), without waiting for the reply. The display verbosity is set to 2 (display only a 'p'), the time between pings is set to 0, and waiting for replies is disabled. None of the other settings are changed.
Usage	hs_nowait
Example	CM/Console/pinghelper>
	hs_nowait

4.13.8.4.hs_wait

Command	hs_wait
Description	Configures the settings for doing high-speed pings (infinite), waiting for the reply. The display verbosity is set to 2 (display only a 'p'), the time between pings is set to 0, and waiting for replies is enabled. None of the other settings are changed.
Usage	hs_wait
Example	CM/Console/pinghelper> hs_wait

4.13.8.5.ip_address

Command	ip_address
Description	Sets or shows the IP address of the device to be pinged.
Usage	ip_address [IpAddress]
Example	CM/Console/pinghelper> Ip_address 20.30.10.1

4.13.8.6.ip_stack

Command	ip_stack
Description	Sets the IP stack number that the pings should be sent to. If 0, then the default stack will be used. The stackNum parameter must correspond to a valid IP stack that has been installed and initialized with an IP address.
Usage	ip_stack [Number{0..255}]
Example	CM/Console/pinghelper> ip_stack 2 // Forces pings to go out IP stack 2.

4.13.8.7.ip_sweep

Command	ip_sweep
Description	Pings all IP addresses on the specified subnet, starting with the address specified, reporting success or failure for each one. It changes the ping settings so that only a single ping is sent. This is often used to discover all of the IP addresses on the subnet. The address will be incremented from 1..254, skipping .0 and .255 since these are often used for local broadcast addresses.
Usage	ip_sweep [Subnet] [StartingIp]
Example	CM/Console/pinghelper> ip_sweep 255.255.255.0 10.24.4.5 // Pings 10.24.4.5 through 10.24.4.254.

4.13.8.8.number_of_pings

Command	number_of_pings
Description	Sets or shows the number of pings to be sent. Note that 0 means infinite (you will need to press a key or type 'stop' to abort). A value of -1 causes the number of pings to be calculated based on the end size, start size, and step amount, so that it will span the range exactly once; num = (end - start + 1) / step.
Usage	number_of_pings [Number{-1..2147483647}]
Example	CM/Console/pinghelper> number_of_pings 3 // Limits the number of pings to 3. number_of_pings 0 // Sets the number of pings to infinite.

	number_of_pings -1	// Calculates the number of pings to span the range.

4.13.8.9.pings

Command	ping
Description	Begins pinging the specified IP address, using the current settings. If the IP address is missing, then it uses the one that was previously set. If you specify -s, then pinging will happen int he background until you type stop. Otherwise, it will poll for a keypress.
Usage	ping [-s] [IpAddress]
Example	CM/Console/pinghelper> ping 10.24.4.3 // Ping until a key is pressed. ping -s // Ping until the user types stop.

4.13.8.10. restore_defaults

Command	restore_defaults
Description	Restores all of the options to their default values (excluding the IP address, which is not modified). The default values cause ping to behave like most host-based ping utilities (3 packets, 64 bytes, wait 5 seconds, etc.).
Usage	restore_defaults
Example	CM/Console/pinghelper> restore_defaults // Ping until a key is pressed.

4.13.8.11. show_*settings*

Command	show_settings
Description	Displays the current ping settings.
Usage	show_settings
Example	CM/Console/pinghelper> show_settings

4.13.8.12. start_size

Command	start_size
Description	Sets or shows the size of the first ping packet that will be sent (including LLC and IP header overhead). The packet size will be increased by the step amount for each packet. This must be between 64..end_size, inclusive.
Usage	start_size [size{64..1518}]
Example	CM/Console/pinghelper>

	start_size 64 //Sets the start size to the minimum allowed.

4.13.8.13. **stats**

Command	stats
Description	Displays the ping statistics summary from the last set of pings. This is the same summary that is displayed at the end of the pings (if verbosity is > 0).
Usage	stats
Example	CM/Console/pinghelper> stats

4.13.8.14. **stats**

Command	step_amount
Description	Sets or shows the amount that the packet size will be increased for each packet. This can be any number (including 0, which means to keep the size constant for every packet). Note that if you set it too large, then the packet size will wrap around to the start size every time, since it will never be allowed to be larger than the end_size. You can also specify a negative number which causes the ping size to start with the end size parameter and step down to the start size, then wrap back around to the end size.
Usage	step_amount [size]
Example	CM/Console/pinghelper> step_amount 1 // Increases the packet size by 1 each time. step_amount -1 // Decreases the packet size by 1 each time.

4.13.8.15. **stats**

Command	stop
Description	Stops the ping that is currently running. This is necessary if you used the -s parameter with ping.
Usage	stop
Example	CM/Console/pinghelper> stop // Stops the ping that is running.

4.13.8.16. **time_between_pings**

Command	time_between_pings
Description	Sets or shows the number of milliseconds that the ping helper will wait before sending the next ping. Note that this does not include time spent waiting for the reply or verifying it, or for time spent printing status information. The actual

	resolution and accuracy of this depends on the system (pSOS generally runs with a 10ms clock tick, so 10ms is the same as 15ms on that system).
Usage	time_between_pings [Milliseconds]
Example	CM/Console/pinghelper> time_between_pings 100 // Waits 100ms before sending the next ping.

4.13.8.17. **verbosity**

Command	verbosity
Description	Sets the level of information that will be displayed while pinging. A higher number provides more information, but also slows down the rate at which pings can be sent. Most host-based ping utilities provide output equivalent to 3. For high-performance, high packet rate pings, values of 1 or 2 are best.
Usage	verbosity [Number{0..3}]
Example	CM/Console/pinghelper> verbosity 0 // Disables printing of all ping status output. verbosity 1 // Displays only the summary of statistics at the end. verbosity 2 // Displays a 'p' every second to show progress, and the summary. verbosity 3 // Displays full ping info for every packet, and the summary.

4.13.8.18. **stats**

Command	verify_enable
Description	Enables/disables verification of ping replies. If enabled, and if waiting for replies is enabled, then if a reply is received, it will verify that it matches the ping that was sent, and that all of the data is intact.
Usage	verify_enable [true|false]
Example	CM/Console/pinghelper> verify_enable 1 // Enables verification of the ping reply.

4.13.8.19. **wait_enable**

Command	wait_enable
Description	Enables/disables waiting for ping replies. If enabled, the ping helper will wait a number of milliseconds for the reply, and will process it if received. This is true even if the timeout is 0ms; it will always check for and process the response. If disabled, then no attempt will be made to check for or process a response. This is generally only of interest when you need to send data very quickly, with no variability.
Usage	wait_enable [true|false]
Example	CM/Console/pinghelper>

	wait_enable 1 // Enables waiting for the ping reply.

4.13.8.20. wait_time

Command	wait_time
Description	Sets or shows the number of milliseconds that the ping helper will wait for a ping response before continuing. This only takes effect if waiting is enabled. The actual resolution and accuracy of this depends on the system (pSOS generally runs with a 10ms clock tick, so 10ms is the same as 15ms on that system).
Usage	wait_time [Milliseconds]
Example	CM/Console/pinghelper> wait_time 5000 // Sets the reply wait timeout to 5 seconds.

4.13.9. Power Management Commands (power)

- Located in:

CM/Console>cd power

CM/Console/power>

- The main instruction set:

Commands:
Client, debug, level,new_client, show, state
Subdirectory:

4.13.9.1.Client

Command	client
Description	Sets the specified client parameter. If client ID is 0, the change will be made to all clients.
Usage	client [ID{0..100}] [enable | timer |state] [value{31}]
Example	CM/Console/power> client 1 enable 0 // Disable client 1. client 1 timer 10 // Sets the timer for client 1 to 10 seconds. client 1 state reduced // Sets the state for client 1 to reduced power.

4.13.9.2.debug

Command	debug
Description	Enable or disable power save debug.

Usage	debug [enable | false]
Example	CM/Console/power> debug true // Enable power save debug. debug false // Disable power save debug.

4.13.9.3.level

Command	level
Description	Shows or modifies the current power save level.
Usage	level [moderate|maximum]
Example	CM/Console/power> level moderate // Sets power save level to moderate reduction (slightly reduced). level maximum // Sets power save level to maximum reduction (reduced as much as possible).

4.13.9.4.new_client

Command	new_client
Description	Installs a new power save client using the specified parameters.
Usage	new_client String{31} reduced power priority{-1..100} full power priority{-1..100} reduced power timer{-1..1000} [async|sync]
Example	CM/Console/power> client foo 1 1 5 // Creates a new client named 'foo' with reduced/full priority of 1/1 and 5 second timer

4.13.9.5.show

Command	show
Description	Display power management information.
Usage	show
Example	CM/Console/power> show

4.13.9.6.*state*

Command	state
Description	Shows or modifies the current power save state.
Usage	state [reduced|full]
Example	CM/Console/power> state full // Sets power save state to full power.

	state reduced	// Sets power save state to reduced power.

4.13.10. System *Command* Table (system)

- Located in:

CM/Console>cd system

CM/Console/system>

The main instruction set:

Commands:
Client, debug, level,new_client, show, state
Subdirectory:

4.13.10.1. diag

Command	diag
Description	Executes diag commands of the system
Usage	diag [-p] [-c] [-s ParmSValue] [-n ParmNValue] [readmem | writemem | clear_debug_counters | show_debug_counters | set_debug_flow | snmp_reset] [Parm2] [Parm3]
Example	CM/Console/system> readmem -s 4 -n 64 // Reads 64 bytes as 32-bit values. 0x80001234 // Write a byte to the address. writemem 0x80001234 0x56 // Clear UTP debug counters. clear_debug_counters // Show debug counters for a selected flow. show_debug_counters // Enable debug counters for the selected set_debug_flow 0 flow. snmp_reset // Reset sockets for all SNMP agents.

4.13.10.2. pktlog

Command	pktlog
Description	his enables/disables logging for packets received from the specified interfaces and being sent to the specified interfaces. If enabled, then the contents of the packet will be displayed, and the forwarder will show info about the HalIf on which it was received/sent, as well as why the packet was dropped. If the from/to interfaces parameters are missing, then all will be done. The interfaces are a bitmask where 0x01 corresponds to the interface at index 0, 0x04 corresponds to the interface at index 2, etc.
Usage	pktlog [-s] [true|false] [from_interfaces] [to_interfaces]

Example	CM/Console/system> pktlog true // Enables packet logging for all interfaces. pktlog false 0x2 // Disables logging of packets from interface 1 (to any). pktlog true 0x1 0x6 // Enables logging of packets received from interface 0 and destined to interfaces 1 or 2. pktlog // Shows the enable/disable state for all interfaces. pktlog -s true // Enables non-verbose logging of packets.

4.13.10.3. set

Command	set
Description	Set state of the system and system functionality.
Usage	set [wdog|exception|fpm_token_depletion_wdog]
Example	CM/Console/system> set wdog // Enables/disable the watchdog timer. set exception // Enables/Disables exception logging. set fpm_token_depletion_wdog // Enabled/Disables the FPM Token Depletion Watchdog.

4.13.10.4. show

Command	show
Description	Shows state of the system.
Usage	show [all | flash | memory | threads | forwarder | blocks | ecos | sockets | fpm | dqm | nonvol | snmp | telnet | ssh]
Example	CM/Console/system> show all // Show all of the below. show flash // Show flash memory info. show memory // Show RAM and buffer info. show threads // Shows the state of active threads in the system. show forwarder // Shows the forwarder's learning tables and forwarder's HAL interfaces. show blocks // Shows the number of serial driver writes blocked due to transmiting too much at once. show ecos // Shows the eCos ip/tcp/udp statistics. show sockets // Shows the eCos socket out of mbuf/cluster socket history tracking logs. show fpm // Displays the configuration settings and statistics of the

		Hardware Free Pool Manager Object.
	show dqm	// Displays the internal state of the DQM interface.
	show nonvol	// Displays the current non-volatile settings values for both the permanent and dynamic sections.
	show snmp	// Displays the current SNMP settings for all agents.
	show telnet	// Displays the current telnet state and information.
	show ssh	// Displays the current ssh state and information.

4.14. HeapManager Commands (HeapManager) Menu

- **Located in:** After the ***CM*** prompt, types "***cd HeapManager***".

  CM> cd HeapManager

  CM/HeapManager>

- Description: Broadcom Heap Manager settings and commands.

The main instruction set:

Commands:
Bcheck,bcheck_crash,bcheck_enable,last_error,maxAlloc,options,stats,threadUsage,time trace,walk,walk_alloc,walk_alloc_time
Subdirectory:

4.14.1. **bcheck**

Command	bcheck
Description	Runs a bounds check in the heap manager (if compiled in).
Usage	bcheck
Example	CM/HeapManager> bcheck

4.14.2. **bcheck_crash**

Command	bcheck_crash
Description	Sets the behavior when an on-the-fly bounds checking error is detected. Turning this on will cause the offending thread to crash after we print relevant information.
Usage	bcheck_crash [true|false]

Example	CM/HeapManager> bcheck // Enables crashing after on-the-fly bcheck error detected.

4.14.3.**bcheck_enable**

Command	bcheck_enable
Description	Turns on-the-fly bounds checking on or off in the heap manager (if compiled in). When this is on, we will validate pointers, seed values, and other heap state during each alloc and free. When off, you must run bcheck manually to detect errors.
Usage	bcheck_enable [true|false]
Example	CM/HeapManager> bcheck_enable true

4.14.4.**last_error**

Command	last_error
Description	Displays the last error that was detected by the heap manager.
Usage	last_error
Example	CM/HeapManager> bcheck // Enables crashing after on-the-fly bcheck error detected.

4.14.5.**maxAlloc**

Command	maxAlloc
Description	Displays the maximum number of bytes that can currently be allocated in a single call to malloc. This takes into account all of the overhead for node tracking and bounds checking, as well as the current fragmentation state of the heap.
Usage	maxAlloc
Example	CM/HeapManager> maxAlloc

4.14.6.**options**

Command	options
Description	Displays heap manager build time options.
Usage	options
Example	CM/HeapManager> options

4.14.7.**stats**

Command	stats

Description	Displays detailed heap manager counters and statistics.
Usage	stats
Example	CM/HeapManager> stats

4.14.8. **time**

Command	time
Description	Print system time to be used for walkAllocTime
Usage	time
Example	CM/HeapManager> time // print system time.

4.14.9. **Trace**

Command	trace
Description	Enables debug tracing for the specified thread ID or all threads if the parameter is 0.
Usage	trace tid [size]
Example	CM/HeapManager> trace 0x80b0a0a0 24 // enable 24 byte alloc tracing for the thread with TID 0x80b0a0a0.

4.14.10. **walk**

Command	walk
Description	Displays all of the free memory blocks.
Usage	walk
Example	CM/HeapManager> walk

4.14.11. **walk_alloc**

Command	walk_alloc
Description	Displays all of the allocated memory blocks. WARNING: This can print a LOT of information!
Usage	walk_alloc [tid count]
Example	CM/HeapManager> walk_alloc 0x80b0a0a0 10 // display the last 10 malloc from task 0x80b0a0a0

4.14.12. **walk_alloc_time**

Command	walk_alloc_time
Description	Displays all of the allocated memory blocks for <tid> between <start> and <end>, optionally printing the callstack.
Usage	walk_alloc_time tid start [end] [callstack]
Example	CM/HeapManager> walk_alloc_time <tid> <start> <end> <callstack>

4.15. Ethernet HAL Commands (enet_hal) Menu

- **Located in:** After the ***CM*** prompt, types "***cd enet_hal***".

  CM> cd enet_hal

  CM/EnetHal>

- Description:To show/change the Ethernet HAL state.
- The main instruction set:

Commands:
Autoneg, down, eth_powerdown, force_link, full_duplex, hal_show, read_mii, show, speed, transmit, up, write_mii
Subdirectory:

4.15.1. autoneg

Command	autoneg
Description	Turns Ethernet AutoNegotiation on or off.
Usage	autoneg [true|false]
Example	***CM/EnetHal>*** autoneg true // Enables Ethernet AutoNegotiation

4.15.2. **down**

Command	down
Description	Sets the interface administratively down (equivalent to setting ifAdminStatus to 'down')
Usage	down

Example	***CM/EnetHal>*** down

4.15.3. eth_powerdown

Command	eth_powerdown
Description	Putting the Ethernet IF into powerdown mode does the following in the specified order: 1 -- Suspends the Ethernet link monitoring task 2 -- Disables interrupts from the EMAC 3 -- Disables the EMAC core 4 -- Disables the clock to the EMAC The Ethernet IF is returned to operation state by negating these actions in reverse order
Usage	eth_powerdown [true|false]
Example	***CM/EnetHal>*** eth_powerdown true // Puts the Ethernet IF in powerdown mode. eth_powerdown false // Restores the Ethernet IF to operational state.

4.15.4. force_link

Command	force_link
Description	Enables/disables the link detection logic, forcing the PHY to think that there is a link when there really isn't.
Usage	force_link [true|false]
Example	***CM/EnetHal>*** force_link // Queries the link-forced state. force_link true // Forces the link. force_link false // Reverts to normal link detection.

4.15.5. full_duplex

Command	full_duplex
Description	If autoneg is off, sets the Ethernet duplex to full/half.
Usage	full_duplex [true|false]
Example	***CM/EnetHal>*** full_duplex true // Sets the link to full duplex

4.15.6. hal_show

Command	hal_show
Description	Causes the Ethernet HAL to display its internal state.

Usage	hal_show
Example	***CM/EnetHal>*** hal_show

4.15.7. read_mii

Command	read_mii
Description	Reads the specified ethernet MII register from the PHY specified.
Usage	read_mii PhyAddr RegAddr
Example	***CM/EnetHal>*** read_mii 1 0x18 // Reads the AUX STATUS REG from the internal PHY on the 3345.

4.15.8. show

Command	show
Description	Causes the HalIf object to display its state.
Usage	show
Example	***CM/EnetHal>*** show

4.15.9. speed

Command	speed
Description	If autoneg is off, sets the Ethernet link speed.
Usage	speed [10|100|1000]
Example	***CM/EnetHal>*** speed 100 // Sets the link speed to 100 Mbps.

4.15.10. transmit

Command	transmit
Description	Transmits packets out the ethernet interface. The packets will have garbage data in them, so you probably don't want to do this on a live network. Packets will be sent as fast as possible unless overridden by the -r flag; you can specify the packet size and/or the number of seconds over which to send packets. If not otherwise specified, it will send 1518 byte packets until the system is power cycled. Flags: -s : The packet size; if not specified, 1518 bytes. -t : Number of seconds you want to transmit; default (0) = infinite. -r : Controls the packet rate. Specify the time (in ms) between each packet. Note that this value will be quantized based on the OS clock

	tick resolution (usually 10ms), so 1ms, 8ms, and 12ms are all the same as 10ms. A value of 0 means 'as fast as possible', i.e. no delay between packets. -p : Specifies the fill pattern for the buffer. The value specified will be used to fill the buffer. If not specified, then the buffer is filled with increasing values.
Usage	transmit [-s PacketSize{64..1518}] [-t NumSeconds] [-r TimeBetweenPacketsMs] [-p FillPattern{0..255}]
Example	***CM/EnetHal>*** Transmit // Sends 1518 byte packets until power cycled. transmit -s 64 // Sends 64 byte packets until power cycled. transmit -t 60 // Sends 1518 byte packets for 1 minute. transmit -r 1000 // Sends 1518 byte packets once a second. transmit -r 10 // Sends 1518 byte packets at 100 packets per second. transmit -p 0xaa // Fills the 1518 byte packet with 0xaa values.

4.15.11. up

Command	up
Description	Sets the interface administratively up (equivalent to setting ifAdminStatus to 'up')
Usage	up
Example	***CM/EnetHal>*** up

4.15.12. write_mii

Command	write_mii
Description	Writes the specified value to the ethernet MII register, using the PHY specified. Note that if you want to mask bits on or off, then you will need to do the math yourself, using read_mii to show the current value.
Usage	write_mii PhyAddr RegAddr Value
Example	***CM/EnetHal>*** write_mii 1 0x18 0x400 // Sets the Force Link bit in the AUX STATUS REG (using the internal PHY on the 3345).

4.16. Event Log commands (event_log) *Menu*

- **Located in:** After the ***CM*** prompt, types "***cd event_log***".

CM> cd event_log

CM/Event Log>

Description:

To manipulate the Event Log.

The main instruction set:

Commands:
Control, flush, log_event, read, show, silent, stop_stress, stress, syslog, verbose_show
Subdirectory:

4.16.1. control

Command	control
Description	Show or modify the contents of the control table.
Usage	control [level{0..8}] [reporting{0..255}]
Example	***CM/Event Log >*** Control // show the control table. control 1 0x80 // set reporting for level 1 to 0x80. control 0 0x00 // set reporting for all levels to 0.

4.16.2. flush

Command	flush
Description	Flush the contents of the event log, including stored events.
Usage	flush
Example	***CM/Event Log >*** flush

4.16.3. log_event

Command	log_event
Description	Log an event with the specified event id to the event log. The event may have up to 4 event-specific text parameters; if your parameter has a space, then enclose it in quotes (e.g. "this is parm 1").
Usage	log_event eventId{15} [evParm1{127}] [evParm2{127}] [evParm3{127}] [evParm4{127}] [evParm5{127}]
Example	***CM/Event Log >*** log_event E102.0 basic.cfg // Logs event E102.0, with the cfg file as evParm1.

4.16.4. read

Command	read
Description	Read the event log from NV storage and rebuild the table.

Usage	read
Example	*CM/Event Log >* read

4.16.5. show

Command	show
Description	Dump the contents of the event log to the console NOTE: for more debug info, turn on log events from the SNMP directory.
Usage	show
Example	*CM/Event Log >* show

4.16.6. silent

Command	silent
Description	If set to true, makes the specified event log element silent (ie nothing printed to the console). Elements which may be specified are 'events', 'traps', and 'syslogs'. If no element is specified, then events, traps, and syslogs will all be affected.
Usage	silent true|false [element{31}]
Example	*CM/Event Log >* silent true // make events, traps, and syslogs all silent. silent true events // make events silent. silent true traps // make traps silent. silent true syslogs // make syslog messages silent.

4.16.7. stop_stress

Command	stop_stress
Description	Stop a stress test if one is in progress.
Usage	stop_stress
Example	*CM/Event Log >* stop_stress

4.16.8. stress

Command	stress
Description	Start a stress test for the selected event log object. If the silent parameter is set to true, events will not be printed to the console. If the number of events is set to 0 (or unspecified), the stress will continue until a stop_stress command is issued. Otherwise, the number of events specified will be logged, then the stress will conclude.

Usage	stress [numevents{0..65535}] [silent]
Example	***CM/Event Log >*** stress silent

4.16.9. syslog

Command	syslog
Description	Set the IP address of the syslog server. Use 0.0.0.0 to inhibit.
Usage	syslog [IpAddress]
Example	***CM/Event Log >*** syslog 11.24.4.3

4.16.10. verbose_show

Command	verbose_show
Description	Print the event log in a more verbose style that does not truncate event text
Usage	verbose_show
Example	***CM/Event Log >*** verbose_show

4.17. FTP Lite Client Commands (ftpLite) Menu

- **Located in:** After the ***CM*** prompt, types "***cd ftpLite***".

```
CM>cd ftpLite
CM/FtpLite>
```

- **Description:** FTP lite Client setting and commands.

The main instruction set:

Commands:
clear_counters, counters, debug, ftp, ip_stack, show_settings
Subdirectory:

4.17.1. clear_counters

Command	clear_counters
Description	Clear the FTP counters from the FTP transfer.
Usage	clear_counters

Example	***CM/FtpLite>*** clear_counters

4.17.2. counters

Command	counters
Description	Displays the FTP counters summary from the FTP transfer.
Usage	counters
Example	***CM/FtpLite>*** counters

4.17.3. debug

Command	debug
Description	Turns debug output on/off
Usage	debug [true | false]
Example	***CM/FtpLite>*** debug true // Turns debug messages on. debug false // Turns debug messages off.

4.17.4. ftp

Command	ftp
Description	Begins FTP to the specified IP address, using the current settings.
Usage	d ftp
Example	***CM/FtpLite>*** ftp // Initiates FTP until a key is pressed.

4.17.5. ip_*stack*

Command	ip_stack
Description	Sets the IP stack number in which the FTP client should run. If 0, then the default stack will be used. The stackNum parameter must correspond to a valid IP stack that has been installed and initialized with an IP address.
Usage	ip_stack [Number{0..255}]
Example	***CM/FtpLite>*** ip_stack 2 // Forces FTP client to run on IP stack 2.

4.17.6. show_settings

Command	show_settings
Description	Displays the current FTP Lite settings.

Usage	show_settings
Example	***CM/FtpLite>*** show_settings

4.18. Message Log *Settings* Commands (msgLog) Menu

- **Located in:** After the ***CM*** prompt, types "***cd msgLog***".

CM>cd msgLog

CM/MsgLog>

- Description: Command for changing Message Log Setting.

The main instruction set:

Commands:
Fields, severities, show_settings
Subdirectory:
[remoteAccess]

4.18.1. fields

Command	fields
Description	Displays or sets the different message fields that are enabled for display. Message field bit definitions: 0x01 -- The severity of the message (INFO, WARNING, ERROR, etc.) 0x02 -- The instance name of the object that generated the message. 0x04 -- The function/method in which the message was generated. 0x08 -- The name of the module/class in which the message was generated. 0x10 -- The system timestamp (Time of Day). 0x20 -- The thread Id. 0x40 -- The system timestamp (millisecond, in hex).
Usage	fields [Bitmask{0x7f}]
Example	***CM/MsgLog>*** fields 0 // Turns off all optional fields. fields 0x7f // Turns on all optional fields.

4.18.2. severities

Command	severities

Description	Displays or sets the different message severity levels that are enabled for display. Message logging bit definitions: 0x00000001 -- Fatal Errors 0x00000002 -- Errors 0x00000004 -- Warnings 0x00000008 -- Initialization 0x00000010 -- Function entry/exit 0x00000020 -- Informational 0x80000000 -- Disable 'always' messages 0x7fffffff -- All messages
Usage	severities [Bitmask]
Example	***CM/MsgLog>*** severities 0 // Turns off all messages (except 'always' messages). severities // Turns off all messages (including 'always' messages). 0x80000000 // Turns on all messages. severities 0x7fffffff

4.18.3. show_settings

Command	show_settings
Description	Displays the current Message Log Settings.
Usage	show_settings
Example	***CM/MsgLog>*** show_settings

4.18.4. *Remote* Access Server Commands (remoteAccess) Menu

Located in: After the ***CM/MsgLog*** prompt, types "*cd remoteAccess*".

CM/MsgLog>cd remoteAccess

CM/MsgLog/RemoteAccess>

The main instruction set:

Commands:
read_default_settings, restart_server, start_server, stop_server
Subdirectory:

4.18.4.1.read_default_settings

Command	read_default_settings
Description	Causes the Remote Access server to read and use the default settings from nonvol. Any existing connections are not changed. This undoes any configuration changes that were made at runtime (e.g. through a MIB, etc).
Usage	read_default_settings [telnet|ssh]
Example	***CM/MsgLog/RemoteAccess >*** read_default_settings telnet // gets the Telnet server to restore default settings from nonvol. read_default_settings ssh // gets the SSH server to restore default settings from nonvol.

4.18.4.2.restart_server

Command	restart_server
Description	Stops, then starts the specified Remote Access server. Any existing connections will be closed. This is the only way to get a Remote Access server to start using new settings from a Remote Access connection (e.g. change the Telnet settings from a Telnet connection).
Usage	restart_server [telnet|ssh]
Example	***CM/MsgLog/RemoteAccess >*** restart_server telnet // Restarts the Telnet server. restart_server ssh // Restarts the Telnet ssh.

4.18.4.3.start_server

Command	start_server
Description	Stops, then starts the specified Remote Access server. Any existing connections will be closed. This is the only way to get a Remote Access server to start using new settings from a Remote Access connection (e.g. change the Telnet settings from a Telnet connection).
Usage	start_server [telnet|ssh]
Example	***CM/MsgLog/RemoteAccess >*** start_server telnet // Starts the Telnet server. start_server ssh // Starts the SSH server.

4.18.4.4.stop_server

Command	stop_server
Description	Stops the specified Remote Access server if it is running. Any existing connections will be closed.

Usage	stop_server [telnet|ssh]
Example	***CM/MsgLog/RemoteAccess >*** stop_server telnet // Stops the Telnet server. stop_server ssh // Stops the SSH server.

4.18.5. all_sizes

Command	all_sizes
Description	Configures the settings for sweeping all packet sizes from 64-1518, with waiting and verification enabled. The time between pings is set to 0 ms, the verbosity is set to full, and the reply wait time is set to 1⁄2 second. The IP address is not changed.
Usage	all_sizes
Example	***CM/Ping>*** all_sizes

4.18.6. end_size

Command	end_size
Description	Sets or shows the size of the largest ping packet that will be sent (including LLC and IP header overhead). After the packet size is increased by the step amount, if it is larger than this value, then the size is reset to the start size. This must be between start_size..1518 (MTU), inclusive.
Usage	end_size [size{64..1518}]
Example	***CM/Ping>*** end_size 1518 // Sets the end size to the maximum allowed.

4.18.7. hs_nowait

Command	hs_nowait
Description	Configures the settings for doing high-speed pings (infinite), without waiting for the reply. The display verbosity is set to 2 (display only a 'p'), the time between pings is set to 0, and waiting for replies is disabled. None of the other settings are change.
Usage	hs_nowait
Example	***CM/Ping>*** hs_nowait // Sets the end size to the maximum allowed.

4.18.8. hs_wait

Command	hs_wait
Description	Configures the settings for doing high-speed pings (infinite), waiting for the reply.

	The display verbosity is set to 2 (display only a 'p'), the time between pings is set to 0, and waiting for replies is enabled. None of the other settings are changed.
Usage	hs_wait
Example	***CM/Ping>*** hs_wait // Sets the end size to the maximum allowed.

4.18.9. ip_address

Command	ip_address
Description	Sets or shows the IP address of the device to be pinged.
Usage	ip_address [IpAddress]
Example	***CM/Ping>*** ip_address 10.24.4.3

4.18.10. ip_stack

Command	ip_stack
Description	Sets the IP stack number that the pings should be sent to. If 0, then the default stack will be used. The stackNum parameter must correspond to a valid IP stack that has been installed and initialized with an IP address.
Usage	ip_stack [Number{0..255}]
Example	***CM/Ping>*** ip_stack 2 // Forces pings to go out IP stack 2.

4.18.11. ip_sweep

Command	ip_sweep
Description	Pings all IP addresses on the specified subnet, starting with the address specified, reporting success or failure for each one. It changes the ping settings so that only a single ping is sent. This is often used to discover all of the IP addresses on the subnet. The address will be incremented from 1..254, skipping .0 and .255 since these are often used for local broadcast addresses.
Usage	ip_sweep Subnet StartingIp
Example	***CM/Ping>*** ip_sweep 255.255.255.0 10.24.4.5 // Pings 10.24.4.5 through 10.24.4.254.

4.18.12. number_of_pings

Command	number_of_pings
Description	Sets or shows the number of pings to be sent. Note that 0 means infinite (you will

	need to press a key or type 'stop' to abort). A value of -1 causes the number of pings to be calculated based on the end size, start size, and step amount, so that it will span the range exactly once; num = (end - start + 1) / step.
Usage	number_of_pings [Number{-1..2147483647}]
Example	***CM/Ping>*** number_of_pings 3 // Limits the number of pings to 3. number_of_pings 0 // Sets the number of pings to infinite. number_of_pings -1 // Calculates the number of pings to span the range.

4.18.13. ping

Command	ping
Description	Begins ping the specified IP address, using the current settings. If the IP address is missing, then it uses the one that was previously set. If you specify -s, then pinging will happen int he background until you type stop. Otherwise, it will poll for a keypress.
Usage	ping [-s] [IpAddress]
Example	***CM/Ping>*** ping 10.24.4.3 // Ping until a key is pressed. ping -s // Ping until the user types stop.

4.18.14. restore_defaults

Command	restore_defaults
Description	Restores all of the options to their default values (excluding the IP address, which is not modified). The default values cause ping to behave like most host-based ping utilities (3 packets, 64 bytes, wait 5 seconds, etc.).
Usage	restore_defaults
Example	***CM/Ping>*** restore_defaults // Ping until a key is pressed. // Ping until the user types stop.

4.18.15. show_settings

Command	show_settings
Description	Displays the current ping settings.
Usage	show_settings
Example	***CM/Ping>*** restore_defaults // Ping until a key is pressed. // Ping until the user types stop.

4.18.16. start_size

Command	start_size
Description	Sets or shows the size of the first ping packet that will be sent (including LLC and IP header overhead). The packet size will be increased by the step amount for each packet. This must be between 64..end_size, inclusive.
Usage	start_size [size{64..1518}]
Example	***CM/Ping>*** start_size 64 // Sets the start size to the minimum allowed.

4.18.17. stats

Command	stats
Description	Displays the ping statistics summary from the last set of pings. This is the same summary that is displayed at the end of the pings (if verbosity is > 0).
Usage	stats
Example	***CM/Ping>*** stats // Sets the start size to the minimum allowed.

4.18.18. step_amount

Command	step_amount
Description	Sets or shows the amount that the packet size will be increased for each packet. This can be any number (including 0, which means to keep the size constant for every packet). Note that if you set it too large, then the packet size will wrap around to the start size every time, since it will never be allowed to be larger than the end_size. You can also specify a negative number which causes the ping size to start with the end size parameter and step down to the start size, then wrap back around to the end size.
Usage	step_amount [size]
Example	***CM/Ping>*** step_amount 1 // Increases the packet size by 1 each time. step_amount -1 // Decreases the packet size by 1 each time.

4.18.19. stop

Command	stop
Description	Stops the ping that is currently running. This is necessary if you used the -s parameter with ping.
Usage	stop
Example	***CM/Ping>*** stop // Stops the ping that is running.

4.18.20. time_between_pings

Command	time_between_pings
Description	Sets or shows the number of milliseconds that the ping helper will wait before sending the next ping. Note that this does not include time spent waiting for the reply or verifying it, or for time spent printing status information. The actual resolution and accuracy of this depends on the system (pSOS generally runs with a 10ms clock tick, so 10ms is the same as 15ms on that system).
Usage	time_between_pings [Milliseconds]
Example	***CM/Ping>*** time_between_pings // Waits 100ms before sending the next ping. 100

4.18.21. verbosity

Command	verbosity
Description	Sets the level of information that will be displayed while pinging. A higher number provides more information, but also slows down the rate at which pings can be sent. Most host-based ping utilities provide output equivalent to 3. For high-performance, high packet rate pings, values of 1 or 2 are best.
Usage	verbosity [Number{0..3}]
Example	***CM/Ping>*** verbosity // Disables printing of all ping status output. 0 // Displays only the summary of statistics at the end. verbosity // Displays a 'p' every second to show progress, and the summary. 1 // Displays full ping info for every packet, and the summary. verbosity 2 verbosity 3

4.18.22. verify_enable

Command	verify_enable
Description	Enables/disables verification of ping replies. If enabled, and if waiting for replies is enabled, then if a reply is received, it will verify that it matches the ping that was sent, and that all of the data is intact.
Usage	verify_enable [true|false]
Example	***CM/Ping>*** verify_enable 1 // Enables verification of the ping reply.

4.18.23. wait_enable

Command	wait_enable
Description	Enables/disables waiting for ping replies. If enabled, the ping helper will wait a number of milliseconds for the reply, and will process it if received. This is true even if the timeout is 0ms; it will always check for and process the response. If disabled, then no attempt will be made to check for or process a response. This is generally only of interest when you need to send data very quickly, with no variability.
Usage	wait_enable [true|false]
Example	***CM/Ping>*** wait_enable 1 // Enables waiting for the ping reply.

4.18.24. wait_time

Command	wait_time
Description	Sets or shows the number of milliseconds that the ping helper will wait for a ping response before continuing. This only takes effect if waiting is enabled. The actual resolution and accuracy of this depends on the system (pSOS generally runs with a 10ms clock tick, so 10ms is the same as 15ms on that system).
Usage	wait_time [Milliseconds]
Example	***CM/Ping>*** wait_time 5000 // Sets the reply wait timeout to 5 seconds.

4.19. Power Management Commands (power) Menu

- **Located in:** After the ***CM*** prompt, types "***cd power***".

CM >cd power

CM/power>

Description:

To show/change power management specific information.

The main instruction set:

Commands:
Client, debug, level, new_client, show, state
Subdirectory:

4.19.1. client

Command	client

Description	Sets the specified client parameter. If client ID is 0, the change will be made to all clients.
Usage	client [ID{0..100}] [enable|timer|state] [value{31}]
Example	***CM/power>*** client 1 enable 0 // Disable client 1. client 1 timer 10 // Sets the timer for client 1 to 10 seconds. client 1 state reduced // Sets the state for client 1 to reduced power.

4.19.2. debug

Command	debug
Description	Enable or disable power save debug.
Usage	debug [enable]
Example	***CM/power>*** debug true // Enable power save debug. debug false // Disable power save debug.

4.19.3. level

Command	level
Description	Shows or modifies the current power save level.
Usage	level [moderate|maximum]
Example	***CM/power>*** level moderate // Sets power save level to moderate reduction (slightly reduced). level maximum // Sets power save level to maximum reduction (reduced as much as possible)

4.19.4. new_client

Command	new_client
Description	Installs a new power save client using the specified parameters.
Usage	new_client String{31} reduced power priority{-1..100} full power priority{-1..100} reduced power timer{-1..1000} [async|sync]
Example	***CM/power>*** client foo 1 1 5 // Creates a new client named 'foo' with reduced/full priority of 1/1 and 5 second timer

4.19.5. show

Command	show
Description	Display power management information.

Usage	show
Example	***CM/power>*** show

4.19.6. state

Command	state
Description	Shows or modifies the current power save state.
Usage	state [reduced|full]
Example	***CM/power>*** state full // Sets power save state to full power. state reduced // Sets power save state to reduced power.

4.20. SNMP Command (snmp) Menu

- **Located in:** After the ***CM*** prompt, types "***cd snmp***".

```
CM >cd snmp
```

```
CM/SNMP>
```

- **Description**: SNMP command.

The main instruction set:

Commands:
cm_cfgfile, cm_filter_test, debug, flush, get, group_members, install_group, key_v3, log, n2m, notify_setup, parent_chain, reset, set, severities, show, test, throttle, tlv_55, trap, view_v1v2
Subdirectory:

4.20.1. cm_cfgfile

Command	cm_cfgfile
Description	Load the specified config file. If IP or path is not specified, then the settings from DOCSIS NV / dhcp settings will be used. NOTE: Only CmSnmpAgent objects support this command.
Usage	cm_cfgfile [IP] [path{254}]
Example	***CM/SNMP>*** cm_cfgfile 10.24.192.200 /home/broadcom/cu.cfg

4.20.2. cm_filter_test

Command	cm_filter_test
Description	Test the DOCSIS filters. Send the specified number of packets from the specified inbound interface to the specified outbound interface. Note that interface values refer to ifIndex values from the ifTable. NOTE: Only CmSnmpAgent objects support this command.
Usage	cm_filter_test inbound ifIndex{1..32} outbound ifIndex{1..32} [num packets {1..1000000}]
Example	***CM/SNMP>*** cm_filter_test 1 2 1 // Send a single packet from IF 1 to IF 2. debug 9 // Set the core agent to max verbosity.

4.20.3. debug

Command	debug
Description	Set the debug level of the core agent.
Usage	debug [Number{0..9}]
Example	***CM/SNMP>*** debug 0 // Set the core agent to be silent. debug 9 // Set the core agent to max verbosity.

4.20.4. flush

Command	flush
Description	Flush the specified SNMP table.
Usage	flush table_name{254}
Example	***CM/SNMP>*** set sysName.0 STRING device.vendor.com

4.20.5. get

Command	get
Description	Get the specified SNMP object. If no index is specified, gets the first instance available. To query a table entry, use [tablename].[index], not [entryname].[index].
Usage	get object.index{254}
Example	***CM/SNMP>*** get sysDescr.0 get ifTable.1

4.20.6. group_members

Command	group_members
Description	List the members of a given group.
Usage	group_members group name{254}
Example	***CM/SNMP>*** group_members iso // Lists the members of the iso (.1) group for the given agent.

4.20.7. install_group

Command	install_group
Description	Install one of the standard DOCSIS 1.1 groups. Supported groups are: docsisManager docsisOperator docsisMonitor docsisUser vacmGroupName is one of the above names. dhPublicKey is any old text string to use as the public key (no spaces). NOTE: Only SnmpV3Agent objects support this command.
Usage	install_group vacmGroupName{31} dhPublicKey{31}
Example	***CM/SNMP>*** install_group docsisManager broadcom

4.20.8. key_v3

Command	key_v3
Description	Get the SNMP v3 keys given the user name. NOTE: Only SnmpV3Agent objects support this command.
Usage	key_v3 userName{31}
Example	***CM/SNMP>*** key_v3 john

4.20.9. log

Command	log
Description	Enable or disable various kinds of SNMP logging.
Usage	log [req|nm|filt|events|thread|trap|time enable]
Example	***CM/SNMP>*** log req true // Enable SNMP request debug. log nm true // Enable SNMP NM authentication debug.

	log filt true // Enable packet filter debug. log events true // Enable event log debug. log thread true // Enable SNMP thread debug.

4.20.10. n2m

Command	n2m
Description	Print the NetToMedia mappings to the console. Note that this is a superset of the ipNetToMediaTable, because it may contain off-net entries as well as on-net ones.
Usage	n2m
Example	***CM/SNMP>*** n2m

4.20.11. notify_setup

Command	notify_setup
Description	Setup Notify Mibs to enable SNMPv3 Notify. Uses default entries. NOTE: Only SnmpV3Agent objects support this command.
Usage	notify_setup
Example	***CM/SNMP>*** notify_setup

4.20.12. parent_chain

Command	parent_chain
Description	Recurse the parent chain for the given object.
Usage	parent_chain object name{254}
Example	***CM/SNMP>*** parent_chain sysDescr // Recurses the parent chain for the sysDescr object for the given agent.

4.20.13. reset

Command	reset
Description	Reset the SNMP agent's socket.
Usage	parent_chain object name{254}
Example	***CM/SNMP>*** reset

4.20.14. set

Command	set
Description	Set the specified SNMP object to the specified value.
Usage	set object.index{254} int|uint|string|oid|ip|mac|inet_addr|hex_string value{254}
Example	***CM/SNMP>*** set sysName.0 STRING device.vendor.com

4.20.15. severities

Command	severities
Description	List SNMP message log app-specific severity bits.
Usage	severities
Example	***CM/SNMP>*** Severities

4.20.16. show

Command	show
Description	Print the SNMP agent's settings to the console.
Usage	show [agent|core|thread|traps]
Example	***CM/SNMP>*** show

4.20.17. test

Command	test
Description	This command is a placeholder to provide an easy way to add temporary commands during development. In released code this command does nothing.
Usage	test [optional IP{254}]
Example	***CM/SNMP>*** test

4.20.18. throttle

Command	throttle
Description	Set the throttle time for packet processing.
Usage	throttle [milliseconds]
Example	***CM/SNMP>*** throttle 100 // Only process 1 packet per 100 milliseconds (10 packets per second). throttle 0 // Apply no packet throttling (default).

4.20.19. **tlv_55**

Command	tlv_55
Description	Enable/disable SNMP management from the CPE. An SnmpCpeAccessEnable (TLV-55) equivalent.
Usage	tlv_55 [enable]
Example	***CM/SNMP>*** tlv_55 true

4.20.20. **trap**

Command	trap
Description	Send the specified trap type using specified snmpVersion to destIp with specified community string snmpVersion -- 2=SNMPv1 trap snmpVersion -- 2=SNMPv2c trap snmpVersion -- 3=Notify
Usage	trap coldStart|warmStart|linkDown|linkUp|authFail [snmpVersion{1..3}] [destIp] [community{254}]
Example	***CM/SNMP>*** trap coldStart 1 10.24.65.171 public // Send a COLD_START SNMPv1 trap to 10.24.65.171 with community 'public'.

4.20.21. view_v1v2

Command	view_v1v2
Description	Set the view used for SNMPv1/v2c queries for the specified agent.
Usage	view_v1v2 community{254}
Example	***CM/SNMP>*** view_v1v2 docsisManagerView

<End of Document>