



AlphaServer ES47/ES80/GS1280

Server Management Command Line Interface

CLI Reference

Version 3.0
October 2003



This document describes the Server Management Command Line Interface (CLI) to hp *AlphaServer* ES47/ES80/GS1280 systems.

The CLI provides access to the whole platform at the most fundamental level. You can display information, such as the system configuration, cabling, and environmental component status. You can update the firmware and power the system or a hard partition on or off, as well as halt and reset. You can create and delete partitions.

The CLI also provides one of the two ways you can access the SRM console or operating system running within the system or a partition.

October 2003

© 2003 Hewlett-Packard Company.

HP shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for HP products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

Regulatory Notices for Series ET2003

FCC Notice

Part 15 of the Federal Communications Commission (FCC) Rules and Regulations has established Radio Frequency (RF) emission limits to provide an interference-free radio frequency spectrum. Many electronic devices, including computers, generate RF energy incidental to their intended function and are, therefore, covered by these rules. These rules place computers and related peripheral devices into two classes, A and B, depending upon their intended installation. Class A devices are those that may reasonably be expected to be installed in a business or commercial environment. Class B devices are those that may reasonably be expected to be installed in a residential environment (i.e., personal computers). The FCC requires devices in both classes to bear a label indicating the interference potential of the device as well as additional operating instructions for the user. The rating label on the device shows which class (A or B) the equipment falls into. Class B devices have an FCC ID on the label. Class A devices do not have an FCC ID on the label. Once the class of the device is determined, refer to the following corresponding statement.

Class A Equipment

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at personal expense.

Class B Equipment

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio or television technician for help.

Modifications

The FCC requires the user to be notified that any changes or modifications made to this device that are not expressly approved by Hewlett-Packard Company may void the user's authority to operate the equipment.

Cables

Connections to this device must be made with shielded cables with metallic RFI/EMI connector hoods in order to maintain compliance with FCC rules and regulations.

Declaration of Conformity for products marked with the FCC logo – United States only

This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Taiwanese Notice

警告使用者:

這是甲類的資訊產品，在居住的環境中使用時，可能會造成射頻干擾，在這種情況下，使用者會被要求採取某些適當的對策。

Japanese Notice

この装置は、情報処理等電波障害自主規制協議会（VCCI）基準に基づくクラス A 情報装置です。この装置を家庭環境で使用すると電波障害を引き起こすことがあります。この場合には、使用者が適切な対策を講じるよう要求されることがあります。

Canadian Notice (Avis Canadien)

Class A Equipment

This Class A digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations. Cet appareil numérique de la classe A respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

Class B Equipment

This Class B digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations. Cet appareil numérique de la classe B respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

European Union Notice

Products with the CE marking comply with the EMC Directive (89/336/EEC) and the Low Voltage Directive (73/23/EEC) issued by the Commission of the European Community and if this product has telecommunication functionality, the R&TTE Directive (1999/5/EC).

Compliance with these directives implies conformity to the following European Norms (in parentheses are the equivalent international standards and regulations):

- EN55022 (CISPR 22) – Electromagnetic Interference
- EN55024 (IEC61000-4-2, 3, 4, 5, 6, 8, 11) – Electromagnetic Immunity
- EN61000-3-2 (IEC61000-3-2) – Power Line Harmonics
- EN61000-3-3 (IEC61000-3-3) – Power Line Flicker
- EN60950 (IEC60950) – Product Safety

Contents

Chapter 1	How to Access the Server Management CLI.....	1-1
1.1	Access the CLI from the SPM	1-1
1.1.1	From the SPM "Actions" Menu Item	1-1
1.1.2	From the SPM Tree View.....	1-2
1.1.3	From the SPM Platform View	1-3
1.2	Access the CLI from the AMU	1-4
1.3	Access the CLI from a PC Telnet session.....	1-4
1.4	Access the CLI from the MBM Backplane.....	1-5
1.5	Accessing the Server Management CLI from the SRM Console	1-7
Chapter 2	Command Line Interface.....	2-1
2.1	Component Addressing	2-1
2.2	Environment Variables	2-1
2.3	Partition Naming and Addressing.....	2-3
2.4	Conventions.....	2-4
2.5	Command Names and Qualifiers	2-4
2.6	Editing.....	2-5
2.7	Online Help	2-6
2.8	MBM Failsafe Loader	2-7
2.8.1	Configuration & Booting Images.....	2-7
2.8.2	Updating Firmware Images.....	2-8
2.8.3	Getting Help	2-8
2.9	CLI Output Messages.....	2-9
Chapter 3	Commands	3-1
3.1	Command Summary.....	3-1
3.2	assign component	3-3
3.3	assign memory	3-5
3.4	bootline	3-7
3.5	build fru.....	3-8
3.6	clear	3-12
3.7	clear alert.....	3-13
3.8	clear display.....	3-14
3.9	clear error	3-15
3.10	clear log	3-17
3.11	clear port.....	3-18
3.12	clear srmenv	3-19
3.13	config cables.....	3-20
3.14	connect.....	3-22
3.15	create partition.....	3-24
3.16	delete partition	3-26
3.17	deposit or d.....	3-28
3.18	disable alert	3-30
3.19	disable remote	3-31
3.20	disable test.....	3-32
3.21	el.....	3-33

3.22	enable alert	3-34
3.23	enable remote.....	3-35
3.24	enable test	3-35
3.25	examine or e.....	3-37
3.26	halt in	3-39
3.27	halt out	3-41
3.28	hangup.....	3-43
3.29	help	3-44
3.30	init modem	3-46
3.31	locate	3-47
3.32	logout.....	3-48
3.33	mbmboot or @	3-49
3.34	modify partition	3-51
3.35	power off.....	3-52
3.36	power on.....	3-54
3.37	reset.....	3-59
3.38	save partition	3-63
3.39	set.....	3-64
3.40	set alert.....	3-65
3.41	set baud.....	3-66
3.42	set dial	3-67
3.43	set escape	3-68
3.44	set flow.....	3-69
3.45	set init	3-70
3.46	set membership	3-71
3.47	set password.....	3-73
3.48	set sys_serial_num	3-74
3.49	set time	3-75
3.50	shell	3-76
3.51	show	3-77
3.52	show cables	3-78
3.53	show duo	3-80
3.54	show error.....	3-84
3.55	show fru	3-88
3.56	show grid	3-92
3.57	show log.....	3-93
3.58	show membership	3-98
3.59	show memory	3-99
3.60	show modem	3-101
3.61	show network.....	3-102
3.62	show partition	3-103
3.63	show power.....	3-106
3.64	show sys_serial_num	3-110
3.65	show system.....	3-111
3.66	show time.....	3-113
3.67	show version.....	3-114
3.68	telnet	3-117
3.69	test	3-119
3.70	test alert	3-121
3.71	test led	3-122
3.72	update.....	3-123
3.73	uptime	3-126

Index **Index-1**

Chapter 1

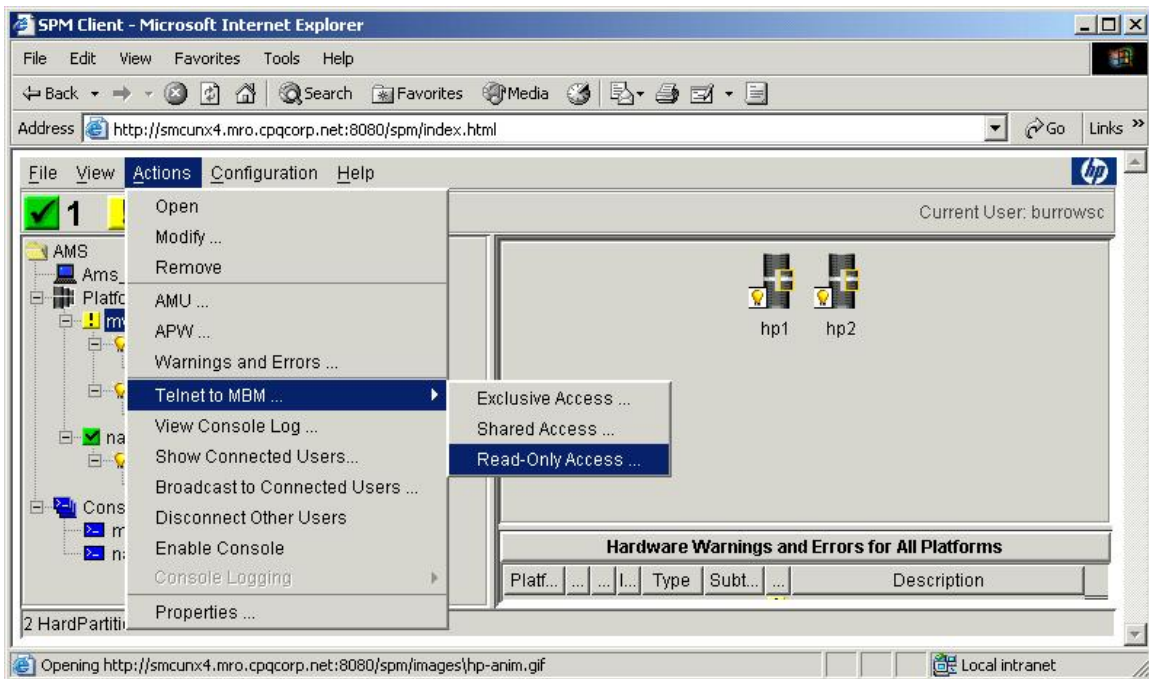
How to Access the Server Management CLI

1.1 Access the CLI from the SPM

If your site has an AlphaServer Management Station (AMS) running the Server Platform Management (SPM) software, and you have administrator or operator access to it, you can use the SPM window to telnet into the Server Management CLI of a particular platform. You can do this in three different ways:

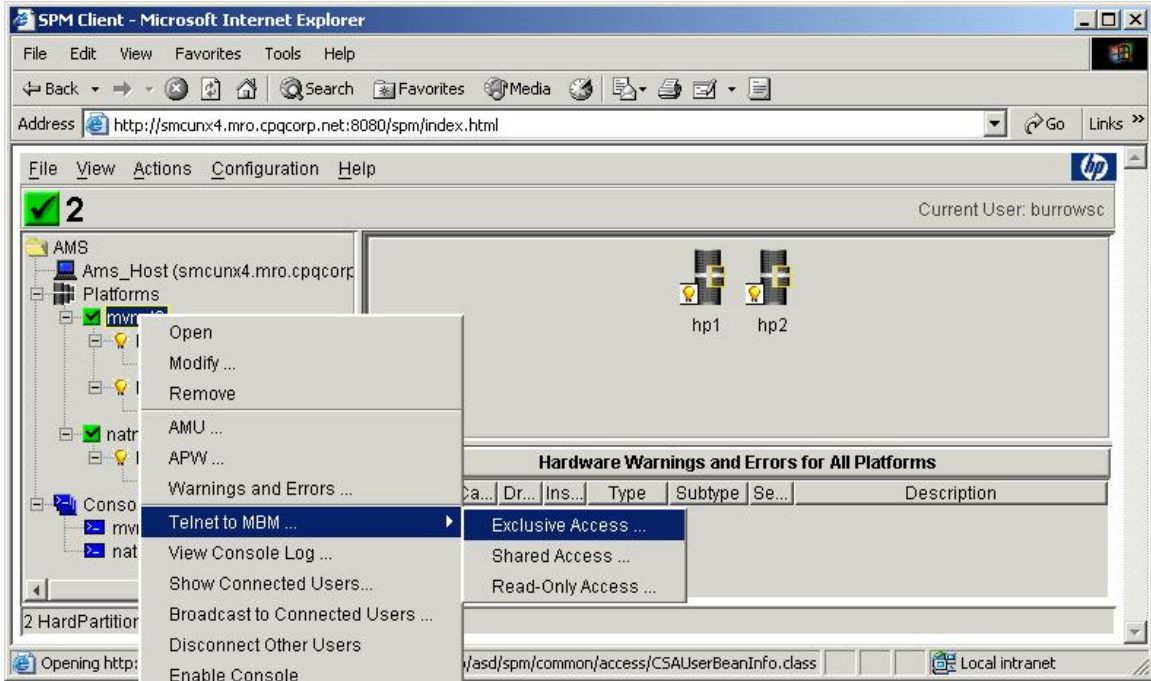
1.1.1 From the SPM "Actions" Menu Item

Once you have logged in to the SPM as a valid user, you can left-click on a platform in either the tree view or the right-hand view. Once you select a platform, you can left-click on the "Actions" menu item to telnet to the Server Management CLI of the platform you selected.



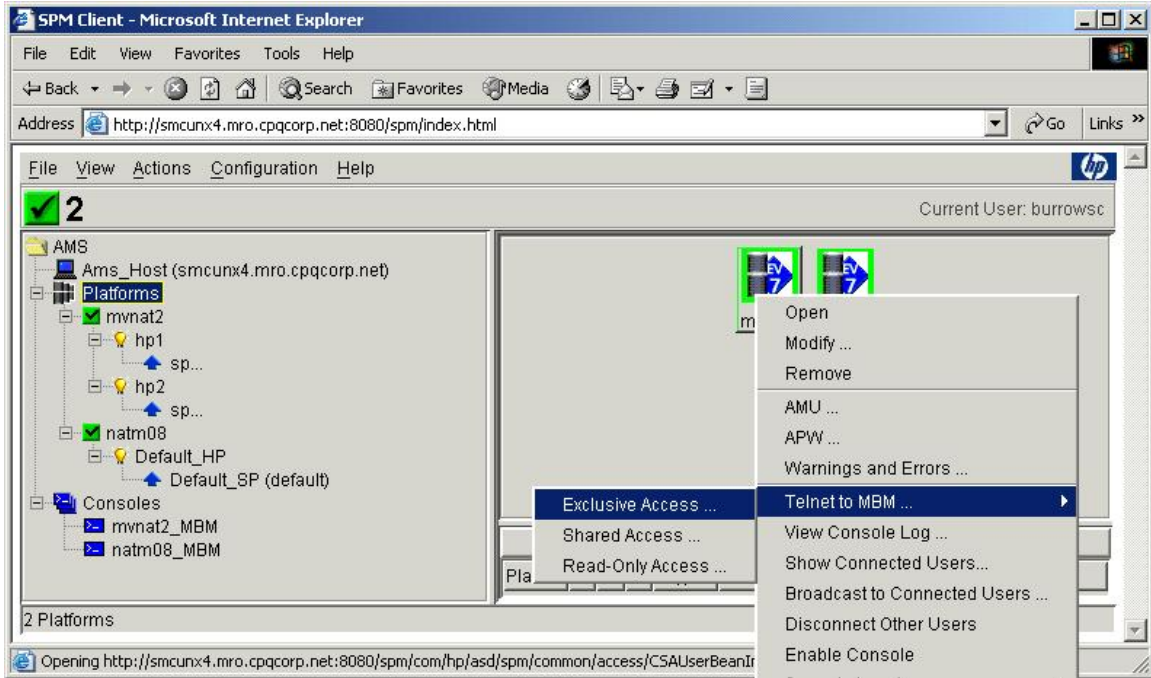
1.1.2 From the SPM Tree View

Once you have logged in as a valid SPM user, you can directly right-click on a platform in the tree view to display a pulldown menu from which you can telnet to the Server Management CLI for that system.



1.1.3 From the SPM Platform View

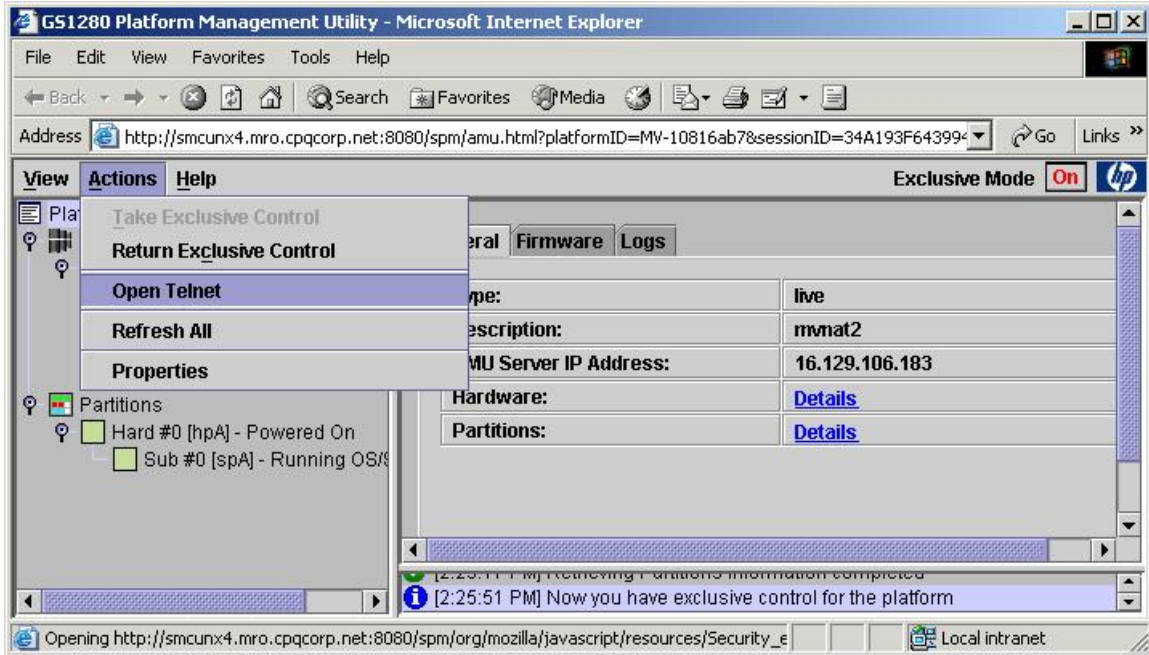
From the right pane of the SPM window, you can right-click on a platform to display a pulldown menu that allows you to telnet to the Server Management CLI of that particular platform.



1.2 Access the CLI from the AMU

You can access the Server Management CLI interface from the AlphaServer Management Utility interface as follows:

1. Click on the "Platform" icon in the tree view. (Remember that the AMU is only attached to a single platform at a time.)
2. Pull down the "Actions" menu and select "Open Telnet" from the options available, as shown below.



1.3 Access the CLI from a PC Telnet session

You can access the Server Management CLI from a PC attached to the Corporate LAN or the Multi-Server LAN by installing KEAterm or similar software on the PC.

Instructions for installing this software on your PC is given in the *ES47/ES80/GS1280 Service Manual*.

1.4 Access the CLI from the MBM Backplane

You can attach a serial terminal directly to an MBM backplane to access the Server Management CLI.

Figure 1-1 shows the serial connector for the MBM backplane on an 8P drawer.

Figure 1-1 8-P Drawer MBM Backplane Serial Connector

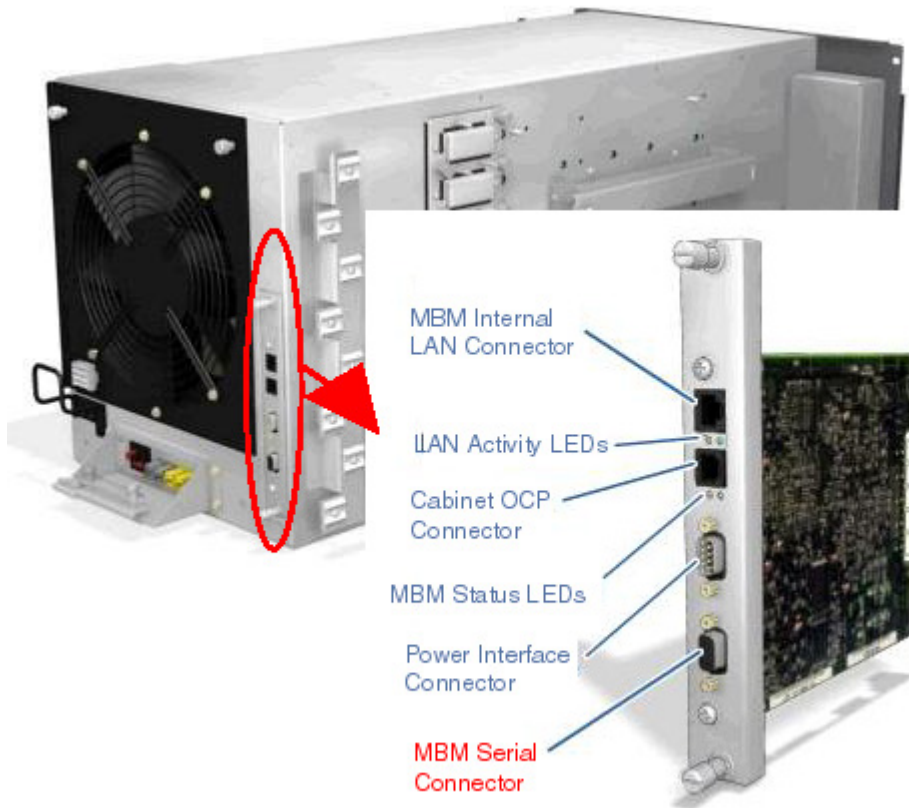
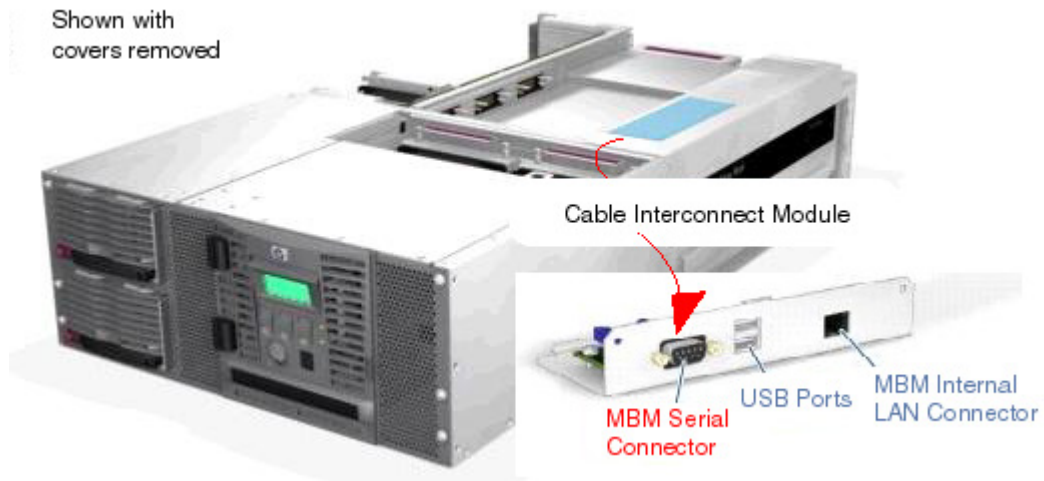


Figure 1-2 shows the serial connector for the MBM backplane on a 2P drawer.

Figure 1-2 2-P Drawer MBM Backplane Connector



1.5 Accessing the Server Management CLI from the SRM Console

If you have accessed the SRM console or operating system for a platform (or for a partition or subpartition within a platform), you can access the Server Management CLI for the platform by entering the escape sequence <Esc><Esc>MBM.

Chapter 2

Command Line Interface

The server management command line interface (CLI) performs these tasks for the ES80 and GS1280 servers:

- Displays system configuration information
- Updates firmware
- Powers on and off, halts, and resets the system or partition
- Provides system partitioning and cabling functions
- Shows the status of fans and power supplies, as well as temperatures
- Implements remote server management
- Connects to the virtual SRM console

Typically the CLI is used on smaller systems that do not use the graphical AlphaServer Management Utility (AMU).

2.1 Component Addressing

Each component is addressed by its cabinet and drawer thumbwheel in hex. The qualifiers -cabinet and -drawer are used throughout the server management CLI. For some commands, a CPU can be specified by its mesh coordinates, -ew and -ns.

Type	Address	Note
8P drawer	-cabinet <cabinet> -drawer <drawer> SBB	cabinet = 0–7, drawer = 0–3
MBM	-cabinet <cabinet> -drawer<drawer> MBM	cabinet = 0–7, drawer = 0–3
DUO	-cabinet <cabinet> -drawer<drawer> DUOn	n = 0–3 in the 8P drawer
CMM	-cabinet <cabinet> -drawer<drawer> CMMn	n = 0–3 in the 8P drawer
CPU	-cabinet <cabinet> -drawer<drawer> CPU n	n = 0–7 in the 8P drawer
CPU	-ew <ew_coord> -ns <ns_coord> CPU	by mesh coordinate (ew, ns)
IOP	-cabinet <cabinet> -drawer <drawer> IOP n	n = 0–7 in the 8P drawer; for IO cabling
PCI drawer	-cabinet <cabinet> -drawer <drawer> PCI	cabinet = 0–F, drawer = 0–F
PBM	-cabinet <cabinet> -drawer <drawer> PBM	cabinet = 0–F, drawer = 0–F
IOR	-cabinet <cabinet> -drawer <drawer> IOR n	n = 0–3 in the PCI drawer

2.2 Environment Variables

Server management environment variables are settings in the MBM, PBM, and CMM that tailor their behaviors. They are analogous to SRM environment variables.

All SM variables are global and stored in the flash. The names are case insensitive ASCII strings with maximum length of 32 characters including NULL. If the variable type is integer, please specify the value in hex.

The command “show **” lists all variables, as shown in Example 2-1.

Example 2-1 The “show *” Command Lists All Environment Variables

```

MBM> show *
cpu_enabled          FFFFFFFF
diag_mask            0
diag_min_flag        off
esc_sequence         ^ [ ^ [ MBM
halt_on_error        on
mfg_mode             off
ocp_text             GS1280
srm_auto_connect     off
MBM>

```

The table below lists environment variables.

Table 2-1 Environment Variables

SM Environment Variable Name	Type or Range	Default Value	Actions	Description
cpu_enabled	128-bit int	32 F's	Modify symbol <code>cpu_enabled</code>	Bit mask of cpus to configure into the partition
diag_mask	96-bit int	24 F's	Modify symbol <code>diag_mask</code>	Selective tests to be run
diag_min_flag	Bool On or off	Off	Modify symbol <code>min_diag_flag</code> and <code>diag_mask</code>	When non-zero, only a minimal set of diagnostics is run. Remember, this is required for semi-hard partitions.
esc_sequence	String (15 chars)	^[^[MBM	Modify symbol <code>mbm_escape_string</code>	A sequence to go back to MBM>
halt_on_error	String On or off	On	Modify symbol <code>halt_on_error</code>	On - any failure in starting a partition halts the startup attempt and returns to the prompt. Off - the server manager automatically maps out the failing components and retries starting the now degraded partition.
mfg_mode	String On or off	Off	Modify symbols <code>envmon_shutdownEnable</code> , <code>envmon_intrusionEnable</code> , and <code>envmon_overtempEnable</code>	On – under this mode, the system will never be turned off regardless of any environmental events. Also assume all covers are closed.
ocp_text	String (20 chars)	product name	Modify the new string to OCP.	Text (usually system name) to be displayed on all drawer OCPs.
part_force_stripping	Bool	Off	Modify <code>force_stripping</code>	On - All partitions are treated

	on or off			as striped. Off - The striping attributed indicated from partition creation is used. The Default_HP is created non-striped, so the force_stripping variable is handy to make that partition behave as a striped partition.
srm_auto_connect	Bool on or off	Off	Modify symbol cli_autoconnect	On - the output from the SRM/OS automatically is directed to the MBM session. At the end of the diagnostic tests, the MBM does a connect command automatically. Off – Stay at the CLI after system powers up

2.3 Partition Naming and Addressing

A key attribute of the ES80/GS1280 server is its ability to be divided into partitions, each of which appears to be a separate computer. A partition contains CPU, memory, and I/O resources.

These servers can be:

- Hard partitioned, with no routing traffic between CPUs.
- Soft partitioned, with routing between all CPUs. All memory access is allowed, and protection is accomplished through software.

The server management is responsible for creating, configuring, testing, saving, and deleting partitions.

The partition name is limited to a maximum of 19 characters and restricted to alphanumeric, underscore, and dash characters. **Free_Pool** is a reserved name for indicating the partition's pool of resources that any partition can allocate. It is legal to have the same subpartition name as the hard partition or to have the same subpartition names under different hard partitions.

A partition must be addressed by name in the format **-hp <hard_partition_name> -sp <sub_partition_name>**.

2.4 Conventions

Convention	Meaning
Fixed- font	Examples are shown in a fixed-width font for clarity and to preserve alignment of output, as they would appear on the console terminal device.
<i>Italic</i>	Italic type is used to express emphasis.
Bold	Command and option keywords embedded in narrative text sections are presented in bold type.
<item>	In command syntax descriptions, angle brackets indicate a placeholder for an item that the user must specify.
[item]	In command syntax descriptions, square brackets enclose optional parameters, qualifiers, or values.
{a, b, c}	In command syntax descriptions, braces containing items separated by commas imply mutually exclusive items. "{a, b, c}" indicates that you can choose from a, b, or c.
{a b c}	In command syntax descriptions, braces containing items separated by vertical bars imply combinatorial items. "{a b c}" indicates that you can choose any combination of a, b, and c.

2.5 Command Names and Qualifiers

The server management CLI interface is case insensitive.

Command names consist of a command verb and, in some cases, a keyword. Each command verb and keyword might have an abbreviation, which is determined by the CLI parser. For example, the minimum match for **show system** is **sh sys**. If a user continues typing more than the minimum, the optional characters must match the full name. This means that **sh syst** will work, but **sh sysa** will not. The optional characters are displayed in square brackets ([]) under the command name in the online help.

The character - introduces a qualifier. Qualifiers may be single characters ("flags") or multiple characters. The exact match has the highest precedence. A multiple character qualifier can be abbreviated as long as its partial match is unique within the command. For instance, **-c** is sufficient for the qualifier **-cabinet** if the command has no other qualifiers starting with the letter "c."

2.6 Editing

The following keys can be used in editing Server Management CLI commands:

Key	Function
Backspace	Delete the character; the cursor moves to the left
Delete	Delete the character; the cursor stays at its position
Down arrow	Get next command
End key	Moves the cursor to the end of the command line
Home key	Moves the cursor to the beginning of the command line
Insert key	Toggles to select "Insert" or "Overwrite" mode
Left arrow	Move the cursor to the left within the command line
Page Up	Reserved for future use
Page Down	Reserved for future use
Right arrow	Moves the cursor to the right within the command line
Ctrl A	Toggle to select "insert" or "overwrite" mode
Ctrl B	Recall the previous line
Ctrl C	Abort the command
Ctrl D	Cancel the current line
Ctrl E	Move the cursor to the end of the command line
Ctrl F	Move the cursor to the right within the command line
Ctrl H	Move the cursor to the beginning of the command line
Ctrl R	Reprint the command line
Ctrl U	Clear the command line

Other rules include:

- Extra spaces are ignored.
- Everything between the "#" character and the end of the line is treated as a comment.
- The previous 16 commands in the current session can be recalled by using the up arrow and down arrow keys.
- The use of double quotes is supported; everything inside the quotes is treated as-is. This is helpful for arguments containing spaces or # (comments).

For example:

```
MBM> build fru CAB0.DRW0.MBM "partnumber" "serialnum" "mbm module"
MBM> sh fru
Fru Name           E Part #           Serial #           Model/Other
CAB0.DRW0.MBM     00 partnumber     serialnum         mbm module
```

- Command abort. If you get tired of a long and complicated command display or action, type CTRL/C (^C). The command is aborted mid-stream.

For example:

```
MBM> sh fru
Fru Name                E Part #          Serial #          Model/Other
CAB0.DRW0.DOCP          00 nightlyFri_0   Fri_43
CAB0.DRW0.MBM           00 partnumber     serialnum         mbm module
CAB0.DRW0.DUO0.CMM     10 nightlyFri_2   Fri_45
CAB0.DRW0.DUO0.DUO     00 nightlyFri_3   Fri_46
~CLI-E-(tCLImodem) command aborted by Ctrl-C
MBM>
```

- Invalid commands are echoed for easy identification.

For example:

```
MBM> crete
~CLI-E-(tCLImodem) unrecognized command verb: crete
MBM> create prt
~CLI-E-(tCLImodem) unrecognized keyword: prt
Usage:  create {partition}
MBM>
```

2.7 Online Help

The brief help for each command displays the syntax and a one-line description. The synopsis line lists all possible options and arguments for the command.

By default, issuing the command **help** displays a list of supported commands, and **help *** lists all commands with their descriptions and syntax. You can also specify a particular command for help. For example:

```
MBM>help show cable

NAME
  sh[ow] cab[le]
FUNCTION
  Display the cable configuration.
SYNOPSIS
  show cable [-ip | -io]

MBM>
```

In the name line, the characters in square brackets ([]) are considered optional. As shown above, the minimum abbreviated form of **show cable** is **sh cab**.

If you issue a command with syntactic errors, an error message followed by the correct command usage is displayed, as shown here:

```
MBM>set time
~CLI-W-(tCLImodem): invalid parameters
Usage: set time <date> <time>

MBM>
```

2.8 MBM Failsafe Loader

If the main MBM firmware image becomes corrupt, the failsafe loader allows the user to recover the system. It also allows the user to load a network-bootable MBM image over the network if configured to do so.

Upon application of VAUX, the failsafe loader is the first piece of firmware to run. It performs power-up diagnostics on the MBM hardware, and then loads/runs the main image. If the main image is missing or corrupt, the failsafe loader will attempt to automatically install the update from the PMU server. If the automatic updating fails, the failsafe loader will sit and wait at the FSL CLI prompt. ("FSL> ")

2.8.1 Configuration & Booting Images

From the FSL CLI, two commands are available that are not available in the main MBM firmware image. **Bootline** will configure the boot settings of MBM. **Mbmbboot** will boot the MBM based on the settings contained in the boot line. The **bootline** menu is shown below.

```
FSL> bootline

'.' = clear field; '-' = go to previous field; ^D = quit

boot device           : flash0
processor number      : 0
host name             : host
file name             : vxWorks
inet on ethernet (e) : 10.250.250.250
inet on backplane (b):
host inet (h)        : 10.253.0.254
gateway inet (g)     :
user (u)             : target
ftp password (pw) (blank = use rsh):
flags (f)            : 0xa0
target name (tn)     :
startup script (s)   :
other (o)            : fei
```

The **bootline** options allow control over the name of the file to boot, the server to get the file from, and startup flags. To boot an image over the network, change the boot device from flash to fei. (Type "fei" and hit enter) To advance to the next option, hit enter without typing anything.

In practice, the only needed options are the boot device, the filename, the host inet (host to download from). The other settings should be left to their default values.

2.8.2 Updating Firmware Images

The failsafe loader supports updating of firmware images. In order to update an image on a micro running its failsafe loader, you must specify its cabinet, drawer, and micro type. For example, to update the main MBM image from the failsafe loader (downloading from MBM1), type the following command:

```
FSL> update mbmfw 10.1.0.1 -cabinet 0 -drawer 0 mbm
```

2.8.3 Getting Help

The failsafe loader supports online help like the normal CLI. Typing “help” at the FSL prompt will produce a list of commands included in the failsafe loader. To get help on a specific command, type “help <cmdname>”.

```
FSL> help
NAME
      help
FUNCTION
      Display information about CLI commands.
SYNOPSIS
      help [<command>, *]
      Command synopsis conventions:
      <item> Implies a placeholder for user specified item.
      <item>... Implies an item or list of items.
      [] Implies optional keyword or item.
      {a,b,c} Implies any one of a, b, c.
      {a|b|c} Implies any combination of a, b, c.
```

The following help topics are available:

clear	clear alert	clear display	clear error
clear log	clear port	d	deposit
e	e1	enable alert	enable remote
enable test	help	logout	reset
rlogin	set	set alert	set baud
set dial	set escape	set flow	set init
set password	set time	shell	show
show cables	show duo	show error	show fru
show log	show memor	show modem	show network
show power	show system	show time	show version
update	uptime		

```
FSL>
```

2.9 CLI Output Messages

The server management CLI may display messages that are not the result of a user command.

The output messages consist of a general format:

```
~Facility-Severity-(taskName): detailed messages
```

Facility	The area that detects the error
Severity	I additional information
	W warning messages
	E error messages
	F fatal operations
taskName	tCLImodem

For example, an error occurred at the user interface

```
MBM> crete
~CLI-E-(tCLImodem) unrecognized command verb: crete
MBM> create prt
~CLI-E-(tCLImodem) unrecognized keyword: prt
Usage: create {partition}
MBM> show sys -blah
~CLI-W-(tCLImodem): unrecognized qualifiers
Usage: show system

MBM>
```

Some common CLI output messages are given in the table below.

Message	Possible Cause
Common Messages	
~CLI-E: input too long	The command line is longer than 80 characters.
~CLI-E: unrecognized command verb	The first word of the command name is invalid.
~CLI-E: unrecognized keyword	The second word of the command name is invalid.
~CLI-E: unrecognized qualifiers	A command switch qualifier either didn't exist, or wasn't enough to be unique.
~CLI-E: conflicting qualifiers	The command switches/qualifiers specified were contradictory.
~CLI-E: too many parameters	Too many command switches/qualifiers were specified.
~CLI-E: insufficient parameters	Not enough command switches/qualifiers were specified.
~CLI-E: invalid parameters	The command switch was invalid
~CLI-E: illegal digit for specified radix	Characters other than 0–9 were used for integer

~CLI-E: numeric value too large and 0–F for hex
The value is overflow.

Command-Specific Messages

- ~CLI-E: illegal target address
 - ~CLI-E: illegal data
 - ~CLI-E: deposit data too large for specified type
 - ~CLI-E: invalid device name
 - ~CLI-E: string too long
 - ~CLI-E: string too short
-

Chapter 3

Commands

3.1 Command Summary

Command	Function
assign {component, memory}	Assign resources to a partition.
bootline	Change the bootline used in boot ROMs.
build fru	Write FRU-specific information to a FRU's EEPROM.
clear {<envar>, alert, display, error, log, port, srmenv}	Clear the SPMU environment variable, the 'alert pending' flag, all error fields in the specified FRU, the screen or serial port, or SRM environment variables for all partitions, for a specific hard partition, or for a specific subpartition within a hard partition.
config cables	Configure or assist cabling.
connect	Connect to the system COM port of a partition.
create partition	Create a hard or subpartition.
delete partition	Delete a hard or subpartition.
deposit or d	Write data to a memory location or CSR, IPR, or I ² C device
disable {alert, remote, test}	Disable the sending of alerts, remote access, or a diagnostic test
el	Event log - show the history of CLI inputs and outputs.
enable {alert, remote, test}	Enable the modem dial-out alert function, remote access to the system, or a diagnostic test.
examine or e	Read a memory location or CSR, IPR, or I ² C device.
halt {in, out}	Halt the entire system, CPU, or partition.
hangup	Terminate the modem or telnet connection.
help	Display the entire list of CLI commands.
init modem	Initialize the modem.
locate	Locate the specified component (system building block, PCI drawer, dynamic duo, or power supply) by blinking its LEDs.
logout	Log out of the remote host.
mbmboot or @	(Works only from the failsafe loader.) Boots the main MBM/PBM image from flash or from the network.
modify partition	Changes the attributes of a hard partition.
power {off, on}	Power off or on a system, component, or partition.
reset	Reset a system, CPU, or partition.
save partition	Save the partition data to the NVRAM.

set {<envar>, alert, baud, dial, escape, flow, init, membership, password, sys serial num, time}	Modify the server management CLI environment.
show {<envar>, cables, duo, error, fru, grid, log, membership, memory, modem, network, partition, power, sys_serial_num, system, time, version}	Show specified system information.
telnet	Log in to a remote host.
test {<test_num>, alert, led}	Start a diagnostic test, or test the modem setup or cable LEDs.
update	Update the firmware.
uptime	Show the amount of time since initialization of all or specified MBMs and PBMs.

3.2 assign component

Assigns CPU and IO resources to a partition.

Note: A CPU and its associated IOs cannot be separated across hard partitions. In order to move the IOP to a different hard partition, move its associated CPU to the targeted partition, and the IOP will automatically follow the CPU to the same destination. The IOP can be moved around between subpartitions within the same hard partition.

Syntax

```
assign component [-cabinet <cabinet> -drawer <drawer>]
                 {CPU<n>, IOP<n>, SBB}
                 -hp <hard_partition> [-sp <sub_partition>]
```

Arguments

CPU<n>	The CPU by cabinet and drawer number. <i>n</i> refers to the CPU number within the 8P drawer and ranges from 0 to 7.
IOP<n>	The IOP by cabinet and drawer number. <i>n</i> refers to the IOP of the 8P drawer and ranges from 0 to 7. A user can assign the IOP only to a subpartition.
SBB	The SBB by cabinet and drawer number.
-hp <hard_partition>	Hard partition name.
-sp <sub_partition>	Subpartition name. Assign to subpartition 0 if no subpartition name is given.

Options

-cabinet <cabinet>	Cabinet number in hex. If not specified, the number defaults to 0.
-drawer <drawer>	Drawer number in hex. If not specified, the number defaults to 0.

Examples

1. Add the entire 8P drawer to the hard partition *account_dept*.

```
MBM> assign comp -ca 0 -dr 3 sbb -hp account_dept
MBM>
```

2. Delete the IO attached to CPU2 of the subpartition *acct_usa*.

```
MBM> assign comp -ca 0 -dr 3 IOP2 -hp account_dept -sp Free_Pool
MBM>
```

3. Add CPU 1 to the subpartition *acct_usa*.

```
MBM> assign comp -ca 0 -dr 3 cpul -hp account_dept -sp acct_usa  
MBM>
```

4. Move IOP1 from a different hard partition, resulting in an error.

```
MBM> assign comp -ca 0 -dr 3 cpul -hp account_dept -sp acct_usa  
  
~CLI-E-(tCLImodem) cannot move IOP across different HP  
MBM> [2003/04/11 10:21:36]  
~PCO-W-(pco)01) Command handler rtn failed for Id:5622 cmd:41a
```

See Also

show partition, show system, create partition, assign memory

3.3 assign memory

Assigns a portion of the hard partition's physical memory to a subpartition. Explicitly assigning memory to a hard partition is unnecessary because the RIMM population associated with each CPU determines the partition's memory size.

Users cannot assign memory to the Free_Pool.

The smallest memory chunk allowed is 8MB.

To deassign memory, specify 0MB as the memory size.

Syntax

```
assign memory      <memory_size> -hp <hard_partition>
                    {-sp <sub_partition>, -com}
```

Arguments

<memory_size>	The memory size in units of GB or MB; must be multiples of 8MB.
-hp <hard_partition>	Hard partition name.
-sp <sub_partition>	Subpartition name.
-com	Assign the memory to community memory, accessible to all subpartition within the hard partition.

Options

None

Examples

1. Assign 16GB of memory to the subpartition *acct_usa*.

```
MBM> assign mem 16GB -hp account_dept -sp acct_usa
MBM>
```

2. Assign 2GB of memory to the community.

```
MBM> assign mem 16GB -hp account_dept -com
MBM>
```

3. Deassign memory from the subpartition *acct_usa*. Memory for this subpartition becomes unspecified.

```
MBM> assign mem 0MB -hp account_dept -sp acct_usa
MBM>
```

See Also

show partition, create partition, assign component

3.4 bootline

This command is usually used from the FSL. Changes the boot line used in the boot ROMs. **bootline** determines the boot device, file name, and, if it is a network boot, the TFTP IP address.

If you are configuring to boot over a network and are booting multiple micros simultaneously, configure each “inet on Ethernet” address to be unique among those micros. You can use the micro IP address.

Syntax

bootline

Arguments

None

Options

None

Example

Change the method of booting to boot from the network by modifying the boot device to be **fei** and “inet on Ethernet” to be **10.0.0.1**. (Make sure that the “host inet” IP address matches the TFTP server interface IP.)

```
FSL> bootline

'.' = clear field; '-' = go to previous field; ^D = quit

boot device      : flash0 fei
processor number : 0
host name        : host
file name        : vxWorks
inet on ethernet (e) : 10.250.250.250 10.0.0.1
inet on backplane (b):
host inet (h)    : 10.253.0.254
gateway inet (g) :
user (u)         : target
ftp password (pw) (blank = use rsh):
flags (f)        : 0xa0
target name (tn) :
startup script (s) :
other (o)        : fei

FSL>
```

See Also

mbmboot, @

3.5 build fru

Sets the part number and serial number of the FRU.

“Inaccessible” errors are usually due to IIC error.

FRU can be specified with qualifiers,
 e.g. 'build fru -cab 0 -draw 2 -duo 1 VRM0.TERM'
 or with the long format as listed in 'show fru' output,
 e.g. 'build fru CAB0.DRW2.DUO1.VRM0.TERM'.
 'build fru -rimm' for RIMM serialization.

RIMM serialization:

A unique RIMM FRU serial number in ASCII is based on the 12 character system serial number concatenated with a 4 numeric alpha sequence number (0-9 then A-Z) determined by the RIMM's location. These 4-byte “CDSIs” are defined as the following:

C - Cabinet number represented by ASCII 0-7.

D - Duo number represented by ASCII hex 0-F within the cabinet.
 It is calculated by (Drawer_num *4 + Duo_num). For example, 9 means drawer 2, duo 1. (Drawer number ranges 0-3. So is Duo number within the drawer)

R - RIMM slot in Duo. ASCII 0-9 for CPU 0 RIMM 0-9.
 ASCII A-J for CPU1 RIMM 0-9.

I - Insertion count (ASCII 0-9, then A-Z).

Firmware can't directly detect the RIMM insertion. A counter keeps track of how many times of RIMM serialization done per Duo RIMMs and is stored in the DUO EEPROM.

RIMMs are usually serialized once. If the RIMMs are moved from one system to another, the ones with mismatched sys_serial_num will be serialized upon request.

Syntax

```
build fru      {{{[-cabinet <cabinet> -drawer <drawer> -duo <duo>] <fru>}
               {cab<n>.[drw<n>.duo<n>].<fru>}}
               {{<part_num> <serial_num> [<model>]},
               {-s <offset> <byte> [<byte>...]}},
               {-rimm}
```

Arguments

fru_name	The FRUs are:	
	COCP	Cabinet OCP
	DOCP	Drawer OCP
	MBM	MBM module
	PCI	PCI backplane

PBM	PBM module
IOR	PCI IO riser
DUO	DUO module
CMM	CMM module
VRM <i>n</i> .CPU	VRM (<i>n</i> = 0-5) for CPUs
VRM <i>n</i> .RIMM	VRM (<i>n</i> = 0-4) for RIMMs
VRM0.TERM	VRM for terminator
RIMM <i>nc</i>	RIMMs (<i>n</i> = 0-9 for slot number, <i>c</i> = 0-1 for CPU processor number)
PS <i>n</i>	PCI power supply (<i>n</i> = 0-1)
PWR <i>n</i>	Power supply tray (<i>n</i> = 0-1) of SBB
PWR <i>n</i> .PS <i>p</i>	Power supply of SBB
CLM	Common Logic Module (of Superdome PS subsystem)

	DUO, CMM, VRM and RIMM require the additional qualifier “-duo” to indicate the residency.
part_num	FRU's part number. This field contains the FRU revision. (This is a 16-character ASCII string with no embedded spaces.)
serial_num	FRU's serial number. This ASCII string must be 10 characters.
model	The FRU's model name or number or alias. This ASCII string may be up to 10 characters. This field is optional and does not apply to third-party FRUs.
offset	The beginning byte offset in hex within this FRU's EEPROM, where the following data bytes are written.
byte...	Up to 16 data bytes to be written. At least one byte must be supplied after the offset.

Options

-cabinet <cabinet>	Cabinet number in hex. If not specified, defaults to 0.
-drawer <drawer>	Drawer number in hex. If not specified, defaults to 0.
-duo <duo>	Dual-CPU module number in decimal.
-s	Write raw data to the EEPROM. Used to apply any FRU-specific data. Checksum is automatically updated after the raw data is written.
-rimm	Build a unique serial number across all RIMMs. The RIMM number is based on the 12-character system serial number concatenated with a 4-digit sequence number determined by the RIMM's location. These four digits, CDSI, are defined as: C - Cabinet number 0 - 7 D - Duo number 0 - F (ASCII hex). It is calculated by (drawer_num * 4 + duo_num) S - RIMM slot 0 - 9 for CPU0, RIMM0-9; and A - J for CPU1, RIMM0-9 I - Insertion count (ASCII 0-9, then A-Z)

Examples

1. Write the manufacture data to the EEPROM of the MBM.

```
MBM> build fru -ca 0 -dr 1 MBM 54-30284-01 NI12312312
MBM>
```

2. Write the manufacture data to the EEPROM of VRM2.CPU (CPU VRM2) on dual-CPU module 3.

```
MBM> build fru -ca 0 -dr 1 -duo 3 VRM2.CPU 20-50928-05.BX01 C213100227
MBM>
or
MBM> build fru CAB0.DRW1.DUO3.VRM2.CPU 20-50928-01.BX01 C213100227
MBM>
```

3. Deposit 0x9C to the location 0x1F of IOR0 EEPROM in cabinet 1, drawer 4.

```
MBM> build fru -ca 1 -dr 4 ior0 -s 1F 9C
or
MBM> build fru CAB1.DRW4.IOR0 -s 1F 9C
MBM>
```

4. Serialize the RIMMs. Unique RIMM serial numbers usually are not programmed by the vendor.

```
MBM> sh fru
Fru Name                E Part #          Serial #           Model/Other
CAB0.DRW0.DOCPP         00 nightlyFri_0   Fri_43
CAB0.DRW0.MBM           00 partnumber     serialnum         mbm module
CAB0.DRW0.DUO0.CMM      10 nightlyFri_2   Fri_45
CAB0.DRW0.DUO0.DUO      00 nightlyFri_3   Fri_46
CAB0.DRW0.DUO0.RIMM00   40 20-1C872-01    CE
CAB0.DRW0.DUO0.RIMM10   40 20-1C872-01    CE
CAB0.DRW0.DUO0.RIMM20   40 20-1C872-01    CE
CAB0.DRW0.DUO0.RIMM30   40 20-1C872-01    CE
CAB0.DRW0.DUO0.RIMM50   40 20-1C872-01    CE
CAB0.DRW0.DUO0.RIMM60   40 20-1C872-01    CE
CAB0.DRW0.DUO0.RIMM70   40 20-1C872-01    CE
CAB0.DRW0.DUO0.RIMM80   40 20-1C872-01    CE
CAB0.DRW0.DUO0.RIMM01   40 20-1C872-01    CE
CAB0.DRW0.DUO0.RIMM11   40 20-1C872-01    CE
CAB0.DRW0.DUO0.RIMM21   40 20-1C872-01    CE
CAB0.DRW0.DUO0.RIMM31   40 20-1C872-01    CE
CAB0.DRW0.DUO0.RIMM51   40 20-1C872-01    CE
CAB0.DRW0.DUO0.RIMM61   40 20-1C872-01    CE
CAB0.DRW0.DUO0.RIMM71   40 20-1C872-01    CE
CAB0.DRW0.DUO0.RIMM81   40 20-1C872-01    CE
CAB0.DRW0.DUO0.VRM0.CPU 00 nightlyFri_4   Fri_47            7F7FF8
CAB0.DRW0.DUO0.VRM1.CPU 00 nightlyFri_5   Fri_48            7F7FF8
CAB0.DRW0.DUO0.VRM2.CPU 00 nightlyFri_6   Fri_49            7F7FF8
CAB0.DRW0.DUO0.VRM3.CPU 00 nightlyFri_7   Fri_50            7F7FF8
CAB0.DRW0.DUO0.VRM4.CPU 00 nightlyFri_8   Fri_51            7F7FF8
CAB0.DRW0.DUO0.VRM2.RIMM 00 nightlyFri_9   Fri_52            7F7FF8
CAB0.DRW0.DUO0.VRM3.RIMM 00 nightlyFri_10  Fri_53            7F7FF8
CAB0.DRW0.DUO0.VRM4.RIMM 00 nightlyFri_11  Fri_54            7F7FF8
CAB0.DRW0.DUO0.VRM0.TERM 00 nightlyFri_12  Fri_55            7F7FA2
CAB0.DRW0.DUO1.CMM      10 nightlyFri_13  Fri_56
CAB0.DRW0.DUO1.DUO      00 nightlyFri_14  Fri_57
~CLI-E-(tCLImodem) command aborted by Ctrl-C
MBM> show sys_serial_num
SYS_SERIAL_NUM          GS1280-32P
```

```

MBM> build fru -rimm
Building RIMM serial numbers...
MBM> sh fru
Fru Name                E Part #                Serial #                Model/Other
CAB0.DRW0.DOCP          00 nightlyFri_0        Fri_43
CAB0.DRW0.MBM           00 partnumber          serialnum              mbm module
CAB0.DRW0.DUO0.CMM     10 nightlyFri_2        Fri_45
CAB0.DRW0.DUO0.DUO     10 nightlyFri_3        Fri_46
CAB0.DRW0.DUO0.RIMM00  06 20-1C872-01        GS1280-32P000L       CE
CAB0.DRW0.DUO0.RIMM10  06 20-1C872-01        GS1280-32P001L       CE
CAB0.DRW0.DUO0.RIMM20  06 20-1C872-01        GS1280-32P002L       CE
CAB0.DRW0.DUO0.RIMM30  06 20-1C872-01        GS1280-32P003L       CE
CAB0.DRW0.DUO0.RIMM50  06 20-1C872-01        GS1280-32P005L       CE
CAB0.DRW0.DUO0.RIMM60  06 20-1C872-01        GS1280-32P006L       CE
CAB0.DRW0.DUO0.RIMM70  06 20-1C872-01        GS1280-32P007L       CE
CAB0.DRW0.DUO0.RIMM80  06 20-1C872-01        GS1280-32P008L       CE
CAB0.DRW0.DUO0.RIMM01  06 20-1C872-01        GS1280-32P00AL       CE
CAB0.DRW0.DUO0.RIMM11  06 20-1C872-01        GS1280-32P00BL       CE
CAB0.DRW0.DUO0.RIMM21  06 20-1C872-01        GS1280-32P00CL       CE
CAB0.DRW0.DUO0.RIMM31  06 20-1C872-01        GS1280-32P00DL       CE
CAB0.DRW0.DUO0.RIMM51  06 20-1C872-01        GS1280-32P00FL       CE
CAB0.DRW0.DUO0.RIMM61  06 20-1C872-01        GS1280-32P00GL       CE
CAB0.DRW0.DUO0.RIMM71  06 20-1C872-01        GS1280-32P00HL       CE
CAB0.DRW0.DUO0.RIMM81  06 20-1C872-01        GS1280-32P00IL       CE
CAB0.DRW0.DUO0.VRM0.CPU 00 nightlyFri_4        Fri_47                7F7FF8
CAB0.DRW0.DUO0.VRM1.CPU 00 nightlyFri_5        Fri_48                7F7FF8
CAB0.DRW0.DUO0.VRM2.CPU 00 nightlyFri_6        Fri_49                7F7FF8
CAB0.DRW0.DUO0.VRM3.CPU 00 nightlyFri_7        Fri_50                7F7FF8
CAB0.DRW0.DUO0.VRM4.CPU 00 nightlyFri_8        Fri_51                7F7FF8
CAB0.DRW0.DUO0.VRM2.RIMM 00 nightlyFri_9        Fri_52                7F7FF8
CAB0.DRW0.DUO0.VRM3.RIMM 00 nightlyFri_10       Fri_53                7F7FF8
CAB0.DRW0.DUO0.VRM4.RIMM 00 nightlyFri_11       Fri_54                7F7FF8
CAB0.DRW0.DUO0.VRM0.TERM 00 nightlyFri_12       Fri_55                7F7FA2
CAB0.DRW0.DUO1.CMM     10 nightlyFri_13       Fri_56
CAB0.DRW0.DUO1.DUO     10 nightlyFri_14       Fri_57
CAB0.DRW0.DUO1.RIMM00  06 20-1C872-01        GS1280-32P010L       CE
CAB0.DRW0.DUO1.RIMM10  06 20-1C872-01        GS1280-32P011L       CE
CAB0.DRW0.DUO1.RIMM20  06 20-1C872-01        GS1280-32P012L       CE
CAB0.DRW0.DUO1.RIMM30  06 20-1C872-01        GS1280-32P013L       CE
CAB0.DRW0.DUO1.RIMM50  06 20-1C872-01        GS1280-32P015L       CE
~CLI-E-(tCLImodem) command aborted by Ctrl-C
MBM>

```

Note: The part and serial numbers are made up in this document.

See Also

build fru, show error, clear error

3.6 clear

Clears the value of a server management environment variable and restores the default value if there is one.

Syntax

```
clear <smv>,  
      *
```

Arguments

smv The server management environment variable.

* Wildcard for all server management environment variables.

Options

None

Example

Restore the default setting of the environment variable auto_connect_srm.

```
MBM> clear srm_auto_connect  
MBM>
```

See Also

show, set

3.7 clear alert

Clears the current alert condition for the modem dial-out.

This command will cause the Server Management CLI on this platform to stop paging the remote operator through the modem. However, it does not prevent server management from issuing alerts to other platform's. If the alert for modem dial-out is not cleared, the Server Management CLI will continue to page the remote operator every 30 minutes. If the sending of alert messages is disabled, the triggering of alerts continues to occur, but nothing will be sent over the modem.

Also, the most recent alert is logged as the alert type. If the system is in secure mode (via the Operator Control Panel keyswitch), the sending of alerts is still allowed, even through remote access into the system is not.

Syntax

clear alert

Arguments

None

Options

None

Examples

This command removes any pending interrupts and allows the capturing of subsequent alert conditions. The command returns one of two messages depending on whether or not an alert is currently pending.

1. With no alerts pending.

```
MBM> clear alert
No alert pending
MBM>
```

2. With alerts pending.

```
MBM> clear alert
Pending alert cleared
MBM>
```

See Also

enable alert, test alert

3.8 clear display

Erases the current display screen on the console terminal.

Syntax

clear display

Arguments

None

Options

None

Example

Clear the terminal display.

```
MBM> clear display  
MBM>
```

3.9 clear error

Clears the TDD and SDD error log entries and checksum errors in the EEPROM located on the specified FRU. "Inaccessible" are usually due to IIC error.

Syntax

```
clear error      {[[-cabinet <cabinet> -drawer <drawer>
                  -duo <duo>] <fru_name>}, -all}
```

Arguments

fru_name The FRUs are:

COCP	Cabinet OCP
DOCP	Drawer OCP
MBM	MBM module
PCI	PCI backplane
PBM	PBM module
IOR	PCI IO riser
SIO	SIO
DUO	Dual-CPU module
CMM	CMM module
VRM n .CPU	VRM ($n = 0-5$) for CPUs
VRM n .RIMM	VRM ($n = 0-4$) for RIMMs
VRM0.TERM	VRM for terminator
RIMM nc	RIMMs ($n = 0-9$ for slot number, $c = 0-1$ for CPU processor number)
PS n	PCI power supply ($n = 0-1$)
PWR n	Power supply tray ($n = 0-1$) of SBB
PWR n .PS p	Power supply of SBB
CLM	Common Logic Module (only exists with Superdome Power Supply subsystem)

DUO, CMM, VRM and RIMM require the additional qualifier "-duo" to indicate the residency.

Options

-cabinet <cabinet>	Cabinet number in hex. If not specified, defaults to 0.
-drawer <drawer>	Drawer number in hex. If not specified, defaults to 0.
-all	All FRUs in the drawer.

Examples

1. This command clears all TDD and SDD errors from a FRU's EEPROM. Use the **show error** command to find the proper FRU name and to verify that errors have been cleared.

```

MBM> show error
FRU Name                E Symptoms
CAB0.COCP                70 FRU Checksum error between bytes 0-62
                        FRU Checksum error between bytes 64-126
                        FRU Checksum error between bytes 128-254
CAB0.DRW0.DOCP           70 FRU Checksum error between bytes 0-62
                        FRU Checksum error between bytes 64-126
                        FRU Checksum error between bytes 128-254
CAB0.DRW0.MBM            10 FRU Checksum error between bytes 0-62
CAB0.DRW0.DUO2.DUO       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DUO2.RIMM00    40 FRU checksum error between bytes 128-254
CAB0.DRW0.DUO2.RIMM10    40 FRU checksum error between bytes 128-254

CAB0.DRW1.PCI            70 FRU Checksum error between bytes 0-62
                        FRU Checksum error between bytes 64-126
                        FRU Checksum error between bytes 128-254
CAB0.DRW1.IOR0           70 FRU Checksum error between bytes 0-62
                        FRU Checksum error between bytes 64-126
                        FRU Checksum error between bytes 128-254

MBM> clear error -ca 0 COCP
MBM> clear error -ca 0 -dr 0 DOCP
MBM> clear error -ca 0 -dr 1 IOR0
MBM> show error
FRU Name                E Symptoms
CAB0.DRW0.MBM            10 FRU Checksum error between bytes 0-62
CAB0.DRW0.DUO2.DUO       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DUO2.RIMM00    40 FRU checksum error between bytes 128-254
CAB0.DRW0.DUO2.RIMM10    40 FRU checksum error between bytes 128-254

CAB0.DRW1.PCI            70 FRU Checksum error between bytes 0-62
                        FRU Checksum error between bytes 64-126
                        FRU Checksum error between bytes 128-254

```

2. Clear errors of all FRUs.

```

MBM> clear error -all
MBM> show error
Cab Dwr Duo Fru        E Symptoms
MBM>

```

See Also

show error, build fru, show fru

3.10 clear log

Erases all error log entries in the MBM and PBM flash. (It takes a few seconds to erase each flash.)

Syntax

```
clear log    {[[-cabinet <cabinet> -drawer <drawer>]
              {MBM, PBM}}, -all}
```

Arguments

None

Options

None

Examples

1. Clear the error log of the MBM in cabinet 0, drawer 0.

```
MBM> clear log -ca 0 -dr 0 MBM
Clearing error log of MBM in Cabinet 0, Drawer 0....
MBM> sh log -ca 0 -dr 0

Error log from MBM in Cabinet 0, Drawer 0
-----
~CLI-E-(tCLImodem) error log is empty

MBM>
```

2. Clear the error logs of all micros.

```
MBM> clear log -all
Clearing error log of MBM in Cabinet 0, Drawer 0....
Clearing error log of MBM in Cabinet 0, Drawer 1....
Clearing error log of MBM in Cabinet 0, Drawer 2....
Clearing error log of MBM in Cabinet 0, Drawer 3....
Clearing error log of PBM in Cabinet 1, Drawer 1....
Clearing error log of PBM in Cabinet 1, Drawer 2....
Clearing error log of PBM in Cabinet 1, Drawer 3....
Clearing error log of PBM in Cabinet 1, Drawer 4....
MBM>
```

See Also

show log

3.11 clear port

Clears the serial port.

Syntax

clear port

Arguments

None

Options

None

Example

Reset the server management CLI port.

```
MBM> clear port  
MBM>
```

See Also

set baud, set flow

3.12 clear srmenv

Clears the SRM environment variables. You can clear SRM variables for all partitions, by hard partition, or by a specific subpartition.

Syntax

```
clear srmenv {-hp <hard partition> [-sp <subpartition>], -all}
```

Arguments

None

Options

hp <hard partition>	Clear SRM environment variables for the hard partition named.
sp <subpartition>	Clear the SRM environment variables for the soft partition named. Clear all subpartitions within the hard partition if no explicit subpartition is given.
all	Clear the SRM environment variables for all partitions.

Examples

1. Clear the SRM environment variables for all subpartitions within a given hard partition.

```
MBM> clear srmenv -hp hp0  
MBM>
```

2. Clear the SRM environment variables for a given subpartition.

```
MBM> clear srmenv -hp hp0 -sp default_sp  
MBM>
```

3.13 config cables

Adds a new cable or verifies the configuration of all cables. This command assists in locating the proper connector pair to connect the cable. The LEDs at each connector are lit until the next connection is issued. This command also tests all cable connections if no parameter is given.

Syntax

```
config cable    [{-c1 <source_cabinet> -d1 <source_drawer> <source_port>}
                  [-c2 <dest_cabinet> -d2 <dest_drawer> <dest_port>]]
                  [-led {on, off}]
```

Arguments

-c1 <source_cabinet>	Cabinet number of the source 8P in hex.
-d1 <source_drawer>	Drawer number of the source 8P in hex.
-c2<dest_cabinet>	Cabinet number of the destination 8P in hex.
-d2 <dest_drawer>	Drawer number of the destination 8P or PCI drawer in hex.
source_port	Either the IP port or IOP. IP port is denoted by N02, N46, S13, S57, E67, W01. IOP is denoted by IOP<n>
dest_port	Either the IP port or PCI IOR. IP port is denoted by N02, N46, S13, S57, E67, W01. PCI IOR is denoted by IOR<n>

Options

-led {on, off} The options on or off turn the LEDs on or off. By default, the LEDs are on.

Examples

1. Verify all cable connections and turn off the LEDs.

```
MBM> config cables}
Configuring all cables...
Completed. Please use 'show cable' to view the configuration.
MBM>
```

2. Configure the cable between the south IP port of 8P drawer 1 on cabinet 0 and the north IP port of 8P drawer 0 in cabinet 1. The LEDs for both ports are lit.

```
MBM> config cables -c1 0 -d1 1 S13 -c2 0 -d2 0 N02            #Turn on LEDs
MBM> config cables -c1 0 -d1 1 S13 -c2 0 -d2 0 N02 -led off #Turn off LEDs
MBM>
```

3. Configure the cable of the south IP port of 8P drawer 1 in cabinet 0. Without providing the destination port, the cabling function finds the matching destination cable according to the mess configuration. The LEDs of both ports are lit.

```
MBM> config cables -c1 0 -d1 1 S13 #Turn on LEDs by default  
MBM>
```

4. Configure the cable between IOP3 of 8P drawer 0 on cabinet 1 and IOR0 of PCI drawer 3 on cabinet 4.

```
MBM> config cables -c1 1 -d1 0 IOP3 -c2 4 -d2 3 IOR0  
MBM>
```

See Also

show cables, test led

3.14 connect

Establishes a system COM1 session to a partition. This command starts a virtual terminal session to the primary CPU in the partition so that the user can access the SRM console or operating system. Enter the escape sequence <ESC><ESC>MBM to return to the server management CLI.

-hp and **-sp** are not needed if the system has only one subpartition (excluding Free_Pool). **-sp** is not needed if there is only one subpartition in a given hard partition.

Each session has the MBM's or PBM's IP address and a unique port number.

The SM environment variable "srm auto_connect" will control the auto connect feature. If it is set to be "on", "connect" is issued automatically upon resetting or powering on a non-partitioned system. Also, the current implementation of the connect command only handles communicating with the SRM on CPU0.

Syntax

```
connect [-hp <hard_partition> [-sp <sub_partition>]]
```

Arguments

-hp <hard_partition>	Hard partition name.
-sp <sub_partition>	Subpartition name.

Options

None

Examples

1. Connect to a non-partitioned system.

```
MBM> connect
```

```
Connecting to partition. Use the sequence ^^[^MBM to return.
```

```
P00>>>
```

2. Connect to the partition *acct_usa* and then return to the server management CLI by typing the escape sequence.

```
MBM> connect -hp account_dept -sp acct_usa
P00>>>
.
Connecting to partition. Use the sequence ^^[MBM to return.
.
P00>>>
.
.
.
P00>>> ^^[MBM
MBM>
```

3. Connect to the only subpartition within the hard partition *hr_dept*.

```
MBM> connect -hp hr_dept
.
Connecting to partition. Use the sequence ^^[MBM to return.
.
P00>>>
```

See Also

hangup

3.15 create partition

Creates a partition. This command adds a new hard partition by providing the hard partition name, maximum number of CPUs, subpartition type, and optional modifiers. It creates a subpartition if the subpartition name is given. No attributes are necessary for a subpartition. (You use assign component or assign memory to define the components of a partition or subpartition.)

If no hard partitions are defined, a Default_HP and Default_SP are created with all CPUs and IOs belonging to the Default_SP. The creation of the first user-created hard partition causes the Default_HP to be deleted. The components of the deleted Default_HP go to the Global Free_Pool. Whenever a hard partition is created, a Default_SP along with the Free_Pool is created under that hard partition. When the first user-defined subpartition is created under a hard partition, the Default_SP is deleted and its components go to the Global Free_Pool.

Syntax

```
create partition      {{-hp <hard_partition> <max_CPUs>
                     <sub_partition_type> [-stripe] },
                     {-hp <hard_partition> -sp <sub_partition>}}
```

Arguments

-hp <hard_partition>	Hard partition name. The name is limited to 19 characters (alphanumeric and underscore only) and it cannot be Default_HP or Free_Pool.
-sp <sub_partition>	Subpartition name. The name is limited to 19 characters (alphanumeric and underscore only) and it cannot be Default_SP or Free_Pool.) . A hard partition is created if no subpartition name is given.
max_CPUs	The maximum number of CPUs that can ever be placed in the hard partition. This value is used to determine the proper routing and memory algorithm.
sub_partition_type	The type of subpartitions in this hard partition: soft soft partitions firm firm partitions semi semi-firm partitions

Options

-stripe Specify memory striping.

Examples

1. Create a hard partition *account_dept* with a maximum of 8 CPUs, define any subsequently created subpartitions as soft partitions, and request memory striping.

```
MBM> create partition -hp account_dept 8 soft -stripe
MBM>
```

2. Create a subpartition under the hard partition *account_dept*.

```
MBM> create partition -hp account_dept -sp acct_usa  
MBM>
```

See Also

show partition, modify partition, delete partition, assign component, assign memory

3.16 delete partition

Destroys a partition. Returns all resources of the specified partition to the Free_Pool. The partition(s) being destroyed must be in the halt or power-off state.

When the last user-defined subpartition within a hard partition is deleted, a Default_SP within that hard partition is created with no components. When the last user-defined hard partition in the system is deleted, a Default_HP and Default_SP are created with all CPUs and IOs belonging to the Default_SP. The user cannot delete the Free_Pool.

Syntax

```
delete partition    {-hp <hard_partition>
                    [-sp <sub_partition>], -all}
```

Arguments

-hp <hard_partition>	Hard partition name.
-sp <sub_partition>	Subpartition name. The hard partition and all of its subpartitions are deleted if no explicit subpartition is given.
-all	All partitions.

Options

None

Examples

1. Delete the subpartition *asia_branch* of hard partition *account_dept*

```
MBM> delete partition -hp account_dept -sp asia_branch
MBM>
```

2. Attempt to delete the subpartition Free_Pool

```
MBM> delete part -hp account_dept -sp Free_Pool
~CLI-E-(tCLImodem) disallow deleting sub partition Free_Pool
Usage: delete partition {-hp <hard_partition> [-sp <sub_partition>],
-all}
MBM>
```

3. Delete the entire hard partition *account_dept*

```
MBM> delete partition -hp account_dept
MBM>
```

4. Delete all partitions. A Default_HP and Default_SP are created.

```
MBM> delete partition -all
MBM>
```

See Also

show partition, save partition, create partition, modify partition

3.17 deposit or d

Writes the contents of a storage location. This command enables users to write data to CPU registers, micro memory, micro IO space, or any I²C device. In order to access any I²C device, users must specify the I²C controller, sub bus, I²C address and the microprocessor it connects to.

Syntax

```
deposit    {[[-cabinet <cabinet> -drawer <drawer>]
             <device> <address> <data> [ -{b, w, l, q}]
             [-{ipr, pmem, io, -iic <ctrl>/<bus>/<iicdev>}]},
             {* <data>}, {-sym <symbol> <data>}} [-n <count>]
```

Arguments

* The set of arguments to the latest **deposit** or **examine** command is used.

device The supported devices are MBM, CMM, PBM, CPU<n>

address The address to be written to, in hex. The default is 0.

data The data to be written, in hex.

Options

-b	Perform a byte transaction.
-w	Perform a word transaction.
-l	Perform a longword transaction.
-q	Perform a quadword transaction. This is the default deposit size and also the default for GPR and IPR accesses.
-n <count>	The number of subsequent locations specified in hex. The address is incremented by the address size qualifier each time. The default is 0. This is not a sticky qualifier.
-ipr	Specifies a CPU IPR.
-pmem	Specifies the physical memory space of a device such as MBM, PBM, CMM and CPU.
-io	Specifies the IO space of a micro such as MBM, PBM and CMM.
-cabinet <cabinet>	Cabinet number in hex. If not specified, the cabinet number from the last deposit or examine command is used.
-drawer <drawer>	Drawer number in hex. If not specified, the drawer number from the last deposit or examine command is used.
-iic	iic A qualifier that introduces an I ² C device.
<ctrl>/<bus>/<iicdev>	ctrl I ² C controller.
	bus I ² C sub bus.
	iicdev I ² C device address in hex with read enable.

Examples

1. Write some value to IPR 0x11 of CPU3 in drawer 1 of cabinet 0. Then read it back.

```
MBM> deposit -ipr -ca 0 -dr 1 CPU3 11 1234123443214321
MBM> e *
0/1/cpu3/ipr:          11 12341234 43214321
MBM>
```

2. Set up configuration register 0 of LM80 on the MBM I²C bus, PS controller 1, sub bus 0 with the iic address of 50 hex.

```
MBM> deposit -ca 0 -dr 3 pbm -iic 1/0/50 0 7
```

3. Write to the MBM memory space

```
MBM> e -ca 0 -dr 2 mbm 20 -n 3 -pmem # examine MBM memory space 0/2/mbm/pmem:
      20 FC
0/2/mbm/pmem:      21 D3
0/2/mbm/pmem:      22 20
0/2/mbm/pmem:      23 00
MBM> deposit * e0 -n 3 # write to MBM memory space
MBM> e * -n 3
0/2/mbm/pmem:      20 E0
0/2/mbm/pmem:      21 E0
0/2/mbm/pmem:      22 E0
0/2/mbm/pmem:      23 E0
MBM>
```

See Also

examine, e

3.18 disable alert

Turns off the modem dial-out function. This command will only disable the modem dial-out. It does not prevent server management from issuing alerts to other platforms. Getting the remote operator's attention is essential to insure the environmental conditions do not damage the hardware.

Syntax

disable alert

Arguments

None

Options

None

Example

Turn off the modem dial-out.

```
MBM> disable alert # not yet implemnted
Disabling alert dial-out
MBM>
```

See Also

enable alert, clear alert, test alert, show modem

3.19 disable remote

Turns off remote access, preventing users from dialing in remotely.

Syntax

disable remote

Arguments

None

Options

None

Example

Do not allow the remote user to log in.

```
MBM> disable remote #not yet implemented
Disabling remote dial-in
MBM>
```

See Also

enable remote, set init, show modem

3.20 disable test

Turns off a diagnostic test. This will disable certain SRM/XSROM tests from running for debug purposes.

Syntax

disable test <test_number>

Argument

test_number The number assigned to a diagnostic test in hex.

Options

None

Example

Disable XSROM test 12, which is Data Pattern March read/write..

```
MBM> show diag_mask
diag_mask          FFFFFFFFFFFFFFFFFFFFFFFF
MBM> disable test 12
MBM> show diag_mask
diag_mask          FFFFFFFFFFFFFFFFFBFFFF
```

See Also

enable test, show

3.21 el

Displays the current event log. Use this command to review recent command history. The event log contains the most recent (up to 2 KB) events that have passed through the MBM and PBM.

Syntax

el

Arguments

None

Options

None

Example

Show the most recent activities.

```
MBM> el[2003/04/10 03:11:02]
~REC-W-(trecTask) Server management group is transitioning.
[2003/04/10 03:11:02]
~REC-W-(trecTask) Server management group is stable.
[2003/04/10 03:11:10]
~ENV-W-(envmon) PBM(cab:01, drw:2) Non Hot Plug Cover was opened
[2003/04/10 03:13:35]
~PCO-I-(pco_02) Preparing to power on partition. HP: HP0
[2003/04/10 03:13:35]
~PCO-I-(pco_03) Preparing to power on partition. HP: HP1
[2003/04/10 03:13:54]
~PCO-I-(pco_02) Running diagnostics on HP: HP0
[2003/04/10 03:13:58]
~PCO-I-(pco_03) Running diagnostics on HP: HP1
[2003/04/10 03:14:38]
~PCO-I-(pco_02) Diagnostics completed on HP: HP0
[2003/04/10 03:14:38]
~PCO-I-(pco_02) HP:HP0 SP:Default_SP Primary is NS:0 EW:0 which is cab:00 drw:0 cpu:0
[2003/04/10 03:14:39]
~PCO-I-(pco_02) Loading SRM on Primary for HP: HP0, SP: Default_SP.
[2003/04/10 03:14:43]
~PCO-I-(pco_02) Powered On HP:HP0
[2003/04/10 03:15:23]
~PCO-I-(pco_03) Diagnostics completed on HP: HP1
[2003/04/10 03:15:24]
~PCO-I-(pco_03) HP:HP1 SP:Default_SP Primary is NS:2 EW:0 which is cab:00 drw:1 cpu:0
[2003/04/10 03:15:24]
~PCO-I-(pco_03) Loading SRM on Primary for HP: HP1, SP: Default_SP.
[2003/04/10 03:15:28]
~PCO-I-(pco_03) Powered On HP:HP1
MBM>
```

See Also

clear display

3.22 enable alert

Turns on the modem dial-out function. This command is triggered whenever an alert condition occurs. An alert is sent every 30 minutes until either alerts are disabled or the alert is cleared.

Syntax

enable alert

Arguments

None

Options

None

Example

Turn on the modem dial-out.

```
MBM> enable alert #not yet implemented
Enabling alert dial-out
MBM>
```

See Also

disable alert, clear alert, test alert, show modem, set alert,
set dial

3.23 enable remote

Turns on remote access privilege. This command allows remote dial-in to the system.

Syntax

enable remote

Arguments

None

Options

None

Example

Enable remote access to the system.

```
MBM> enable remote    #not yet implemented
Enabling remote dial-in
MBM>
```

See Also

disable remote, set init, show modem

3.24 enable test

Turns on a diagnostic test.

3.24.1.1 Syntax

enable test <test_number>

3.24.1.2 Argument

test_number A diagnostic test number in hex.

3.24.1.3 Options

None

3.24.1.4 Example

Enable XSROM test 12, which is Data Pattern March read/write).

```
MBM> set diag_min_flag on
MBM> show diag_mask
diag_mask          000000010003000300030000
MBM> enable test 12
MBM> show diag_mask
diag_mask          000000010003000300070000
MBM>
```

See Also

disable test, show

3.25 examine or e

Reads the contents of a storage location. This command enables users to read the data from CPU registers, micro memory, micro IO space or any I²C device. In order to access any I²C device, a user must specify the I²C controller, sub bus, I²C address, and the microprocessor it connects to. Users can also write to any MBM/PBM symbol, mainly for debugging purposes.

Syntax

```
examine      [[[-cabinet <cabinet> -drawer <drawer>]
                <device> <address>
                [-{b, w, l, q}] [-{ipr, pmem, io,
                iic <ctrl>/<bus>/<iicdev>}]],
                {*}, {-sym <symbol>}] [-n <count>]
```

Arguments

* The latest set of arguments to the **examine** command is used.
device The supported devices are MBM, CMM, PBM, IIC, GPR, IPR
address The address to be read. The default is 0.

Options

-b	Perform a byte transaction.
-w	Perform a word transaction.
-l	Perform a longword transaction.
-q	Perform a quadword transaction. This is the default for GPR and IPR accesses.
-n <count>	The number of subsequent locations, in hex. The address is incremented by the address size qualifier each time. The default is 0.
-ipr	Specifies a CPU IPR.
-pmem	Specifies the physical memory space of a device such as MBM, PBM, CMM, or CPU.
-io	Specifies the IO space of a micro such as MBM, PBM or CMM.
-cabinet <cabinet>	Rack number in hex. If not specified, the cabinet number from the last deposit or examine command is used.
-drawer <drawer>	Box number in hex. If not specified, the drawer number from the last deposit or examine command is used.
-iic	iic A qualifier that introduces an I ² C device.
<ctrl>/<bus>/<iicdev>	ctrl I ² C controller. bus I ² C sub bus. iicdev I ² C device address in hex with read enable.

Examples

1. Write some value to IPR 0x11 of CPU3 in drawer 1 of cabinet 0. Then read it back.

```
MBM> deposit -ipr -ca 0 -dr 1 CPU3 11 1234123443214321
```

```
MBM> e *
0/1/cpu3/ipr:11 12341234 43214321
MBM>
```

2. Read the temperature of the LM80 on the MBM I²C bus, PS controller 1, sub bus 0 with the I²C address 0x50.

```
MBM> e -ca 0 -dr 3 MBM -iic 1/0/50 27 -b
0/3/mbm/iic/1/0/50: 27 19
MBM>
```

3. Read from and write to the MBM memory space.

```
MBM> e -ca 0 -dr 2 mbm 20 -n 2 -pmem # examine MBM memory space
0/2/mbm/pmем: 20 F4
0/2/mbm/pmем: 21 FB
0/2/mbm/pmем: 22 20
MBM> d * e0 -n 2 # write to MBM memory space
MBM> e * -n 2
0/2/mbm/pmем: 20 E0
0/2/mbm/pmем: 21 E0
0/2/mbm/pmем: 22 E0
MBM>
```

4. Read from and write to the MBM memory space.

```
MBM> e -sym dumppkt_enable
(28ccf8) dumppkt_enable: 0
MBM>
```

See Also

deposit, d

3.26 halt in

Asserts halt to a system with no partitions, or to a specified CPU or partition.

Syntax

```
halt in  [{{[-cabinet <cabinet> -drawer <drawer>] CPU<n>},
         {-ew <ew_coord> -ns <ns_coord> CPU},
         {-hp <hard_partition> [-sp <sub_partition>], -all}]
```

Arguments

CPU<n> The CPU by cabinet and drawer number. *n* refers to the CPU number within the 8P drawer and ranges from 0 to 7.

CPU The CPU by mesh coordinate. The qualifiers of **-ew** and **-ns** must be used in conjunction with it.

Options

-cabinet <cabinet>	Cabinet number in hex. If not specified, the number defaults to 0.
-drawer <drawer>	Drawer number in hex. If not specified, the number defaults to 0.
-ew <ew_coord>	East-West coordinate of the CPU in hex.
-ns <ns_coord>	North-South coordinate of the CPU in hex.
-hp <hard_partition>	Hard partition name.
-sp <sub_partition>	Subpartition name. All subpartitions under the hard partition are affected if none is specified.
-all	All partitions.

Examples

1. Assert system halt to a non-partitioned system or a system with only one subpartition.

```
MBM> halt in
MBM>
```

2. Assert halt to subpartition *acct_usa* of hard partition *account_dept*.

```
MBM> halt in -hp account_dept -sp acct_usa
MBM>
```

3. Halt all subpartitions in hard partition *account_dept*.

```
MBM> halt in -hp account_dept
MBM>
```

4. Halt all partitions.

```
MBM> halt in -all  
MBM>
```

5. Halt an individual CPU by its physical location (cabinet and drawer).

```
MBM> halt in -ca 0 -dr 2 CPU0  
MBM>
```

6. Halt an individual CPU by its mesh coordinate.

```
MBM> halt in -ew 1 -ns 0 CPU0  
MBM>
```

See Also

halt out

3.27 halt out

Deasserts halt to a system, CPU, or partition. The halt condition is broadcast throughout the entire system if no partition is specified.

Note: `halt` does not apply to `Free_Pool`.

Syntax

```
halt out [ {[-cabinet <cabinet> -drawer <drawer>] CPU<n>},
          {-ew <ew_coord> -ns <ns_coord> CPU},
          {-hp <hard_partition> [-sp <sub_partition>], -all} ]
```

Arguments

`CPU<n>` The CPU by cabinet and drawer number. *n* refers to the CPU number within the 8P drawer and ranges from 0 to 7.

`CPU` The CPU by mesh coordinate. The qualifiers of **-ew** and **-ns** must be used in conjunction with it.

Options

<code>-cabinet <cabinet></code>	Cabinet number in hex. If not specified, the number defaults to 0.
<code>-drawer <drawer></code>	Drawer number in hex. If not specified, the number defaults to 0.
<code>-ew <ew_coord></code>	East-West coordinate of the CPU in hex.
<code>-ns <ns_coord></code>	North-South coordinate of the CPU in hex.
<code>-hp <hard_partition></code>	Hard partition name.
<code>-sp <sub_partition></code>	Subpartition name. All subpartitions under the hard partition are affected if none is specified.
<code>-all</code>	All partitions.

Examples

1. Deassert system halt to a non-partitioned system.

```
MBM> halt out
MBM>
```

2. Deassert halt to subpartition `acct_usa` of hard partition `account_dept`.

```
MBM> halt out -hp account_dept -sp acct_usa
MBM>
```

3. Deassert halt to all subpartitions in hard partition `account_dept`.

```
MBM> halt out -hp account_dept
MBM>
```

4. Deassert halt to all partitions.

```
MBM> halt out -all  
MBM>
```

5. Deassert halt to an individual CPU by physical location (cabinet and drawer).

```
MBM> halt out -ca 0 -dr 2 CPU0  
MBM>
```

6. Deassert halt to an individual CPU by mesh coordinate.

```
MBM> halt out -ew 1 -ns 0 CPU0  
MBM>
```

See Also

halt in

3.28 hangup

Terminates a console or telnet session. This command will halt an SRM **connect** or MBM **telnet** access session.

Syntax

```
hangup    {{-hp <hard_partition> [-sp <subpartition>}},
          {[-cabinet <cabinet> -drawer <drawer>] {MBM, PBM} <port>}}
```

Arguments

-hp <hard_partition>	Hard partition name
-sp <subpartition>	Subpartition name. Optional if there is only one subpartition within the given hard partition.
MBM	The micro MBM
PBM	The micro PBM
port	The port number that the MBM/PBM is connected to. Port 23 is for CLI and port 322 for Vxworks shell.

Options

-cabinet <cabinet>	Cabinet number in hex. Defaults to 0 if not specified.
-drawer <drawer>	Drawer number in hex. Defaults to 0 if not specified.

Examples

1. Terminate the console connection.

```
MBM> hangup -hp Default_HP
MBM>
```

2. Terminate a telnet session..

```
MBM> hangup -ca 1 -dr 0 PBM 23
MBM>
```

See Also

connect, telnet

3.29 help

Displays information about server management CLI commands. Describes command conventions and available commands by default; includes a detailed description and syntax for a given command.

Syntax

help [**<command>**, *]

Arguments

command The command to be displayed.
* Display all commands.

Options

None

Examples

1. List the help topics.

```
MBM> help
NAME
    help
FUNCTION
    Display information about CLI commands.
SYNOPSIS
    help [<command>, *]
    Command synopsis conventions:
    <item> Implies a placeholder for user specified item.
    <item>... Implies an item or list of items.
    [] Implies optional keyword or item.
    {a,b,c} Implies any one of a, b, c.
    {a|b|c} Implies any combination of a, b, c.
```

The following help topics are available:

assign component	assign memory	bootline	build fru
clear	clear alert	clear display	clear error
clear log	clear port	clear srmenv	config cables
connect	create partition	d	delete partition
deposit	disable alert	disable remote	disable test
e	el	enable alert	enable remote
enable test	examine	halt in	halt out
hangup	help	init modem	locate
logout	modify partition	power off	power on
reset	save partition	set	set alert
set baud	set dial	set escape	set flow
set init	set membership	set password	set sys_serial_num
set time	show	show cables	show duo
show error	show fru	show grid	show log
show membership	show memory	show modem	show network
show partition	show power	show system	show sys_serial_num
show time	show version	telnet	test
test alert	test led	update	uptime

MBM>

2. Help on a particular command. The characters in square brackets ([]) are optional.

```
MBM> help show cable
NAME
    sh[ow] cab[le]
FUNCTION
    Display the cable configuration.
SYNOPSIS
    show cable [-ip | -io]

MBM>
```

3.30 init modem

Initializes the modem using the user-defined initialization string.

Syntax

init modem

Arguments

None

Options

None

Examples

Initialize the modem and allow remote access to the system if the initialization succeeds.

1. A modem is present and the initialization completes.

```
MBM> init modem      #not yet implemented
Initializing modem...passed
Modem initialized
MBM>
```

2. The initialization fails.

```
MBM> init modem      #not yet implemented
Error during Modem Initialization
MBM>
```

See Also

set init, show modem

3.31 locate

Locate the specified component. This command will cause LEDs to blink in the specified cabinet (blue and yellow LEDs), drawer (yellow LED), or module (yellow LED). The LEDs blink for 30 seconds at one-second intervals to help you locate the component of SBB, PCI, duo, and power.

Syntax

locate [-cabinet <cabinet> -drawer <drawer>] {SBB, PCI, DUO<n>, PWR<n>}

Arguments

SBB	The entire system building block.
PCI	The PCI drawer by cabinet and drawer number.
DUO<n>	The dynamic duo by cabinet and drawer number. The value of n refers to the dynamic duo within the 8P drawer and ranges from 0 to 3.
PWR<n>	SBB power interface module. The value n is 0 for 48-V power supply. n may change for future power supply types.

Options

-cabinet <cabinet>	Cabinet number, in hex. The cabinet number is required to identify the component and defaults to 0 if not specified.
-drawer <drawer>	Drawer number, in hex. The drawer number is required to identify the component and defaults to 0 if not specified.

Example

Find the SBB with drawer ID 1 in Cabinet 0.

```
MBM> locate -ca 0 -dr 1 sbb
MBM>
```

3.32 logout

Exits from the remote host. This command logs out of the remote host MBM or PBM. If a remote login is active (via **telnet**), it is stopped.

Syntax

logout

Arguments

None

Options

None

Example

Telnet to a remote MBM with IP address 10.254.1.0, and then log out from it.

```
MBM> telnet 10.254.1.0
Connecting to 10.254.1.0
Exit sequence is ^^[MBM.
Welcome - GS1280 Server Manager - X2.2-144

MBM> logout

Closed connection.
MBM>
```

See Also

telnet

3.33 mbmboot or @

NOTE: This command works only from within the MBM Failsafe Loader.

This command boots the main MBM/PBM image from flash or from the network.

Syntax

mbmboot

or

@

Arguments

None

Options

None

Example

Boot the main MBM image

```
FSL> mbmboot
boot device           : flash
unit number          : 0
processor number      : 0
host name             : host
file name             : vxWorks
inet on ethernet (e) : 10.250.250.250
host inet (h)         : 10.253.0.254
user (u)              : target
flags (f)             : 0xa0
other (o)             : fei

06 07 08 09 Attaching interface lo0...done

Adding 5254 symbols for standalone.

                VxWorks

Copyright 1984-1998 Wind River Systems, Inc.

                CPU: AMD SC520 CDP
                VxWorks: 5.4.2
                BSP version: 1.2/0
                Creation date: Apr 16 2002
                WDB: Ready.
```

Marver Server Management X1.0-11202 Starting up

```
Image built on Apr 16 2002 at 12:52:30
-> Running POST ...0A 0B 0C 0D 0E
Cabinet number: 0
Drawer number: 3
Micro type: 0
Node IP address: 10.3.0.1
probeQ: 2ad934 grpTask[i].RcvQ:2ad934...
...
```

See Also

bootline

3.34 modify partition

Modify existing hard partition attributes. This command will change the hard partition attributes, including maximum CPUs and memory striping.

Syntax

modify partition -hp <hard_partition> {CPU <max_CPUs | stripe {on, off}}

Arguments

-hp <hard_partition>	Hard partition name
-CPU <max_CPUs>	The maximum number of CPUs that can ever be placed in the hard partition. This value is used to determine the proper routing and memory algorithm.
-stripe {on, off}	Turn on or turn off memory striping

Options

None

Example

1. Modify hard partition of account_dept by allowing more CPUs.

```
MBM> modify partition -hp account_dept -cpu 64
MBM>
```

See Also

show partition, create partition

3.35 power off

Turns off the power to a system, component, or partition. This command powers off a non-partitioned system if no arguments are given, or it powers off a specific hard partition or component.

Use the **show power** command to confirm the power state.

Syntax

```
power off    [ {-cabinet <cabinet> -drawer <drawer>} {DUO<n>, IOR<n>, PCI,
                SBB}},
                {-hp <hard_partition>, -all} ]
```

Arguments

DUO<n> The dual-CPU module by cabinet and drawer number. *n* refers to the module within the 8P drawer and ranges from 0 to 3.

IOR<n> IOR by cabinet and drawer number. *n* refers to the IO riser within the PCI drawer and ranges from 0 to 3 for the Expanded I/O drawer.

PCI The PCI drawer by cabinet and drawer number.

SBB The entire system building block.

Options

-cabinet <cabinet>	Cabinet number in hex. If not specified, the number defaults to 0.
-drawer <drawer>	Drawer number in hex. If not specified, the number defaults to 0.
-hp <hard_partition>	Hard partition name.
-all	All partitions.

Examples

1. Power off a system with only one hard partition. This is the only case in which the **power off** command can be used without arguments.

```
MBM> power off
[2003/04/10 03:09:43]
~PCO-I-(pco 04) Preparing to power off partition. HP: Default_HP
MBM>
```

2. Issuing the **power off** command without arguments returns an error if the system has more than one hard partition.

```
MBM> p off
~CLI-E-(tCLImodem) need to specify partition or component
Usage: power off [{-cabinet <cabinet> -drawer <drawer>} {DUO<n>, PCI, SBB}},
                {-hp <hard_partition>}, {-all}]
MBM>
```

3. Power off hard partition *account_dept*.

```
MBM> power off -hp account_dept
[2003/04/10 03:09:43]
~PCO-I-(pco_01) Preparing to power off partition. HP: account_dept
MBM>
```

4. Power off all partitions.

```
MBM> power off -all
[2003/04/10 03:09:43]
~PCO-I-(pco_04) Preparing to power off partition. HP: HP0
[2003/04/10 03:09:44]
~PCO-I-(pco_00) Preparing to power off partition. HP: HP1
[2003/04/10 03:09:44]
~PCO-I-(pco_03) Preparing to power off partition. HP: HP2
[2003/04/10 03:09:45]
~PCO-I-(pco_01) Preparing to power off partition. HP: HP3
MBM>
```

5. Power off dual-processor module 3 in drawer 2 of cabinet 0.

```
MBM> power off -ca 0 -dr 2 duo3
MBM>
```

6. Power off 8P drawer 2 of cabinet 0.

```
MBM> power off -ca 0 -dr 2 sbb
MBM>
```

7. Power off PCI drawer 2 of cabinet 0xA.

```
MBM> power off -ca 1 -dr 2 pci
MBM>
```

See Also

power on, show power

3.36 power on

Turns on the power to a system, component, or partition.

If no arguments are given, this command powers on a system with one hard partition. By using arguments, you can power on a specific hard partition or component.

When there is only one hard partition, the screen display shows the entire power up sequence like the grid and diagnostics. However, if more than one hard partition is defined in the database, the hard partitions are powered on in parallel, a maximum of four hard partitions at a time. The detailed power up progress messages like the grid and the diagnostics messages, are sent to the console session and do NOT come out on the MBM session.

Given two hard partitions (Yankees and RedSox) each with two sub-partitions (Jeter and Bernie in Yankees, and Pedro and Noma in RedSox) issuing either “power on -all”, “power on -hp RedSox”, or “Power on -hp Yankees” will power up the partitions and send the output to the console session. When a user connects to the console session (`conn -hp RedSox -sp Pedro`) the powerup messages (the grid and diagnostics) will be displayed. All of the sub-partition console buffers will contain the grid and diagnostics power up messages.

Any diagnostic failures from any hard partition are delivered to the MBM session for high visibility. They also are sent to the console session.

Use the **show power** command to confirm the power state.

Syntax

```
power on [ {[-cabinet <cabinet> -drawer <drawer>] {DUO<n>, IOR<n>, PCI, SBB}},
         {-hp <hard_partition>, -all} ] [-xsrom]
```

Arguments

DUO<n> The dual-processor module by cabinet and drawer number. *n* refers to the module within the 8P drawer and ranges from 0 to 3.

IOR<n> IOR by cabinet and drawer number. *n* refers to the IO riser within the PCI drawer and ranges from 0 to 3 for the Expanded I/O shelf. (Note: Do not leave the IOR power on if its connected CPU is off.)

PCI The PCI drawer by cabinet and drawer number.

SBB The entire system building block.

Options

-cabinet <cabinet> Cabinet number in hex. If not specified, the number defaults to 0.

-drawer <drawer> Drawer number in hex. If not specified, the number defaults to 0.

-hp <hard_partition> Hard partition name.

-all All partitions.

-xsrom Power up until the level at which XROM tests complete.


```

Running test 16, Interprocessor Memory Access ... on 21 EV7s
Running test 40, Local I/O Device Interrupts ... on 22 EV7s
Running test 41, Local Interval Timer Interrupts ... on 22 EV7s
Running test 42, Local Interprocess Interrupts ... on 22 EV7s
Running test 43, Software Alerts ... on 1 EV7s
Running test 46, Other Local Interrupt Bits ... on 22 EV7s
[2003/04/16 00:08:40]
~PCO-I-(pco_01) Diagnostics completed on HP: Default_HP
[2003/04/16 00:08:40]
~PCO-I-(pco_01) HP:Default_HP SP:Default_SP Primary is NS:0 EW:0 which is
cab:00 drw:0 cpu:0
Running test 50, Loop on Secondary Routine ... on 21 EV7s (SP:Default_SP)
[2003/04/16 00:08:41]
~PCO-I-(pco_01) Loading SRM on Primary for HP: Default_HP, SP: Default_SP.
Running test 50, Loop on Secondary Routine ... on 0 EV7s
(SP:Free_Pool)[2003/04/16 00:08:45]
~PCO-I-(pco_01) Powered On HP:Default_HP
MBM>

```

2. Issuing the **power on** command without arguments returns an error if the system has more than one hard partition.

```

MBM> p on
~CLI-E-(tCLImodem) need to specify partition or component
Usage: power on [{"-cabinet <cabinet> -drawer <drawer>"} {DUO<n>, PCI, SBB}],
                [{"-hp <hard_partition>, -all"} | [-xsrom]]
MBM>

```

3. Power on hard partition *account_dept* in a multiple-partitioned system:

```

MBM> power on -hp account_dept
[2003/04/16 00:14:56]
~PCO-I-(pco_02) Preparing to power on partition. HP: account_dept
[2003/04/16 00:15:15]
~PCO-I-(pco_02) Running diagnostics on HP: account_dept
[2003/04/16 00:15:31]
~DIA-W-(pco_02) Test 17 [T12 with isolation info (MFG)] failed on cpu [NS: 0 EW:
1] which is cab:00 drw:0 cpu:2
[2003/04/16 00:16:21]
~PCO-I-(pco_02) Diagnostics completed on HP: account_dept
[2003/04/16 00:16:21]
~PCO-I-(pco_02) HP:account_dept SP:Default_SP Primary is NS:0 EW:0 which is
cab:00 drw:0 cpu:0
[2003/04/16 00:16:21]
~PCO-I-(pco_02) Loading SRM on Primary for HP: account_dept, SP: Default_SP.
[2003/04/16 00:16:25]
~PCO-I-(pco_02) Powered On HP:account_dept
MBM>

```

4. Power on all partitions.

```

MBM> power on -all
[2003/04/14 14:13:30]
~PCO-I-(pco_01) Preparing to power on partition. HP: hp0
[2003/04/14 14:13:31]
~PCO-I-(pco_02) Preparing to power on partition. HP: hp1
[2003/04/14 14:13:45]
~PCO-I-(pco_02) Running diagnostics on HP: hp1
[2003/04/14 14:13:48]
~PCO-I-(pco_01) Running diagnostics on HP: hp0
[2003/04/14 14:14:45]
~PCO-I-(pco_02) Diagnostics completed on HP: hp1
[2003/04/14 14:14:45]~PCO-I-(pco_02) HP:hp1 SP:Default_SP Primary is NS:2 EW:0
which is cab:00 drw:1 cpu:0
[2003/04/14 14:14:45]~PCO-I-(pco_02) Loading SRM on Primary for HP: hp1, SP:
Default_SP.

```


8. Power on PCI drawer 2 of cabinet 1.

```
MBM> power on -ca 1 -dr 2 pci  
MBM>
```

See Also

power off, show power, reset, connect

3.37 reset

Resets a CPU or a partition. This command resets the entire system if it is not partitioned. For a partitioned system, the user must specify which partition to reset. It also enables a user to selectively reset a CPU.

When there is only one hard partition, the screen display shows the entire reset sequence like the grid and diagnostics. However, if more than one hard partition is defined in the database, the hard partitions are reset on in parallel, a maximum of four hard partitions at a time. The detailed progress messages like the grid and the diagnostics are sent to the console session and do NOT come out on the CLI session. When a user connects to the console session, the progress messages will be displayed. Any diagnostic failures from any hard partition are delivered to the CLI session for high visibility. They also are sent to the console session.

NOTE: Subpartitions cannot be reset individually.

Syntax

```
reset [ [-cabinet <cabinet> -drawer <drawer>] CPU<n>},
      {-ew <ew_coord> -ns <ns_coord> CPU},
      {{-hp <hard_partition>, -all} [-xsrom] }
      {-micro {[-cabinet <cabinet> -drawer <drawer>] <micro>}, -all}]
```

Arguments

CPU<n> The CPU by cabinet and drawer number. *n* refers to the CPU number within the 8P drawer and ranges from 0 to 7.
 CPU The CPU by mesh coordinate. The qualifiers **-ew** and **-ns** must be used.
 micro The micro to be warm reset. It can be CMM<n>, MBM or PBM.

Options

-cabinet <cabinet>	Cabinet number in hex. If not specified, the number defaults to 0.
-drawer <drawer>	Drawer number in hex. If not specified, the number defaults to 0.
-ew <ew_coord>	East-West coordinate of the CPU in hex.
-ns <ns_coord>	North-South coordinate of the CPU in hex.
-hp <hard_partition>	Hard partition name.
-micro	Specifies resetting micro.
-all	All partitions or all micros.
-xsrom	Reset the system or partition(s) to the level at which XROM tests complete.

Examples

1. Reset a system with only one hard partition.

```

MBM> reset
[2003/04/16 00:06:30]
~PCO-I-(pco_01) Preparing to reset partition. HP: Default_HP
[2003/04/16 00:06:43]
~PCO-I-(pco_01) Configuring for 22 CPUs for HP:0 Default_HP
  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
  .w..w..w..w..w..w..w..w.....
0  wP--P--P--F--P--P--P--Fw.....
  .|..|..|..|..|..|..|..|.....
  .|..|..|..|..|..|..|..|.....
1  wP--P--P--F--P--P--P--Fw.....
  .|..|..|..|..|..|..|..|.....
  .|..|..|..|..|..|..|..|.....
2  wP--P--P--F--P--P--F--Fw.....
  .|..|..|..|..|..|..|..|.....
  .|..|..|..|..|..|..|..|.....
3  wP--P--P--F--P--P--F--Fw.....
  .w..w..w..w..w..w..w..w.....
  .|..|..|..|..|..|..|..|.....
  .|..|..|..|..|..|..|..|.....
4  .....
  .....
5  .....
  .....
6  .....
  .....
7  .....
  .....
  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
[2003/04/16 00:06:54]
~PCO-I-(pco_01) Running diagnostics on HP: Default_HP
Running test 10, Initialize RAMBUS ... on 22 EV7s
Running test 11, Initialize Memory ... on 21 EV7s
Running test 12, Data Pattern March read/write ... on 21 EV7s
Running test 13, RAID channel Test ... on 11 EV7s
Running test 14, Single Bit Error ... on 21 EV7s
Running test 15, Double Bit Error ... on 21 EV7s
Running test 20, Init IO7 and Start Clocks ... on 4 EV7s
Running test 21, IO7 Data Path (Scratch CSR) ... on 4 EV7s
Running test 22, IO Single Bit Error checkers ... on 4 EV7s
Running test 23, IO Double Bit Error checkers ... on 4 EV7s
Running test 24, IO Timer Expirations ... on 4 EV7s
Running test 25, IO up-hose SBE checkers ... on 4 EV7s
Running test 26, IO up-hose DBE checkers ... on 4 EV7s
Running test 27, IO7 pass2 data mover test ... on 4 EV7s
Running test 30, Configure RBOX Routes ... on 22 EV7s
Running test 31, Clear Errors / Enable Routes ... on 22 EV7s
Running test 32, Route Test: N S E W ... on 22 EV7s (North)
Running test 32, Route Test: N S E W ... on 22 EV7s (South)
Running test 32, Route Test: N S E W ... on 22 EV7s (East )
Running test 32, Route Test: N S E W ... on 22 EV7s (West )
Running test 33, Inverse Route Setup ... on 22 EV7s
Running test 32, Route Test: N S E W ... on 22 EV7s (North)
Running test 32, Route Test: N S E W ... on 22 EV7s (South)
Running test 32, Route Test: N S E W ... on 22 EV7s (East )
Running test 32, Route Test: N S E W ... on 22 EV7s (West )
Running test 33, Inverse Route Setup ... on 22 EV7s
Running test 34, Single Bit Error checker ... on 22 EV7s
Running test 35, Double Bit Error checker ... on 22 EV7s
Running test 31, Clear Errors / Enable Routes ... on 22 EV7s
Running test 16, Interprocessor Memory Access ... on 21 EV7s
Running test 40, Local I/O Device Interrupts ... on 22 EV7s
Running test 41, Local Interval Timer Interrupts ... on 22 EV7s

```

```

Running test 42, Local Interprocess Interrupts ... on 22 EV7s
Running test 43, Software Alerts ... on 1 EV7s
Running test 46, Other Local Interrupt Bits ... on 22 EV7s
[2003/04/16 00:08:40]
~PCO-I-(pco_01) Diagnostics completed on HP: Default_HP
[2003/04/16 00:08:40]
~PCO-I-(pco_01) HP:Default_HP SP:Default_SP Primary is NS:0 EW:0 which is
cab:00 drw:0 cpu:0
Running test 50, Loop on Secondary Routine ... on 21 EV7s (SP:Default_SP)
[2003/04/16 00:08:41]
~PCO-I-(pco_01) Loading SRM on Primary for HP: Default_HP, SP: Default_SP.
Running test 50, Loop on Secondary Routine ... on 0 EV7s (SP:Free_Pool)
MBM>

```

2. Try to reset a system with more than one hard partition.

```

MBM> reset
~CLI-E-(tCLImodem) need to specify partition or component
Usage: reset [{[-cabinet <cabinet> -drawer <drawer>] CPU<n>},
           {-ew <ew_coord> -ns <ns_coord> CPU},
           {{-hp <hard_partition>, -all} [-xsrom]},
           {-micro {[-cabinet <cabinet> -drawer <drawer>]CMM<n>,PBM,MBM},
            -all}]
MBM>

```

3. Reset hard partition *account_dept* and stop after the XSROM tests.

```

MBM> reset -hp account_dept -xsrom
[2003/04/16 00:14:56]
~PCO-I-(pco_02) Preparing to reset partition. HP: account_dept
[2003/04/16 00:15:15]
~PCO-I-(pco_02) Running diagnostics on HP: account_dept
[2003/04/16 00:16:21]
~PCO-I-(pco_02) Diagnostics completed on HP: account_dept
[2003/04/16 00:16:21]
~PCO-I-(pco_02) HP:account_dept SP:Default_SP Primary is NS:0 EW:0 which is
cab:00 drw:0 cpu:0
[2003/04/16 00:16:21]
~PCO-I-(pco_02) Loading SRM on Primary for HP: account_dept, SP: Default_SP.
MBM>

```

4. Reset all partitions.

```

MBM> reset -all
[2003/04/14 14:13:30]
~PCO-I-(pco_01) Preparing to reset partition. HP: hp0
[2003/04/14 14:13:31]
~PCO-I-(pco_02) Preparing to reset partition. HP: hp1
[2003/04/14 14:13:45]
~PCO-I-(pco_02) Running diagnostics on HP: hp1
[2003/04/14 14:13:48]
~PCO-I-(pco_01) Running diagnostics on HP: hp0
[2003/04/14 14:14:45]
~PCO-I-(pco_02) Diagnostics completed on HP: hp1
[2003/04/14 14:14:45]
~PCO-I-(pco_02) HP:hp1 SP:Default_SP Primary is NS:2 EW:0 which is
cab:00 drw:1 cpu:0
[2003/04/14 14:14:45]
~PCO-I-(pco_02) Loading SRM on Primary for HP: hp1, SP: Default_SP.
[2003/04/14 14:14:48]
~PCO-I-(pco_01) Diagnostics completed on HP: hp0
[2003/04/14 14:14:48]
~PCO-I-(pco_01) HP:hp0 SP:Default_SP Primary is NS:0 EW:0 which is
cab:00 drw:0 cpu:0

```

```
[2003/04/14 14:14:48]
~PCO-I-(pco_01) Loading SRM on Primary for HP: hp0, SP: Default_SP.
MBM>
MBM> conn -hp hp0 -sp default_sp

Connecting to partition. Use the sequence ^^[MBM to return.
~PCO-I-(pco_01) Preparing to reset partition. HP: hp0
~PCO-I-(pco_01)

Configuring for 8 CPUs for HP:0 hp0
. . . . (more progress messages)
.
P00>>>
```

5. Reset CPU 3 in drawer 2 of cabinet 0.

```
MBM> reset -ca 0 -dr 2 cpu3
MBM>
```

6. Reset the CPU with mesh coordinate (2, 3).

```
MBM> reset -ew 2 -ns 3 cpu
MBM>
```

7. Reset micro CMM 0 of drawer 2 in cabinet 0.

```
MBM> reset -ca 0 -dr 2 cmm0 -micro
MBM>
```

8. Reset all micros.

```
MBM> reset -micro -all
Resetting all micros....
MBM>
```

See Also

power on, connect

3.38 save partition

Saves the partition database. This command saves the volatile partition state to the non-volatile storage NVRAM. All partition data is saved if no explicit partition is given. An entire hard partition must be saved; a particular subpartition cannot.

Syntax

```
save partition          [-hp <hard_partition>, -all]
```

Arguments

None

Options

-hp <hard_partition>	Hard partition name.
-all	All partitions.

Examples

1. Save all subpartitions under the hard partition *account_dept*.

```
MBM>save partition -hp account_dept
MBM>
```

2. Save all partitions.

```
MBM>save partition
    or
MBM>save partition -all
MBM>
```

See Also

show partition, create partition, delete partition

3.39 set

Sets the value of a server management environment variable. This command is analogous to the SRM **set** command; however, these settings are used to control the behavior of the Server Management subsystem.

If the variable type is "integer", specify values in hex.

Note: See the section on environment variables for details.

Syntax

```
set <smv> <value>
```

Arguments

smv	Variable name. Names are case insensitive and can consist of a maximum of 32 characters including NULL.
value	The value assigned to the variable. If the variable is of type integer, specify values in hex.

Options

None

Example

Set the feature that automatically connects to the SRM during power up.

```
MBM> set srm_auto_connect 1  
MBM>
```

See Also

show, clear

3.40 set alert

Sets the modem alert string. This string should be set to a meaningful value, such as the system serial number or remote access phone number. The maximum string length is 63 characters. The alert string is sent to the modem after the dial string.

Syntax

```
set alert [<alert_string>]
```

Argument

alert_string	The ASCII string that is sent to the modem when an alert condition is triggered. It is sent after the dial string is sent. Issuing the set alert command without an alert string causes the current alert string to be erased.
--------------	---

Options

None

Example

Set up the alert string.

```
MBM> set alert 15085551212#;  
MBM00
```

Note: The # and ; characters are required to terminate the alert string. Without them the alert may not be sent to a numeric paging system and dial-in may be disabled.

See Also

set dial, enable alert, test alert, show modem

3.41 set baud

Changes the serial port baud rate. This command enables users to change the baud rate of the modem port. The baud rate is nonvolatile.

Syntax

set baud <baud_rate>

or

set <modem_baud> <baud_rate>

Argument

baud_rate The baud rate for the specified channel; one of 2400, 3600, 4800, 7200, 9600, 19200, 38400, and 57600.

Options

None

Example

Set the baud rate of the modem port to 38.4k.

```
MBM> set baud 38400  
MBM>
```

See Also

clear port, set flow, show modem

3.42 set dial

Sets the modem dial string. The dial string is the target phone number and any appropriate modem commands required by the target. It is nonvolatile across system resets and power cycles. The maximum string length is 63 characters. The dial string is sent to the modem before the alert string.

Because modem commands disallow mixed cases, all alphabetic characters in the dial string are converted to uppercase.

Syntax

```
set dial    [<dial_string>]
```

Argument

dial_string	The ASCII string that is sent to the modem when an alert condition is triggered. If no argument is given, the dial string is erased.
-------------	--

Options

None

Example

Set up the dial string to call a numeric paging service.

```
MBM> set dial atdt9,15085551212#;  
MBM>
```

See Also

set alert, enable alert, test alert, show modem

3.43 set escape

Changes the server management CLI escape sequence.

This command enables the user to change the escape sequence used for invoking the server management CLI. The escape sequence can be any character string, not to exceed 15 characters, and it is case insensitive. A typical sequence consists of two or more control characters. It is recommended that control characters be used in preference to ASCII characters. The escape sequence is stored in the MBM's or PBM's EEPROM. The default escape sequence is `^[^]SPMU`. The escape key on a PC keyboard is equivalent to `^[` (control left-bracket). Either `<esc>` or `^[` is acceptable.

Use the **show modem** command to confirm the escape sequence before exiting the server management CLI.

Syntax

set escape [`<esc_sequence>`]

Argument

`esc_sequence` The escape sequence for entering the server management CLI. The default escape sequence is restored if no argument is provided.

Options

None

Example

The following example consists of two instances of the Esc key and the letters **FUN**. The **F** is not displayed when you set the escape sequence because it is preceded by the escape character.

```
MBM> set escape un
MBM>
```

See Also

show modem

3.44 set flow

Changes the flow control of the modem port. The flow control type is nonvolatile.

Syntax

set flow <flow_control>

or

set <modem_flow> <flow_control>
Where modem_flow is a server management environment variable.

Argument

flow_control The supported flow control types for the modem port: **none**, **software**, **hardware**, or **both**.

Options

None

Example

Set the flow control of the modem port.

```
MBM> set flow hardware  
MBM>
```

See Also

set baud, clear port, show modem

3.45 set init

Sets the initialization string that is sent to the modem when the **enable remote** command is issued. The initialization string is stored in the MBM's EEPROM and has a maximum length of 31 characters.

Because modem commands disallow mixed cases, all alphabetic characters entered in the init string are converted to uppercase. It also automatically configures the modem's flow control according to the flow control setting and enables the modem carrier detect features to monitor the modem connectivity.

Syntax

```
set init [<init_string>]
```

Argument

init_string	The ASCII string that is sent to the modem when the system management CLI attempts to initialize it for remote access and sending of alerts. The init string is erased if no parameter is given.
-------------	--

Options

None

Example

Set up the initialization string.

```
MBM> set init ate0v0&c1s0=3  
MBM>
```

See Also

enable remote, show modem

3.46 set membership

Set the membership for the MBM/PBM micros.

This command will allow the user to add, subtract, or auto detect the set of micros in the system configuration. Use the **set membership** command to update the current set of micros found on the private LAN.

Syntax

```
set membership      {{-add [-cabinet <cabinet> -drawer <drawer>] {MBM, PBM}},
                   {-sub [-cabinet <cabinet> -drawer <drawer>] {MBM, PBM}},
                   {-auto}}
```

Arguments

- MBM** The micro MBM. This is required only if the add or subtract option is specified in conjunction with the cabinet and drawer.
- PBM** The micro PBM. This is required only if the add or subtract option is specified in conjunction with the cabinet and drawer.

Options

- add** Add a micro to the membership. This option requires the cabinet, drawer, and a micro argument (the micro does not have to be in the LAN for this command to succeed).
- auto** Auto detects the micros found in the private LAN.
- cabinet <cabinet>** Cabinet number in hex. Cabinet number is required to identify the component and defaults to 0 if it's not specified.
- drawer <drawer>** Drawer number in hex. Drawer number is required to identify the component and defaults to 0 if it's not specified.
- subtract** Subtract a micro from the list of micros. This option requires the cabinet, drawer, and a micro argument (the micro must be in the list of micros in order for this command to succeed).

Examples

1. Example of the "-add" option

```
MBM> set membership -add -ca 0 -dr 2 PBM
~GRP-E-(grp_IAMAlive) MISSING NEIGHBOR timeout from Cab: 0, Drawer: 5
Cab Drw Module      MicroID   IP Address      In Group
0  0  MBM      100000a   10.0.0.1        TRUE
0  1  MBM      100010a   10.1.0.1        TRUE
0  2  PBM      2fe0a     10.254.2.0      FALSE
0  5  PBM      5fe0a     10.254.5.0      TRUE
MBM> ~GRP-W-(grp_IAMAlive) Leaving Group ID: 100010a.a
~GRP-W-(grp_Create) Joined Group ID: 100000a.b
~GRP-W-(grp_Monitor) Leaving Group ID: 100000a.b
~GRP-W-(grp_Monitor) Joined Group ID: 100010a.c
[2003/04/17 15:51:27]
~REC-W-(trecTask) Server management group is transitioning.
~GRP-W-(grp_Monitor) Leaving Group ID: 100010a.c
~GRP-W-(grp_Monitor) Joined Group ID: 100010a.d
```

AlphaServer ES47/ES80/GS1280 Server Management CLI Reference

```
interrupt: ~GRP-I-(interrupt), GROUP HAS FORMED ID:100010a.d
interrupt: ~GRP-I-(interrupt), GROUP IS STABLE ID:100010a.d
[2003/04/17 15:51:55]
~REC-W-(trecTask) Server management group is stable.
MBM>
```

2. Example of the “-sub” option

```
MBM> set member -sub -ca 0 -dr 2 pbm
~GRP-E-(grp_IAMAlive) MISSING NEIGHBOR timeout from Cab: 0, Drawer: 5
Cab Drw Module      MicroID  IP Address      In Group
0  5  PBM      5fe0a    10.254.5.0      TRUE
0  1  MBM      100010a  10.1.0.1        TRUE
0  0  MBM      100000a  10.0.0.1        TRUE
MBM> ~GRP-W-(grp_IAMAlive) Leaving Group ID: 100010a.7
~GRP-W-(grp_Create) Joined Group ID: 100000a.8
~GRP-W-(grp_Monitor) Leaving Group ID: 100000a.8
~GRP-W-(grp_Monitor) Joined Group ID: 100010a.9
[2003/04/17 15:48:46]
~REC-W-(trecTask) Server management group is transitioning.
~GRP-W-(grp_Monitor) Leaving Group ID: 100010a.9
~GRP-W-(grp_Monitor) Joined Group ID: 100010a.a
interrupt: ~GRP-I-(interrupt), GROUP HAS FORMED ID:100010a.a
interrupt: ~GRP-I-(interrupt), GROUP IS STABLE ID:100010a.a
[2003/04/17 15:49:13]
~REC-W-(trecTask) Server management group is stable.
~GRP-W-(grp_Probe) MBM/PBM cab:00 drw:2 is not in the member list.
~GRP-W-(grp_Probe) Use Set Membership -add -ca 0 -dr 2 PBM

MBM>
```

3. Example of the “-auto” option (also the default if nothing is specified)

```
MBM> Set Membership -auto
~GRP-E-(grp_IAMAlive) MISSING NEIGHBOR timeout from Cab: 0, Drawer: 5
Cab Drw Module      MicroID  IP Address      In Group
0  5  PBM      5fe0a    10.254.5.0      TRUE
0  2  PBM      2fe0a    10.254.2.0      TRUE
0  1  MBM      100010a  10.1.0.1        TRUE
0  0  MBM      100000a  10.0.0.1        TRUE
MBM> ~GRP-W-(grp_IAMAlive) Leaving Group ID: 100010a.d
~GRP-W-(grp_Create) Joined Group ID: 100000a.e
~GRP-W-(grp_Monitor) Leaving Group ID: 100000a.e
~GRP-W-(grp_Monitor) Joined Group ID: 100010a.f
[2003/04/17 15:54:40]
~REC-W-(trecTask) Server management group is transitioning.
~GRP-W-(grp_Monitor) Leaving Group ID: 100010a.f
~GRP-W-(grp_Monitor) Joined Group ID: 100010a.10
interrupt: ~GRP-I-(interrupt), GROUP HAS FORMED ID:100010a.10
interrupt: ~GRP-I-(interrupt), GROUP IS STABLE ID:100010a.10
[2003/04/17 15:55:08]
~REC-W-(trecTask) Server management group is stable.
MBM>
```

See Also

show membership

3.47 set password

Sets the remote access password. The password has a maximum length of 15 characters. The default password is **mvfirmware**. The password is nonvolatile and stored in the EEPROM of the MBM or PBM.

Syntax

set password <password>

Argument

password The password that must be entered by a remote session to gain access to the system.

Options

None

Example

Change the remote access password.

```
MBM> set password mvfw  
MBM>
```

See Also

show modem

3.48 set sys_serial_num

Sets the system serial number. The serial number has a maximum length of 12 characters. This command converts input to upper case and propagates the serial number to all EEPROMs except RIMMs.

Syntax

```
set sys_serial_num <serial_number>
```

Argument

serial_number	The system serial number, limited to 12 characters.
---------------	---

Options

None

Example

Set the system serial number.

```
MBM> sh sys_serial
sys_serial_num is not set!
MBM> set sys_serial_num marvel_fw16P
Updating SYS_SERIAL_NUM in FRU EEPROMs....
MBM>
MBM> sh sys_serial
sys_serial_num          MARVEL_FW16P
MBM>
```

See Also

show sys_serial_num

3.49 set time

Sets the common base time. By default, this command sends the current base time to the primary CPU's MBM, which then sends out a time synchronizer message to all MBMs and PBMs to get this new base time. This command enables the user to set up both date and time or just one of them.

You can use **show time** to confirm your changes.

Warning: Changing the base time will affect the time on all existing partitions.

Syntax

set time <date> | <time>

Arguments

date The format is yyyy/mm/dd.
time The format is hh:mm:ss.

Options

None

Examples

1. Change the date and time.

```
MBM> set time 2002/04/16 17:20:30
MBM> show time
Common base time: 2002/04/16 17:20:34

HP Name           SP Name           Delta Time (Days hh:mm:ss)
hp0                Default_SP        +00 +00:+00:+00
hpl                Default_SP        -01 -01:-01:-01
MBM>
```

2. Change the time only.

```
MBM> set time 14:45:00
MBM>
```

3. Change the date only.

```
MBM> set time 2003/08/16
MBM>
```

See Also

show time

3.50 shell

Starts a vxworks shell.

Note: This command will be removed when the MBM/PBM firmware is not shipped with the vxworks shell.

Syntax

shell

Arguments

None

Options

None

Example

Spawn a new shell task to go back to the shell prompt.

```
MBM> shell
10.0.0.1->
```

3.51 show

Displays the value of one or all server management environment variables. Integer variables are displayed in hex.

Syntax

```
show    {<smv>, *}
```

Arguments

smv The environment variable name. Lists all environment variables if no variable name is provided.

* List all environment variables.

Options

None

Example

Display a particular variable value.

```
MBM> show srm_auto_connect
MBM> srm_auto_connect      1
```

See Also

set, clear

3.52 show cables

Displays the cabling configuration. This command shows the physical cabling layout. Interprocessor (IP) cables connect external 8P pairs of CPU routing ports. The six IP ports are two North, two South, one East and one West. There are eight IO cables in a drawer. Each corresponds to an IO7.

For IP cables, each system building block is labeled "SBB(c, d)," where "c" is the cabinet number and "d" is the drawer number in that cabinet. Connections are indicated by dashes. Each pair of matching symbols indicates a wrap-around. "X" means wrong connection and "x" means expected but missing connection. "?" means unknown cable status.

For IO cables, dashes indicate a connection between the IOR and the IO port of the CPU. Otherwise, there is no connection.

There are four IORs per Expanded I/O shelf; otherwise, there is only one IOR per standard PCI drawer.

Syntax

show cables [-ip | -io]

Arguments

None

Options

- ip Show only the IP cables between 8P drawers.
- io Show only the PCI IO cables.

Example

Show all cables of an 8P system.

```
MBM> sh cable
IP Cabling: Each System Building Block is represented by SBB(Cabinet, Drawer)
Each pair of matching symbols indicates wrap-around
('X' - wrong connection, 'x' - expected but missing connection)
      A0  A1                A2  A3
      |  |                |  |
0a---SBB(0,0)-----SBB(0,2)---0a
      |  |                |  |
      |  |                |  |
1a---SBB(0,1)-----SBB(0,3)---1a
      |  |                |  |
      A0  A1                A2  A3
```

IO cabling between IORs of the PCI drawer and CPU IOPs

PCI drawer				SBB		
Cab	Drw	IOR		Cab	Drw	IOP
1	1	0	-----	0	0	0
1	2	0	-----	0	3	0
1	3	0	-----	0	1	0
1	4	0	-----	0	2	2

MBM>

See Also

config cables

3.53 show duo

Shows the configuration and state of dual processor modules.

When calculating the total CPU memory, the RAID RIMMs are excluded. RIMM4 is redundant if RIMMs 0 - 4 are fully populated. RIMM9 is also redundant if RIMMs 5-9 are fully populated.

Syntax

show duo [-cabinet < cabinet > -drawer < drawer > [duo<n>]]

Argument

duo<n> The dual processor module by cabinet and drawer number. The optional *n* is the dual processor module within the 8P drawer, a value from 0 to 3.

Options

-cabinet< cabinet > Cabinet number in hex. If this qualifier is not used, all cabinets are shown.
 -drawer< drawer > Drawer number in hex. If this qualifier is not used, all drawers are shown.

Example

1. Show all dual CPU modules in the system.

```
MBM> sh duo
DUO0 is in Cabinet 0, Drawer 0, Slot 0
CMM Status: OK
CMM Power state: ON
```

	CPU0	CPU1
CPU State:	SRM	XSROM
CPU Diag status:	TBD	TBD
CPU Memory:	1024MB (exclude RAID)	1024MB (exclude RAID)
RIMM0:	128MB	128MB
RIMM1:	128MB	128MB
RIMM2:	128MB	128MB
RIMM3:	128MB	128MB
RIMM4:	0MB (Non-existent)	0MB (Non-existent)
RIMM5:	128MB	128MB
RIMM6:	128MB	128MB
RIMM7:	128MB	128MB
RIMM8:	128MB	128MB
RIMM9:	0MB (Non-existent)	0MB (Non-existent)
CPU Revision:	EV7 2.1	EV7 2.1
CPU Cache:	1.5 MB	1.5 MB
CPU Speed:	1000 MHz	1000 MHz

```
DUO1 is in Cabinet 0, Drawer 0, Slot 1
CMM Status: OK
CMM Power state: ON
```

	CPU2	CPU3
CPU State:	XSROM	XSROM
CPU Diag status:	TBD	TBD
CPU Memory:	1024MB (exclude RAID)	1024MB (exclude RAID)

```

RIMM0:      128MB      128MB
RIMM1:      128MB      128MB
RIMM2:      128MB      128MB
RIMM3:      128MB      128MB
RIMM4:      0MB (Non-existent) 0MB (Non-existent)
RIMM5:      128MB      128MB
RIMM6:      128MB      128MB
RIMM7:      128MB      128MB
RIMM8:      128MB      128MB
RIMM9:      0MB (Non-existent) 0MB (Non-existent)
CPU Revision: EV7 2.1      EV7 2.1
CPU Cache:   1.5 MB      1.5 MB
CPU Speed:   1000 MHz     1000 MHz

```

DUO2 is in Cabinet 0, Drawer 0, Slot 2

CMM Status: OK
CMM Power state: ON

	CPU4	CPU5
	-----	-----
CPU State:	XSROM	XSROM
CPU Diag status:	TBD	TBD
CPU Memory:	1024MB (exclude RAID)	1024MB (exclude RAID)
RIMM0:	128MB	128MB
RIMM1:	128MB	128MB
RIMM2:	128MB	128MB
RIMM3:	128MB	128MB
RIMM4:	0MB (Non-existent)	0MB (Non-existent)
RIMM5:	128MB	128MB
RIMM6:	128MB	128MB
RIMM7:	128MB	128MB
RIMM8:	128MB	128MB
RIMM9:	0MB (Non-existent)	0MB (Non-existent)
CPU Revision:	EV7 2.1	EV7 2.1
CPU Cache:	1.5 MB	1.5 MB
CPU Speed:	1000 MHz	1000 MHz

DUO3 is in Cabinet 0, Drawer 0, Slot 3

CMM Status: OK
CMM Power state: ON

	CPU6	CPU7
	-----	-----
CPU State:	XSROM	XSROM
CPU Diag status:	TBD	TBD
CPU Memory:	1024MB (exclude RAID)	1024MB (exclude RAID)
RIMM0:	128MB	128MB
RIMM1:	128MB	128MB
RIMM2:	128MB	128MB
RIMM3:	128MB	128MB
RIMM4:	0MB (Non-existent)	0MB (Non-existent)
RIMM5:	128MB	128MB
RIMM6:	128MB	128MB
RIMM7:	128MB	128MB
RIMM8:	128MB	128MB
RIMM9:	0MB (Non-existent)	0MB (Non-existent)
CPU Revision:	EV7 2.1	EV7 2.1
CPU Cache:	1.5 MB	1.5 MB
CPU Speed:	1000 MHz	1000 MHz

DUO0 is in Cabinet 0, Drawer 1, Slot 0

CMM Status: OK
CMM Power state: ON

	CPU0	CPU1
	-----	-----
CPU State:	XSROM	XSROM
CPU Diag status:	TBD	TBD
CPU Memory:	1024MB (exclude RAID)	1024MB (exclude RAID)
RIMM0:	128MB	128MB
RIMM1:	128MB	128MB
RIMM2:	128MB	128MB
RIMM3:	128MB	128MB
RIMM4:	0MB (Non-existent)	0MB (Non-existent)

AlphaServer ES47/ES80/GS1280 Server Management CLI Reference

```

RIMM5:      128MB      128MB
RIMM6:      128MB      128MB
RIMM7:      128MB      128MB
RIMM8:      128MB      128MB
RIMM9:      0MB (Non-existent) 0MB (Non-existent)
CPU Revision: EV7 2.1      EV7 2.1
CPU Cache:   1.5 MB      1.5 MB
CPU Speed:   1000 MHz     1000 MHz

```

DUO1 is in Cabinet 0, Drawer 1, Slot 1

CMM Status: OK

CMM Power state: ON

	CPU2	CPU3
	-----	-----
CPU State:	XSROM	XSROM
CPU Diag status:	TBD	TBD
CPU Memory:	1024MB (exclude RAID)	1024MB (exclude RAID)
RIMM0:	128MB	128MB
RIMM1:	128MB	128MB
RIMM2:	128MB	128MB
RIMM3:	128MB	128MB
RIMM4:	0MB (Non-existent)	0MB (Non-existent)
RIMM5:	128MB	128MB
RIMM6:	128MB	128MB
RIMM7:	128MB	128MB
RIMM8:	128MB	128MB
RIMM9:	0MB (Non-existent)	0MB (Non-existent)
CPU Revision:	EV7 2.1	EV7 2.1
CPU Cache:	1.5 MB	1.5 MB
CPU Speed:	1000 MHz	1000 MHz

DUO2 is in Cabinet 0, Drawer 1, Slot 2

CMM Status: OK

CMM Power state: ON

	CPU4	CPU5
	-----	-----
CPU State:	XSROM	XSROM
CPU Diag status:	TBD	TBD
CPU Memory:	1024MB (exclude RAID)	1024MB (exclude RAID)
RIMM0:	128MB	128MB
RIMM1:	128MB	128MB
RIMM2:	128MB	128MB
RIMM3:	128MB	128MB
RIMM4:	0MB (Non-existent)	0MB (Non-existent)
RIMM5:	128MB	128MB
RIMM6:	128MB	128MB
RIMM7:	128MB	128MB
RIMM8:	128MB	128MB
RIMM9:	0MB (Non-existent)	0MB (Non-existent)
CPU Revision:	TBD	TBD
CPU Cache:	TBD	TBD
CPU Speed:	TBD	TBD

DUO3 is in Cabinet 0, Drawer 1, Slot 3

CMM Status: OK

CMM Power state: ON

	CPU6	CPU7
	-----	-----
CPU State:	XSROM	XSROM
CPU Diag status:	TBD	TBD
CPU Memory:	1024MB (exclude RAID)	1024MB (exclude RAID)
RIMM0:	128MB	128MB
RIMM1:	128MB	128MB
RIMM2:	128MB	128MB
RIMM3:	128MB	128MB
RIMM4:	0MB (Non-existent)	0MB (Non-existent)
RIMM5:	128MB	128MB
RIMM6:	128MB	128MB
RIMM7:	128MB	128MB
RIMM8:	128MB	128MB
RIMM9:	0MB (Non-existent)	0MB (Non-existent)

```

CPU Revision:      EV7 2.1                EV7 2.1
CPU Cache:        1.5 MB                  1.5 MB
CPU Speed:        1000 MHz                 1000 MHz
MBM>

```

2. Show a specific dual CPU module.

```

MBM> sh duo -ca 0 -dr 1 duo2
DUO2 is in Cabinet 0, Drawer 1, Slot 2
CMM Status: OK
Duo Power state: ON

```

	----- CPU4 -----	----- CPU5 -----
CPU State:	SRM	XSROM
CPU Diag status:	TBD	TBD
CPU Memory:	1024MB (exclude RAID)	1024MB (exclude RAID)
RIMM0:	128MB	128MB
RIMM1:	128MB	128MB
RIMM2:	128MB	128MB
RIMM3:	128MB	128MB
RIMM4:	128MB	128MB
RIMM5:	128MB	128MB
RIMM6:	128MB	128MB
RIMM7:	128MB	128MB
RIMM8:	128MB	128MB
RIMM9:	128MB	128MB
CPU Revision:	EV7 2.1	EV7 2.1
CPU Cache:	1.5 MB	1.5 MB
CPU Speed:	1000 Mhz	1000 Mhz

MBM>

See Also

show system, show power, show memory

3.54 show error

Displays errors logged in the FRU EEPROM. This command will display any TDD failures logged in a FRU's IIC EEPROM. "Inaccessible" is usually due to IIC error.

Syntax

```
show error [-cabinet <cabinet> -drawer <drawer>]
```

Arguments

None

Options

-cabinet <cabinet> Cabinet number in hex. If not specified, all cabinets are shown.

-drawer <drawer> Drawer number in hex. If not specified, all drawers are shown.

Examples

1. List errors from all FRUs.

```
MBM> sh error
FRU Name
CAB0.DRW0.DU00.CMM          10 FRU checksum error between bytes 0-62
CAB0.DRW0.DU00.RIMM00       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU00.RIMM10       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU00.RIMM20       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU00.RIMM30       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU00.RIMM50       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU00.RIMM60       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU00.RIMM70       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU00.RIMM80       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU00.RIMM01       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU00.RIMM11       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU00.RIMM21       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU00.RIMM31       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU00.RIMM51       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU00.RIMM61       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU00.RIMM71       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU00.RIMM81       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU01.CMM          10 FRU checksum error between bytes 0-62
CAB0.DRW0.DU01.RIMM00       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU01.RIMM10       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU01.RIMM20       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU01.RIMM30       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU01.RIMM50       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU01.RIMM60       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU01.RIMM70       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU01.RIMM80       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU01.RIMM01       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU01.RIMM11       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU01.RIMM21       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU01.RIMM31       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU01.RIMM51       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU01.RIMM61       40 FRU checksum error between bytes 128-254
CAB0.DRW0.DU01.RIMM71       40 FRU checksum error between bytes 128-254
```



```

CAB0.DRW3.DU00.CMM      10 FRU checksum error between bytes 0-62
CAB0.DRW3.DU00.RIMM00   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU00.RIMM10   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU00.RIMM20   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU00.RIMM30   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU00.RIMM50   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU00.RIMM60   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU00.RIMM70   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU00.RIMM80   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU00.RIMM01   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU00.RIMM11   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU00.RIMM21   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU00.RIMM31   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU00.RIMM51   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU00.RIMM61   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU00.RIMM71   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU00.RIMM81   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU01.CMM      10 FRU checksum error between bytes 0-62
CAB0.DRW3.DU01.RIMM00   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU01.RIMM10   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU01.RIMM20   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU01.RIMM30   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU01.RIMM50   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU01.RIMM60   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU01.RIMM70   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU01.RIMM80   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU01.RIMM01   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU01.RIMM11   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU01.RIMM21   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU01.RIMM31   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU01.RIMM51   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU01.RIMM61   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU01.RIMM71   40 FRU checksum error between bytes 128-254
CAB0.DRW3.DU01.RIMM81   40 FRU checksum error between bytes 128-254
CAB1.COCP                36 FRU checksum error between bytes 0-62
                        FRU checksum error between bytes 64-126
                        TDD - Type: 127 Test: 255 SubTest: 255 Error: 255
                        94: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
                        SDD - Type: 15 LastLog: 3 Overwrite: 3
MBM>

```

2. List errors from FRUs in Cabinet 1.

```

MBM> sh error -ca 1
FRU Name          E Symptoms
CAB1.COCP         36 FRU checksum error between bytes 0-62
                  FRU checksum error between bytes 64-126
                  TDD - Type: 127 Test: 255 SubTest: 255 Error: 255
                  94: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
                  SDD - Type: 15 LastLog: 3 Overwrite: 3
MBM>

```

3. List errors from FRUs in Drawer 1 in Cabinet 0.

```

MBM> sh error -ca 0 -dr 1
FRU Name          E Symptoms
CAB0.DRW1.DU00.CMM 10 FRU checksum error between bytes 0-62
CAB0.DRW1.DU00.RIMM00 40 FRU checksum error between bytes 128-254
CAB0.DRW1.DU00.RIMM10 40 FRU checksum error between bytes 128-254
CAB0.DRW1.DU00.RIMM20 40 FRU checksum error between bytes 128-254
CAB0.DRW1.DU00.RIMM30 40 FRU checksum error between bytes 128-254
CAB0.DRW1.DU00.RIMM50 40 FRU checksum error between bytes 128-254
CAB0.DRW1.DU00.RIMM60 40 FRU checksum error between bytes 128-254
CAB0.DRW1.DU00.RIMM70 40 FRU checksum error between bytes 128-254
CAB0.DRW1.DU00.RIMM80 40 FRU checksum error between bytes 128-254
CAB0.DRW1.DU00.RIMM01 40 FRU checksum error between bytes 128-254
CAB0.DRW1.DU00.RIMM11 40 FRU checksum error between bytes 128-254
CAB0.DRW1.DU00.RIMM21 40 FRU checksum error between bytes 128-254
CAB0.DRW1.DU00.RIMM31 40 FRU checksum error between bytes 128-254

```

```
CAB0.DRW1.DUO0.RIMM41 40 FRU checksum error between bytes 128-254
CAB0.DRW1.DUO0.RIMM51 40 FRU checksum error between bytes 128-254
CAB0.DRW1.DUO0.RIMM61 40 FRU checksum error between bytes 128-254
CAB0.DRW1.DUO0.RIMM71 40 FRU checksum error between bytes 128-254
CAB0.DRW1.DUO0.RIMM81 40 FRU checksum error between bytes 128-254
CAB0.DRW1.DUO0.RIMM91 40 FRU checksum error between bytes 128-254
MBM>
```

See Also

clear error, build fru, show fru

3.55 show fru

Displays information on every FRU.

Syntax

```
show fru [-cabinet <cabinet> -drawer <drawer>] [-s, -e]
```

Arguments

None

Options

-cabinet <cabinet>	Cabinet number in hex. If not specified, all cabinets are shown.
-drawer <drawer>	Drawer number in hex. If not specified, all drawers are shown.
-s	Display the system serial number stored for the FRUs.
-e	Display only the FRUs with errors

Examples

1. Display FRU information for all components with an I²C EEPROM. Note that part numbers and serial numbers are for illustrative purposes.

```
MBM> sh fru
Fru Name                E Part #                Serial #                Model/Other
CAB0.DRW0.DOCF          00 nightlyFri_0        Fri_43
CAB0.DRW0.MBM           00 partnumber          serialnum              mbm module
CAB0.DRW0.DU00.CMM      10 nightlyFri_2        Fri_45
CAB0.DRW0.DU00.DUO      10 nightlyFri_3        Fri_46
CAB0.DRW0.DU00.RIMM00   06 20-1C872-01        GS1280-32P000L        CE
CAB0.DRW0.DU00.RIMM10   06 20-1C872-01        GS1280-32P001L        CE
CAB0.DRW0.DU00.RIMM20   06 20-1C872-01        GS1280-32P002L        CE
CAB0.DRW0.DU00.RIMM30   06 20-1C872-01        GS1280-32P003L        CE
CAB0.DRW0.DU00.RIMM50   06 20-1C872-01        GS1280-32P005L        CE
CAB0.DRW0.DU00.RIMM60   06 20-1C872-01        GS1280-32P006L        CE
CAB0.DRW0.DU00.RIMM70   06 20-1C872-01        GS1280-32P007L        CE
CAB0.DRW0.DU00.RIMM80   06 20-1C872-01        GS1280-32P008L        CE
CAB0.DRW0.DU00.RIMM01   06 20-1C872-01        GS1280-32P00AL        CE
CAB0.DRW0.DU00.RIMM11   06 20-1C872-01        GS1280-32P00BL        CE
CAB0.DRW0.DU00.RIMM21   06 20-1C872-01        GS1280-32P00CL        CE
CAB0.DRW0.DU00.RIMM31   06 20-1C872-01        GS1280-32P00DL        CE
CAB0.DRW0.DU00.RIMM51   06 20-1C872-01        GS1280-32P00FL        CE
CAB0.DRW0.DU00.RIMM61   06 20-1C872-01        GS1280-32P00GL        CE
CAB0.DRW0.DU00.RIMM71   06 20-1C872-01        GS1280-32P00HL        CE
CAB0.DRW0.DU00.RIMM81   06 20-1C872-01        GS1280-32P00IL        CE
CAB0.DRW0.DU00.VRM0.CPU 00 nightlyFri_4        Fri_47                7F7FF8
CAB0.DRW0.DU00.VRM1.CPU 00 nightlyFri_5        Fri_48                7F7FF8
CAB0.DRW0.DU00.VRM2.CPU 00 nightlyFri_6        Fri_49                7F7FF8
CAB0.DRW0.DU00.VRM3.CPU 00 nightlyFri_7        Fri_50                7F7FF8
CAB0.DRW0.DU00.VRM4.CPU 00 nightlyFri_8        Fri_51                7F7FF8
CAB0.DRW0.DU00.VRM2.RIMM 00 nightlyFri_9        Fri_52                7F7FF8
```

CAB0.DRW0.DU00.VRM3.RIMM	00	nightlyFri_10	Fri_53	7F7FF8
CAB0.DRW0.DU00.VRM4.RIMM	00	nightlyFri_11	Fri_54	7F7FF8
CAB0.DRW0.DU00.VRM0.TERM	00	nightlyFri_12	Fri_55	7F7FA2
CAB0.DRW0.DU01.CMM	10	nightlyFri_13	Fri_56	
CAB0.DRW0.DU01.DUO	10	nightlyFri_14	Fri_57	
CAB0.DRW0.DU01.RIMM00	06	20-1C872-01	GS1280-32P010L	CE
CAB0.DRW0.DU01.RIMM10	06	20-1C872-01	GS1280-32P011L	CE
CAB0.DRW0.DU01.RIMM20	06	20-1C872-01	GS1280-32P012L	CE
CAB0.DRW0.DU01.RIMM30	06	20-1C872-01	GS1280-32P013L	CE
CAB0.DRW0.DU01.RIMM50	06	20-1C872-01	GS1280-32P015L	CE
CAB0.DRW0.DU01.RIMM60	06	20-1C872-01	GS1280-32P016L	CE
CAB0.DRW0.DU01.RIMM70	06	20-1C872-01	GS1280-32P017L	CE
CAB0.DRW0.DU01.RIMM80	06	20-1C872-01	GS1280-32P018L	CE
CAB0.DRW0.DU01.RIMM01	06	20-1C872-01	GS1280-32P01AL	CE
CAB0.DRW0.DU01.RIMM11	06	20-1C872-01	GS1280-32P01BL	CE
CAB0.DRW0.DU01.RIMM21	06	20-1C872-01	GS1280-32P01CL	CE
CAB0.DRW0.DU01.RIMM31	06	20-1C872-01	GS1280-32P01DL	CE
CAB0.DRW0.DU01.RIMM51	06	20-1C872-01	GS1280-32P01FL	CE
CAB0.DRW0.DU01.RIMM61	06	20-1C872-01	GS1280-32P01GL	CE
CAB0.DRW0.DU01.RIMM71	06	20-1C872-01	GS1280-32P01HL	CE
CAB0.DRW0.DU01.RIMM81	06	20-1C872-01	GS1280-32P01IL	CE
CAB0.DRW0.DU01.VRM0.CPU	00	nightlyFri_15	Fri_58	7F7FA2
CAB0.DRW0.DU01.VRM1.CPU	00	nightlyFri_16	Fri_59	7F7FA2
CAB0.DRW0.DU01.VRM2.CPU	00	nightlyFri_17	Fri_60	7F7FA2
CAB0.DRW0.DU01.VRM3.CPU	00	nightlyFri_18	Fri_61	7F7FA2
CAB0.DRW0.DU01.VRM4.CPU	00	nightlyFri_19	Fri_62	7F7FA2
CAB0.DRW0.DU01.VRM2.RIMM	00	nightlyFri_20	Fri_63	7F7FF8
CAB0.DRW0.DU01.VRM3.RIMM	00	nightlyFri_21	Fri_64	7F7FF8
CAB0.DRW0.DU01.VRM4.RIMM	00	nightlyFri_22	Fri_65	7F7FA2
CAB0.DRW0.DU01.VRM0.TERM	00	nightlyFri_23	Fri_66	7F7FF8
CAB0.DRW1.DOCBP	00	nightlyFri_24	Fri_67	
CAB0.DRW1.MBM	00	nightlyFri_25	Fri_68	
CAB0.DRW1.DU00.CMM	10	nightlyFri_26	Fri_69	
CAB0.DRW1.DU00.DUO	10	nightlyFri_27	Fri_70	
CAB0.DRW1.DU00.RIMM00	06	20-1C872-01	GS1280-32P040L	CE
CAB0.DRW1.DU00.RIMM10	06	20-1C872-01	GS1280-32P041L	CE
CAB0.DRW1.DU00.RIMM20	06	20-1C872-01	GS1280-32P042L	CE
CAB0.DRW1.DU00.RIMM30	06	20-1C872-01	GS1280-32P043L	CE
CAB0.DRW1.DU00.RIMM50	06	20-1C872-01	GS1280-32P045L	CE
CAB0.DRW1.DU00.RIMM60	06	20-1C872-01	GS1280-32P046L	CE
CAB0.DRW1.DU00.RIMM70	06	20-1C872-01	GS1280-32P047L	CE
CAB0.DRW1.DU00.RIMM80	06	20-1C872-01	GS1280-32P048L	CE
CAB0.DRW1.DU00.RIMM01	06	20-1C872-01	GS1280-32P04AL	CE
CAB0.DRW1.DU00.RIMM11	06	20-1C872-01	GS1280-32P04BL	CE
CAB0.DRW1.DU00.RIMM21	06	20-1C872-01	GS1280-32P04CL	CE
CAB0.DRW1.DU00.RIMM31	06	20-1C872-01	GS1280-32P04DL	CE
CAB0.DRW1.DU00.RIMM41	06	20-1C872-01	GS1280-32P04EL	CE
CAB0.DRW1.DU00.RIMM51	06	20-1C872-01	GS1280-32P04FL	CE
CAB0.DRW1.DU00.RIMM61	06	20-1C872-01	GS1280-32P04GL	CE
CAB0.DRW1.DU00.RIMM71	06	20-1C872-01	GS1280-32P04HL	CE
CAB0.DRW1.DU00.RIMM81	06	20-1C872-01	GS1280-32P04IL	CE
CAB0.DRW1.DU00.RIMM91	06	20-1C872-01	GS1280-32P04JL	CE
CAB0.DRW1.DU00.VRM0.CPU	00	nightlyFri_28	Fri_71	7F7FF8
CAB0.DRW1.DU00.VRM1.CPU	00	nightlyFri_29	Fri_72	7F7FF8
CAB0.DRW1.DU00.VRM2.CPU	00	nightlyFri_30	Fri_73	7F7FF8
CAB0.DRW1.DU00.VRM3.CPU	00	nightlyFri_31	Fri_74	7F7FF8
CAB0.DRW1.DU00.VRM2.RIMM	00	nightlyFri_32	Fri_75	7F7FF8
CAB0.DRW1.DU00.VRM3.RIMM	00	nightlyFri_33	Fri_76	7F7FF8
CAB0.DRW1.DU00.VRM4.RIMM	00	nightlyFri_34	Fri_77	7F7FF8
CAB0.DRW1.DU00.VRM0.TERM	00	nightlyFri_35	Fri_78	7F7FF8
CAB0.DRW1.DU01.CMM	10	nightlyFri_36	Fri_79	
CAB0.DRW1.DU01.DUO	10	nightlyFri_37	Fri_80	
CAB0.DRW1.DU01.RIMM00	06	20-1C872-01	GS1280-32P050L	CE
CAB0.DRW1.DU01.RIMM10	06	20-1C872-01	GS1280-32P051L	CE
CAB0.DRW1.DU01.RIMM20	06	20-1C872-01	GS1280-32P052L	CE
CAB0.DRW1.DU01.RIMM30	06	20-1C872-01	GS1280-32P053L	CE
CAB0.DRW1.DU01.RIMM50	06	20-1C872-01	GS1280-32P055L	CE
CAB0.DRW1.DU01.RIMM60	06	20-1C872-01	GS1280-32P056L	CE
CAB0.DRW1.DU01.RIMM70	06	20-1C872-01	GS1280-32P057L	CE
CAB0.DRW1.DU01.RIMM80	06	20-1C872-01	GS1280-32P058L	CE
CAB0.DRW1.DU01.RIMM01	06	20-1C872-01	GS1280-32P05AL	CE

AlphaServer ES47/ES80/GS1280 Server Management CLI Reference

```

CAB0.DRW1.DUO1.RIMM11    06 20-1C872-01      GS1280-32P05BL    CE
CAB0.DRW1.DUO1.RIMM21    06 20-1C872-01      GS1280-32P05CL    CE
CAB0.DRW1.DUO1.RIMM31    06 20-1C872-01      GS1280-32P05DL    CE
CAB0.DRW1.DUO1.RIMM51    06 20-1C872-01      GS1280-32P05FL    CE
CAB0.DRW1.DUO1.RIMM61    06 20-1C872-01      GS1280-32P05GL    CE
CAB0.DRW1.DUO1.RIMM71    06 20-1C872-01      GS1280-32P05HL    CE
CAB0.DRW1.DUO1.RIMM81    06 20-1C872-01      GS1280-32P05IL    CE
CAB0.DRW1.DUO1.VRM0.CPU  00 nightlyFri_38     Fri_81             7F7FF8
CAB0.DRW1.DUO1.VRM1.CPU  00 nightlyFri_39     Fri_82             7F7FF8
CAB0.DRW1.DUO1.VRM2.CPU  00 nightlyFri_40     Fri_83             7F7FF8
CAB0.DRW1.DUO1.VRM3.CPU  00 nightlyFri_41     Fri_84             7F7FF8
CAB0.DRW1.DUO1.VRM4.CPU  00 nightlyFri_42     Fri_85             7F7FF8
CAB0.DRW1.DUO1.VRM2.RIMM 00 nightlyFri_43     Fri_86             7F7FF8
CAB0.DRW1.DUO1.VRM3.RIMM 00 nightlyFri_44     Fri_87             7F7FF8
CAB0.DRW1.DUO1.VRM4.RIMM 00 nightlyFri_45     Fri_88             7F7FF8
CAB0.DRW1.DUO1.VRM0.TERM 00 nightlyFri_46     Fri_89             7F7FA2
CAB1.COCP                 36 Unrecognized FRU Format (ff)
CAB1.DRW0.PWR0            00 nightlyFri_95     Fri_138
CAB1.DRW1.PWR0            00 nightlyFri_96     Fri_139
CAB1.DRW1.DOCP            00 nightlyFri_97     Fri_140
CAB1.DRW1.PBM             00 nightlyFri_98     Fri_141
CAB1.DRW1.PCI             00 nightlyFri_99     Fri_142
CAB1.DRW1.PS0             00 nightlyFri_100    Fri_143            7F7FA2
CAB1.DRW1.PS1             00 nightlyFri_101    Fri_144            7F7FA2
CAB1.DRW1.IOR0            00 nightlyFri_102    Fri_145
CAB1.DRW2.PWR0            00 nightlyFri_103    Fri_146
CAB1.DRW2.DOCP            00 nightlyFri_104    Fri_147
CAB1.DRW2.PBM             00 nightlyFri_105    Fri_148
CAB1.DRW2.PCI             00 nightlyFri_106    Fri_149
CAB1.DRW2.PS0             00 nightlyFri_107    Fri_150            7F7FA2
CAB1.DRW2.PS1             00 nightlyFri_108    Fri_151            7F7FA2
CAB1.DRW2.IOR0            00 nightlyFri_109    Fri_152
CAB1.DRW3.PWR0            00 nightlyFri_110    Fri_153
CAB1.DRW3.DOCP            00 nightlyFri_111    Fri_154
CAB1.DRW3.PBM             00 nightlyFri_112    Fri_155
CAB1.DRW3.PCI             00 nightlyFri_113    Fri_156
CAB1.DRW3.PS1             00 nightlyFri_114    Fri_157            7F7FA2
CAB1.DRW3.IOR0            00 nightlyFri_115    Fri_158
CAB1.DRW4.DOCP            00 nightlyFri_116    Fri_159
CAB1.DRW4.PBM             00 nightlyFri_117    Fri_160
CAB1.DRW4.PCI             00 nightlyFri_118    Fri_161
CAB1.DRW4.PS1             00 nightlyFri_119    Fri_162            7F7FA2
CAB1.DRW4.IOR0            00 nightlyFri_120    Fri_163
MBM>

```

2. Display FRU information for all FRUs in Cabinet 1.

```

MBM> show fru -ca 1
Fru Name                E Part #                Serial #                Model/Other
CAB1.COCP                36 Unrecognized FRU Format (ff)
CAB1.DRW0.PWR0           00 nightlyFri_95        Fri_138
CAB1.DRW1.PWR0           00 nightlyFri_96        Fri_139
CAB1.DRW1.DOCP           00 nightlyFri_97        Fri_140
CAB1.DRW1.PBM            00 nightlyFri_98        Fri_141
CAB1.DRW1.PCI            00 nightlyFri_99        Fri_142
CAB1.DRW1.PS0            00 nightlyFri_100       Fri_143              7F7FA2
CAB1.DRW1.PS1            00 nightlyFri_101       Fri_144              7F7FA2
CAB1.DRW1.IOR0           00 nightlyFri_102       Fri_145
CAB1.DRW2.PWR0           00 nightlyFri_103       Fri_146
CAB1.DRW2.DOCP           00 nightlyFri_104       Fri_147
CAB1.DRW2.PBM            00 nightlyFri_105       Fri_148
CAB1.DRW2.PCI            00 nightlyFri_106       Fri_149
CAB1.DRW2.PS0            00 nightlyFri_107       Fri_150              7F7FA2
CAB1.DRW2.PS1            00 nightlyFri_108       Fri_151              7F7FA2
CAB1.DRW2.IOR0           00 nightlyFri_109       Fri_152
CAB1.DRW3.PWR0           00 nightlyFri_110       Fri_153
CAB1.DRW3.DOCP           00 nightlyFri_111       Fri_154
CAB1.DRW3.PBM            00 nightlyFri_112       Fri_155
CAB1.DRW3.PCI            00 nightlyFri_113       Fri_156

```

```

CAB1.DRW3.PS1          00 nightlyFri_114    Fri_157          7F7FA2
CAB1.DRW3.IOR0        00 nightlyFri_115    Fri_158
CAB1.DRW4.DOCP        00 nightlyFri_116    Fri_159
CAB1.DRW4.PBM         00 nightlyFri_117    Fri_160
CAB1.DRW4.PCI         00 nightlyFri_118    Fri_161
CAB1.DRW4.PS1         00 nightlyFri_119    Fri_162          7F7FA2
CAB1.DRW4.IOR0        00 nightlyFri_120    Fri_163
MBM>

```

3. Display only FRUs with errors in Cabinet 1

```

MBM> show fru -ca 1
Fru Name                E Part #                Serial #                Model/Other
CAB1.OCP                 36 Unrecognized FRU    Format (ff)
MBM>

```

4. Display the system serial number stored in the FRUs in Cabinet 1

```

MBM> sh fru -ca 1 -s
Fru Name                Sys_Serial_Num
CAB1.COCP                GS1280-32P
CAB1.DRW0.PWR0           GS1280-32P
CAB1.DRW1.PWR0           GS1280-32P
CAB1.DRW1.DOCP           GS1280-32P
CAB1.DRW1.PBM            GS1280-32P
CAB1.DRW1.PCI            GS1280-32P
CAB1.DRW1.PS0            GS1280-32P
CAB1.DRW1.PS1            GS1280-32P
CAB1.DRW1.IOR0           GS1280-32P
CAB1.DRW2.PWR0           GS1280-32P
CAB1.DRW2.DOCP           GS1280-32P
CAB1.DRW2.PBM            GS1280-32P
CAB1.DRW2.PCI            GS1280-32P
CAB1.DRW2.PS0            GS1280-32P
CAB1.DRW2.PS1            GS1280-32P
CAB1.DRW2.IOR0           GS1280-32P
CAB1.DRW3.PWR0           GS1280-32P
CAB1.DRW3.DOCP           GS1280-32P
CAB1.DRW3.PBM            GS1280-32P
CAB1.DRW3.PCI            GS1280-32P
CAB1.DRW3.PS1            GS1280-32P
CAB1.DRW3.IOR0           GS1280-32P
CAB1.DRW4.DOCP           GS1280-32P
CAB1.DRW4.PBM            GS1280-32P
CAB1.DRW4.PCI            GS1280-32P
CAB1.DRW4.PS1            GS1280-32P
CAB1.DRW4.IOR0           GS1280-32P
MBM>

```

See Also

build fru, show error, clear error

3.57 show log

Displays errors logged in the MBM or PBM flash. The events logged might be the following:

- Server management micro detected errors
- Hardware environmental events. For example, fan, voltage, temperature, power supply failure, etc.
- EV7 diagnostic failures
- EV7 logout frame

Error logs are organized by entries. By default, it lists all entries. However, certain entries can be extracted with qualifiers.

Syntax

```
show log      [-cabinet <cabinet> -drawer <drawer> [MBM, PBM]]
              [-start <n>] [-end <n>], [-entry <n>] [, -last <n>, -since <date>]
              [-reverse] [search <string>]
              [{-full, -dump, -summary, -brief}]
```

Arguments

MBM The micro MBM (optional).
PBM The micro PBM (optional).

Options

-brief	This keeps the user from getting screens full of data unless they want it.
-cabinet<cabinet>	Cabinet number in hex. This qualifier is optional to extract the micros on this cabinet. Without this qualifier, show all cabinets.
-drawer<drawer>	Drawer number in hex. This qualifier is optional to extract the micros with this drawer number. Without this qualifier, show all drawers.
-dump	This will display the log as a series of hex long words so the raw data can be examined. This helps debugging EV7 PAL logout frame and other binary data.
-end<n>	The number of the last error entry to be displayed. The default is the last entry.
-entry	This will display the single log entry specified.
-full	Display the full logout frame and other lengthy packets.
-last<n>	The most recent <n> entries to be displayed.
-reverse	Display entries in reverse order. That means that the most current entries are displayed first.

-search <string>	Extract only text entries that match the given string (case insensitive). If the string contains spaces, please enclose it with double quotes "".
-since <date>	Displays entries after a given date, which is in the format <i>yyyy/mm/dd</i> .
-start<n>	The number of the first error entry to be displayed. The default is the first entry.
-summary	Display a count of the number of text, diag, logout, sysevent, and binary logs in the MBM/PBM.

Examples

1. Show the entire system environment summary while the system is on, then off.

```
MBM> sh log

Error log from MBM in Cabinet 0, Drawer 0
-----
Entry 0: At 2002/04/16 19:04:21
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
Entry 1: At 2002/04/16 19:18:55
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
Entry 2: At 2002/04/16 19:25:18
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
Entry 3: At 2002/04/17 19:27:55
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
Entry 4: At 2002/04/17 19:32:13
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
Entry 5: At 2002/04/17 19:35:03
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
Entry 6: At 2002/04/18 19:45:36
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)

Error log from MBM in Cabinet 0, Drawer 1
-----
~CLI-E-(tCLImodem) error log is empty

Error log from MBM in Cabinet 0, Drawer 2
-----
Entry 11: At 2002/04/18 22:51:08
~ENV-W-(envmon) Fan 0 (Front Fan) warning:1814RPM, from MBM(0,2)
Entry 12: At 2002/04/18 22:51:08
~ENV-W-(envmon) Fan 1 (Back Fan) warning:1795RPM, from MBM(0,2)
Entry 13: At 2002/04/18 22:51:24
~ENV-I-(envmon) Fan 0 (Front Fan) returned to normal, from MBM(0,2)
Entry 14: At 2002/04/18 22:51:24
~ENV-I-(envmon) Fan 1 (Back Fan) returned to normal, from MBM(0,2)

Error log from PBM in Cabinet 1, Drawer 1
-----
Entry 0: At 2081/09/20 23:37:17
~ENV-W-(envmon) Cover 0 (Hot Plug) is open, from PBM(1,1)
Entry 1: At 2081/09/20 23:37:17
~ENV-W-(envmon) Cover 1 (Non Hot Plug) is open, from PBM(1,1)

Error log from PBM in Cabinet 1, Drawer 2
-----
~CLI-E-(tCLImodem) error log is empty

MBM>
```

2. Extract specific entries from a micro.

```
MBM> sh log -ca 0 -dr 0 -start 2 -end 4

Error log from MBM in Cabinet 0, Drawer 0
-----
Entry 2: At 2002/04/17 19:25:18
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
Entry 3: At 2002/04/17 19:27:55
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
Entry 4: At 2002/04/17 19:32:13
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)

MBM>
```

3. Extract the five most recent entries from a specific micro.

```
MBM> sh log -ca 0 -dr 4 -last 5
Error log from PBM in Cabinet 0, Drawer 4
-----
Entry 459: At 2056/06/23 18:52:29
~ENV-E-(envmon) PS 0 (PS0) AC failure, from PBM(0,4)
Entry 460: At 2056/06/23 18:54:44
~ENV-W-(envmon) Cover 0 (Hot Plug) is open, from PBM(0,4)
Entry 461: At 2056/06/23 18:54:46
~ENV-E-(envmon) PS 0 (PS0) AC failure, from PBM(0,4)
Entry 462: At 2056/06/23 19:24:15
~ENV-W-(envmon) Cover 0 (Hot Plug) is open, from PBM(0,4)
Entry 463: At 2056/06/23 19:24:17
~ENV-E-(envmon) PS 0 (PS0) AC failure, from PBM(0,4)

MBM>
```

4. Display the most recent entries first.

```
MBM> sh log -reverse

Error log from MBM in Cabinet 0, Drawer 0
-----
Entry 6: At 2002/04/17 19:45:36
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
Entry 5: At 2002/04/17 19:35:03
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
Entry 4: At 2002/04/17 19:32:13
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
Entry 3: At 2002/04/17 19:27:55
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
Entry 2: At 2002/04/17 19:25:18
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
Entry 1: At 2002/04/17 19:18:55
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
Entry 0: At 2002/04/17 19:04:21
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
```

5. Search for fan failure event.

```
MBM> sh log -search "front fan"

Error log from MBM in Cabinet 0, Drawer 2
-----
Entry 11: At 2002/04/18 22:51:08
~ENV-W-(envmon) Fan 0 (Front Fan) warning:1814RPM, from MBM(0,2)
Entry 13: At 2002/04/18 22:51:24
~ENV-I-(envmon) Fan 0 (Front Fan) returned to normal, from MBM(0,2)
```

6. Display using the "-dump" option

```

MBM>
MBM> show log -dump -ca 0 -dr 0 mbm

Error log from MBM in Cabinet 0, Drawer 0
-----
Entry 0: At 2002/11/25 16:50:10
  Text Report
01a37780:                               457e003c *           < ~E*
01a37790: 492d564e 6e65282d 6e6f6d76 424d2029 *NV-I-(envmon) MB*
01a377a0: 6163284d 30303a62 7264202c 29303a77 *M(cab:00, drw:0)*
01a377b0: 31535020 6e614620 73616820 65656220 * PS1 Fan has bee*
01a377c0: 6572206e 65766f6d *n remove.....*
Entry 1: At 2002/11/25 17:07:51
  Text Report
01a37780:                               457e003a *           : ~E*
01a37790: 492d564e 6e65282d 6e6f6d76 424d2029 *NV-I-(envmon) MB*
01a377a0: 6163284d 30303a62 7264202c 29303a77 *M(cab:00, drw:0)*
01a377b0: 31535020 6e614620 73616820 65656220 * PS1 Fan has bee*
01a377c0: 6461206e *n ad.....*
Entry 2: At 2002/11/25 17:08:01
  Text Report
01a37780:                               457e003e *           > ~E*
01a37790: 492d564e 6e65282d 6e6f6d76 424d2029 *NV-I-(envmon) MB*
01a377a0: 6163284d 30303a62 7264202c 29303a77 *M(cab:00, drw:0)*
01a377b0: 31535020 6e614620 74657220 656e7275 * PS1 Fan returne*
01a377c0: 6f742064 726f6e20 *d to nor.....*
Entry 3: At 2002/11/25 17:08:08
  Text Report
01a37780:                               457e0038 *           8 ~E*
01a37790: 492d564e 6e65282d 6e6f6d76 424d2029 *NV-I-(envmon) MB*
01a377a0: 6163284d 30303a62 7264202c 29303a77 *M(cab:00, drw:0)*
01a377b0: 30535020 73616820 65656220 6572206e * PS0 has been re*
01a377c0: 65766f6d *move.....*
Entry 4: At 2002/11/25 17:08:08
  Text Report
01a37780:                               457e0036 *           6 ~E*
01a37790: 492d564e 6e65282d 6e6f6d76 424d2029 *NV-I-(envmon) MB*
01a377a0: 6163284d 30303a62 7264202c 29303a77 *M(cab:00, drw:0)*
01a377b0: 31535020 73616820 65656220 6461206e * PS1 has been ad*
Entry 5: At 2002/11/25 17:08:14
  Text Report
01a37780:                               457e003c *           < ~E*
01a37790: 492d564e 6e65282d 6e6f6d76 424d2029 *NV-I-(envmon) MB*
01a377a0: 6163284d 30303a62 7264202c 29303a77 *M(cab:00, drw:0)*
01a377b0: 30535020 6e614620 73616820 65656220 * PS0 Fan has bee*
01a377c0: 6572206e 65766f6d *n remove.....*
Entry 6: At 2002/11/25 17:08:34
  Text Report
01a37780:                               457e003c *           < ~E*
01a37790: 492d564e 6e65282d 6e6f6d76 424d2029 *NV-I-(envmon) MB*
01a377a0: 6163284d 30303a62 7264202c 29303a77 *M(cab:00, drw:0)*
01a377b0: 30535020 6e614620 73616820 65656220 * PS0 Fan has bee*
01a377c0: 6572206e 65766f6d *n remove.....*

```

```
MBM>
```

7. Example of the "-summary" option.

```

MBM>
MBM> show log -summary -ca 0 -dr 0 mbm

Error log from MBM in Cabinet 0, Drawer 0
-----
9 Text Reports in Log

MBM>

```

8. Extract events after a given date.

```
MBM> show log -ca 0 -dr 0 mbm -since 2002/04/17
Error log from MBM in Cabinet 0, Drawer 0
-----
Entry 4: At 2002/04/17 19:32:13
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
Entry 5: At 2002/04/17 19:35:03
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
Entry 6: At 2002/04/18 19:45:36
~ENV-W-(envmon) Cover 0 (Main Drawer) is open, from MBM(0,0)
MBM>
```

See Also

clear log

3.58 show membership

Show membership – display the list of micros in the data base. This command will display all the micros in the data base. The micros listed may not reflect the current set of micros that would be listed as seen in the **show network** command. If a new micro is added and the **show membership** command lists the new micro, then the group will be reformed with the newly added micro.

Syntax

show membership

Arguments

None

Options

None

Example

```
MBM>
MBM> show membership
Cab Drw Module      MicroID   IP Address   In Group
1   3   PBM       103fe0a  10.254.3.1   TRUE
1   4   PBM       104fe0a  10.254.4.1   TRUE
0   2   MBM       100020a  10.2.0.1     TRUE
0   3   MBM       100030a  10.3.0.1     TRUE
0   1   MBM       100010a  10.1.0.1     TRUE
1   2   PBM       102fe0a  10.254.2.1   TRUE
1   1   PBM       101fe0a  10.254.1.1   TRUE
0   0   MBM       100000a  10.0.0.1     TRUE
MBM>
```

See Also

set membership, show network

3.59 show memory

Displays memory by physical location or by partitioning. By default, this command displays total physical memory, which is based on RIMM size and population. After assigning resources to individual partitions, memory is divided into chunks.

Syntax

show memory [-physical, -partition]

Arguments

None

Options

-physical Show memory per CPU and total memory for the system.
 -partition Show memory by partitioning assignment.

Examples

1. Show the entire system memory.

```
MBM> sh mem # same as show mem -phys
Cab Drw CPU Memory Size
0 0 0 1024MB
0 0 1 1024MB
0 0 2 1024MB
0 0 3 1024MB
0 0 4 1024MB
0 0 5 1024MB
0 0 6 1024MB
0 0 7 1024MB
0 1 0 1024MB
0 1 1 1024MB
0 1 2 1024MB
0 1 3 1024MB
0 1 4 1024MB
0 1 5 1024MB
0 1 6 1024MB
0 1 7 1024MB
0 2 0 1024MB
0 2 1 1024MB
0 2 2 1024MB
0 2 3 1024MB
0 2 4 1024MB
0 2 5 1024MB
0 2 6 1024MB
0 2 7 1024MB
0 3 0 1024MB
0 3 1 1024MB
0 3 2 1024MB
```

```
0 3 3 1024MB
0 3 4 1024MB
0 3 5 1024MB
0 3 6 1024MB
0 3 7 1024MB
```

Total Physical Memory: 32768MB (32.000GB)

MBM>

2. Show memory by partitions.

```
MBM> sh mem -part
```

```
Hard Partition: HP No.= 0 with total physical memory 8192MB (8.000GB)
```

```
Sub Partition: SP No.=255 with total memory 0MB in 0 chunks:
```

```
Sub Partition: SP No.= 0 with total memory 8192MB in 16 chunks:
```

```
chunk 0, size 16MB, start address 0x00000000,00000000
chunk 1, size 1008MB, start address 0x00000000,01000000
chunk 2, size 16MB, start address 0x00000004,00000000
chunk 3, size 1008MB, start address 0x00000004,01000000
chunk 4, size 16MB, start address 0x00000008,00000000
chunk 5, size 1008MB, start address 0x00000008,01000000
chunk 6, size 16MB, start address 0x0000000c,00000000
chunk 7, size 1008MB, start address 0x0000000c,01000000
chunk 8, size 16MB, start address 0x00000020,00000000
chunk 9, size 1008MB, start address 0x00000020,01000000
chunk 10, size 16MB, start address 0x00000024,00000000
chunk 11, size 1008MB, start address 0x00000024,01000000
chunk 12, size 16MB, start address 0x00000028,00000000
chunk 13, size 1008MB, start address 0x00000028,01000000
chunk 14, size 16MB, start address 0x0000002c,00000000
chunk 15, size 1008MB, start address 0x0000002c,01000000
```

```
Community: No.= 0 with total memory 0MB in 0 chunks:
```

```
Hard Partition: HP No.= 1 with total physical memory 0MB (0.000GB)
```

```
Sub Partition: SP No.=255 with total memory 0MB in 0 chunks:
```

```
Sub Partition: SP No.= 0 with total memory 0MB in 0 chunks:
```

```
Community: No.= 0 with total memory 0MB in 0 chunks:
```

MBM>

See Also

show partition, show system, show duo

3.60 show modem

Displays remote management settings. This command displays modem strings, password information, alerts, baud rates, and other server management status.

Syntax

show modem

Arguments

None

Options

None

Examples

Display remote management status and information.

```
MBM> show modem
```

```
Remote System Management Settings
```

```
SM escape sequence : TBD
OCP power switch   :
OCP secure switch  :
Remote user        :
Remote access      : TBD
Remote alert       : TBD
Modem baud         : TBD
Modem flow control : TBD
Modem password     : TBD
Modem init string  : TBD
Modem dial string  : TBD
Modem alert string : TBD
Alert pending      :
Most recent alert  :
MBM>
```

3.61 show network

Displays all MBM, PBM, CMM, and CPU IP addresses on the local LAN.

Syntax

show network

Arguments

None

Options

None

Example

Display the system topology

```
MBM> sh net
Cab Drw Module      IP Address      Parent IP
0  0  MBM             10.0.0.1        0.0.0.0
0  0    CMM0          10.0.1.0        10.0.0.1
0  0    CPU0          10.0.1.1        10.0.1.0
0  0    CPU1          10.0.1.2        10.0.1.0
0  0    CMM1          10.0.2.0        10.0.0.1
0  0    CPU2          10.0.2.1        10.0.2.0
0  0    CPU3          10.0.2.2        10.0.2.0
0  1  MBM             10.1.0.1        0.0.0.0
0  1    CMM0          10.1.1.0        10.1.0.1
0  1    CPU0          10.1.1.1        10.1.1.0
0  1    CPU1          10.1.1.2        10.1.1.0
0  1    CMM1          10.1.2.0        10.1.0.1
0  1    CPU2          10.1.2.1        10.1.2.0
0  1    CPU3          10.1.2.2        10.1.2.0
1  1  PBM             10.254.1.1      0.0.0.0
1  2  PBM             10.254.2.1      0.0.0.0
1  3  PBM             10.254.3.1      0.0.0.0
1  4  PBM             10.254.4.1      0.0.0.0
0  0  PMUS           10.253.0.1
-  -  DHCP           10.253.0.1-253
-  -  VMS            10.253.0.254
MBM>
```

See Also

show system, show membership

3.62 show partition

Displays partition attributes and resources. This command confirms the partition settings. By default it lists all partitions. The "Status" column for IOR lists either "P" or "F", where P indicates that the IOR passed XSROM tests, and F indicates that the IOR failed XSROM tests and was disabled.

Syntax

```
show partition [{-hp <hard_partition> [-sp <sub_partition>}], -all [-brief]
```

Arguments

None

Options

-hp <hard_partition>	Hard partition name.
-sp <sub_partition>	Subpartition name. All subpartitions under the hard partition are displayed if no subpartition is specified.
-all	All partitions.
-brief	Highlight the partition attributes, number of CPUs and connected IOs

Examples

1. List all partitions.

```
MBM> show part

-----
Hard Partition : HP Name = USA_branch, HP No.= 0, SP count = 2
Attributes     : max CPUs = 32, SP type = soft, Non-stripe
Power State    : Power OffPhysical Memory: 8192MB (8.000GB)

Community Memory: 0MB (0.000GB)
Sub Partition: HP Name = USA_branch, HP No.= 0
               SP Name = hr, SP No.= 0
               State = Not Running, Telnet port = 323
Assigned Memory: 1024MB (1.000GB)

CPUs:
  Cab  Drw  CPU  (NS,EW)  PID  Type
  ---  ---  ---  ---      ---  ---
  0    0    0    ( 0,0 )    0    Primary
  0    0    2    ( 0,1 )    2    Non-primary
  0    0    4    ( 0,2 )    4    Non-primary
  0    0    6    ( 0,3 )    6    Non-primary
  0    0    1    ( 1,0 )    1    Non-primary
  0    0    3    ( 1,1 )    3    Non-primary
  0    0    5    ( 1,2 )    5    Non-primary
  0    0    7    ( 1,3 )    7    Non-primary

IOPs:
      SBB
  Cab  Drw  IOP  (NS,EW)  -----  PCI Drawer  Status
  ---  ---  ---  ---      ---
  0    0    0    ( 0,0 )    1    1    0    P
  0    0    2    ( 0,1 )
  0    0    4    ( 0,2 )
  0    0    6    ( 0,3 )
  0    0    1    ( 1,0 )
  0    0    3    ( 1,1 )
  0    0    5    ( 1,2 )
  0    0    7    ( 1,3 )

Sub Partition: HP Name = USA_branch, HP No.= 0
               SP Name = Free_Pool, SP No.= 255

Free Memory: 0MB (0.000GB)

CPUs: None           IOPs: None

MBM>
```

2. Show only one subpartition.

```

MBM> sh part -hp USA_branch -sp hr
-----
Hard Partition : HP Name = USA_branch, HP No.= 0, SP count = 2
Attributes     : max CPUs = 32, SP type = soft, Non-stripe
Power Stae     : Power Off
Physical Memory: 8192MB (8.000GB)

Community Memory: 0MB (0.000GB)

Sub Partition: HP Name = USA_branch, HP No.= 0
               SP Name = hr, SP No.= 0
               State = Not Running, Telnet port = 323

Assigned Memory: 1024MB (1.000GB)

CPUs:
  Cab  Drw  CPU  (NS,EW)  PID  Type
  0    0    0   ( 0,0 )    0   Primary
  0    0    2   ( 0,1 )    2   Non-primary
  0    0    4   ( 0,2 )    4   Non-primary
  0    0    6   ( 0,3 )    6   Non-primary
  0    0    1   ( 1,0 )    1   Non-primary
  0    0    3   ( 1,1 )    3   Non-primary
  0    0    5   ( 1,2 )    5   Non-primary
  0    0    7   ( 1,3 )    7   Non-primary

IOPs:
      SBB
      Cab  Drw  IOP  (NS,EW)  -----  PCI Drawer
      0    0    0   ( 0,0 )    1    1    0    P
      0    0    2   ( 0,1 )
      0    0    4   ( 0,2 )
      0    0    6   ( 0,3 )
      0    0    1   ( 1,0 )
      0    0    3   ( 1,1 )
      0    0    5   ( 1,2 )
      0    0    7   ( 1,3 )

MBM>

```

3. Show partitions in brief format.

```

MBM> sh part -b
Partition Name      CPU Count  IO Count  State      Telnet Port
hp0                 32 max    2         Power Off
Default_SP          16        2         Not Running 323
hp1                 32 max    2         Power Off
Default_SP          8         1         Not Running 324
MBM>

```

See Also

create partition, delete partition

3. Show the detailed environment of drawer 0 in cabinet 0.

```

MBM> sh power -ca 0 -dr 0 -detail
SBB: cabinet = 0, drawer = 0
Power state: ON
Temperatures(in centigrade):
    MBM: 29.0 (P)    OCP: 23.0 (P)
Fans(in RPM):
    Front Fan: 2008 (P)    Back Fan: 2008 (P)
Power Interface: cabinet = 1, drawer = 0, type = WPI, slot = 0, status = OK
Power Supplies:
    PWR0.PS0: ON          PWR0.PS1: ...          PWR0.PS2: ON
Voltages:
    MBM 2.5V: 2.457V (P)  MBM 3.3V: 3.294V (P)  MBM 5.0V: 4.915V (P)

DUO0: cabinet = 0, drawer = 0
Power state: ON
Temperatures(in centigrade):
    CMM: 21.5 (P)    CPU0: 31.1 (P)          CPU1: 28.1 (P)
Voltages:
    EV7 Core 1.575V: 1.580V (P)    Term Volt 1.835V: 1.870V (P)
    RIMM Volt 2.523V: 2.496V (P)    EV79 Core 0.000V: 0.000V (P)
    CMM Volt 3.300V: 3.230V (P)
VRMs:
    CPU    RIMM  Term
    012345 01234 0
    P P P P P . . . P P P P P

DUO1: cabinet = 0, drawer = 0
Power state: ON
Temperatures(in centigrade):
    CMM: 26.0 (P)    CPU0: 31.6 (P)          CPU1: 30.1 (P)
Voltages:
    EV7 Core 1.575V: 1.570V (P)    Term Volt 1.783V: 1.830V (P)
    RIMM Volt 2.523V: 2.496V (P)    EV79 Core 0.000V: 0.000V (P)
    CMM Volt 3.300V: 3.213V (P)
VRMs:
    CPU    RIMM  Term
    012345 01234 0
    P P P P . . . P P P P P

MBM>

```

4. Show the detailed environment of PCI drawer 3 in cabinet 0.

```

MBM> sh power -ca 0 -dr 3 pci -detail
PCI Drawer: cabinet = 0, drawer = 3
Temperatures(in centigrade):
    PBM: 27.5 (P)    PCI_0: 24.5 (P)    PCI_1: 25.0 (P)
    OCP: 24.5 (P)    PS0: 24.0 (P)    PS1: 27.0 (P)
Fans(in RPM):
    Fan 0: 2721 (P) Fan 1: 2766 (P) Fan 2: 2766 (P)
Power Supplies:
    PS0: ON          PS1: ON
Voltages:
    PBM 2.5V: 2.470V (P)    PBM 3.3V: 3.294V (P)    PBM 5.0V: 4.915V (P)
    PCI 1.5V: 1.500V (P)    PCI 2.5V: 2.540V (P)    PCI 3.3V: 3.329V (P)
    PCI 5.0V: 5.127V (P)    PCI 12.0V:12.129V (P)    PCI VAUX: 5.048V (P)
    IOR0 1.5V: 1.500V (P)    IOR0 1.8V: 1.790V (P)

MBM>

```

5. Show the detailed environment of X-shelf PCI drawer 4 in cabinet 0.

```

MBM> sh power -ca 0 -dr 4 pci -detail
PCI Drawer: cabinet = 0, drawer = 4
Power state: ON
Temperatures(in centigrade):
    PBM: 27.5 (P)    PCI_0: 23.5 (P)    PCI_1: 24.0 (P)

```

```
    OCP: 25.0 (P)   PS0: 24.0 (P)   PS1: ...
Fans(in RPM):
    Fan 0: 2678 (P) Fan 1: 2616 (P) Fan 2: 2721 (P)
Power Supplies:
    PS0: ON        PS1: ...
Voltages:
    PBM 2.5V: 2.457V (P)   PBM 3.3V: 3.294V (P)   PBM 5.0V: 4.915V (P)
    PCI 1.5V: 1.500V (P)   PCI 2.5V: 2.540V (P)   PCI 3.3V: 3.294V (P)
    PCI 5.0V: 5.100V (P)   PCI 12.0V:12.129V (P)   PCI VAUX: 4.995V (P)
    IOR0 1.5V: 1.500V (P)   IOR0 1.8V: 1.790V (P)   IOR1 1.5V: 1.500V (P)
    IOR1 1.8V: 1.810V (P)   IOR2 1.5V: 1.490V (P)   IOR2 1.8V: 1.780V (P)
    IOR3 1.5V: 0.000V (P)   IOR3 1.8V: 0.000V (P)
MBM>
```

See Also

power on, power off

3.64 show sys_serial_num

Displays the system serial number. System serial number is limited to 12 characters and does not apply to RIMMs.

Syntax

show sys_serial_num

Arguments

None

Options

None

Examples

1. Show the system serial number.

```
MBM> sh sys_serial_num
sys_serial_num          MARVEL_FW16P
MBM>
```

2. Show the system serial number, which has not been set.

```
MBM> sh sys_serial
sys_serial_num is not set!
MBM>
```

See Also

set sys_serial_num

3.65 show system

Displays a summary of system components.

- P Current status Pass
- F Current status Fail
- D Current status Disabled for RIMMs
- p Past status Pass (currently not powered on)
- f Past status Fail (currently not powered on)
- d Past status Disabled for RIMMs
- ? Status unknown (typically status not obtained due to I²C error)
- . Not present or not applicable

MBM and PBM are powered by VAUX, they are always on with the status either P or F.

The 'IOP0' and 'IOP1' columns indicate which IOR is connected to the IOP.

The 'IOR<n>' column indicate which IOP connected to the IOR. If nothing is displayed for IOR1, IOR2 or IOR3, either there is a standard PCI drawer with at most 1 IOR, or an Expanded PCI drawer with a possible maximum of 4 IORs.

The 'Powered SBB' column indicates which SBB is being powered by the power supply sub-system.

Syntax

show system

Arguments

None

Options

None

Example

Display system summary.

```
MBM> sh sys
SBBs (System Building Block):
  Cab Drw Duo MBM CMM CPUs CPU0 RIMMs CPU1 RIMMs IOP0 IOP1
                   01 0123456789 0123456789 (Cab,Drw).ior (Cab,Drw).ior
  0 0 . P
  0 0 0 . P PP PPPP.PPPP. PPPP.PPPP. PCI( 1,1 ).0 .....
  0 0 1 . P PP PPPP.PPPP. PPPP.PPPP. ....
  0 0 2 . Filler .....
  0 0 3 . Filler .....
  0 1 . P
  0 1 0 . P PP PPPP.PPPP. PPPPPPPPPP PCI( 1,3 ).0 .....
  0 1 1 . P PP PPPP.PPPP. PPPP.PPPP. ....
  0 1 2 . Filler .....
```

AlphaServer ES47/ES80/GS1280 Server Management CLI Reference

```

0 1 3 . Filler .....
0 2 . P .....
0 2 0 . P PP PPPP.PPPP. PPPP.PPPP. ....
0 2 1 . P PP PPPP.PPPP. PPPP.PPPP. PCI( 1,4 ).0 .....
0 2 2 . Filler .....
0 2 3 . Filler .....
0 3 . P .....
0 3 0 . P PP PPPP.PPPP. PPPP.PPPP. PCI( 1,2 ).0 .....
0 3 1 . P PP PPPP.PPPP. PPPP.PPPP. ....
0 3 2 . Filler .....
0 3 3 . Filler .....
PCI Drawers:
Cab Drw PBM IORs          IOR0          IOR1          IOR2          IOR3
          0123 (Cab,Drw).iop (Cab,Drw).iop (Cab,Drw).iop (Cab,Drw).iop
1 1 P P          SBB( 0,0 ).0
1 2 P P          SBB( 0,3 ).0
1 3 P P          SBB( 0,1 ).0
1 4 P P          SBB( 0,2 ).2
SBB Power Sub-systems:
Cab Drw Type WPI/SDI Power Supply Powered SBB
          0123 0123456789AB SBB(cab,drw)
1 0 WPI P... PPP..... SBB( 0,0 )
1 1 WPI P... PPP..... SBB( 0,1 )
1 2 WPI P... PPP..... SBB( 0,2 )
1 3 WPI P... PPP..... SBB( 0,3 )
MBM> p off -all
MBM> sh sys
SBBs (System Building Block):
Cab Drw Duo MBM CMM CPUs CPU0 RIMMs CPU1 RIMMs IOP0 IOP1
          01 0123456789 0123456789 (Cab,Drw).ior (Cab,Drw).ior
0 0 . P .....
0 0 0 . p pp pppp.pppp. pppp.pppp. PCI( 1,1 ).0 .....
0 0 1 . p pp pppp.pppp. pppp.pppp. ....
0 0 2 . Filler .....
0 0 3 . Filler .....
0 1 . P .....
0 1 0 . p pp pppp.pppp. pppppppppp PCI( 1,3 ).0 .....
0 1 1 . p pp pppp.pppp. pppp.pppp. ....
0 1 2 . Filler .....
0 1 3 . Filler .....
0 2 . P .....
0 2 0 . p pp pppp.pppp. pppp.pppp. ....
0 2 1 . p pp pppp.pppp. pppp.pppp. PCI( 1,4 ).0 .....
0 2 2 . Filler .....
0 2 3 . Filler .....
0 3 . P .....
0 3 0 . p pp pppp.pppp. pppp.pppp. PCI( 1,2 ).0 .....
0 3 1 . p pp pppp.pppp. pppp.pppp. ....
0 3 2 . Filler .....
0 3 3 . Filler .....
PCI Drawers:
Cab Drw PBM IORs          IOR0          IOR1          IOR2          IOR3
          0123 (Cab,Drw).iop (Cab,Drw).iop (Cab,Drw).iop (Cab,Drw).iop
1 1 P p          SBB( 0,0 ).0
1 2 P p          SBB( 0,3 ).0
1 3 P p          SBB( 0,1 ).0
1 4 P p          SBB( 0,2 ).2
SBB Power Sub-systems:
Cab Drw Type WPI/SDI Power Supply Powered SBB
          0123 0123456789AB SBB(cab,drw)
1 0 WPI p... ppp..... SBB( 0,0 )
1 1 WPI p... ppp..... SBB( 0,1 )
1 2 WPI p... ppp..... SBB( 0,2 )
1 3 WPI p... ppp..... SBB( 0,3 )
MBM>

```

See Also

show network, show duo, show memory

3.66 show time

Shows the common base time and the delta time. This command shows the common base time that is shared by all microprocessors. The primary MBM provides the base time. This command also shows the time specific to each partition. If delta times are not set, the Default_SP of the Default_HP displays -1 for each delta time field.

Syntax

show time

Arguments

None

Options

None

Example

Show the common base time and delta times of all subpartitions.

```
MBM> sh time
Common base time: 2002/04/17 10:59:07

HP Name           SP Name           Delta Time (Days hh:mm:ss)
hp0                Default_SP        +00 +00:+00:+00
hp1                Default_SP        -01 -01:-01:-01
MBM>
```

See Also

set time

3.67 show version

Displays the firmware versions. The firmware modules are CMMFW, CMMFSL, CMMFPGA, SRROMFW, XSROMFW, SRMFW, MBMFW, MBMFSL, PBMFW, PBMFSL, PBMFPGA, XSHFPGA, and MBMFPGA. MBMFPGA applies to the ES80 server only. XSHFPGA applies to the XShelf PCI drawer only.

An asterisk at the end of a firmware module name (for example, CMMFW*) indicates that a backup copy is stored in the MBM flash.

Image revision is a combination of fields no bigger than a 16 byte null terminated string that takes on the format

<x><mmm>.<nnn>-<ssss>

where:

<x>	V	Released version
	T	Test version; pre-release checkout; internally distributed and supported
	X	Unsupported - not for distribution
<mmm>		1 to 3 digits reflecting the major version followed by a dot.
<nnn>		1 to 3 digits reflecting the minor version followed by a dash.
<sssss>		1 to 5 digits reflecting the build number.

Syntax

```
show version      [-run] [-cabinet <cabinet> -drawer <drawer>]
                  [CMM<n>, PBM, MBM]
                  [-fw <fw_module>]
```

Arguments

CMM<n>	The dual-processor module by cabinet and drawer number, with n referring to the module number, ranging from 0 to 3.
PBM	The flash of PBM by cabinet and drawer number.
MBM	The flash of MBM by cabinet and drawer number..

Options

-run	Display the versions of running firmware, which is in RAM.
-cabinet <cabinet>	Cabinet number in hex. If this qualifier is not used, all cabinets are shown.
-drawer <drawer>	Drawer number in hex. If this qualifier is not used, all drawers are shown.
-fw <fw_module>	Display the specified firmware module.

Examples

1. Display all firmware revisions.

```

MBM> sh ver
Local MBM(cab:00, drw:0) FW version X2.2-144 built on Aug 10 2003 at 22:26:07

Cab Drw Micro   FW Module           Flash Firmware Revision
0  0  MBM           MBMFW               X2.2-144
                   MBMFSL              V2.1-1
                   PBMFPGA*            X4.1-01
                   XSHFPGA*            X3.1-03
                   CMMFW*              T2.3-2
                   CMMFSL*             V2.1-22
                   CMMFPGA*            V114
                   SROMFW*             V1.0-7
                   XSROMFW*            T1.0-12
                   SRMFW*             X6.6-2914
0  0  CMM0          CMMFW               T2.3-2
                   CMMFSL              V2.1-22
                   CMMFPGA            V114
                   SROMFW             V1.0-7
                   XSROMFW            T1.0-12
                   SRMFW             X6.6-2914
0  0  CMM1          CMMFW               T2.3-2
                   CMMFSL              V2.1-22
                   CMMFPGA            V114
                   SROMFW             V1.0-7
                   XSROMFW            T1.0-12
                   SRMFW             X6.6-2914
0  1  MBM           MBMFW               X2.2-144
                   MBMFSL              V2.1-1
                   PBMFPGA*            X4.1-01
                   XSHFPGA*            X3.1-03
                   CMMFW*              T2.3-2
                   CMMFSL*             V2.1-22
                   CMMFPGA*            V114
                   SROMFW*             V1.0-7
                   XSROMFW*            T1.0-12
                   SRMFW*             X6.6-2914
0  1  CMM0          CMMFW               T2.3-2
                   CMMFSL              V2.1-22
                   CMMFPGA            V114
                   SROMFW             V1.0-7
                   XSROMFW            T1.0-12
                   SRMFW             X6.6-2914
0  1  CMM1          CMMFW               T2.3-2
                   CMMFSL              V2.1-22
                   CMMFPGA            V114
                   SROMFW             V1.0-7
                   XSROMFW            T1.0-12
                   SRMFW             X6.6-2914
1  1  PBM           PBMFW               X2.2-144
                   PBMFSL              V2.1-1
                   PBMFPGA            X4.1-01
                   CMMFW*              T2.3-2
                   CMMFSL*             V2.1-22
                   CMMFPGA*            V114
                   SROMFW*             V1.0-7
                   XSROMFW*            T1.0-12
                   SRMFW*             X6.6-2914
1  2  PBM           PBMFW               X2.2-144
                   PBMFSL              V2.1-1
                   XSHFPGA            X3.1-03
                   CMMFW*              T2.3-2
                   CMMFSL*             V2.1-22
                   CMMFPGA*            V114
                   SROMFW*             V1.0-7
                   XSROMFW*            T1.0-12

```

```
SRMFW* X6.6-2914
MBM>
```

2. Show the firmware version by cabinet and drawer:

```
MBM> sh vers -ca 0 -dr 0 mbm
Local MBM(cab:00, drw:0) FW version X2.2-144 built on Aug 10 2003 at 22:26:07
Cab Drw Micro FW Module Flash Firmware Revision
0 0 MBM MBMFW X2.2-144
MBMFSL V2.1-1
PBMFPGA* X4.1-01
XSHFPGA* X3.1-03
CMMFW* T2.3-2
CMMFSL* V2.1-22
CMMFPGA* V114
SRMFW* V1.0-7
XSROMFW* T1.0-12
SRMFW* X6.6-2914
MBM>
```

3. Show the firmware version by a particular module:

```
MBM> sh version -fw mbmfw
Local MBM(cab:00, drw:0) FW version X2.2-144 built on Aug 10 2003 at 22:26:07
Cab Drw Micro FW Module Flash Firmware Revision
0 0 MBM MBMFW X2.2-144
0 1 MBM MBMFW X2.2-144
1 1 PBM PBMFW X2.2-144
1 2 PBM PBMFW X2.2-144
MBM>
```

See Also

update

3.68 telnet

Log in to a remote host. Users can telnet to the Vxworks shell (port 322) or CLI session of a remote MBM/PBM.

Syntax

```
telnet <ip_address> [port]
```

Arguments

ip_address IP address of MBM or PBM

port The port number to connect to. The default is 23.

Options

None

Examples

1. Telnet to a CLI session, then exit using escape sequence.

```
MBM>
MBM> telnet 10.254.2.0
Connecting to 10.254.2.0.
Exit sequence is ^^[MBM.
Welcome - GS1280 Server Manager - T2.1-6
```

```
MBM>
MBM> # type ^^[MBM to close the connection
Closed connection.
MBM>
```

2. Telnet to a CLI session, then exit using **logout** command.

```
MBM>
MBM> telnet 10.254.2.0
Connecting to 10.254.2.0.
Exit sequence is ^^[MBM.
Welcome - GS1280 Server Manager - T2.1-6
MBM>
MBM> logout

Closed connection.
MBM>
```

3. Telnet to a Vxworks shell for debugging, then exit

```
MBM> telnet 10.1.0.1 322
Connecting to 10.1.0.1.
Exit sequence is ^^[MBM.
.
.
.
10.1.0.1->logout
Closed connection.
MBM>
```

See Also

logout, connect, hangup

3.69 test

Executes a diagnostic test on the specified CPU or hard partition.

This command sends a request to the CMM to start a test on its client CPU. The CPU can be specified by either cabinet and drawer number or by east-west and north-south coordinates.

Without specifying CPU or hard partition, the test will be executed on the default hard partition if the system has only one hard partition.

In order to run an individual test, a partition must be reset or powered on to XSROM level.

Syntax

```
test <test_num> [ {[-cabinet <cabinet> -drawer <drawer>] CPU<n>},
                {-ew <ew_coord> -ns <ns_coord> CPU},
                {-hp <hard_partition> } ]
                [<parameters>...] [-p <pass_count>]
```

Arguments

test_num A diagnostic test number.
parameters... Four parameters associated with the test if needed.

Options

-cabinet <cabinet>	Cabinet number in hex. If not specified, defaults to 0.
-drawer <drawer>	Drawer number in hex. If not specified, defaults to 0.
CPU<n>	The CPU by cabinet and drawer number. <i>n</i> refers to the CPU number in the 8P drawer and ranges from 0 to 7.
-ew <ew_coord>	East-West coordinate of the CPU in hex.
-ns <ns_coord>	North-South coordinate of the CPU in hex.
CPU	The CPU by mesh coordinate. The qualifiers -ew and -ns must be used.
-hp <hard_partition>	Hard partition name. Tests are invoked on all CPUs in the hard partition.
-p	Pass count. The default is 1. If 0, then run until CTRL/C (^C) is pressed.

Examples

1. Run test 11 on CPU2 in drawer 3 of cabinet 2.

```
MBM> test 11 -ca 2 -dr 3 CPU2
MBM>
```

or

```
MBM> test 11 -ew 5 -ns b CPU
MBM>
```

2. Run test 13 on all CPUs in the hard partition Default_HP

```
MBM> test 13
Running test 13, RAID channel Test ... on 2 EV7s
[2003/04/18 15:22:38]
~DIA-W-(pco_04) Test 13 [RAID channel Test] failed on cpu [NS: 0 EW: 0] which i0
[2003/04/18 15:22:38]
  BEGIN DIAGNOSTIC TEST FAILURE INFO BEGIN
  Test target cabinet:00 drawer:0 CPU0
  test number: 13 (hex) [RAID channel Test]
  test status: 03
  rsvd1: 00
  result length: 003e
  revision: T1.0-9_ocla
  error number: 01 -- Write, Verify Read error
  rsvd2: 00
  error format: 01
  severity code: 01
  FRU1: 0
  FRU2: 0
  FRU3: 0
  FRU4: 0
  P1: ffff8000.00000000 Expected data (written)
  P2: 00000000.00000000 Actual data received (read)
  P3: ffff8000.00000000 Failing Address
  P4: 00000000.00000000 (null)
  P5: 00000000.00000000 (null)
  P6: 00000000.00000000 (null)
  P7: 00000000.00000000 (null)
  P8: 00000000.00000000 (null)
  FRU suspect: EV7, Memory
  FRU extra: Ev7 not capable of RAID cab:00 drw:0 CPU0
  END DIAGNOSTIC TEST FAILURE INFO END
MBM>
```

See Also

power on, reset, enable test, disable test

3.70 test alert

Tests the modem setup and connection. This command simulates an alert condition by immediately sending an alert. The modem must already be initialized and the sending of alerts must be enabled before this command can be issued.

Syntax

test alert

Arguments

None

Options

None

Example

Test the modem connection.

```
MBM> test alert  
TBD  
MBM>
```

See Also

enable alert, set alert, set dial, clear alert

3.71 test led

Tests all IP and IO cable port LEDs. This command turns the cable LEDs on or off. Setting it for 0 seconds turns the LEDs off. The default is to blink the LEDs for 60 seconds. The maximum is 3600 seconds.

Syntax

test led [<seconds>]

Argument

seconds The amount of time to leave the cable LEDs on or to turn them off.

Options

None

Examples

1. Blink the cable LEDs for the default amount of time (60 seconds).

```
MBM> test led
MBM>
```

2. Turn off the cable LEDs.

```
MBM> test led 0
MBM>
```

See Also

config cables

3.72 update

Updates the firmware on any component. This command updates the flash image on any of the nodes currently responding on the server management network, or it loads the temporary image into RAM. By default, it updates the flash.

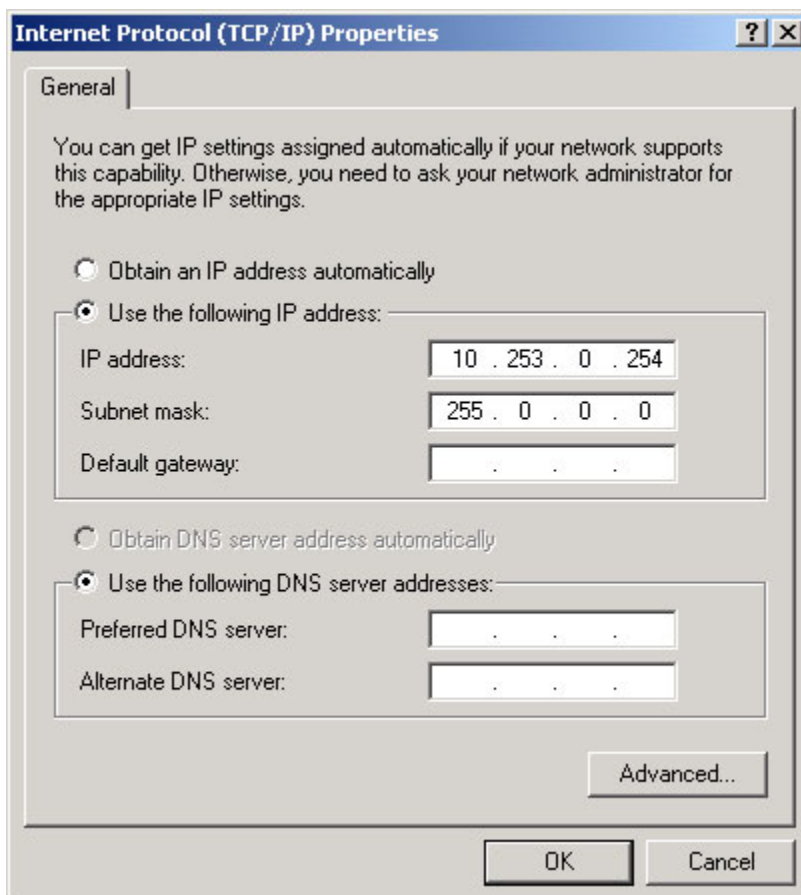
The firmware modules are CMMFW, CMMFSL, CMMFPGA, SRROMFW, XSROMFW, SRMFW, MBMFW, MBMFSL, PBMFW, PBMFSL, PBMFPGA, and MBMFPGA. MBMFPGA applies to the ES80 server only.

The same firmware is used to update all modules if no particular device is specified.

MBM and PBM share the same firmware image. Updating the MBMFW or MBMFSL also updates PBMFW or PBMFSL, and the reverse is true.

NOTE: This command uses TFTP communication between the MBM (which requests the image), and a connected PC (which serves the image). The individual update images are included on the AlphaServer ES47/ES80/GS1280 Firmware CD.

The PC must be connected to the Server Management hub via an Ethernet adapter. This connection can be static using the address "10.253.0/254" or dynamic using the DHCP protocol. To set up the PC for this select Start/Settings/Control Panel/Network and Dial Up Connections/Local Area Connection/ Properties, Internet Protocol (TCP/IP)/ Properties. The examples shown in this document use the static address "10.253.0.254".



If using DHCP, check "Obtain an IP Address automatically".

Start the TFTP Server on the PC by invoking tftpd32.exe. A copy is found on the AlphaServer ES47/ES80/GS1280 Firmware CD, to be copied to a convenient directory on your local PC. Then change the settings to point the directory containing the files to be upgraded.

Syntax

```
update <fw_module> <ip_address> [-file <file_name>]
        [-cabinet <cabinet> -drawer <drawer>] [<device>]
```

Arguments

fw_module Firmware module to be updated.
ip_address TFTP server IP address for file transfer.
device A specific device to be updated (optional). The device is MBM, PBM, or CMM<n>. If n is not specified, all CMM images in the 8P drawer are updated.

Options

-file <file_name> The optional firmware file name. If not supplied, the default file is used.
-cabinet <cabinet> Cabinet number in hex. If this qualifier is not used, all cabinets are shown.
-drawer <drawer> Drawer number in hex. If this qualifier is not used, all drawers are shown.

Examples

1. Update the CMM firmware in all flashes.

```
MBM> update cmmfw 10.253.0.254 -fi cmm3_t2_2_4.bin
Update of module CMMFW completed on node 10.0.0.1
Update of module CMMFW completed on node 10.0.1.0
Update of module CMMFW completed on node 10.0.2.0
Update of module CMMFW completed on node 10.0.3.0
Update of module CMMFW completed on node 10.0.4.0
Update of module CMMFW completed on node 10.1.0.1
Update of module CMMFW completed on node 10.1.1.0
Update of module CMMFW completed on node 10.1.2.0
Update of module CMMFW completed on node 10.1.3.0
Update of module CMMFW completed on node 10.1.4.0
Update of module CMMFW completed on node 10.254.2.0
Update of module CMMFW completed on node 10.254.5.0
Update is complete!
MBM>
```

2. Update all MBM and PBM images.

```
MBM> update mbmfw 10.253.0.254 -fi mbm_T2_1_7.bin
Update of module MBMFW completed on node 10.0.0.1
Update of module MBMFW completed on node 10.1.0.1
Update of module MBMFW completed on node 10.254.2.0
Update of module MBMFW completed on node 10.254.5.0
Update is complete!
MBM>
```

See Also

show version

3.73 uptime

Displays the elapsed time that micros have run since initialization. This command currently displays the up time of all MBMs and PBMs.

Syntax

```
uptime [-cabinet <cabinet_num> -drawer <drawer_num> ]
```

Arguments

MBM The MBM micro (optional).

PBM The PBM micro (optional).

Options

-cabinet <cabinet> Cabinet number in hex. If this qualifier is not used, all cabinets are shown.

-drawer <drawer> Drawer number in hex. If this qualifier is not used, all drawers are shown.

Examples

1. Display the uptime of all MBMs and PBMs.

```
MBM> uptime
Cab Drw Module      Uptime (Days hh:mm:ss)
0  0  MBM             0 15:14:31
0  1  MBM             0 15:41:12
0  2  MBM             0 15:41:12
0  3  MBM             0 00:00:19
1  1  PBM             0 15:41:12
1  2  PBM             0 15:41:12
1  3  PBM             0 15:41:12
1  4  PBM             0 15:41:12
MBM>
```

2. Display the uptime of the PBM in cabinet 0 with drawer ID 1.

```
MBM> uptime -ca 1 -dr 1
Cab Drw Module      Uptime (Days hh:mm:ss)
1  1  PBM             0 15:32:27
MBM>
```

Index

A

Alerts

 disable alert..... 3-30

 enable alert 3-34

assign component command..... 3-3

assign memory command 3-5

B

bootline command 3-7

build fru command 3-8

C

clear alert command 3-13

clear command 3-12

clear display command..... 3-14

clear error command 3-15

clear log command 3-17

clear port command 3-18

Commands

 assign component..... 3-3

 assign memory..... 3-5

 bootline..... 3-7

 build fru 3-8

 clear 3-12

 clear alert 3-13

 clear display 3-14

 clear error..... 3-15

 clear log..... 3-17

 clear port 3-18

 clear srmenv..... 3-19

 config cable..... 3-20

 connect..... 3-22

 create partition 3-24

 delete partition 3-26

 deposit or d 3-28

 disable alert..... 3-30

 disable remote..... 3-31

 disable test..... 3-32

 el 3-33

 enable alert 3-34

 enable remote 3-35

 enable test..... 3-35

 examine or e 3-37

 halt in..... 3-39

 halt out 3-41

 hangup 3-43

 help 3-44

 init modem..... 3-46

 locate..... 3-47

 logout 3-48

 mbmboot or @ 3-49

 modify partition..... 3-51

 power off 3-52

 power on 3-54

 reset 3-59

 save partition..... 3-63

 set..... 3-64

 set alert..... 3-65

 set baud..... 3-66

 set dial 3-67

 set escape 3-68

 set flow 3-69

 set init 3-70

 set membership..... 3-71

 set password 3-73

 set sys_serial_num 3-74

 set time..... 3-75

 shell 3-76

 show 3-77

 show cable 3-78

 show duo 3-80

 show error 3-84

 show fru..... 3-88

 show grid..... 3-92

 show log 3-93

 show membership 3-98

 show memory 3-99

 show modem..... 3-101

 show network 3-102

 show partition 3-103

 show power 3-106

 show sys_serial_num..... 3-110

 show system..... 3-111

 show time 3-113

 show version 3-114

 telnet..... 3-117

 test..... 3-119

 test alert..... 3-121

 test led..... 3-122

 update 3-123

 uptime..... 3-126

Commands..... 3-1

config cable command..... 3-20

connect command..... 3-22

create partition command 3-24

D

delete partition command 3-26

deposit command..... 3-28

disable alert command..... 3-30

disable remote command 3-31

disable test command..... 3-32

E

el command 3-33

enable alert command 3-34

enable remote command 3-35

enable test command 3-35

examine command	3-37	set membership	3-71
H		set password command.....	3-73
halt in command	3-39	set sys_serial_num command	3-74
halt out command	3-41	set time command.....	3-75
hangup command.....	3-43	show cable command	3-78
help command	3-44	show command.....	3-77
I		show duo command.....	3-80
init modem command	3-46	show error command	3-84
P		show fru command.....	3-88
Partitions		show log command.....	3-93
create partition	3-24	show membership.....	3-98
delete partition	3-26	show memory command.....	3-99
save partition.....	3-63	show modem command.....	3-101
show partition.....	3-103	show network command	3-102
power off command	3-52	show partition command.....	3-103
power on command	3-54	show power command.....	3-106
R		show sys_serial_num command.....	3-110
Remote		show system command	3-111
disable remote.....	3-31	show time command.....	3-113
enable remote	3-35	show version command	3-114
reset command.....	3-59	T	
S		test alert command	3-121
save partition command	3-63	test command	
set alert command	3-65	disable test	3-32
set baud command	3-66	enable test.....	3-35
set command	3-64	test command	3-119
set dial command.....	3-67	test led command.....	3-122
set escape command	3-68	U	
set flow command.....	3-69	update command	3-123
set init command.....	3-70	uptime command	3-126