HP StorageWorks

1000/1500 Modular Smart Array Command Line Interface user guide

This guide details the Command Line Interface (CLI), which is used to configure and manage the following products:
— HP StorageWorks 1000 Modular Smart Array (MSA1000)
— HP StorageWorks 1500 Modular Smart Array (MSA1500)

Part number: 347282–004
Fourth edition: June 2007

invent

Legal and notice information

HP StorageWorks 1000/1500 Modular Smart Array Command Line Interface user guide

Contents

Figures

Tables

About this guide

This user guide provides information to help you use the MSA Command Line Interface (CLI).

Intended audience

This book is intended for use by administrators with a moderate amount of SAN-management experience.

Related documentation

In addition to this guide, see the following related documents:

- MSA1500 compatibility guide
- *HP StorageWorks Modular Smart Array 1500 installation and configuration overview* (printed poster)
- *HP StorageWorks 1500 Modular Smart Array installation guide*
- *HP StorageWorks 1500 Modular Smart Array maintenance and service guide*
- MSA1000 compatibility guide
- *HP StorageWorks Modular Smart Array 1000 installation and configuration overview* (printed poster)
- *HP StorageWorks 1000 Modular Smart Array installation guide*
- *HP StorageWorks 1000 Modular Smart Array maintenance and service guide*
- *HP Array Configuration Utility user guide*
- *Configuring Arrays on HP Smart Array Controllers reference guide*

These and other HP documents can be found on the HP documents website: http://www.docs.hp.com.

Document conventions and symbols

Table 1 Document conventions

Convention	Element
Medium blue text: Figure 1	Cross-reference links and e-mail addresses
Medium blue, underlined text (http://www.hp.com)	Website addresses
Bold font	• Key names • Text typed into a GUI element, such as into a box • GUI elements that are clicked or selected, such as menu and list items, buttons, and check boxes
Italics font	Text emphasis
`Monospace font`	• File and directory names • System output • Code • Text typed at the command-line
`Monospace, italic font`	• Code variables • Command line variables
`Monospace, bold font`	Emphasis of file and directory names, system output, code, and text typed at the command line

⚠ **WARNING!** Indicates that failure to follow directions could result in bodily harm or death.

△ **CAUTION:** Indicates that failure to follow directions could result in damage to equipment or data.

IMPORTANT: Provides clarifying information or specific instructions.

NOTE: Provides additional information.

TIP: Provides helpful hints and shortcuts.

HP technical support

Telephone numbers for worldwide technical support are listed on the HP support website: http://www.hp.com/support/.

Collect the following information before calling:

- Technical support registration number (if applicable)
- Product serial numbers
- Product model names and numbers
- Applicable error messages
- Operating system type and revision level
- Detailed, specific questions

For continuous quality improvement, calls may be recorded or monitored.

HP strongly recommends that customers sign up online using the Subscriber's choice website at http://www.hp.com/go/e-updates.

- Subscribing to this service provides you with e-mail updates on the latest product enhancements, newest versions of drivers, firmware documentation updates, and instant access to numerous other product resources.
- After signing up, you can quickly locate your products by selecting **Business support** and then **Storage** under Product Category.

HP-authorized reseller

For the name of your nearest HP-authorized reseller:

- In the United States, call 1-800-345-1518.
- Elsewhere, see the HP website: http://www.hp.com. Click **Contact HP** to find locations and telephone numbers.

Helpful websites

For third-party product information, see the following HP websites:

- http://www.hp.com
- http://www.hp.com/go/storage
- http://www.hp.com/support/
- http://www.docs.hp.com

1 Overview and setup

The Command Line Interface (CLI) is used to configure and manage the MSA array controller and its storage. Some configuration and management tasks include configuring storage units (LUNs), limiting access to the storage, and viewing information on MSA components (controller, unit, and cache).

In addition, the CLI can be used to display system setup information and status. It may also provide information on devices that are attached to the controller.

NOTE: The CLI is available for all supported operating systems.

The CLI is an out-of-band utility, accessed through a host computer connected to the serial port of an MSA controller.

This chapter discusses:

About the CLI

To use the CLI, enter a command string at the CLI prompt (`CLI>`). Commands must use a specific, preset syntax. After a command is entered and executed, the results are displayed at the CLI prompt.

When using the CLI, the following keystrokes have special meaning:

Table 2 CLI special keys

Keyboard keys	Arrow key	Meaning
Ctrl+B	Left arrow	Move the cursor back one character
Ctrl+F	Right arrow	Move the cursor forward one character
Ctrl+P	Up arrow	Recall the previous command in the command buffer
Ctrl+N	Down arrow	Recall the next command in the command buffer

NOTE: The CLI uses a zero-based numbering system. For example, LUN number assignments begin with 0.

CLI commands in redundant configurations

When an MSA has two controllers, the same firmware image is run on both controllers. The two controllers communicate with each other through a PCI bus called the inter-controller link (ICL). Each controller has a serial port and has a CLI available to the users who connect the serial port to a serial terminal.

Some CLI commands are entered from one controller's CLI prompt but are executed from the other controller. Command syntax instructs the controller to accept user input, pass the command to the other controller, and then display the result.

The following keywords are used in the CLI to indicate a specific controller:

- `this_controller`—is included in command syntax to refer to the controller that the CLI is connected to.
- `other_controller`—is included in command syntax to refer to the other controller in the MSA.

CLI command syntax

As previously mentioned, commands are entered at the CLI prompt. Commands are **not** case sensitive and must be typed out in full.

CLI command strings include the basic command plus specific command options, some of which are mandatory and some of which are optional.

The CLI does not support line-continuation characters. If all characters of a command do not fit on one line of the CLI, let them wrap around to the next line on the screen. The maximum command length is 255 characters.

Example command

```
add unit 0 data="disk101-disk103" raid_level=0
```

This example command has three portions—the basic command, plus two required command options.

Basic command

```
add unit <#> data="diskrange" raid_level=r
```

The basic command includes a word or phrase used to instruct the controller. Commands usually contain a verb with a noun. Every CLI command must begin with a basic command.

Command options

`#`—the number to assign to the LUN.

`data="diskrange"`—the number of an individual drive or range of drives to incorporate into the LUN. Disks are identified by box number and bay number. For example, disk101 through disk103 identifies disks 1 through 3 in box number 1.

raid_level=`r`—the RAID fault-tolerance level to use, where `r` represents:

`0` = RAID 0 (no fault tolerance)
`1` = RAID 1 (mirroring)
`5` = RAID 5 (distributed parity)
`6` = RAID 6 (advanced data guarding (ADG))

An option is defined as words or phrases listed after the basic command that supply necessary information to support the command. If an option is required but not entered, the CLI command string is considered invalid. If an option is available but not required, a default value is used.

Overview of CLI storage configuration procedures

When using the CLI to initially configure the MSA controller and its storage, use the following sequence:

1. Create the LUNs.
   See "LUN-related commands" on page 25 for command descriptions.

   NOTE: OpenVMS environments must assign a unique ID number to each LUN. See "Assigning a name or ID to a LUN" on page 28 for instructions.

2. Enter array controller settings, including global parameters.
   See "Global commands" on page 35 for command descriptions.

   NOTE: OpenVMS environments must assign a unique ID number to each array controller. See "Setting the controller ID" on page 36 for instructions.

3. Enter connection information about the HBAs with access to the MSA.
   See "Host connection commands" on page 40 for command descriptions.

   NOTE: Each host accessing the storage must identify its operating system (profile type.) See "Changing the profile of a connection" on page 41 for instructions.

4. (Optional) Limit access to the storage.
   See "Access Control List commands" on page 43 for command descriptions.

Accessing the CLI

The CLI is accessed through a host computer connected to the serial port of an MSA controller.

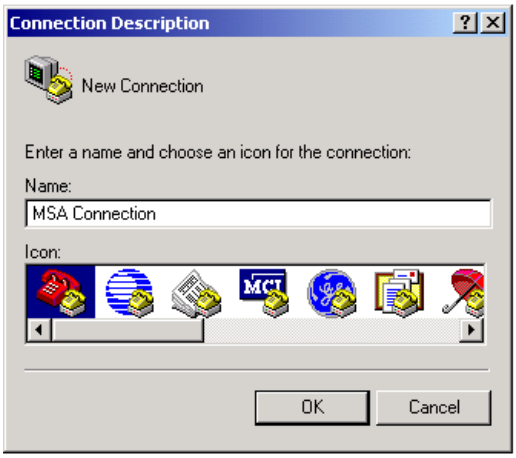
Use the following steps to establish a serial connection to the controller:

1. Make sure there is an MSA controller installed in Slot 1 of the unit
   (Slot 1 is located on the front right of the unit). In multipathing configurations, two MSA controllers must be installed.
2. Connect the MSA controller to a host using the custom serial cable included in the shipping carton of the MSA.

   An additional or replacement custom serial cable can be ordered using part number 259992-001.
3. Set up a terminal emulator.

   Depending on the host operating system, different emulators are available. For example, Linux uses Minicom and Microsoft Windows uses HyperTerminal. Setup procedures for emulators differ—the following instructions are for setting up HyperTerminal:

   a. Click **Start > All Programs > Accessories > Communications > HyperTerminal** to open HyperTerminal.

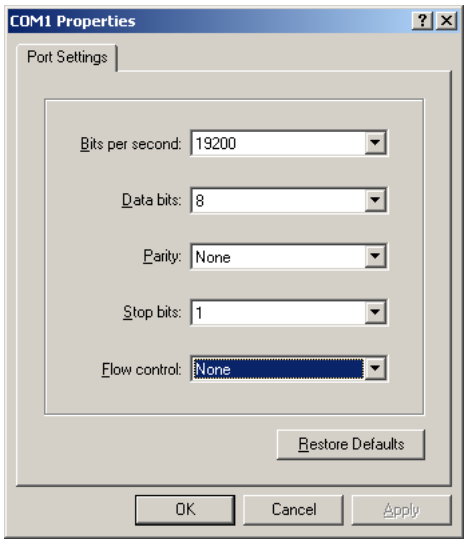


   b. In the HyperTerminal **New Connection** dialog box, type a name to associate with the connection between the controller and the host, then click **OK**.

      The **Connect To** dialog box is displayed.

   c. Expand the **Connect using** drop-down box, select a COM port, and then click **OK**.

The **COM Properties** dialog box is displayed.



d. Enter the following settings, and then click **OK**.

Bits per Second: `19200`
Data bits: `8`
Parity: `None`
Stop bits: `1`
Flow control: `None`

The CLI input screen is displayed.

e. Press **Enter** several times to display the command prompt (`CLI>`).

Commands can now be entered at this CLI prompt.

2 Using the CLI

Managing and configuring the MSA and its storage includes configuring LUNs, entering connection information, limiting access to the storage, and viewing information about the configuration.

This chapter is organized in the order in which the commands are used when configuring a new MSA installation.

Help commands

Help commands can be used to display a list of all possible commands or a detailed description of a specific command. Displayed information may include required command syntax, a brief definition, the number of characters for a command option value, or a list of allowed/disallowed characters.

The basic command is `help`. It can be modified to narrow the request.

Basic command

```
help [CommandNoun|CommandVerb] [display_all]
```

Command options

`CommandNoun|CommandVerb`—information about the specified command verb or noun will be displayed.

`display_all`—a list of all currently supported commands will be displayed.

Example `help` command and response

```
CLI> help


Possible command verbs:
                    help                add
copy                change              download
delete              migrate             expand
extend              accept              rename
set                 locate              show
disable             start               stop
clear               override

Possible command nouns:
unit                 units               connection
acl                  profile             mode
spare                firmware            unit_id
this_controller      this_controller_id  other_controller
other_controller_id  globals             prompt
disk                 bus                 box
all                  cancel              connections
version              disks               tech_support
standby              preferred_path      auto_path_change
proxystats           this_controller_hard_address
perf
cacheinfo            taskstats           debug
eventlot             aculock
```

Example `help display_all` command and response

```
CLI> help display_all

Displaying list of all currently supported CLI commands:
?                                   help
add unit                            add units
add connection                      add acl
add profile                         copy profile
change mode                         add spare
download firmware                   delete unit
delete units                        delete connection
delete acl                          delete profile
delete spare                        migrate unit
expand unit                         extend unit
accept unit                         accept units
rename connection                   set unit_id
set this_controller                 set this_controller_id
set other_controller                set other_controller_id
set globals                         set acl
set connection                      set prompt
set unit                            locate disk
locate unit                         locate bus
locate box                          locate all
locate cancel                       show connections
show unit                           show units
show unit_id                        show this_controller
show other_controller               show version
show disks                          show version -all
show acl                            show globals
show tech_support                   show profile
disable other_controller            disable standby
Show preferred_path                 disable this_controller
show auto_path_change               set preferred_path
show proxystats                     set auto_path_change
set this_controller_hard_address    show this_controller_hard_address
start perf                          show perf
clear perf                          stop perf
show taskstats                      show cacheinfo
show eventlog                       set debug
show aculock                        override aculock
```

Show commands

The `show` command is used to view MSA configuration and status information. A few samples are provided in the following paragraphs.

Basic command

```
show [acl] [aculock] [cacheinfo] [connections] [disks] [eventlog] [globals]
[other_controller] [perf] [preferred_path] [profile] [proxystats] [taskstats]
[tech_support] [this_controller] [unit #] [unit_id #] [units] [version]
```

Command options

`acl`—displays information about the access control lists, including whether they are enabled or disabled, the connection name, wwpn, and LUNs they are permitted to access.

`aculock`—displays the status of the ACU lock.

`cacheinfo`—displays MSA controller cache information.

`connections`—displays information about the HBA connections to the MSA, including the user-defined name, wwpn, and operating system profile.

`disks`—displays information about the hard drives connected to the MSA, including: disk number, enclosure box and bay number, enclosure bus number, disk size, LUN in which the disk is used, and disks assigned as spares.

`eventlog`—displays MSA event log.

`globals`—displays global information about the MSA chassis, including the user-defined system name, rebuild/expand priority settings, read/write cache settings, and temperature readings.

`other_controller`—displays information about the MSA controller, including: the user-defined controller ID, hardware build number, controller firmware version, SCSI ID, redundancy settings, cache settings, and battery information.

`perf`—displays MSA performance analysis information.

`preferred_path`—(dual-controller, active/active configurations only) displays the controller path for the LUNs.

`profile`—displays information about the profiles assigned to the HBAs connected to the MSA.

`proxystats`—(dual-controller, active/active configurations only) displays information about commands that have been passed from one controller to the other through the intercontroller link (ICL) during the past 30 minutes.

`taskstats`—displays MSA task statistics information.

`tech_support`—displays a summary of system information, executing the following commands in a batch: `show disks`, `show units`, `show this_controller`, `show other_controller`, `show cacheinfo`, `show globals`, `show version`, `show profile`, `show acl`, `show connections`, and `show eventlog`.

`this_controller`—displays information about the MSA controller, including: the user-defined controller ID, hardware build number, controller firmware version, SCSI ID, redundancy settings, cache settings, and battery information.

`unit #`—displays information about a specific LUN, including: the LUN ID, status, list of hard drives included in the LUN, list of hard drives assigned as spares to the LUN, RAID level, and LUN size.

`unit_id #`—displays the user-defined name assigned to a LUN.

`units`—displays information about all configured LUNs, including: the LUN ID, status, list of hard drives included in the LUN, list of hard drives assigned as spares to the LUN, RAID level, and LUN size.

`version`—displays version information, including: hardware build, controller firmware, and fan module firmware.

`version -all`—displays individual hard drive firmware versions, in addition to what the `version` command shows.

Example `show disks` command and response for an MSA1500

```
CLI> show disks
Disk List:    (box,bay)   (B:T:L:)    Size         Speed          Units
Enclosure 1: SATA                     SATA         PROLIANT 8LCE
Disk101       (1,01)      (0:03:01)   250.0GB      160 MB/s       none
Disk102       (1,02)      (0:03:02)   250.0GB      160 MB/s       none
Disk103       (1,03)      (0:03:03)   250.0GB      160 MB/s       none
Disk104       (1,04)      (0:03:04)   250.0GB      160 MB/s       none
Disk105       (1,05)      (0:03:05)   250.0GB      160 MB/s       none
Disk106       (1,06)      (0:03:06)   250.0GB      160 MB/s       none
Disk107       (1,07)      (0:03:07)   250.0GB      160 MB/s       none
Enclosure 2: SCSI                     PROLIANT     PROLIANT 4LEE
Disk201       (2,01)      (1:00:00)   250.0GB      160 MB/s       none
Disk202       (2,02)      (1:01:00)   250.0GB      160 MB/s       none
Disk203       (2,03)      (1:02:00)   250.0GB      160 MB/s       none
Disk204       (2,04)      (1:03:00)   250.0GB      160 MB/s       none
Disk205       (2,05)      (1:04:00)   250.0GB      160 MB/s       none
Disk206       (2,06)      (1:05:00)   250.0GB      160 MB/s       none
```

Example `show unit #` command and response

```
CLI> show unit 1
Unit 1:
In PDLA mode, Unit 1 is LUN 2; In VSA mode, Unit 1 is LUN 1
Unit Identifier:
Device Indentifier:              600805F3-00006B20-AE277D4B-B0D100F7
Preferred Path:                  Controller 1 (this controller)
Cache Status:                    Enabled
Max Boot Partition:              Disabled
Volume Status:                   VOLUME OK
Parity Init Status:              3% complete
5 Data Disk(s) used by lun 1:
  Disk107:                       Box 1, Bay 07,   (B: T: L: 2;04:07)
  Disk108:                       Box 1, Bay 08,   (B: T: L: 2;04:08)
  Disk207:                       Box 2, Bay 07,   (B: T: L: 2;04:07)
  Disk208:                       Box 2, Bay 08,   (B: T: L: 2;04:08)
Spare Disk(s) used by lun 1:
  No spare drive is designated
Logical Volume Raid Level:       DISTRIBUTED PARITY FAULT TOLERANCE
                                 (RAID5)
                                 stripe_size=16KB
Logical Volume Capacity:         173,658MB
```

Example `show globals` command and response

```
CLI> show globals

Global Parameters:
  System Name:        MSA-1
  Rebuild Priority:   medium
  Expand Priority:    medium
  Surface Delay:      3.0 seconds
Total Cache:          256MB
  50% Read Cache:     128 MB
  50% Write Cache:    128 MB
Temperature:
  EMU:                30 Celsius, 86 Fahrenheit
  PS1:                40 Celsius, 104 Fahrenheit
  PS2:                40 Celsius, 104 Fahrenheit
```

Example `show version` command and response

```
CLI> show version

MSA 1500 Firmware Revision:       6.86b1
MSA 1500 Firmware Revision:       7
Fibre Module AutoRev:             0x020000
SCSI I/O Module Bus 0 Revision:   2.02
SCSI I/O Module Bus 1 Revision:   2.02
SCSI I/O Module Bus 2 Revision:   2.02
SCSI I/O Module Bus 3 Revision:   2.02
Fan Control Module A Revision:    2.38
Fan Control Module B Revision:    2.38
```

Example `show version -all` command and response

```
CLI> show version -all

MSA 1500 Firmware Revision:       5.10b414 (SGA0434084)
MSA 1500 Firmware Revision:       7 [AutoRev: 0x010000]
Fibre Module AutoRev:             0x020000
SCSI I/O Module Bus 0 Revision:   2.02
SCSI I/O Module Bus 1 Revision:   2.02
SCSI I/O Module Bus 2 Revision:   2.02
SCSI I/O Module Bus 3 Revision:   2.02
Fan Control Module A Revision:    2.38
Fan Control Module B Revision:    2.38
 Box 1a PROLIANT 4L7E*DB  Rev:    2.30 (Z8LCLMPY47)
 Box 1b PROLIANT 4L7E*DT  Rev:    CP20 (Z8LCLMPY47)
 Box 3b PROLIANT 4L7E*DT  Rev:    CP20 (TERJE2BOX2)
 Box 3a PROLIANT 4L7E*DB  Rev:    2.30 (TERJE2BOX2)

Disk Drives:
B  T  L BOX BAY    GB Model       Rev.          Serial No.
0  0  0  1   1     146.8 COMPAQ   BD146863B3    HPB8 B8LX86HM
0  0  0  1   1     146.8 COMPAQ   BD146863B3    HPB8 B8LX8EFM
0  8  0  1   7     146.8 COMPAQ   BD14686225    HPB6 P01P4707AGE0431
```

Example `show this_controller` command and response

```
CLI> show this_controller

Controller 1 (right controller):
  MSA1500 c Hewlett-Packard xxx Version 6.86 Build 122 Hardware 7
  Component Enclosure.
  Controller Identifier:
  NODE_ID=yyyyyyyy-yyyyyyyy
  SCSI_VERSION=SCSI-3
  Supported Redundancy Mode: Active/Standby Asm-Active/Active
  Current Redundancy Mode: Asym-Active/Active
  Device Port SCSI address 6
  Terminal speed for the CLI is set to 19200.
Host Port_1:
  REPORTED PORT_ID YYYYYYYYY-YYYYYYYYY
  PORT_1_TOPOLOGY=F_Port
Cache:
  128 megabytes read cache 128 megabytes write cache Version 2
  Cache is GOOD, and Cache is enabled
  No unflushed data in cache
Battery:
  Module #1 is fully charged and turned off.
Controller Up Time:
  5 Days 06 Hours 01 Minutes 51 Seconds
Health:
  Surface Scan:   Complete.
  Rebuild Status: Complete.
  Expansion:      Complete.
```

Example `show connections` command and response

NOTE: Use the `show connections` command to verify that all connections to the MSA are recognized and defined, including the profile type.

```
CLI> show connections


Connection Name: <unknown>
  Host WWNN = 11111111-11111111
  Host WWPN = 22222222-22222222
  Profile Name = Windows
  Unit Offset = 0
  Controller 1 Port 1 Status = Online
  Controller 2 Port 1 Status = Offline

Connection Name: <unknown>
  Host WWNN = 33333333-33333333
  Host WWPN = 44444444-44444444
  Profile Name = Windows
  Unit Offset = 0
  Controller 1 Port 1 Status = Online
  Controller 2 Port 1 Status = Offline
```

Example `show preferred_path` command and response

```
CLI> show preferred_path

Controller 1 (this_controller) is the current optimal path for these units;
0,1,4,5

Controller 2 (other_controller) is the current optimal path for these units;
2,3,6,7

Implicit (automatic load-based) LUN ownership changes are currently
DISABLED.

Controller 1 (this_controller) is the explicit power-up path for these
units; 0,1,4,5

Controller 2 (other_controller) is the explicit power-up path for these
units; 2,3,6,7
```

LUN-related commands

The following commands are used to physically locate specific hard drives in a LUN as well as create, delete, and modify storage LUNs.

- Flashing LEDs/locating hard drives
- Creating LUNs
- Assigning a name or ID to a LUN
- Adding a spare to an existing LUN
- Deleting LUNs
- Deleting spares
- Recognizing a failed unit
- Expanding an array
- Extending a LUN
- Migrating a LUN to a different RAID level
- Changing the cache setting for a LUN
- Setting the preferred path for a LUN
- Enabling/disabling automatic path switching

Flashing LEDs/locating hard drives

A variety of commands are available for physically locating specific hard drives. You can precisely locate all drives attached to the MSA, all drives in an exact storage enclosure, all drives on a specific SCSI bus, all drives in a specific LUN, or a particular hard drive.

When these commands are executed, the LEDs of the requested drives will blink. These LEDs are visible from the front of the MSA and its attached storage enclosures.

NOTE: If a time limit is not included with the `locate` command, the LEDs will blink for 30 seconds.

The basic command verb is `locate`, but a variety of command nouns and command options are available to customize the request.

Basic command

```
locate [all] [box x] [bus x] [cancel] [disk diskxxx] [disk diskxxx-diskyyy]
[time=t] [unit x]
```

Command options

`all`—all drives connected to the MSA storage subsystem will be located.

`box x`—the number of the storage enclosure whose disks you want to locate. This option applies only to the MSA1000.

1=the MSA1000 drive shelf
2=the storage enclosure attached to SCSI port A
3=the storage enclosure attached to SCSI port B

`bus x`—the number of the bus whose disks you want to locate. This option applies only to the MSA1000.

`cancel`—immediately turns off all location LEDs.

`disk diskxxx`—a specific drive to locate. Disks are identified by box number and bay number.

`disk diskxxx-diskyyy`—a range of drives to locate. Disks are identified by box number and bay number.

`time=t`—(optional) the length of time to flash the LEDs, where *t* represents the number of seconds. This option must be preceded by another option such as `all`.

`unit x`—the number of the LUN whose disks you want to locate.

Creating LUNs

A LUN is a logical storage unit comprised of one or more hard drives.

When a LUN is initially created, the LUN Unit ID is automatically set to the same number assigned to the LUN. If desired, this LUN ID name can be changed. See "Assigning a name or ID to a LUN" on page 28 for procedural information.

The basic command is `add unit`. A variety of options are available to modify the command.

Basic command

```
add unit[s] <#> <data="diskrange"> <raid_level=r> [spare="diskrange"]
[stripe_size=s] [size=s<GB|MB>] [max_boot=<enable/disable>]
[cache=<enable/disable>]
```

Command options

#—the number to assign to the LUN.

`data="diskrange"`—the number of an individual drive or range of drives to incorporate into the LUN. Disks are identified by box number and bay number. For example, disk110 identifies disk 10 in box number 1, and disk101-disk105 identifies disks 1 through 5 in box number 1.

NOTE: Quotation marks must be entered both before and after the data disk drives that are to be included in the LUN.

`raid_level=r`—the RAID fault-tolerance level to use, where r represents:

`0` = RAID 0 (no fault tolerance)
`1` = RAID 1 (mirroring)
`5` = RAID 5 (distributed parity)
`6` = Advanced Data Guarding (ADG)

NOTE: If more than one pair of drives are included in a RAID 1 array, the data is striped across the first half of the drives in the array, and then each drive is mirrored to a drive in the remaining half of the drives for fault tolerance. This method is referred to as RAID 1+0.

`stripe_size=s`—(optional) the stripe size to assign (in KB), where *s* represents 8, 16, 32, 64, 128, or 256 Kilobytes.

`RAID 0` uses stripe sizes 8, 16, 32, 64, 128, and 256 (Default: 128 KB)
`RAID 1` uses stripe sizes 8, 16, 32, 64, 128, and 256 (Default: 128 KB)
`RAID 5` uses stripe sizes 8, 16, 32, and 64 (Default: 16 KB)
`RAID 6` uses stripe sizes 8, 16, 32, and 64 (Default: 16 KB)

`size=syy`—(optional) how much of the available space on the indicated drives is to be used for the LUN, where `s` represents the LUN size, and yy indicates MB or GB. (When GB is entered, all return displays will be converted to MB.) If no size is specified, the maximum available space of the included disks, up to 2 TB, will be assigned to the unit.

`spare="diskrange"`—(optional) disk(s) to assign as a spare to the unit, where *diskrange* represents the disk number. More than one disk can be assigned as a spare to a LUN. And, the same drive can be assigned as an available spare to several different LUNs.

`maxboot=enable|disable`—(optional) changes the size of the boot partition.

`Enable` = (default) 8-GB boot partition
`Disable` = 4-GB boot partition

`cache=enable|disable`—(optional) determines whether to use the array controller cache for the LUN.

`Enable` = (default) use the array controller cache
`Disable` = do not use the array controller cache

IMPORTANT: Maintain a record of the units as they are created. These unit numbers are used in other CLI commands. In addition to recording the unit number, the drives included, RAID type and size, **record the order in which they are created**.

Example command and response to create a single LUN from a group of drives

```
CLI> add unit 0 data="disk101-disk107 disk110" raid_level=6 stripe_size=64


First volume to be configured on these drives.
Logical Unit size = 69460 MB
RAID overhead = 0 MB
Total space occupied by new unit = 69460 MB
Free space left on this volume = 0 MB
Unit 0 is created successfully.
```

Example command and response to create multiple LUNs on a group of drives

```
CLI> add unit 1 data="disk111-disk114" raid_level=5 stripe_size=32
size=1000mb


First volume to be configured on these drives.
The logical unit size has been adjusted by 4MB for optimal performance.
Logical Unit size = 996 MB
RAID overhead = 498 MB
Total space occupied by new unit = 1494 MB
Free space left on this volume = 24533 MB
Unit 1 is created successfully.


CLI> add unit 2 data="disk111-disk114" raid_level=5 stripe_size=32
size=2000mb

Logical Unit size = 2000 MB
RAID overhead = 1000 MB
Total space occupied by new unit = 3000 MB
Free space left on this volume = 21533 MB
Unit 2 is created successfully.

CLI> add unit 3 data="disk111-disk114" raid_level=5 stripe_size=16
size=4000mb

Logical Unit size = 4000 MB
RAID overhead = 2000 MB
Total space occupied by new unit = 6000 MB
Free space left on this volume = 15533 MB
Unit 3 is created successfully.
```

Example command and response to create a LUN with an assigned spare

```
CLI> add unit 4 data="disk211-disk212" raid_level=1 spare="disk213"


First volume to be configured on these drives.
Logical Unit size = 69460 MB
RAID overhead = 69460 MB
Total space occupied by new unit = 138920 MB
Free space left on this volume = 0 MB
Unit 4 is created successfully.
```

Assigning a name or ID to a LUN

If desired (or required by your operating system), each LUN can be assigned a unique name or ID, in addition to its number. These user-defined names make it easier to identify specific LUNs in other configuration procedures.

NOTE: OpenVMS systems require each LUN to have a unique ID. No two devices in the entire SAN for the OpenVMS systems may share ID numbers. LUNs in different storage systems must have different IDs. LUN IDs in OpenVMS systems must be numeric.

Basic command

```
set unit_id <#> <identifier>
```

Command options

#—the number of the LUN that is being assigned a name.

`identifier`—name or number to assign to the specified LUN. The ID can be up to 230 alphanumeric characters or a decimal number in the range of 0 though 65535. If spaces are included in the name, quotation marks are required.

Example command and response

```
CLI> set unit_id 1 1523

Device identifier "1523" created.
```

Adding a spare to an existing LUN

Basic command

```
add spare unit=<#> <diskrange>
```

Command options

#—the unit to assign the spare to. This is the same number that was given to the unit when it was created with the `add unit` command.

`diskrange`—indicates the drives to assign as the spare to the LUN. Disks are identified by box number and bay number.

Example command and response

```
CLI> add spare unit=2 disk112
  Available write cache lines: 0x37FD (592458:592420)
  Available read cache lines: 0x37EE
  Using write cache lines 0x0 to 0x1BFD
  Hiding write cache lines 0x1BFE to 0x37FC
  Using read cache lines 0x37FD to 0x53F3
  Hiding read cache lines 0x53F4 to 0x6FEA
  Spare drive(s) has been added. Use 'show unit 0' to confirm.
```

Deleting LUNs

NOTE: A LUN cannot be deleted if there are other LUNs after it. For example, if LUNs 0-3 are configured, LUN 1 cannot be deleted without also deleting LUNs 2 and 3.

After a LUN is deleted, its unit number goes unused until manually assigned to a new LUN. Unit numbers are not automatically reassigned when a LUN is deleted.

Basic command

```
delete unit <#> [-y]
```

Command options

#—the unit or range of units to delete. This is the same number that was given to the unit when it was created with the `add unit` command.

`-y`—(optional) no user confirmation is required before the deletion is made.

Example command and response

```
CLI> delete unit 4


Data will be lost after the unit is deleted.
Do you still want to DELETE unit 4 (Y/N)? Y
Please wait while unit 4 is being deleted…
Unit 4 is deleted successfully.
```

Deleting spares

Basic command

```
delete spare unit <#> <diskrange>
```

Command options

`unit #`—the unit that will no longer have access to the spare. This is the same number that was given to the unit when it was created with the `add unit` command.

`diskrange`—the spare drive or range of spares to remove from use. Disks are identified by box number and bay number.

Example command

```
CLI> delete spare unit=2 disk109
```

Recognizing a failed unit

If all drives of a previously failed unit are in working order, use this command to change the state of the unit back to `VOLUME_OK`.

To accept media exchange on a unit marked as failed:

Basic command

```
accept unit <#>
```

Command options

#—the unit that you want to activate, where # represents the unit number.

Expanding an array

△ **CAUTION:** In an HP-UX environment, expanding a LUN, as described in this section, is not supported. See the HP-UX documentation for instructions on expanding LUNs on an HP-UX system.

To add additional physical drives to an array:

NOTE: Because this command affects the entire array, all LUNs made from the array are also affected.

Basic command

```
expand unit <#> <diskrange>
```

Command options

#—one of the LUN within the target array.

NOTE: Any LUN in the array can be entered to identify the array; the space is added to the array and not the LUN.

`diskrange`—the physical disks to add to the array. Disks are identified by box number and bay number.

Example command and response

```
CLI> expand unit 4 disk204-disk207


The actual new array capacity will be 3000MB.
The array with Unit 4 is being expanded.
Use "show unit 4" to monitor progress.
```

Extending a LUN

△ **CAUTION:** In an HP-UX environment, extending a LUN as described in this section is not supported. See the HP-UX documentation for instructions on extending LUNs on an HP-UX system.

To incorporate unused, available space in an array to a specific LUN:

Basic command

```
extend unit <#> [new_size=syy|add_size=syy]
```

Command options

#—the LUN to which the space will be added.

`add_size=syy`—how much of the available space in the array to add to the LUN, where s represents the size, and yy indicates GB, MB, or KB. If no size is specified, the maximum available space of the included disks will be assigned to the array.

`new_size=syy`—can be used instead of `add_size` to enter the total new size of the LUN, where `s` represents the size, and yy indicates GB, MB, or KB.

Example command and response

```
CLI> extend unit 2 add_size=1000mb


The actual new volume size will be 1992MB.
Unit 2 is being extended.
Use "show unit 2" to monitor progress.
```

Migrating a LUN to a different RAID level

While obeying the constraints of included number of drives and applicable stripe sizes, you can migrate an array from one RAID level to another. If you attempt to move to an unsupported RAID configuration for an array, an error message is displayed.

NOTE: Before changing the RAID level or stripe size of a LUN, verify there is available, unused space on the array. Migrating from one RAID level to another may require additional space for parity and organizational purposes.

Basic command

```
migrate unit <#> [raid_level=r] [stripe_size=s]
```

Command options

#—the number of the LUN to modify.

`raid_level=r`—the RAID level to migrate to. RAID levels include:

`0` = RAID 0 (no fault tolerance)
`1` = RAID 1 (mirroring)
`5` = RAID 5 (distributed parity)
`6` = RAID 6 (advanced data guarding (ADG))

`stripe_size=s`—(optional) the stripe size to use. Options include:

`RAID 0` can use stripe sizes 8, 16, 32, 64, 128, and 256 (Default: 128 KB)
`RAID 1` can use stripe sizes 8, 16, 32, 64, 128, and 256 (Default: 128 KB)
`RAID 5` can use stripe sizes 8, 16, 32, and 64 (Default: 16KB)
`RAID 6` can use stripe sizes 8, 16, 32, and 64 (Default: 16 KB)

Example command and response

```
CLI> migrate unit 0 raid_level=5 stripe_size=32
```

```
The RAID level of Unit 0 will now be 5.
Unit 0 is being migrated.
Use "show unit 0" to monitor progress.
```

Changing the cache setting for a LUN

To enable or disable the array accelerator cache for a specific LUN:

Basic command

```
set unit <#> <cache=[enable|disable]>
```

Command options

#—the number of the LUN to modify. In this example, LUN 0 will be modified.

`cache=enable|disable`—either enables or disables the use of the array accelerator cache for the specified LUN.

Example command and response

```
CLI> set unit 0 cache=enable

Cache for unit 0 has been enabled.
```

Setting the preferred path for a LUN

NOTE: This command is available only with active/active firmware (v6.x or later).

Configures an explicit preferred controller path for specified logical units. Host requests that are directed to the non-preferred controller will be delayed slightly, due to data and command transfer through the inter-controller link.

If implicit path changes are enabled (see "Enabling/disabling automatic path switching" on page 34), controller firmware may automatically switch controller paths for unit(s) based on host request load.

Basic command

```
set preferred_path [controller] [reset] <unit_list>
```

Command options

`controller`—indicate the controller by entering `this_controller`, `other_controller`, `controller=1`, or `controller=2`.

`reset`—restore the last explicitly configured path for the specified LUNs.

`unit_list`—indicate the LUNs to set by entering `all`, the unit number of a specific LUN, or a range of unit numbers.

Example command and response

```
CLI> set preferred_path this_controller 0-9

SetPreferredPath start... change_map=0x2FF, lun_to_ctlr_map=0x3FF
POST_Do_Msg: 67 CACHE HARDWARE TEMPORARILY DISABLED
Quiescing host I/O for LUN map 0xFFEFF...
Host/proxy I/O quiesced...
Quiescing host I/O for LUN map 0xFFF00100...
Host/proxy I/O quiesced...
Quiescing cache flush task...
Quiescing background I/O...
Background quiesced...
Flipping ownership(s)... change_map=0x2FF, lun_to_ctlr_map=0x3FF
POST_Do_Msg: 41 REDUNDANCY ACTIVE ACTIVE CONTROLLER
Preferred path for unit 0 is now controller 1 (this_controller)
Preferred path for unit 1 is now controller 1 (this_controller)
Preferred path for unit 2 is now controller 1 (this_controller)
Preferred path for unit 3 is now controller 1 (this_controller)
Preferred path for unit 4 is now controller 1 (this_controller)
Preferred path for unit 5 is now controller 1 (this_controller)
Preferred path for unit 6 is now controller 1 (this_controller)
Preferred path for unit 7 is now controller 1 (this_controller)
Preferred path for unit 9 is now controller 1 (this_controller)
Resuming I/O...
I/O Resumed...
POST_Do_Msg: 65 CACHE HARDWARE ENABLED
```

Enabling/disabling automatic path switching

NOTE: This command is available only with active/active firmware (v6.x or later).

If implicit path changes are enabled, controller firmware will automatically determine and, as needed, switch controller paths for units based on host request load, according to the specified preferences.

Basic command

```
set auto_path_change <enable|disable> [prp=#] [mnr=#] [twp=#] [twl=#]
```

Command options

`enable|disable`—enables or disables implicit path switching.

`prp`—specify a proxied percentage (# = 51-100). This is the minimum percentage of proxied requests required to trigger implicit path change.

`mnr`—specify the minimum number of requests that must be issued to the LUN before path changes are enabled (the number is rounded up to the next highest power of two).

`twp`—specify the minimum number of minutes (0-255) to wait at power-up before changing LUN ownership.

`twl`—specify the minimum number of minutes (0-255) to wait before changing LUN ownership again for same unit.

Example command and response

```
CLI> set auto_path_change disable

Implicit (automatic load-based) LUN ownership changes are currently
DISABLED.


CLI> set auto_path_change enable

Implicit (automatic load-based) LUN ownership changes are currently ENABLED.

Implicit LUN ownership change preferences are as follows:

Minimum % of proxied requests to trigger implicit path change..:  75%

Minimum number of requests to LUN before path change enabled...:  65536

Minimum time to wait before changing ownership after power-up..:  30 minutes

Minimum time between ownership changes for LUN.................:  30 minutes


CLI> set auto_path_change twp=15

Implicit (automatic load-based) LUN ownership changes are currently ENABLED.

Implicit LUN ownership change preferences are as follows:

Minimum % of proxied requests to trigger implicit path change..:  75%

Minimum number of requests to LUN before path change enabled...:  65536

Minimum time to wait before changing ownership after power-up..:  15 minutes

Minimum time between ownership changes for LUN.................:  30 minutes
```

Global commands

The CLI provides the ability to configure the MSA controller, enter global settings, and perform system-wide commands, including:

- Setting global variables
- Setting the controller ID
- Setting hard addressing
- Changing the CLI prompt
- Updating MSA firmware
- Disabling a controller for removal
- Overriding the ACU lock
- Enabling or disabling debug console logging

Setting global variables

A single command with a variety of parameters is used to set the expand priority, the read/write ratio, and the system name.

Basic command

```
set globals [expand_priority=p] [rebuild_priority=r] [system_name="name"]
[read_cache=r] [write_cache=w]
```

Command options

`expand_priority=p`—the expand priority, where `p` represents the priority setting of low, medium, or high. Used when expanding an array to set the priority of array expansions in relation to input/output operations. See below for detailed information about these settings.

`rebuild_priority=p`—the rebuild priority, where `p` represents the priority setting of low, medium, or high. Used when rebuilding an array to set the priority of an array rebuild in relation to input/output operations. See below for detailed information about these settings.

- `low`—expansion or rebuild takes place only when the array controller is not busy handling normal I/O requests. This setting has minimal effect on normal I/O operations. However, there is an increased risk that data will be lost if another physical drive fails while the rebuild is in progress.
- `medium`—expansion or rebuild occurs for half of the time, and normal I/O requests are handled during the rest of the time.
- `High`—rebuild or expansion occurs at the expense of normal I/O operations. Although system performance is affected, this setting provides better data protection because the array is vulnerable to additional drive failures for a shorter time.

`system_name="name"`—the system name, where *"name"* represents any user defined phrase, up to 20 alphanumeric characters. Quotation marks are required around the name.

`read_cache=r`—the read cache, where r specifies the read cache as a percentage of the total cache.

`write_cache=w`—the write cache, where r specifies the write cache as a percentage of the total cache.

NOTE: The sum of `read_cache and write_cache` must equal 100.

Example command and response

```
CLI> set globals expand_priority=medium
rebuild_priority=medium system_name="MSA-1"
read_cache=50 write_cache=50


Global
Parameters:
System Name:        MSA-1
Rebuild Priority:   medium
Expand Priority:    medium
Total Cache:        256MB
50% Read Cache:     256MB
50% Write Cache:    256MB
```

Setting the controller ID

This command can be used to assign a unique name to each controller installed in the MSA.

NOTE: In OpenVMS environments, a unique identifier must be assigned to each MSA controller. The value must be numeric.

Basic command

```
set this_controller_id <identifier>
```

-or-

```
set other_controller_id <identifier>
```

Command options

`identifier`—the user-defined ID for the controller. The ID can be up to 230 alphanumeric characters or a decimal number in the range of 0 though 65535. If spaces are included in the name, quotation marks are required.

Example command and response

```
CLI> set this_controller_id MSA-1

Controller identifier MSA-1 created.
```

Setting hard addressing

To enable or disable Fibre Channel Arbitrated Loop hard addressing on an MSA controller:

Basic command

```
set this_controller_hard_address <enable|disable> <loop_id>
```

Command options

`enable|disable`—enables or disables hard addressing on the controller.

`loop_id`—a decimal number between 0 and 125 to assign the loop identifier.

Example command and response

```
CLI> set this_controller_hard_address enable 1
Hard Addressing is enabled.
Loop ID = 1, ALPA = 0xE8
```

Changing the CLI prompt

To temporarily change the CLI prompt from the default of "CLI>":

Basic command

```
set prompt <new_prompt>
```

Command options

new_prompt—the user-defined name for the prompt (up to 24 alphanumeric characters).

Example command and response

```
CLI> set prompt MSA-1

MSA-1>
```

Updating MSA firmware

Periodically, HP releases updated versions of MSA controller firmware. Updates may include: additional features and functions, performance enhancements, and fixes to known issues. In addition to MSA1000 and MSA1500 components, this command can also be used to update components on attached MSA20 storage enclosures.

NOTE: Because this updating method uses an out-of-band serial connection to the MSA, it is substantially slower than in-band Fibre Channel updating methods.

If possible, HP recommends using the Fibre-Channel updating method for your operating system.

Operating system	In-band utility (Fibre Channel)	Out-of-band utility (Serial)
Windows ProLiant host	MSA Flash Utility	MSA CLI
Linux ProLiant host	MSA Flash Utility	MSA CLI
HP-UX host	SCSI Command Utility (SCU)	MSA CLI
	Array Configuration Utility CLI (ACU-CLI)	
OpenVMS host	MSA_Util	MSA CLI not tested
Tru64 UNIX host	SCSI CAM Utility (SCU)	MSA CLI not supported

CAUTION: Before using this method to update the firmware, make note of the following requirements:

- A full backup of *all* data on the MSA array being updated is required.
- A copy of the array configuration, including all the information used to create the units, is needed for the update.
- If it has been more than six months since you restarted your MSA storage system, HP recommends that you power-cycle the MSA (power off, and then power on) before updating the firmware to ensure that you are working with a fresh system.
- The MSA must be power-cycled after updating the firmware to activate the new firmware.
- When performing an online update for active/passive configurations, ensure that the front-right controller is the active one. The firmware update may not work properly if the front-left controller is active during the update, especially if the update is performed via serial cable.
- Existing firmware on MSA1000 controllers must be v4.32 or higher.
- Existing firmware on MSA1500 controllers must be v4.94 or higher.
- Host computer must support the 1k Xmodem (Ymodem) protocol.
- Because firmware updates require a restart of the MSA and its attached storage enclosures, perform MSA controller firmware updates only during a scheduled maintenance window.

- For newly installed MSA, do not perform a firmware update until the controller batteries are fully charged.
- For existing MSA, do not perform a firmware update until you have confirmed that the "host mode" or "profile" for each connection is correctly set. The host mode identifies the operating system of each HBA connection to the storage. Do not use the "default" setting. If the host mode is not properly set, hosts may lose access to the storage or experience other difficulties after the update. Use the `show connections` and `set connection` commands to view and change the profile.
- After performing an update, confirm that the "host mode" or "profile" for each connection is still set correctly. Use the `show connections` and `set connection` commands to view and change the profile.
- HP-UX environments must have an MSA Fibre link to a host.

IMPORTANT: If you encounter any problems during the firmware update process, stop and contact HP technical support. See "HP technical support" on page 9 for support contact information.

1. Schedule a maintenance window for the update.
2. Obtain the firmware files (.bin option) and save it to a temporary location on the host. Firmware updates are provided on the **Software, firmware & drivers** page of MSA websites:
   MSA1000: http://www.hp.com/go/msa1000
   MSA1500: http://www.hp.com/go/msa1500cs
3. Perform a *full* backup of the data on the array.
4. If the MSA has not been restarted in the last six months, power-cycle the array. This ensures that you are working with a fresh system before beginning the firmware update.
5. Open an emulator session to access the CLI.
6. Enter the following command:
   ```
   download firmware offline
   ```
   This command performs an update and does not automatically restart the MSA. Through this updating method, the MSA controller and fan modules can be updated. This method is supported for use in both single-controller and dual-controller configurations.
7. On the emulator window menu bar, select **Transfer > Send File**.
   a. Click **Browse**, navigate to the directory in which you placed the firmware files, and then select the firmware file.
   b. Expand the Protocol drop-down box, and then select **1K Xmodem**.
   c. Click **Send**. A 1K Xmodem status window is displayed.

   NOTE: During the updating process, the following messages are displayed on the MSA controller LCD panel:
   ```
   FIRMWARE FLASH STARTED
   FIRMWARE FLASH DONE
   ```
8. Wait for a completion message to be displayed.
9. Restart the MSA by pressing the power/standby button on the front of the MSA. Newly downloaded firmware cannot be accessed until the MSA is restarted.

NOTE: In dual-controller configurations, firmware on the two controllers is compared each time the MSA chassis is restarted. If the versions are mismatched, the system prompts to clone the firmware on the controller with the latest version over to the controller with the earlier version firmware. The following message is displayed on the LCD panel of the controller with the earlier firmware:

```
07 CLONE FIRMWARE ? '<' = NO, '>' = YES
```

Press the **>** button on the LCD panel to clone the firmware. During the cloning process, informational messages are displayed on the controller LCD panels. When the cloning process is complete, the just-updated controller automatically restarts.

10. Before resuming I/O access to the MSA, verify that the MSA and its storage arrays are online.

Disabling a controller for removal

In a dual-controller system, this command disables one of the controllers to prepare it for removal. When a controller is disabled, all resources being processed by that controller are automatically failed over to the remaining controller. After a controller has been successfully disabled, the LCD panel displays a message stating that it is safe for that controller to be removed.

NOTE: This command is available only in dual-controller configurations.

Basic command

```
disable other_controller
```

-or-

```
disable this_controller
```

-or-

```
disable standby
```

Overriding the ACU lock

To prevent simultaneous use of more than one management interface, the system automatically places a lock on the MSA CLI whenever the Array Configuration Utility (ACU) or the Array Configuration Utility - Command Line Interface (ACU-CLI) is managing the MSA.

This command can disable the lock and allow the CLI to be used at the same time that the ACU or ACU-CLI is active.

△ **CAUTION:** Turn off the ACU lock only under strict supervision. Multiple configuration sessions may lead to contradictory commands, unintended configuration changes, inappropriate actions, or a disrupted status.

Basic command

```
override aculock <on|off>
```

Enabling or disabling debug console logging

To allow additional system level messages to appear on the CLI console during failure events (such as hard drive, power supply, or fan module failures):

Basic command

```
set debug <on|off>
```

Host connection commands

Each time the MSA is powered on, all active HBA connections to the MSA are automatically detected and identified by their WWPN.

Before a host accesses the storage of the MSA, you must identify the operating system (host mode) of each connection. HP also recommends assigning a user-defined name to each connection, to make the identification and setup of each connection easier.

After connections are established, be sure to restrict access to the LUNs to specific hosts. The CLI uses an Access Control List (ACL) to enter the list of LUNs a host can access. See "Access Control List commands" on page 43 for more information about SSP and ACLs.

Each of the following commands is used to enter and manage the connections:

- Adding a connection
- Changing the profile of a connection
- Changing the user-defined name of a connection
- Deleting the user-defined name of a connection
- Creating a new profile
- Moving to a different profile
- Changing the profile mode

NOTE: Use the `show connections` command to display information about each HBA connected to the MSA, including connection name, WWPN, and profile. See "Show commands" on page 19 for an example display.

Adding a connection

If the host is not yet connected to the MSA, but WWPN of the HBA is known, you can use this command to manually add, name, and identify the connection.

Basic command

```
add connection <ConnectionName> <wwpn=xxxxxxxx-xxxxxxxx> [profile=profile_name]
[offset=offset_value]
```

Command options

`ConnectionName`—the user-defined name to give the connection, up to 16 alphanumeric characters.

`wwpn=xxxxxxxx-xxxxxxxx`—the WWPN of the active HBA inside the host attached to the MSA. The show connections command can be used to obtain the WWPN.

`profile=profile_name`—the platform of the host. Options include:

- `Default`—is not operating-system specific and should not be used; use a valid pre-defined name.
- `Windows`
- `OpenVMS`
- `Tru64`
- `Linux`
- `Solaris`
- `Netware`
- `HP`

`offset=offset_value`—(default: 0) the unit offset for assigning logical volumes.

Example command and response

```
CLI> add connection MSA-1 wwpn=12345678-12345678 profile=windows

Connection MSA-1 has been added successfully.
Profile Windows is set for the new connection.
```

Changing the profile of a connection

To change the operating-system profile associated with an existing connection (or to change the HBA of a connection):

NOTE: The connection profile is sometimes referred to as the Host Mode.

Basic command

```
set connection <ConnectionName> [wwpn=xxxxxxxx-xxxxxxxx] [profile=profile_name]
[offset=offset_value]
```

Command options

`ConnectionName`—the name of the connection to modify.

`wwpn=xxxxxxxx-xxxxxxxx`—the WWPN of the connection to modify, using the WWPN of the HBA. The "show connections" can be used to obtain the WWPN.

`profile=profile_name`—the platform of the host. Options include:

- `Default`—is not operating-system specific and should not be used; use a valid pre-defined name.
- `Windows`
- `OpenVMS`
- `Tru64`
- `Linux`
- `Solaris`
- `Netware`
- `HP`

`offset=offset_value`—(default: 0) the unit offset for assigning logical volumes.

Example command and response

```
CLI> set connection MSA-1 profile=windows


The profile of connection MSA-1 is set to Windows successfully.
```

Changing the user-defined name of a connection

To change the user-defined name associated with a connection:

Basic command

```
rename connection <old_connection_name> <new_connection_name>
```

Command options

`old_connection_name`—the name of the connection that is to be changed.

`new_connection_name`—the new name to assign to the connection, up to 16 alphanumeric characters.

Example command and response

```
CLI> rename connection abc xyz


Connection(s) has been renamed successfully.
```

Deleting the user-defined name of a connection

To remove the user-defined name associated with a connection to a host:

Basic command

```
delete connection <connection_name>
```

Command options

`connection_name`—the nickname that was assigned to the HBA within the host.

Example command and response

```
CLI> delete connection MSA-1


Connection(s) has been deleted successfully.
```

Creating a new profile

This command creates a new host profile.

Basic command

```
add profile <profileName>
```

Command options

`profileName`—the name of a new profile.

Moving to a different profile

This command copies the profile settings of a valid, pre-defined profile into a different profile.

NOTE: If you are migrating from active/passive to active/active firmware and did not set your profile (it is still set to `default`), you can use this command to copy the settings for a valid profile (such as `Windows`) into the default profile.

However, HP recommends using the `set connection` command to properly set the host connection. For details, see "Changing the profile of a connection" on page 41.

Basic command

```
copy profile <sourceName> <destinationName>
```

Command options

`sourceName`—the name of an existing, valid profile whose settings you want to copy.

`destinationName`—the name of the profile (such as `Default`) that is importing the settings.

Pre-defined profiles include:

`Default`—is not operating-system specific and should not be used; use a valid pre-defined name.

`Windows`

`OpenVMS`

`Tru64`

`Linux`

`Solaris`

`Netware`

`HP`

Example command

```
CLI> copy profile Windows default
```

Changing the profile mode

This command changes a host profile mode based on the number of the mode that is displayed in the `show profile` command.

△ **CAUTION:** Use only under the advise of an authorized customer support technician.

Basic command

```
change mode <profileName> <hostModeNumber>
```

Command options

`profileName`—the name of an existing profile.

`hostModeNumber`—the number that is displayed next to the mode in the `show profile` command.

Access Control List commands

When multiple hosts access the storage of the MSA, it becomes necessary to restrict access to LUNs to specific hosts. The CLI uses an Access Control List (ACL) to enter the list of LUNs a host can access.

NOTE: Even in the same OS environment, if multiple servers and HBAs have access to the same LUN, all of the HBAs must be of the same model.

Each of the following commands is used to set up and manage the ACL:

- Viewing the ACL
- Adding to the ACL
- Deleting information from the ACL

Each of these tasks is defined in the following sections.

NOTE: Use the `show connections` command to display the connection name, defined profile type, and the WWPN of each HBA connected to the MSA.

The connection name or the WWPN is required when entering ACL information. See "Show commands" on page 19 for a sample display.

Viewing the ACL

To display the current ACL:

Basic command

```
show acl
```

Example command and response

```
CLI> show acl


ACL is enabled:
Connection      WWPN                  Units
ABC             11111111-22222222     0,1,2
XYZ             33333333-44444444     3,4
Inaccessible Units: 5,6
```

Adding to the ACL

Two methods of entering LUN assignments are available:

- Adding to the ACL using the connection name
- Adding to the ACL using the WWPN

Basic command

```
add acl <connection=name> <unit=#>
```

-or-

```
add acl <connection=wwpn=xxxxxxxx-xxxxxxxx> <unit=#>
```

Command options

`connection=name`—the name of the connection to grant access.

`connection=wwpn=xxxxxxxx-xxxxxxxx`—the wwpn of the connection to grant access.

`unit=#`—the LUN or range of LUNs to assign to the indicated host. Access to all units can be granted by entering `all`.

NOTE: The ACL is automatically activated when the first entry is made.

Example command and response

```
CLI> add acl connection=wwpn=12345678-12345678 unit=2


Allowing 12345678-12345678 access to unit 2.
```

Deleting information from the ACL

After the ACL has been set up, access permissions for certain LUNs may need to be removed.

Two methods of removing access to previously assigned LUNs are available:

- Deleting information from the ACL using the connection name
- Deleting information from the ACL using the WWPN

Basic command

```
delete acl <connection=name> <unit=#>
```

-or-

```
delete acl <connection=wwpn=xxxxxxxx-xxxxxxxx> <unit=#>
```

Command options

`connection=name`—the name of the connection for which you want to remove access.

`connection=wwpn=xxxxxxxx-xxxxxxxx`—the WWPN of the connection for which you want to remove access.

`unit=#`—the LUNs, or range of LUNs, to remove from the ACL of the indicated host. Access to all units can be removed by entering `all`.

NOTE: If all entries in the ACL are being deleted, immediately after the last entry is deleted, all connected hosts have unlimited access to the storage.

Example command and response

```
CLI> delete acl wwpn=12345678-12345678 unit=1


Disallowing 12345678-12345678 access to unit 1.
```

A Storage overview

Arrays and logical drives

The capacity and performance of a single physical (hard) drive is adequate for home users. However, business users demand higher storage capacities, higher data transfer rates, and greater protection against data loss when drives fail.

Connecting extra physical drives (Figure 1) to a system increases the total storage capacity but has no effect on the efficiency of read/write (R/W) operations. Data can still be transferred to only one physical drive at a time.

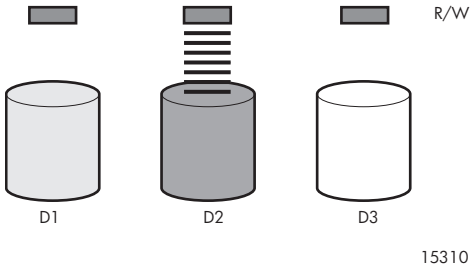


Figure 1 Multiple physical drives (D1, D2, and D3) in a system

An array controller combines several physical drives into one or more virtual units called logical drives, which have superior performance, capacity, and/or fault-tolerant features than separate physical drives. The read/write heads of all included physical drives are active simultaneously, reducing the total time required for data transfer.

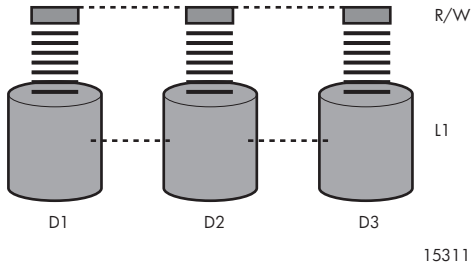


Figure 2 Multiple physical drives (D1, D2, and D3) configured into one logical drive (L1)

Because the read/write heads are active simultaneously, the same amount of data is written to each drive during any given time interval. Each unit of data is called a block, and adjacent blocks form a set of data stripes across all physical drives in that logical drive (Figure 3).

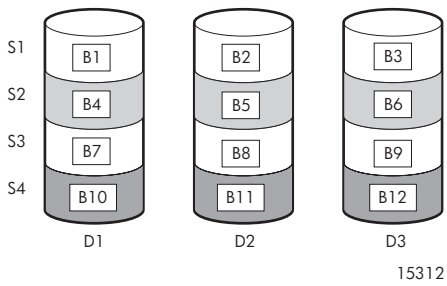


Figure 3 Data striping (S1-S4) and data blocks (B1-B12) on multiple physical drives (D1, D2, and D3)

For data in the logical drive to be readable, the data block sequence must be the same in every stripe. This sequencing process is performed by the array controller, which sends the data blocks to the drive write heads in the correct order.

A natural consequence of the striping process is that each physical drive in a given logical drive will contain the same amount of data. If one physical drive has a larger capacity than other physical drives in the same logical drive, the extra capacity is wasted, because it cannot be used by the logical drive.

The group of physical drives containing the logical drive is called a drive array, or array. Because all physical drives in an array are commonly configured into just one logical drive, the term array is often used as a synonym for logical drive. However, an array can contain several logical drives, each of a different size (Figure 4).

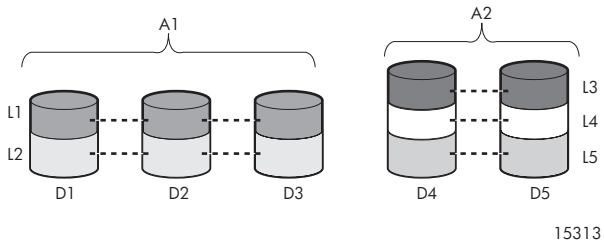


Figure 4 Two arrays (A1, A2) containing five logical drives (L1 through L5) spread across five physical drives (D1 through D5)

Each logical drive in an array is distributed across all of the physical drives within the array. A logical drive can also extend across more than one storage enclosure attached to the array system.

Drive failure, although rare, is potentially catastrophic. For example, in Figure 4, failure of any one physical drive in an array causes every logical drive in the array to suffer irretrievable data loss.

To protect against data loss due to physical drive failure, logical drives are usually configured with fault tolerance.

Fault-tolerance methods

To protect against data loss due to physical drive failure, logical drives are usually configured with fault tolerance. The following configuration types are available:

- RAID 0—Data striping only (no fault tolerance)
- RAID 1+0—Drive mirroring
- RAID 5—Distributed data guarding
- RAID 6 (ADG)—Advanced data guarding

For any configuration except RAID 0, further protection against data loss can be achieved by assigning a drive as an online spare. This drive contains no data and is connected to the same controller as the array. When any other physical drive in the array fails, the controller automatically rebuilds information data protection. (In the unlikely event that another drive in the array fails while data is being rewritten to the spare, the logical drive will still fail.)

A spare is assigned to an array and is automatically assigned to all logical drives in the same array. You do not need to assign a separate spare to each array; you can configure one hard drive to be the spare for several arrays.

RAID 0—No fault tolerance

A RAID 0 configuration (Figure 3) provides no protection against data loss when a drive fails. However, it is useful for rapid storage of large amounts of non-critical data (for printing or image editing, for example), or when cost is the most important consideration.

Table 3 RAID 0 features

Advantages	Disadvantages
Highest write performance of all RAID methods.	All data on the logical drive is lost if a physical drive fails.
Lowest cost per unit of stored data of all RAID methods.	Cannot use an online spare.
All drive capacity is used to store data (none is needed for fault tolerance).	Can only preserve data by backing it up to external drives.

RAID 1+0—Drive mirroring

In a RAID 1+0 configuration, data on a physical hard drive is duplicated to a second drive.

NOTE: When only two hard drives are included in the array, this fault-tolerance method is called RAID 1. When more than two hard drives are included in the array, this fault-tolerance method is called RAID 1+0. RAID 1 is not supported on the MSA1510i storage system.

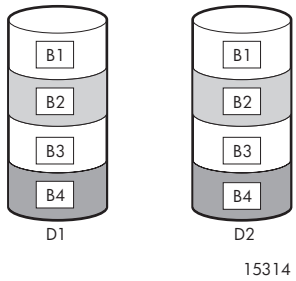


Figure 5 RAID 1 array, with two physical hard drives (D1 and D2)

When the array has more than two physical drives, drives are mirrored in pairs (Figure 6).

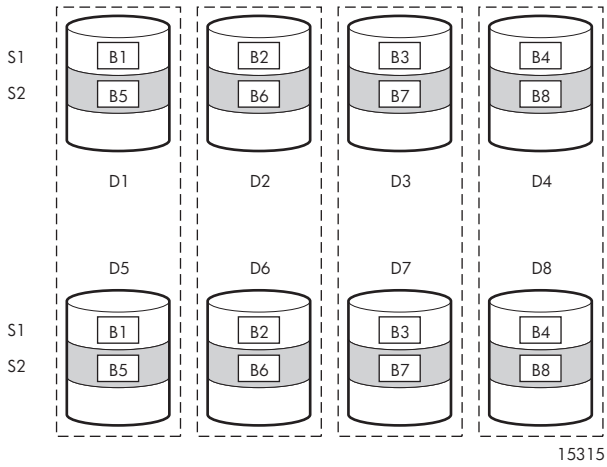


Figure 6 RAID 1+0 array, with eight physical hard drives (D1 through D8)

In each mirrored pair, the physical drive that is not busy answering other requests answers any read requests that are sent to the array. (This behavior is called load balancing.) If a physical drive fails, the remaining drive in the mirrored pair can still provide all the necessary data. Several drives in the array can fail without incurring data loss, as long as no two failed drives belong to the same mirrored pair.

RAID 1+0 is useful when high performance and data protection are more important than the cost of physical drives.

Table 4 RAID 1, RAID 1+0 features

Advantages	Disadvantages
Highest read and write performance of any fault-tolerant configuration.	Expensive (half of the drives are used for fault tolerance).
No loss of data as long as no failed drive is mirrored to another failed drive.	Only half of total drive capacity usable for data storage.

RAID 5—Distributed data guarding

In this method, a block of parity data is calculated for each stripe from the data that is in all other blocks within that stripe. The blocks of parity data are distributed across every physical drive within the logical drive (Figure 7). When a physical drive fails, data that was on the failed drive can be calculated from the data on the remaining drives and the parity data. This recovered data is written to the assigned spare or to a replacement drive in a process called a rebuild.

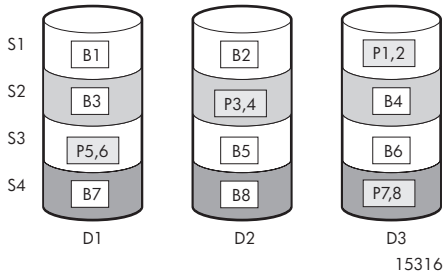


Figure 7 RAID 5 array, with three physical hard drives (D1, D2, and D3) showing distributed parity information (Px,y)

This configuration is useful when cost, performance, and data availability are equally important.

Table 5 RAID 5 features

Advantages	Disadvantages
Highest read performance.	Relatively low write performance.
No loss of data if one physical drive fails.	Loss of data if a second drive fails before data from the first failed drive is rebuilt.
More usable drive capacity than RAID 1+0, because parity information requires the storage space equivalent to one physical drive.	

RAID 6—Advanced data guarding

RAID 6 (also called RAID ADG) is similar to RAID 5, because both methods generate and store parity information to protect against data loss caused by drive failure. With RAID 6, however, two different sets of parity data are distributed across the physical drives, allowing data to be preserved even if two drives fail. Each set of parity data uses up a capacity equivalent to that of one of the constituent drives, as shown in Figure 8.

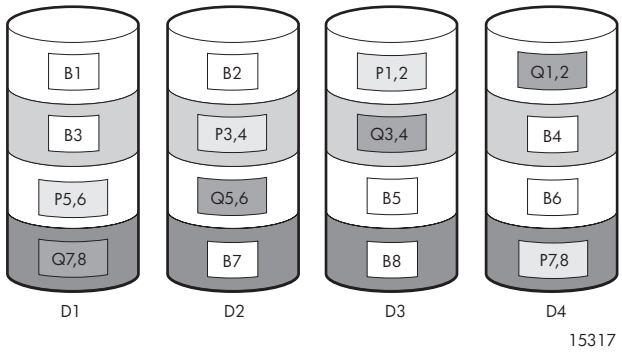


Figure 8 RAID 6 array, with four physical hard drives (D1, D2, D3, and D4) showing distributed parity information (Px,y)(Qx,y)

This method is most useful when data loss is unacceptable, but cost is also an important factor. The probability that data loss will occur when arrays are configured with RAID 6 is less than when they are configured with RAID 5.

Table 6 RAID 6 features

Advantages	Disadvantages
High read performance.	Relatively low write performance (lower than RAID 5), because of the need to create two sets of parity data.
High data availability—Any two drives can fail without loss of critical data.	
More drive capacity is usable than with RAID 1+0—Parity information requires only the storage equivalent to two physical drives.	

Comparison of RAID methods

Table 7 summarizes important features of the different RAID levels.

Table 7 Summary of RAID methods

	RAID 0	RAID 1+0	RAID 5	RAID 6
Alternative name	Striping	Mirroring	Distributed Data Guarding (DDG)	Advanced Data Guarding (ADG)
Usable drive space*	100%	50%	67% to 93%	50% to 96%
Usable drive space formula	n	n/2	(n-1)/n	(n-2)/n
Minimum number of physical drives	1	2	3	4
Tolerates physical drive failure?	No	Yes	Yes	Yes
Tolerates simultaneous failure of more than one physical drive?	No	Only if no two failed drives are in a mirrored pair	No	Yes
Read performance	High	High	High	High
Write performance	High	Medium	Low	Low
Relative cost	Low	High	Medium	Medium

*Values for usable drive space are calculated with these assumptions:

- All physical drives in the array have the same capacity.
- Online spares are not used.
- No more than 14 physical drives are used per array for RAID 5.
- No more than 56 drives are used with RAID 6.

Choosing a RAID level

Use Table 8 to help you determine the best RAID level for your environment.

Table 8 Choosing a RAID level

Most important characteristic	Also important	Suggested RAID level
Fault tolerance	Cost effectiveness	RAID 6
	I/O performance	RAID 1+0
Cost effectiveness	Fault tolerance	RAID 6*
	I/O performance	RAID 5 (RAID 0 if fault tolerance is not required)
I/O performance	Cost effectiveness	RAID 5 (RAID 0 if fault tolerance is not required)
	Fault tolerance	RAID 1+0

Index