DIGITAL StorageWorks HSZ50 Array Controller

HSOF Version 5.1 CLI Reference Manual

Part Number: EK–HSCLI–RM. B01

March 1997

Software Version: HSOF 5.1

Digital Equipment Corporation
Maynard, Massachusetts

March 1997

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 instruction manual, 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 his own expense. Restrictions apply to the use of the local-connection port on this series of controllers; failure to observe these restrictions may result in harmful interference. Always disconnect this port as soon as possible after completing the setup operation. Any changes or modifications made to this equipment may void the user's authority to operate the equipment.

Warning!
This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

Achtung!
Dieses ist ein Gerät der Funkstörgrenzwertklasse A. In Wohnbereichen können bei Betrieb dieses Gerätes Rundfunkstörungen auftreten, in welchen Fällen der Benutzer für entsprechende Gegenmaßnahmen verantwortlich ist.

Avertissement!
Cet appareil est un appareil de Classe A. Dans un environnement résidentiel cet appareil peut provoquer des brouillages radioélectriques. Dans ce cas, il peut être demandé à l' utilisateur de prendre les mesures appropriées.

Table of Contents

Glossary

Index

Figures

Related Documents

The following table lists some of the documents related to the use of this product.

Document title	Part number
DECevent Installation Guide	AA–Q73JA–TE
StorageWorks BA350–MA Controller Shelf User's Guide	EK–350MA–UG
StorageWorks Configuration Manager for DEC OSF/1 Installation Guide	AA–QC38A–TE
StorageWorks Configuration Manager for DEC OSF/1 System Manager's Guide for HSZterm	AA–QC39A–TE
StorageWorks Solutions Configuration Guide	EK–BA350–CG
StorageWorks Solutions Shelf and SBB User's Guide	EK–BA350–UG
StorageWorks Solutions SW300-Series RAID Enclosure Installation and User's Guide	EK–SW300–UG
StorageWorks SW500-Series Cabinet Installation and User's Guide	EK–SW500–UG
StorageWorks SW800-Series Data Center Cabinet Installation and User's Guide	EK–SW800–UG
The RAIDBOOK—A Source for RAID Technology	RAID Advisory Board
Polycenter Console Manager User's Guide	Computer Associates
VAXcluster Systems Guidelines for VAXcluster System Configurations	EK–VAXCS–CG
16-Bit SBB User’s Guide	EK-SBB16-UG
7-Bit SBB Shelf (BA356 Series) User’s Guide	EK-BA356-UG
SBB User’s Guide	EK-SBB35-UG

1 CLI Overview

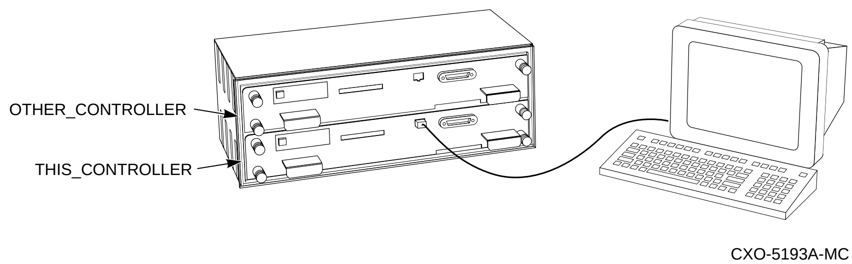
CLI Overview

The command line interpreter (CLI) is one of the user interfaces to the StorageWorks™ array controllers in your subsystem. The CLI commands allow you to manage the subsystem by viewing and modifying the configuration of the controllers and the devices attached to them. You also use the CLI to start controller diagnostic and utility programs.

CLI commands for configuring and viewing the controllers use the relative terms THIS_CONTROLLER and OTHER_CONTROLLER. THIS_CONTROLLER always refers to the controller that is running the current CLI session. You may have a maintenance terminal or PC directly connected to the terminal port on the front of the controller, or you may be using a virtual terminal connection through the host bus. In either case, THIS_CONTROLLER is the one that you are interacting with directly. This is shown in Figure 1–1.

The controller with the terminal connection is THIS_CONTROLLER.

Figure 1–1 Identifying this_controller, other_controller



OTHER_CONTROLLER always refers to the controller that is *not* running the current CLI session—the one that is not connected to a terminal or terminal session.

Accessing and Exiting the CLI

Note

The maintenance terminal port on the controller is meant to be used only for initial configuration and required maintenance functions. Operation of this port may cause harmful radio frequency interference. Always disconnect the terminal and cable when you have finished working with the controller.

You can access the CLI from a maintenance terminal connected to the front bezel of the controller (local connection) or through the host operating system using a VAXcluster™ system console (VCS), a diagnostic and utility protocol (DUP), or an HSZterm (remote connection). You *must* use a local connection to perform initial controller configuration, such as setting the controller ID. Once you have completed initial configuration and the controller is visible on the host bus, you can perform all other configuration through a remote connection.

In a dual-redundant controller configuration, you can view and set the configurations of both controllers with one local or remote connection. Use THIS_CONTROLLER commands to change the configuration on the controller that is running the CLI session. Use OTHER_CONTROLLER commands to view or change the configuration on the partner controller.

If you are using a remote DUP connection to the CLI, enter the EXIT command at the CLI prompt to close the connection. You do not need to exit from a local connection.

Capturing Your CLI Session

If you are using a remote terminal connection via DUP on an OpenVMS™ system and you specify the /LOG switch on your command line, a log file of your CLI session is created. You must use the EXIT command to exit the CLI in order to close and print the log file.

Entering CLI Commands

Use the following tips and techniques when entering CLI commands:

- Commands are not case sensitive.
- With few exceptions, you only need to enter enough of each command to make the command unique (usually three characters). For example, SHO is equivalent to SHOW.
- While the controller is processing a command, you can enter succeeding commands without waiting for the CLI prompt. The controller will process each command when it has completed the previous command. (A unit that is experiencing heavy I/O load may be slow to respond to CLI commands.)
- You can enter only one switch (or "qualifier") with each SET command. To enter multiple switches for the same device or unit, you must use multiple SET commands.

You can recall and edit the last five commands. This feature can save time and help prevent mistakes when you need to enter similar commands during configuration. Use the following keys to recall and edit commands:

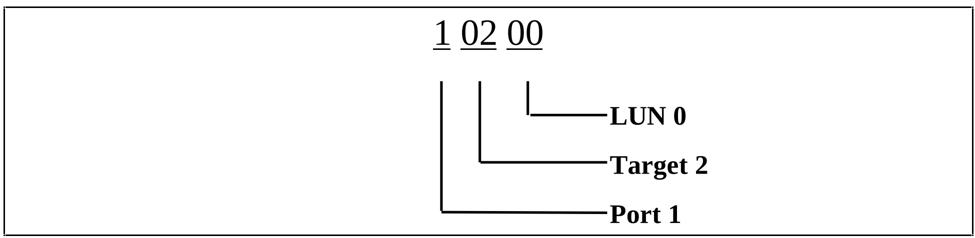
Key	Function
Up Arrow or Ctrl/B, Down Arrow or Ctrl/N	Steps backward and forward through the five most recent CLI commands.
Left arrow or Ctrl/D, Right arrow or Ctrl/F	Moves the cursor left or right in a command line.
Ctrl/E	Moves the cursor to the end of the line.
Ctrl/H or Backspace	Moves the cursor to the beginning of the line.
Ctrl/J or Linefeed	Deletes the word to the left of the cursor.
Ctrl/U	Deletes characters from the beginning of the line to the cursor.
Ctrl/A	Toggles between insert mode and overstrike mode. The default setting is insert mode, which allows you to insert characters at the cursor location, moving the existing characters to the right. Overstrike mode replaces existing characters when you type. The CLI returns to insert mode at the beginning of each line.
Ctrl/R	Recalls the contents of the command line. This is especially helpful if the system issues a message that interrupts your typing.

Specifying the Device PTL

Units accessed by the controller are commonly called logical units (LUN). Each logical unit has a number that is device addressable through a target. Often, a command requires that a device's port-target-LUN (PTL) address be entered. The PTL address is a five-digit address by which the controller identifies the location of the device.

- P—Designates the controller's SCSI port number (1through 6).
- T—Designates the target identification (ID) number of the device. Valid target ID numbers for a single-controller configuration are 0 through 6. Valid target ID numbers for a dual-redundant-controller configuration are 0 through 5.
- L—Designates the LUN of the device (must be 0).

Place one space between the port number, target number, and the two-digit LUN number when entering the PTL address. An example of a PTL address follows:



Note

DIGITAL UNIX® requires passthrough units to be at LUN 0. OpenVMS accepts LUN numbers 0 through 7.

Syntax

Commands to the controller must match the following command structure:

COMMAND PARAMETER SWITCHES

Command—Instructs the controller to do something. For example, the SET command instructs the controller to set something.

Parameter—The object that receives the command and is assigned the task of processing the command.

Switches—Reserved words that have a special function. For example, the member switch indicates that the term after it is the name or number of the storage member.

2
CLI Commands

Descriptions of all CLI Commands

Required Parameters

Optional Switches

Examples

ADD CDROM

Adds a CD–ROM drive to the controller configuration and names the drive.

Format

`ADD CDROM` *`CDROM-name SCSI-location`*

Parameters

CDROM-name
Specifies a name for the CD–ROM drive. You use this name with the ADD UNIT command to identify the CD–ROM drive as a host-addressable unit.

The name must start with a letter (A–Z) and can then consist of up to eight more characters made up of letters A-Z, numbers 0-9, periods (.), dashes (-), or underscores (_), for a total of nine characters.

SCSI-location
The *SCSI-location* parameter assigns a PTL address to the CD-ROM that is used by the controller. See page , for an explanation of the PTL numbering system.

Examples

To add a CD–ROM drive at port 1, target 0, LUN 0, named CD_PLAYER:

```
CLI> ADD CDROM CD_PLAYER 1 0 0
```

See also

ADD UNIT
DELETE
SHOW CDROM

ADD DISK

Adds a disk drive to the controller configuration and names the drive.

Format

`ADD DISK` *`disk-name SCSI-location`*

Parameters

disk-name
Specifies a name for the disk drive. You use this name when adding the disk drive to a storageset or with the ADD UNIT command to create a single-disk host-addressable unit.

The name must start with a letter (A–Z) and can then consist of up to eight more characters made up of letters A-Z, numbers 0-9, periods (.), dashes (-), or underscores (_), for a total of nine characters.

SCSI-location
The *SCSI-location* parameter assigns a PTL address to the disk drive. See page , for an explanation of the PTL numbering system.

Switches

TRANSPORTABLE
NOTRANSPORTABLE (Default)
Specify the TRANSPORTABLE switch for any disk drive that you want to move to and from StorageWorks environments with the data intact.

Note

Use the TRANSPORTABLE switch only for disk drives that must be moved to StorageWorks environments with data intact.

A TRANSPORTABLE disk drive does not contain any special information space and can be moved to a non-StorageWorks environment with its data intact. You cannot use a TRANSPORTABLE disk drive in any type of storageset, such as a stripeset or RAIDset.

Specify NOTRANSPORTABLE for all disk drives that will be used in RAIDsets, stripesets, and mirrorsets. Also use the NOTRANSPORTABLE switch for all single-disk units that will be used exclusively in a StorageWorks or HSC™ controller.

The controller makes a small portion of nontransportable disk drives inaccessible to the host and uses the space to store information (metadata) that improves data reliability, error detection, and recovery. Because of this metadata, only StorageWorks controllers can retrieve user data from nontransportable disk drives.

Examples

To add nontransportable DISK100 at port 1, target 0, LUN 0:

```
CLI> ADD DISK DISK100 1 0 0
```

To add transportable disk DISK230 at port 2, target 3, LUN 0:

```
CLI> ADD DISK DISK230 2 3 0 TRANSPORTABLE
```

See also

DELETE
LOCATE
SHOW DISKS
SET *disk-name*

ADD MIRRORSET

Adds a mirrorset to the controller configuration and names the mirrorset. The number of members is set to the number of devices specified in the command.

Format

```
ADD MIRRORSET mirrorset-name disk-name1 [disk-nameN]
```

Parameters

mirrorset-name
Specifies a name for the mirrorset. You use this name with the ADD UNIT command to identify the mirrorset as a host-addressable unit.

The name must start with a letter (A–Z) and can then consist of up to eight more characters made up of letters A-Z, numbers 0-9, periods (.), dashes (-), or underscores (_), for a total of nine characters.

disk-name1 disk-nameN
The names of the disk drives that make up the mirrorset. A mirrorset contains 1 to 6 disk drives.

Switches

COPY=NORMAL (Default)
COPY=FAST

Note

A normal mirrorset member is one whose entire contents are the same as all other normal members.

The COPY switch allows you to set the speed at which the controller copies data from normal mirrorset members to new members.

Select NORMAL to prioritize other controller operations over the copy operation. The controller uses relatively few resources to perform the copy, and there is little impact on performance.

Select FAST when the copy operation must take precedence over other controller operations. The controller uses more resources and the copy takes less time, but overall controller performance is reduced during the copy.

POLICY=BEST_FIT
POLICY=BEST_PERFORMANCE (Default)
NOPOLICY
The POLICY switch allows you to set the criteria the controller uses to choose a replacement member from the spareset when a mirrorset member fails.

Select BEST_FIT to choose a replacement device from the spareset that most closely matches the capacities of the remaining members. If more than one device in the spareset is the correct size, the controller selects the device that gives the best performance.

Select BEST_PERFORMANCE to choose a replacement device from the spareset that results in the best performance (the device should be on a different port than existing members). If more than one device in the spareset has the best performance, the controller selects the device that most closely matches the size of the remaining members.

READ_SOURCE=ROUND_ROBIN
READ_SOURCE=LEAST_BUSY (Default)
READ_SOURCE=disk-name
The READ_SOURCE switch allows you to control which mirrorset member is used by the controller to satisfy a read request.

Select ROUND_ROBIN to cause the controller to direct read requests to each NORMAL mirrorset member in sequential membership order. No preference is given to any NORMAL member.

Select LEAST_BUSY to cause the controller to direct read requests to the NORMAL mirrorset member with the least busy work queue.

Select the *disk-name* of a specific member to cause the controller to direct all read requests to that member. If that member fails out of the mirrorset, the controller reverts to the LEAST_BUSY method for mirrorset read requests.

Examples

To add DISK100, DISK210, and DISK320 as a mirrorset with the name MIRR1:

```
CLI> ADD DISK DISK100 1 0 0
CLI> ADD DISK DISK210 2 1 0
CLI> ADD DISK DISK320 3 2 0
CLI> ADD MIRRORSET MIRR1 DISK100 DISK210 DISK320
CLI> INITIALIZE MIRR1
CLI> ADD UNIT D305 MIRR1
```

See also

ADD DISK
DELETE
INITIALIZE
MIRROR
REDUCE
SHOW MIRRORSETS
UNMIRROR

ADD PASSTHROUGH

Creates a passthrough container (command disk) to allow direct access to a device. HSZ™ controllers use passthrough containers to communicate with tape drives and tape loaders.

Format

ADD PASSTHROUGH *passthrough-name SCSI-location*

Parameters

passthrough-name
Specifies a name for the passthrough container. You use this name with the ADD UNIT command to identify the passthrough container as a host-addressable unit. You may want to use a name that indicates the type of device served by the passthrough container, such as "TAPE" or "LOADER."

The name must start with a letter (A–Z) and can then consist of up to eight more characters made up of letters A-Z, numbers 0-9, periods (.), dashes (-), or underscores (_), for a total of nine characters.

loader-name
The name of the tape loader device that receives passthrough commands. This is the name that was given to the device with the ADD PASSTHROUGH command.

SCSI-location
The *SCSI-location* parameter assigns a PTL address to the drive or loader.
See page 1–6, for an explanation of the PTL numbering system.

Examples

To add tape drive TAPE230 at port 2, target 3, LUN 0 on an HSZ controller:

Note

DIGITAL UNIX requires HSZ passthrough units to be at LUN 0 (the third digit in the unit number). OpenVMS accepts LUN numbers 0-7.

```
CLI> ADD PASSTHROUGH TAPE230 2 3 0
CLI> ADD UNIT P100 TAPE230
```

See also

ADD UNIT
DELETE
SHOW PASSTHROUGH

ADD RAIDSET

Creates a RAIDset from 3 to 14 disks.

Format

ADD RAIDSET *RAIDset-name disk-name1 disk-name2 [disk-nameN]*

Parameters

RAIDset-name
Specifies a name for the RAIDset. You use this name with the ADD UNIT command to identify the RAIDset as a host-addressable unit.

The name must start with a letter (A–Z) and can then consist of up to eight more characters made up of letters A-Z, numbers 0-9, periods (.), dashes (-), or underscores (_), for a total of nine characters.

disk-name1 disk-name2 disk-nameN
The disks that will make up the RAIDset. A RAIDset can contain 3 to 14 member disks.

Switches

POLICY=BEST_FIT
POLICY=BEST_PERFORMANCE (Default)
NOPOLICY
The POLICY switch allows you to set the criteria the controller uses to choose a replacement member from the spareset when a RAIDset member fails.

Select BEST_FIT to choose a replacement device from the spareset that most closely matches the capacities of the remaining members. If more than one device in the spareset is the correct size, the controller selects the device that gives the best performance.

Select BEST_PERFORMANCE to choose a replacement device from the spareset that results in the best performance (the device should be on a different port than existing members). If more than one device in the spareset has the best performance, the controller selects the device that most closely matches the size of the remaining members.

Select NOPOLICY to prevent the controller from replacing a failed disk drive. This causes the RAIDset to run in a reduced state until a BEST_FIT or BEST_PERFORMANCE policy is selected, or a member is manually replaced in the RAIDset (see SET *RAIDset-name*).

RECONSTRUCT=NORMAL (Default)
RECONSTRUCT=FAST
The RECONSTRUCT switch allows you to set the speed at which the controller reconstructs the data on a new RAIDset member that has replaced a failed member.

Select NORMAL to balance other controller operations against the reconstruct operation. The controller uses relatively few resources to perform the reconstruct, and there is little impact on performance.

Select FAST when the reconstruct operation must take precedence over other controller operations. The controller uses more resources and the reconstruct takes less time, but overall controller performance is reduced during the reconstruct.

REDUCED
NOREDUCED (Default)
The REDUCED switch allows you to add a RAIDset that is missing one member. You only need to use the REDUCED switch if you are re-adding a reduced RAIDset to the subsystem. The NOREDUCED setting is the default and indicates that all RAIDset members that make up the RAIDset are being specified, such as when creating a new RAIDset.

Examples

To create RAIDset RAID9 with disks DISK100, DISK210, and DISK320:

```
CLI> ADD DISK DISK100 1 0 0
CLI> ADD DISK DISK210 2 1 0
CLI> ADD DISK DISK320 3 2 0
CLI> ADD RAIDSET RAID9 DISK100 DISK210 DISK320
CLI> INITIALIZE RAID9
CLI> ADD UNIT D204 RAID9
```

To create RAIDset RAID8 with disks DISK100, DISK210, and DISK320, and use the BEST_FIT replacement policy:

```
CLI> ADD DISK DISK100 1 0 0
CLI> ADD DISK DISK210 2 1 0
CLI> ADD DISK DISK320 3 2 0
CLI> ADD RAIDSET RAID8 DISK100 DISK210 DISK320 POLICY=BEST_FIT
CLI> INITIALIZE RAID8
CLI> ADD UNIT D205 RAID8
```

This example shows creating a three-member RAIDset from the members of a four-member RAIDset that was already reduced. Note that you must not initialize the RAIDset, because it was initialized in its previous location.

```
CLI> ADD DISK DISK100 1 3 0
CLI> ADD DISK DISK210 2 4 0
CLI> ADD DISK DISK320 3 5 0
CLI> ADD RAIDSET RAID6 DISK130 DISK240 DISK350 REDUCED
CLI> ADD UNIT D205 RAID6
```

See also

ADD UNIT
DELETE
SET RAIDSET
SHOW RAIDSET
INITIALIZE

ADD SPARESET

Adds a disk drive to the spareset and initializes the metadata on the drive. The spareset is a pool of disk drives available to the controller to replace failing members of RAIDsets and mirrorsets.

Format

`ADD SPARESET` *`disk-name`*

Parameters

disk-name
The name of the disk drive to add to the spareset. You can add only one drive to the spareset with each command.

Examples

To add DISK220 and DISK330 to the spareset:

```
CLI> ADD DISK DISK220 2 2 0
CLI> ADD DISK DISK330 3 3 0
CLI> ADD SPARESET DISK220
CLI> ADD SPARESET DISK330
```

See also

DELETE SPARESET
SET FAILEDSET
SHOW SPARESET

ADD STRIPESET

Creates a stripeset out of 2 to 14 disks or mirrorsets.

Format

ADD STRIPESET *stripeset-name container-name1 [container-nameN]*

Parameters

stripeset-name
Specifies a name for the stripeset. You use this name with the ADD UNIT command to identify the stripeset as a host-addressable unit.

The name must start with a letter (A–Z) and can then consist of up to eight more characters made up of letters A-Z, numbers 0-9, periods (.), dashes (-), or underscores (_), for a total of nine characters.

container-name1 container-nameN
The names of the disk drives or mirrorsets that make up the stripeset. A stripeset can be made up of from 2 to 14 containers.

Examples

To create stripeset STRIPE1 with three disks: DISK100, DISK210, and DISK320:

```
CLI> ADD DISK DISK100 1 0 0
CLI> ADD DISK DISK210 2 1 0
CLI> ADD DISK DISK320 3 2 0
CLI> ADD STRIPESET STRIPE1 DISK100 DISK210 DISK320
CLI> INITIALIZE STRIPE1
CLI> ADD UNIT D403 STRIPE1
```

The next example shows creating a two-member striped mirrorset (a stripeset whose members are mirrorsets). Note that you only need to initialize the stripeset; you do not need to initialize the mirrorsets individually.

```
CLI> ADD DISK DISK100 1 0 0
CLI> ADD DISK DISK210 2 1 0
CLI> ADD DISK DISK320 3 2 0
CLI> ADD DISK DISK430 4 3 0
CLI> ADD MIRRORSET MR1 DISK100 DISK210
CLI> ADD MIRRORSET MR2 DISK320 DISK430
CLI> ADD STRIPESET STRIPE1 MR1 MR2
CLI> INITIALIZE STRIPE1
CLI> ADD UNIT D304 STRIPE1
```

See also

ADD UNIT
ADD MIRRORSET
DELETE
INITIALIZE
SHOW STRIPESET

ADD UNIT

Identifies a device, container, or partition as a host-addressable logical unit. The controller maps all requests from the host to the logical unit number as requests to the container specified in the ADD UNIT command.

You must initialize the disk drive or container before you can add it as a unit.

Format

`ADD UNIT` *`unit-number container-name`*

Parameters

unit-number
A number such as D104, where D represents a device-type letter, 1 is one of the controller target IDs, and the 4 is a logical unit number (the middle number is always 0).

The device type letter is either “D” for disk containers (including CD–ROMs) or “P” for passthrough containers created from tape drives and loaders.

The controller target ID must be in the range 0 to 7, and must be one of the IDs set with the SET THIS_CONTROLLER ID and SET OTHER_CONTROLLER ID commands. You must leave out the target ID if it is 0, and only use the logical unit number.

The logical unit number must be in the range 0 to 7. You can assign up to eight logical units for each controller target ID.

Table 2–1 HSZ Unit Numbering breaks out the components in HSZ unit numbering.

Table 2–1 HSZ Unit Numbering

Unit number	Device type	Target ID	LUN
D401	disk	4	1
P100	passthrough	1	0
D5	disk	0	5

On a partitioned container, all of the partitions must be addressed through a single controller. Thus, the unit numbers for all partitions on a container must have the same controller target ID.

container-name
The name of the container that is used to create the unit.

Switches

Table 2–2 lists all switches for the ADD UNIT command and shows which switches can be used with each type of device and storageset. Descriptions of the switches follow the table.

Table 2–2 Switches for the ADD UNIT command

Switch	RAID - set	Stripe-set	Mirror-set	NoTrans Disk	Trans Disk	CD–ROM	Pass-through
PARTITION=partition_number	✓	✓	✓	✓			
MAXIMUM_CACHED_TRANSFER	✓	✓	✓	✓	✓	✓	
PREFERRED_PATH NOPREFERRED_PATH (Default)	✓	✓	✓	✓	✓	✓	✓
READ_CACHE (Default) NOREAD_CACHE	✓	✓	✓	✓	✓	✓	
RUN (Default) NORUN	✓	✓	✓	✓	✓	✓	
WRITE_PROTECT NOWRITE_PROTECT (Default)	✓	✓	✓	✓	✓		
WRITEBACK_CACHE NOWRITEBACK_CACHE (Default)	✓	✓	✓	✓			

Note

The RUN and NORUN switches cannot be specified for partitioned units.

MAXIMUM_CACHED_TRANSFER=n
MAXIMUM_CACHED_TRANSFER=32 (Default)

The MAXIMUM_CACHED_TRANSFER switch allows you to set the largest transfer (in blocks) to be cached by the controller. The controller will not cache any transfers over this size. Valid values are 1–1024.

PREFERRED_PATH=THIS_CONTROLLER
PREFERRED_PATH=OTHER_CONTROLLER
NOPREFERRED_PATH (Default)

Note

All partitions on a container must be addressed through the same controller. If you set PREFERRED_PATH for one partition, all partitions on that container will inherit the same path.

The PREFERRED_PATH switch allows you to balance I/O load by specifying the controller through which the unit will be accessed. If you set NOPREFERRED_PATH for a unit, it can be accessed through either controller.

The controllers only use the PREFERRED_PATH setting if they are in a dual-redundant configuration. If one controller fails, all the devices are accessed through the remaining controller, ignoring the PREFERRED_PATH setting.

Note

The PREFERRED_PATH and NOPREFERRED_PATH switches are valid only for HSZ controllers in dual-redundant multiple bus-failover configurations. The preferred path for units on dual-redundant HSZ controllers is determined by the first digit in the unit number (the controller target ID) specified in the ADD UNIT command.

When the failed controller is restarted, the drives automatically return to the controller specified by the PREFERRED_PATH switch.

You can specify the PREFERRED_PATH switch on a single controller; however, the switch will not take effect until you add a second controller and configure the two controllers for dual-redundancy.

READ_CACHE (Default)
NOREAD_CACHE

The READ_CACHE switch allows you to enable or disable the use of read cache with a unit.

Read caching improves performance in almost all situations, so it is generally good to leave it enabled. However, under certain workloads, like a backup, there may be a low probability for a cache hit and it may be beneficial to turn read caching off to remove the overhead of caching.

RUN (Default)
NORUN

The RUN switch allows you to enable and disable a unit's availability to the host.

Select NORUN to make a unit unavailable to the host and to cause any user data for that unit to be flushed from the write-back cache to the disk drives. The devices that make up the unit are still spun up.

Select RUN to make a unit available to the host.

Note

The RUN and NORUN switches cannot be specified for partitioned units.

WRITE_PROTECT
NOWRITE_PROTECT (Default)

Note

Write protect disables the writing of any new data by the host. However, the controller may still write to a write-protected RAIDset to satisfy a reconstruct pass or to reconstruct a newly replaced member.

The WRITE_PROTECT switch allows you to ensure that data on a unit is not overwritten.

Select WRITE_PROTECT to prevent write operations to the unit.

Select NOWRITE_PROTECT to allow both reads and writes to the unit.

WRITEBACK_CACHE
NOWRITEBACK_CACHE (Default)

Note

WRITEBACK_CACHE requires that write-back cache modules be installed in the subsystem and that the licensed feature be enabled through FLS.

The WRITEBACK_CACHE switch allows you to enable or disable the use of write-back cache with a unit.

Select WRITEBACK_CACHE for all new RAIDsets and mirrorsets, and for all other units for which you want this feature. The write-back cache improves performance for write operations to the unit.

Note

It may take up to 5 minutes for the controller to flush unwritten data from the cache once you disable write-back caching.

Select NOWRITEBACK_CACHE for those units for which you want all writes to go directly to devices without being cached.

Considerations When Using Write-back Caching

The following list summarizes considerations you must be aware of when using write-back cache.

Two conditions will cause data contained within write-back cache to be lost: if power from the main power supply and the external cache battery is interrupted, or if the cache module is removed before the controller flushes the data from the write-back cache.

- When restarted, the controller attempts to flush any unflushed data within write-back cache to the devices. However, by specifying the IGNORE_ERRORS or IMMEDIATE_SHUTDOWN switch, you allow data to reside in write-back cache when the controller is turned off, regardless of any errors detected.
- RAIDsets and mirrorsets require data to be stored in write-back cache to accomodate the write hole and to increase performance—without regard to the WRITEBACK_CACHE switch setting.

If data is contained within the write-back cache while the subsystem is shut down, do not perform any hardware changes until after the controller flushes the data to the devices.

When restarted, the controller attempts to flush any unflushed data within the write-back cache to the devices. However, by specifying the IGNORE_ERRORS or IMMEDIATE_SHUTDOWN switch, you allow data to reside in write-back cache when the controller is turned off, regardless of any errors detected.

Examples

To create disk unit D102 from the single disk DISK100 and set unit access to be through THIS_CONTROLLER:

```
CLI> ADD DISK DISK100 1 0 0
CLI> INITIALIZE DISK100
CLI> ADD UNIT D102 DISK100 PREFERRED_PATH=THIS_CONTROLLER
```

To create disk unit D107 from RAIDset RAID9 and enable write-back caching for the unit:

```
CLI> ADD DISK DISK110 1 1 0
CLI> ADD DISK DISK210 2 1 0
CLI> ADD DISK DISK310 3 1 0
CLI> ADD DISK DISK410 4 1 0
CLI> ADD RAIDSET RAID9 DISK110 DISK210 DISK310 DISK410
CLI> INITIALIZE RAID9
CLI> ADD UNIT D107 RAID9 WRITEBACK_CACHE
```

See also

CREATE_PARTITION
DELETE *unit-number*
SET *unit-number*
SHOW UNITS

CLEAR_ERRORS CLI

Stops the display of current or previous error messages at the CLI prompt.

Note

This command does not clear the error conditions, it only stops displaying the errors at the CLI prompt.

The controller displays any error messages before the CLI prompt. After you correct the error condition, you must issue the CLEAR_ERRORS CLI command or restart the controller to clear the error message.

Format

```
CLEAR_ERRORS CLI
```

Examples

To clear the message “All NVPM components initialized to their default settings.” from the CLI prompt:

```
CLI> All NVPM components initialized to their default settings.
CLI> CLEAR_ERRORS CLI
```

See also

CLEAR_ERRORS INVALID_CACHE
CLEAR_ERRORS LOST_DATA
CLEAR_ERRORS UNKNOWN
CLEAR_ERRORS UNWRITEABLE_DATA

CLEAR_ERRORS INVALID_CACHE

Clears an INVALID_CACHE error and allows the controller and cache to resume operation.

A controller presents an INVALID CACHE error during initialization if it detects a mismatch between the cache information on the controller and the cache information on the cache module. The controller allows only a limited set of CLI commands until you clear the error, to prevent you from giving commands that may make the situation worse. The CLEAR_ERRORS INVALID_CACHE command either clears the information in the controller module or the information in the cache module.

Format

`CLEAR_ERRORS` *`controller`* `INVALID_CACHE` *`data_retention`*

You must completely spell out "INVALID_CACHE."

Parameters

controller
Identifies the controller for which to clear the INVALID_CACHE condition. You must specify either THIS_CONTROLLER or OTHER_CONTROLLER.

data_retention
Specifies whether to keep the cache data and overwrite the controller information (NODESTROY_UNFLUSHED_DATA) or keep the controller information and discard the cache data (DESTROY_UNFLUSHED_DATA).

Use the NODESTROY_UNFLUSHED_DATA parameter in the following situations:

- You have replaced the controller module.
- The controller's nonvolatile memory (NVMEM) has lost its memory, possibly from an NVMEM battery failure.
- You removed the cache module while there was unflushed data.

Use the DESTROY_UNFLUSHED_DATA parameter in the following situations:

- You have replaced the cache module.
- Any other reason not listed above.

Examples

To clear an invalid cache error on THIS_CONTROLLER after replacing a controller module:

```
CLI> CLEAR_ERRORS THIS_CONTROLLER INVALID_CACHE
NODESTROY_UNFLUSHED_DATA
```

To clear an invalid cache error on the OTHER_CONTROLLER after replacing a cache module:

```
CLI> CLEAR_ERRORS OTHER_CONTROLLER INVALID_CACHE
DESTROY_UNFLUSHED_DATA
```

See also

CLEAR_ERRORS CLI
CLEAR_ERRORS LOST_DATA
CLEAR_ERRORS UNKNOWN
CLEAR_ERRORS UNWRITEABLE_DATA

CLEAR_ERRORS LOST_DATA

Clears the lost data error on a unit.

Note

Issuing the CLEAR_ERRORS LOST_DATA command for one partition on a container will clear the lost data for all partitions on the container.

If a write-back cache module is removed or fails while it contains data that has not been written to disk, a LOST DATA error is reported on the unit. The CLEAR_ERRORS LOST_DATA command clears the lost data error but does not recover the lost data. It may take up to 5 minutes to clear lost data.

Format

CLEAR_ERRORS *unit-number* LOST_DATA

You must completely spell out "LOST_DATA".

Parameters

unit-number
Specifies the unit number that will have the lost data error cleared. The *unit-number* is the name given the unit when it was created using the ADD UNIT command.

Examples

To clear the lost data error on disk unit D103:

CLI> **CLEAR_ERRORS D103 LOST_DATA**

See also

CLEAR_ERRORS CLI
CLEAR_ERRORS INVALID_CACHE
CLEAR_ERRORS UNKNOWN
CLEAR_ERRORS UNWRITEABLE_DATA

CLEAR_ERRORS UNKNOWN

Clears the UNKNOWN error from a device.

If a device has a failure such that the controller marks the device as UNKNOWN, the controller does not check the device again to see if it has been repaired or if the failure condition has been rectified. You must use this command for the controller to recognize the device after you correct the condition that caused the device to be marked UNKNOWN.

Format

`CLEAR_ERRORS` *`device-name`* `UNKNOWN`

You must completely spell out "UNKNOWN."

Parameters

device-name
Specifies the name of the device with the UNKNOWN error.

Examples

To cause the controller to recognize DISK300, a previously UNKNOWN device:

```
CLI> CLEAR_ERRORS DISK300 UNKNOWN
```

See also

CLEAR_ERRORS CLI
CLEAR_ERRORS INVALID_CACHE
CLEAR_ERRORS LOST_DATA
CLEAR_ERRORS UNWRITEABLE_DATA

CLEAR_ERRORS UNWRITEABLE_DATA

Clears the unwriteable data error on a unit.

Caution

This command causes loss of user data.

If a container fails in a way that user data in the write-back cache cannot be written to the container, the controller reports an unwriteable data error. The CLEAR_ERRORS UNWRITEABLE_DATA command removes the data from the cache and clears the unwriteable data error.

Format

`CLEAR_ERRORS` *`unit-number`* `UNWRITEABLE_DATA`

You must completely spell out "UNWRITEABLE_DATA."

Parameters

unit-number
Specifies the unit number that will have the unwriteable data error cleared. The *unit-number* is the name given the unit when it was created using the ADD UNIT command.

Examples

To clear the unwriteable data error on disk unit D103:

```
CLI> CLEAR_ERRORS D103 UNWRITEABLE_DATA
```

See also

CLEAR_ERRORS CLI
CLEAR_ERRORS INVALID_CACHE
CLEAR_ERRORS LOST_DATA
CLEAR_ERRORS UNKNOWN
RETRY_ERRORS UNWRITEABLE_DATA

CREATE_PARTITION

Marks part of a disk or storageset to be used as a separately addressable unit. You can divide any disk or storageset into as many as six partitions, each of which can be separately presented to the host.

Format

`CREATE_PARTITION` *`container-name`* `SIZE=`*`n`*

Parameters

container-name

Note

Once you have partitioned a container, you cannot reunify it without reinitializing the container.

The name of the disk or storageset from which you want to create a partition. This is the name given the disk or storageset when it was created using the ADD command (ADD DISK, ADD STRIPESET, and so forth). You can partition any disk, stripeset, mirrorset, striped mirrorset, or RAIDset. You must initialize the container before creating partitions.

SIZE=percent
SIZE=LARGEST
Specifies the size of the partition to be created as a percentage of the total size of the disk or storageset.

Specify a percentage of the total container capacity to set the partition to a particular size, or to evenly divide the disk or storageset. The resulting partition will be slightly smaller than the size you specify, to accomodate controller information and so that each partition ends with a complete stripe.

Specify LARGEST to have the controller create the largest partition possible from unused space on the disk or storageset. You must also use the LARGEST setting when creating the last partition on a container, because the space will not be equal to an exact percentage value.

Examples

To create RAIDset R9 and divide it into four parts:

```
CLI> ADD DISK DISK100 1 0 0
CLI> ADD DISK DISK210 2 1 0
CLI> ADD DISK DISK320 3 2 0
CLI> ADD RAIDSET RAID9 DISK100 DISK210 DISK320
```

```
CLI> INITIALIZE RAID9
CLI> CREATE_PARTITION RAID9 SIZE=25
CLI> CREATE_PARTITION RAID9 SIZE=25
CLI> CREATE_PARTITION RAID9 SIZE=25
CLI> CREATE_PARTITION RAID9 SIZE=LARGEST
CLI> ADD UNIT D101 RAID9 PARTITION=1
CLI> ADD UNIT D102 RAID9 PARTITION=2
CLI> ADD UNIT D103 RAID9 PARTITION=3
CLI> ADD UNIT D104 RAID9 PARTITION=4
```

See also

ADD UNIT
DELETE *unit-number*
DESTROY PARTITION
SHOW

DELETE *container-name*

Deletes a container from the list of known containers. You cannot delete a container that is in use by a higher-level container. For example, you cannot delete a disk that is a member of a RAIDset, or a RAIDset that is declared as a unit. You must first delete the higher-level containers.

Format

`DELETE container-name`

Parameters

container-name

Note

You cannot delete the spareset and failedset containers.

Specifies a name that identifies the container. This is the name given the container when it was created using the ADD command (ADD DISK, ADD STRIPESET, and so forth).

Examples

To delete disk drive DISK100:

```
CLI> DELETE DISK100
```

To delete stripeset STRIPE1:

```
CLI> DELETE STRIPE1
```

To delete RAIDset RAID9:

```
CLI> DELETE RAID9
```

See also

DELETE FAILEDSET
DELETE SPARESET
UNMIRROR

DELETE *device-name*

Deletes a device from the controller configuration.

Format

`DELETE` *`device-name`*

Parameters

device-name

The name of the device to delete.

Examples

To delete disk device number 120:

CLI> **DELETE DISK120**

To delete tape device number 130:

CLI> **DELETE TAPE130**

See also

SET *device-name*

DELETE FAILEDSET

Removes a disk drive from the failedset. The failedset contains the drives that were removed from RAIDsets and mirrorsets either because they failed or were manually removed via the SET command. The DELETE FAILEDSET command removes disk drives from the failedset, typically before you remove them physically from the shelf for testing, repair, or replacement.

You should consider disk drives in the failedset to be defective. Test the disk drives, then either repair them or replace them.

Format

`DELETE FAILEDSET` *`disk-name1`*

Parameters

disk-name1
The disk drive name to delete from the failedset. You can delete one disk at a time from the failedset.

Examples

To delete DISK220 from the failedset:

```
CLI> DELETE FAILEDSET DISK220
```

See also

SET FAILEDSET
SHOW FAILEDSET

DELETE SPARESET

Removes a disk drive from the spareset. The spareset is a pool of drives available to the controller to replace failing members of RAIDsets and mirrorsets.

Format

`DELETE SPARESET` *`disk-name1`*

Parameters

disk-name1
The names of the disk drive to remove from the spareset. You can remove one disk at a timefrom the spareset.

Examples

To remove DISK110 from the spareset:

`CLI>` **`DELETE SPARESET DISK230`**

See also

ADD SPARESET
SHOW SPARESET

DELETE *unit-number*

Deletes a logical unit from the controller configuration. The device, storageset, or partition will no longer be addressable by the host. If write-back caching was enabled for the unit, the controller flushes any user data from the cache to the devices before deleting the unit.

If a unit has cache errors, you must first clear the errors with the CLEAR_ERRORS UNWRITEABLE_DATA or CLEAR_ERRORS LOST_DATA commands.

Format

```
DELETE unit-number
```

Parameters

unit-number
Specifies the logical unit number that is to be deleted. The *unit-number* is the name given the unit when it was created using the ADD UNIT command.

Examples

To delete passthrough unit number 103:

```
CLI> DELETE P103
```

See also

ADD UNIT
CLEAR_ERRORS LOST_DATA
CLEAR_ERRORS UNWRITEABLE_DATA
DESTROY_PARTITION

DESTROY_PARTITION

Marks a used partition as free, and consolidates it with any adjacent free partitions. User data on the partition is lost.

You cannot destroy a partition that has a unit number assigned; you must first use the DELETE *unit-number* command to delete the unit.

Format

`DESTROY_PARTITION` *`container-name`* `PARTITION=`*`partition-number`*

Parameters

container-name
The name of the disk or storageset that contains the partition that you want to destroy. This is the name given the disk or storageset when it was created using the ADD command (ADD DISK, ADD STRIPESET, and so forth).

partition-number

Note

Be sure to use the SHOW command to identify the correct partition before using the DESTROY_PARTITION command.

The number of the partition that you want to destroy. You can use the SHOW command to find partition numbers.

Examples

To delete the unit for partition 2 on RAIDset RAID9 and destroy the partition:

```
CLI> DELETE D102
CLI> DESTROY_PARTITION RAID9 PARTITION=2
```

See also

CREATE_PARTITION
DELETE *unit-number*
SHOW

DIRECTORY

Lists the diagnostics and utilities available on THIS_CONTROLLER.The diagnostics and utilities displayed may not appear in the order shown in the example below.

Format

```
DIRECTORY
```

Examples

To display directory listing:

```
CLI> DIRECTORY
DILX   V51Z   D
DSTAT  V51Z   D
VTDPY  V51Z   D
HSUTIL V51Z   D
C_SWAP V51Z   D
DIRECT V51Z   D
 CFMENU V51Z   D
CHVSN  V51Z   D
CLCP   V51Z   D
CLONE  V51Z   D
CONFIG V51Z   D
FMU    V51Z   D
```

See also

HELP
RUN

EXIT

Exits the CLI and breaks the remote connection.

When you enter the EXIT command from a remote connection, the connection is broken and control is returned to the host. When you enter the EXIT command from a local connection, the CLI restarts, displaying the copyright notice, the controller type, and the last fail packet.

Format

```
EXIT
```

Examples

To exit from a virtual terminal connection:

```
CLI> EXIT
Control returned to host
$
```

HELP

Displays a brief description of using the question mark (?) to obtain help on any command or CLI function.

Format

```
HELP
```

Examples

To see help about using the HELP command:

```
CLI> HELP
Help may be requested by typing a question mark (?) at the CLI
prompt. This will print a list of all available commands
For further information you may enter a partial command and type a
space followed by a "?" to print a list of all available options at
that point in the command. For example:

         SET THIS_CONTROLLER ?

Prints a list of all legal SET THIS_CONTROLLER commands
```

To get help on the SET command, using the (?) facility:

```
CLI> SET ?
Your options are:
FAILOVER
OTHER_CONTROLLER
NOFAILOVER
THIS_CONTROLLER
FAILEDSET
Unit number or container name
```

See also

DIRECTORY

INITIALIZE

Initializes a container so that you can create a logical unit from it. During initialization, a small amount of disk space is reserved for controller metadata and is made inaccessible to the host.

Caution

The INITIALIZE command destroys all user data on the container, unless you specify the NODESTROY switch.

If you set a single-disk container as TRANSPORTABLE, any metadata is destroyed on the device and the full device is accessible to the host.

You must use the INITIALIZE command before you:

- Create a unit from a newly installed disk
- Create a unit from a newly created RAIDset, stripeset, or mirrorset

You do not need to use the INITIALIZE command when you:

- Create a new unit from the same disks that were previously initialized as another unit, such as when a unit is moved
- Add a RAIDset with the REDUCED switch

Format

`INITIALIZE` *`container-name`*

Parameters

container-name
Specifies the container (disk, stripeset, mirrorset, or RAIDset) to initialize.

Switches

CHUNKSIZE=n
CHUNKSIZE=DEFAULT (Default)
Specifies the chunksize to be used for RAIDsets and stripesets. You can specify the chunk size in blocks (CHUNKSIZE=*n*), or you can let the controller determine the optimal chunk size (CHUNKSIZE=DEFAULT). The CHUNKSIZE switch does not apply to mirrorsets.

DESTROY(Default)
NODESTROY
The NODESTROY switch instructs the controller not to overwrite the user data and forced error metadata during the initialization. Only use the NODESTROY switch when you want to create a unit out of devices that have been reduced from mirrorsets. This allows the data on the container to be reused for a disk, stripeset, or mirrorset unit. (The NODESTROY switch is ignored for RAIDsets.)

SAVE_CONFIGURATION
NOSAVE_CONFIGURATION (Default)
The SAVE_CONFIGURATION switch instructs the controller to keep a copy of the controller configuration on the disk devices that are being initialized. Should you replace the controller in a single-controller configuration, the configuration information will be read from a device and loaded into the new controller.

Specify SAVE_CONFIGURATION when initializing any disk device or container on which you want to store a copy of the controller configuration. If you use the switch for a multi-device storageset, such as a stripeset, the complete information is stored on each device in the storageset.

Note

DIGITAL recommends that the SAVE_CONFIGURATION switch only be used for single controller configurations. (Use the SET FAILOVER COPY= command to save configuration information for dual-redundant configurations.)

Specify NOSAVE_CONFIGURATION for devices and storagesets on which you do not want to store a copy of the controller configuration.

A device that is initialized with the SAVE_CONFIGURATION switch has slightly less storage space than an identical device that is not used to store the controller configuration.

SAVE_CONFIGURATION only requires one disk be initialized with this option, but more may be used, if desired. DIGITAL *does not* recommend initializing *all* disks with the SAVE_CONFIGURATION switch, because every update to nonvolatile memory causes writes to all disks and can affect performance adversely.

SAVE_CONFIGURATION is not available for upgrades of software or hardware, and will not perform inter-platform conversions. For example, you cannot use SAVE_CONFIGURATION to upgrade from HSOF V3.1 to V5.1 or from an HSZ40 array controller to an HSZ50 array controller.

Examples

To initialize single-disk container DISK100 and save a copy of the controller configuration on it:

```
CLI> ADD DISK DISK100 1 0 0
CLI> INITIALIZE DISK100 SAVE_CONFIGURATION
CLI> ADD UNIT D102 DISK100
```

To initialize stripeset STRIPE1 with the default chunk size (note that if you do not specify the chunksize, the controller initializes the unit with the default chunksize):

```
CLI> ADD DISK DISK110 1 1 0
CLI> ADD DISK DISK210 2 1 0
CLI> ADD DISK DISK310 3 1 0
CLI> ADD STRIPESET STRIPE1 DISK110 DISK210 DISK310
CLI> INITIALIZE STRIPE1
CLI> ADD UNIT D204 STRIPE1
```

To initialize RAIDset RAID9 with a chunk size of 20:

```
CLI> ADD DISK DISK120 1 2 0
CLI> ADD DISK DISK220 2 2 0
CLI> ADD DISK DISK320 3 2 0
CLI> ADD RAIDSET RAID9 DISK120 DISK220 DISK320
CLI> INITIALIZE RAID9 CHUNKSIZE=20
CLI> ADD UNIT D301 RAID9
```

To initialize DISK440 and preserve the data after removing it (REDUCE) from a mirrorset:

```
CLI> REDUCE DISK440
CLI> INITIALIZE DISK440 NODESTROY
CLI> ADD UNIT D201 DISK440
```

See also

ADD UNIT

LOCATE

Identifies configured units, storagesets, and devices by flashing the amber device fault LED on the front of the StorageWorks building block (SBB). The device fault LED will flash once per second until turned off with the LOCATE CANCEL command. You also can use the LOCATE command as a lamp test.

Note

The device fault LED flashes at a faster rate on any device that is in the failedset. The different flashing rate should help you distinguish between LOCATEd devices and failed devices. The device fault LED on failed devices will continue to flash after you issue the LOCATE CANCEL command.

Format

```
LOCATE
```

Switches

ALL
Turns on the amber device fault LEDs of all configured devices. This switch also can be used as a lamp test. Use LOCATE CANCEL to turn off the LEDs.

CANCEL
Turns off all amber device fault LEDs that were turned on with the LOCATE command.

DISKS
Turns on the amber device fault LEDs of all configured disk drives. Use LOCATE CANCEL to turn off the LEDs.

UNITS
Turns on the amber device fault LEDs of all devices used by units. This command is useful to determine which devices are not currently configured into logical units. Use LOCATE CANCEL to turn off device the LEDs.

PTL SCSI-location
Turns on the amber device fault LED on the device at the given SCSI location. *SCSI-location* is specified in the form PTL where:

P designates the port (1–6 for 6-port controllers, or 1–3 for 3-port controllers).

T designates the target ID of the device, (0–6 with a single controller, or 0–5 with dual-redundant controllers).

L designates the LUN of the device (must be 0 or 1).

When entering the PTL, you must put at least one space between the port, target, and LUN numbers.

Note

Although you can specify the PTL location or name of any device in the subsystem, not all devices have a device fault LED, and so may not seem to respond to the LOCATE command.

---

container-name
Turns on the amber fault LEDs on the devices that make up *container-name*. If a device name is given, the device's LED is lit. If a storageset name is given, all device LEDs that make up the storageset are lit. Use LOCATE CANCEL to turn off the LEDs.

unit-number
Turns on the amber fault LEDs on the devices that make up *unit-number*. Use LOCATE CANCEL to turn off the LEDs.

Examples

To turn on the device fault LED on device DISK100 and then turn it off:

```
CLI> LOCATE DISK100
CLI> LOCATE CANCEL
```

To turn on the device fault LEDs on all devices that make up disk unit number 102:

```
CLI> LOCATE D102
```

To turn on the device fault LEDs on all configured disk devices:

```
CLI> LOCATE DISKS
```

To turn off the device fault LEDs on all devices:

```
CLI> LOCATE CANCEL
```

MIRROR

Creates a one-member mirrorset from a single disk. Use this command on devices that are already members of higher level containers (stripesets or units). Use the ADD MIRRORSET command to create a mirrorset from devices that are not already members of higher level containers.

After you convert the device to a mirrorset, increase the nominal number of members with the SET *mirrorset-name* MEMBERSHIP=*number-of-members* command. Use the SET *mirrorset-name* REPLACE=*disk-name* command to actually add more members to the mirrorset.

Format

MIRROR *disk-name mirrorset-name*

Parameters

disk-name

Note

If you MIRROR the members of a stripeset, you cannot use the stripeset with any version of controller software prior to Version 2.5.

Specifies the name of the disk to convert to a one-member mirrorset. The disk must be part of a unit.

mirrorset-name
Specifies a name for the mirrorset.

The name must start with a letter (A–Z) and can then consist of up to eight more characters made up of letters A-Z, numbers 0-9, periods (.), dashes (-), or underscores (_), for a total of nine characters.

Switches

COPY=NORMAL (Default)
COPY=FAST

Note

A normal mirrorset member is one whose entire contents are the same as all other normal members.

The COPY switch allows you to set the speed at which the controller copies data from normal mirrorset members to new members.

Select NORMAL to prioritize other controller operations over the copy operation. The controller uses relatively few resources to perform the copy, and there is little impact on performance.

Select FAST when the copy operation must take precedence over other controller operations. The controller uses more resources and the copy takes less time, but overall controller performance is reduced during the copy.

POLICY=BEST_FIT
POLICY=BEST_PERFORMANCE
NOPOLICY (Default)
Specifies the replacement policy to be used when a mirrorset member fails.

Select BEST_FIT to choose a replacement device from the spareset that most closely matches the capacities of the remaining members. If more than one device in the spareset is the correct size, the controller selects the device that gives the best performance.

Select BEST_PERFORMANCE to choose a replacement device from the spareset that results in the best performance (the device should be on a different port than existing members). If more than one device in the spareset has the best performance, the controller selects the device that most closely matches the size of the remaining members.

Select NOPOLICY to prevent the controller from replacing a failed disk drive. This causes the mirrorset to run with less than the nominal number of members until a BEST_FIT or BEST_PERFORMANCE policy is selected, or a member is manually replaced in the mirrorset.

Examples

The commands in this example create a one-member mirrorset from each member of a stripeset. The succeeding commands set the nominal number of members in each mirrorset to two and add a second disk to each mirrorset. Note that you do not need to initialize the mirrorsets or add them as units; the higher-level structure of the stripeset is carried down to the mirrorsets.

```
CLI> ADD DISK DISK110 1 1 0
CLI> ADD DISK DISK210 2 1 0
CLI> ADD DISK DISK310 3 1 0
CLI> ADD STRIPESET STRIPE1 DISK110 DISK210 DISK310
LI> INITIALIZE STRIPE1
CLI> ADD UNIT D102 STRIPE1

CLI> MIRROR DISK110 MIRROR1
CLI> SET MIRROR1 MEMBERSHIP=2
CLI> SET MIRROR1 REPLACE=DISK220
```

```
CLI> MIRROR DISK210 MIRROR2
CLI> SET MIRROR2 MEMBERSHIP=2
CLI> SET MIRROR2 REPLACE=DISK320

CLI> MIRROR DISK310 MIRROR3
CLI> SET MIRROR3 MEMBERSHIP=2
CLI> SET MIRROR3 REPLACE=DISK120
```

See also

ADD MIRRORSET
REDUCE
SHOW MIRRORSETS
UNMIRROR

REDUCE

Note

The nominal number of members in a mirrorset is the number given in the SET *mirrorset-name* MEMBERSHIP=*number-of-members* command. The actual number of members may be less than the nominal number if you do not add devices to the mirrorset or if you remove a member. The actual number of members can never be greater than the nominal number of members.

Removes member disk drives from mirrorsets and decreases the nominal number of members in the mirrorsets. The controller does not put removed members in the failedset, as with the SET *mirrorset-name* REMOVE=*disk-name* command. When using the REDUCE command to split off a copy of a striped mirrorset, you must reduce all mirrorsets at the same time with one command. The CLONE program does this automatically.

The disks to be removed do not need to be members of the same mirrorset. However, the devices *must* be part of the same unit (for example, the same striped mirrorset). The controller pauses I/O to the unit while it removes the specified mirrorset members.

Note

A NORMAL member is a mirrorset member whose entire contents are guaranteed to be the same as all other NORMAL members. Only normal members can be reduced. An error is displayed if you attempt to reduce the only normal mirrorset member.

Note that for each mirrorset that you reduce, there must be at least one remaining NORMAL member after the reduction. If this is not true for all disk-names specified, then none of the mirrorsets are reduced.

Format

```
REDUCE disk-name1 [disk-nameN]
```

Parameters

*disk-name*1
The name of the NORMAL mirrorset member to be removed.

[*disk-name*N]
The name of additional NORMAL mirrorset members to be removed from the same unit.

Examples

To remove DISK210, DISK250, and DISK420 from their respective mirrorsets:

```
CLI> SHOW STRIPE1
Name       Storageset          Uses              Used by
-----------------------------------------------------------
STRIPE1    stripeset           MIRR1
                               MIRR2
                               MIRR3

CLI> SHOW MIRRORSETS
Name       Storageset          Uses               Used by
-----------------------------------------------------------
MIRR1      mirrorset           DISK110            STRIPE1
                               DISK210
MIRR2      mirrorset           DISK120            STRIPE1
                               DISK250
MIRR3      mirrorset           DISK330            STRIPE1
                               DISK420

CLI> REDUCE DISK210 DISK250 DISK420

CLI> SHOW MIRRORSETS
Name       Storageset          Uses               Used by
-----------------------------------------------------------
MIRR1      mirrorset           DISK110            STRIPE1
MIRR2      mirrorset           DISK120            STRIPE1
MIRR3      mirrorset           DISK330            STRIPE1
```

See also

ADD MIRRORSET
MIRROR
RUN CLONE
SHOW MIRRORSET
SET *mirrorset-name*

RENAME

Renames a container.

Format

`RENAME` *`old-name new-name`*

Parameters

old-name
The existing name for the container.

new-name
The new name for the container.

The name must start with a letter (A–Z) and can then consist of up to eight more characters made up of letters A-Z, numbers 0-9, periods (.), dashes (-), or underscores (_), for a total of nine characters.

Examples

To rename DISK0 to DISK100:

```
CLI> SHOW DISKS
Name          Type          Port Targ  Lun          Used by
-----------------------------------------------------------
DISK0         disk             1    0    0            D100
DISK110       disk             1    1    0            D110


CLI> RENAME DISK0 DISK100


CLI> SHOW DISKS
Name          Type          Port Targ  Lun          Used by
-----------------------------------------------------------
DISK100       disk             1    0    0            D100
DISK110       disk             1    1    0            D110
```

RESTART THIS_CONTROLLER and RESTART OTHER_CONTROLLER

Flushes all user data from the specified controller's write-back cache (if present) and restarts the controller. THIS CONTROLLER is the controller that is running the current CLI session. The OTHER_CONTROLLER is the controller that is *not* running the current CLI session.

Format

```
RESTART OTHER_CONTROLLER
RESTART THIS_CONTROLLER
```

Switches

Caution

The IGNORE_ERRORS and IMMEDIATE switches cause the controller to keep unflushed data in the write-back cache until it restarts and is able to write the data to devices. Do not perform any hardware changes and do not let the cache batteries run down before the controller can flush the cache.

IGNORE_ERRORS
NOIGNORE_ERRORS (Default)
The IGNORE_ERRORS and NOIGNORE_ERRORS switches instruct the the controller how to respond to write-back cache errors.

Caution

The IGNORE_ERRORS switch may cause data to remain in write-back cache.

Specify IGNORE_ERRORS to instruct the controller to ignore any write-back cache errors. Such errors can result from data contained within write-back cache that cannot be written to the devices or lost data errors.

Specify NOIGNORE_ERRORS to instruct the controller not to run the self-test program if write-back cache errors are detected.

IMMEDIATE_SHUTDOWN
NOIMMEDIATE_SHUTDOWN
The IMMEDIATE_SHUTDOWN and NOIMMEDIATE_SHUTDOWN switches are used to instruct the controller when to shutdown when running the self-test program.

Caution

The IMMEDIATE_SHUTDOWN switch instructs the controller to immediately shutdown, without regard to any data contained within write-back cache.

Select IMMEDIATE_SHUTDOWN to instruct the controller to run the self-test program immediately without checking for online devices or without flushing user data from write-back cache to devices.

Select NOIMMEDIATE_SHUTDOWN to instruct the controller to flush data from write-back cache before running the self-test program.

Examples

To restart THIS_CONTROLLER as long as it does not have any units online:

```
CLI> RESTART THIS_CONTROLLER
```

To restart the OTHER_CONTROLLER even if there are units online to the OTHER_CONTROLLER:

```
CLI> RESTART OTHER_CONTROLLER OVERRIDE_ONLINE
```

See also

SELFTEST OTHER_CONTROLLER
SELFTEST THIS_CONTROLLER
SHUTDOWN OTHER_CONTROLLER
SHUTDOWN THIS_CONTROLLER

RETRY_ERRORS UNWRITEABLE_DATA

If a container fails in a way that data in the write-back cache cannot be written to the container, an unwriteable data error is reported. If possible, correct the condition that is causing the unwriteable data and try the write operation again. The RETRY_ERRORS UNWRITEABLE_DATA command causes the controller to try to write the unwriteable data. No data is lost if the retry fails.

Format

`RETRY_ERRORS UNWRITEABLE_DATA` *`unit-number`*

Parameters

unit-number
Specifies the logical unit number for which to try the write operation. The *unit-number* is the name given the unit when it was created using the ADD UNIT command.

Examples

To attempt to write the cached data that was previously marked unwriteable to disk unit D103:

`CLI>` **`RETRY_ERRORS UNWRITEABLE_DATA D103`**

See also

CLEAR_ERRORS UNWRITEABLE_DATA

RUN

Starts a diagnostic or utility program on THIS_CONTROLLER. You can run diagnostics and utilities *only* on THIS_CONTROLLER.

Format

`RUN` *`program-name`*

Parameters

program-name
The name of the diagnostic or utility to be run. DILX and CLONE are examples of utilities and diagnostics that can be run from the CLI.

Examples

To run the DILX diagnostic:

```
CLI> RUN DILX
Disk Inline Exerciser - version 2.0 . . .
```

See also

DIRECTORY

SELFTEST OTHER_CONTROLLER and SELFTEST THIS_CONTROLLER

Flushes all user data from the specified controller's write-back cache (if present) and shuts down the controller, then restarts it in self-test mode. You must press the controller reset (//) button to take the controller out of self-test mode.

THIS_CONTROLLER is the controller that is running the current CLI session. The OTHER_CONTROLLER is the controller that is *not* running the current CLI session.

Format

```
SELFTEST OTHER_CONTROLLER
SELFTEST THIS_CONTROLLER
```

Switches

Caution

The IGNORE_ERRORS and IMMEDIATE_SHUTDOWN switches cause the controller to keep unflushed data in the write-back cache until it restarts and is able to write the data to devices. Do not perform any hardware changes and do not let the cache batteries run down before the controller can flush the cache.

IGNORE_ERRORS
NOIGNORE_ERRORS (Default)
The IGNORE_ERRORS and NOIGNORE_ERRORS switches instruct the the controller how to respond to write-back cache errors.

Caution

The IGNORE_ERRORS switch may cause data to remain in write-back cache.

Specify IGNORE_ERRORS to instruct the controller to ignore any write-back cache errors. Such errors can result from data contained within write-back cache that cannot be written to the devices or lost data errors.

Specify NOIGNORE_ERRORS to instruct the controller not to run the self-test program if write-back cache errors are detected.

IMMEDIATE_SHUTDOWN
NOIMMEDIATE_SHUTDOWN
The IMMEDIATE_SHUTDOWN and NOIMMEDIATE_SHUTDOWN switches are used to instruct the controller when to shutdown when running the self-test program.

Caution

The IMMEDIATE_SHUTDOWN switch instructs the controller to immediately shutdown, without regard to any data contained within write-back cache.

Select IMMEDIATE_SHUTDOWN to instruct the controller to run the self-test program immediately without checking for online devices or without flushing user data from write-back cache to devices.

Select NOIMMEDIATE_SHUTDOWN to instruct the controller to flush data from write-back cache before running the self-test program.

Examples

To start the self-test on THIS_CONTROLLER, as long as it does not have any units online:

```
CLI> SELFTEST THIS_CONTROLLER
```

To start the self-test on the OTHER_CONTROLLER even if the controller cannot flush all data from the write-back cache:

```
CLI> SELFTEST OTHER_CONTROLLER IGNORE_ERRORS
```

See also

RESTART OTHER_CONTROLLER
RESTART THIS_CONTROLLER
SHUTDOWN OTHER_CONTROLLER
SHUTDOWN THIS_CONTROLLER

SET *device-name*

Changes the transportable characteristics of a disk drive, or sets the maximum data transfer rate between the controller and a device.

Format

`SET` *`device-name`*

Parameters

device-name
The name of the device to change.

Switches

TRANSFER_RATE_REQUESTED=ASYNCHRONOUS
TRANSFER_RATE_REQUESTED=DEFAULT (Default: 10MHZ)
TRANSFER_RATE_REQUESTED=10MHZ
TRANSFER_RATE_REQUESTED=5MHZ
Specifies the maximum data transfer rate for the controllers to use in communicating with the device. You may need to limit the transfer rate to accommodate long cables between the controllers and the device, such as with a tape library.

TRANSPORTABLE
NOTRANSPORTABLE (Default)
Specify the TRANSPORTABLE switch for any disk drive that you want to move to and from StorageWorks environments with the data intact.

Note

Use the TRANSPORTABLE switch only for disk drives that must be moved to StorageWorks environments with data intact.

A TRANSPORTABLE disk drive does not contain any special information space and can be moved to a non-StorageWorks environment with its data intact. You cannot use a TRANSPORTABLE disk drive in any type of storageset, such as a stripeset or RAIDset.

Specify NOTRANSPORTABLE for all disk drives that will be used in RAIDsets, stripesets, and mirrorsets. Also use the NOTRANSPORTABLE switch for all single-disk units that will be used exclusively in a StorageWorks or HSC™ controller.

The controller makes a small portion of nontransportable disk drives inaccessible to the host and uses the space to store information (metadata) that improves data reliability, error detection, and recovery. Because of this metadata, only StorageWorks controllers can retrieve user data from nontransportable disk drives.

Examples

To set DISK130 to transportable:

```
CLI> SET DISK130 TRANSPORTABLE
```

To limit the data transfer rate to 5MHz for LOADER200:

```
CLI> SET LOADER200 TRANSFER_RATE_REQUESTED=5MHZ
```

See also

ADD DISK
SHOW DISKS

SET FAILEDSET

Changes the automatic spare policy for the failedset.

Format

```
SET FAILEDSET
```

Switches

AUTOSPARE
NOAUTOSPARE (Default)
The AUTOSPARE switch allows you to enable automatic moving of replaced disk drives from the failedset to the spareset. When a disk drive in the failedset is replaced with a factory-fresh drive and this feature is enabled, the controller automatically removes that disk drive from the failedset and adds it to the spareset, where it is available as a hot spare for RAIDsets and mirrorsets.

Specify AUTOSPARE to enable the feature. Specify NOAUTOSPARE to disable the feature.

Examples

To enable the automatic spare feature:

CLI> **SET FAILEDSET AUTOSPARE**

See also

SHOW FAILEDSET

SET FAILOVER

Note

Both controllers must be in NOFAILOVER mode before you use the SET FAILOVER command.

Places THIS_CONTROLLER and the OTHER_CONTROLLER into a dual-redundant (failover) configuration. This allows both controllers to access the storage devices, providing higher throughput. If one of the two controllers fails, the devices and cache (if any) attached to the failed controller become available to and accessible through the operating controller.

Format

`SET FAILOVER COPY=`*`configuration-source`*

Parameters

COPY=THIS_CONTROLLER
COPY=OTHER_CONTROLLER

Caution

The device configuration information from the specified controller will overwrite the information on the target controller. Make sure you know which controller has the good configuration information *before* entering this command.

Specifies the controller that contains the source configuration for the copy. The other controller will restart when the command is completed.

Specify THIS_CONTROLLER to copy all the device configuration information from the controller running the CLI session to the other controller.

Specify OTHER_CONTROLLER to copy all the device configuration information from the controller that is *not* running the current CLI session to THIS_CONTROLLER (the controller that *is* running the current CLI session).

Due to the amount of information that must be passed between the two controllers, this command may take up to 1 minute to complete.

Examples

To place the controllers into a dual-redundant configuration and copy configuration information from the controller running the current CLI session to the other controller:

```
CLI> SET FAILOVER COPY=THIS_CONTROLLER
```

See also

SET MULTIBUS_FAILOVER
SET NOFAILOVER

SET *mirrorset-name*

Changes the characteristics of a mirrorset, including adding and removing members.

Format

```
SET mirrorset-name
```

Parameters

mirrorset-name
The name of the mirrorset that will have its characteristics modified. This is the name given to the mirrorset when it was created with the ADD MIRRORSET command.

Switches

MEMBERSHIP=number-of-members
Sets the nominal number of members of a mirrorset to the number specified by *number-of-members* (1 through 6).

If you increase the number of members and autosparing is turned on (by specifying POLICY=BEST_FIT or POLICY=BEST_PERFORMANCE), the controller automatically adds disk drives from the spareset to the mirrorset until either the new number of members is reached, or there are no more suitable disk drives in the spareset.

If you increase the number of members and autosparing is turned off (by specifying NOPOLICY), you must use the REPLACE switch to bring the mirrorset up to the new nominal number of members.

You cannot set the nominal number of members to be lower than the actual number of members. Use the REMOVE switch to reduce the number of devices that are part of the mirrorset.

REPLACE=disk-name

Note

You cannot specify any other switches to the SET *mirrorset -name* command when you use the REPLACE switch.

Adds a disk member into an existing mirrorset provided the following conditions are met:

- The replacement policy is set to NOPOLICY.
- The nominal number of members is greater than the actual number of members.
- The disk member to be replaced is as large as the existing members of the mirrorset.

If these conditions are met, the device specified by *disk-name* is added to the mirrorset specified by *mirrorset-name*. The nominal number of members does not change.

REMOVE=disk-name

Note

> You cannot specify any other switches to the SET *mirrorset -name* command when you use the REMOVE switch.

Removes members from an existing mirrorset. The disk drive specified by *disk-name* is removed from the mirrorset specified by *mirrorset-name*. The removed disk drive is added to the failedset and a new member is autospared into the mirrorset (if applicable).

Note

> A NORMAL mirrorset member's entire contents are the same as all other NORMAL members.

If the mirrorset will not have a remaining NORMAL or NORMALIZING member after the disk drive is removed, an error is reported and no action is taken.

(NORMALIZING members only exist when a mirrorset is first created. One member is identified as NORMAL, and all other *original* mirrorset members are marked as NORMALIZING. All new data that is written to the mirrorset is written to all members. All data on the NORMAL member that existed before the mirrorset was created is copied to the NORMALIZING members. When all the blocks on the members are the same, the NORMALIZING members are marked as NORMAL. Members may also be marked as NORMALIZING if cache data is lost.)

Note that the REMOVE switch does not change the nominal number of members in the mirrorset. If the mirrorset has a replacement policy and there are acceptable disk drives in the spareset, the controller adds disk drives from the spareset to the mirrorset to make the actual number of members equal the nominal number of members.

POLICY=BEST_FIT
POLICY=BEST_PERFORMANCE (Default)
NOPOLICY
The SET *mirrorset-name* POLICY=*policy-type* command specifies the replacement policy to be used when a mirrorset member within the mirrorset fails.

Select BEST_FIT to choose a replacement device from the spareset that most closely matches the capacities of the remaining members. If more than one device in the spareset is the correct size, the controller selects the device that gives the best performance.

Select BEST_PERFORMANCE to choose a replacement device from the spareset that results in the best performance (the device should be on a different port than existing members). If more than one device in the spareset has the best performance, the controller selects the device that most closely matches the size of the remaining members.

NOPOLICY retires a failing device from the mirrorset without selecting a replacement. This causes the mirrorset to run with less than the nominal number of members until a BEST_FIT or BEST_PERFORMANCE policy is selected, or a member is manually replaced in the mirrorset.

READ_SOURCE=ROUND_ROBIN
READ_SOURCE=LEAST_BUSY (Default)
READ_SOURCE=disk-name
The READ_SOURCE switch allows you to control which mirrorset member is used by the controller to satisfy a read request.

Select ROUND_ROBIN to cause the controller to direct read requests to each NORMAL mirrorset member in sequential membership order. No preference is given to any NORMAL member.

Select LEAST_BUSY to cause the controller to direct read requests to the NORMAL mirrorset member with the least busy work queue.

Select the disk-name of a specific member to cause the controller to direct all read requests to that member. If that member fails out of the mirrorset, the controller reverts to the LEAST_BUSY method for mirrorset read requests.

COPY=NORMAL (Default)
COPY=FAST

Note

A normal mirrorset member is one whose entire contents are the same as all other normal members.

The COPY switch allows you to set the speed at which the controller copies data from normal mirrorset members to new members.

Select NORMAL to prioritize other controller operations over the copy operation. The controller uses relatively few resources to perform the copy, and there is little impact on performance.

Select FAST when the copy operation must take precedence over other controller operations. The controller uses more resources and the copy takes less time, but overall controller performance is reduced during the copy.

Examples

To change the replacement policy of mirrorset MIRR1 to BEST_FIT:

```
CLI> SET MIRR1 POLICY=BEST_FIT
```

The following example removes member DISK300 from mirrorset MIRR1 that was created above. If the mirrorset has a replacement policy and an acceptable disk drive is in the spareset, the controller automatically adds the spare disk drive to the mirrorset.

```
CLI> SET MIRR1 REMOVE=DISK300
```

To add disk drive DISK320 to the mirrorset MIRR1:

```
CLI> SET MIRR1 REPLACE=DISK320
```

A copy operation begins immediately on DISK320.

See also

ADD MIRRORSET
MIRROR
REDUCE
SHOW MIRRORSET
UNMIRROR

SET MULTIBUS_FAILOVER

Note

Both controllers must be in NOFAILOVER mode before you use the SET MULTIBUS_FAILOVER command.

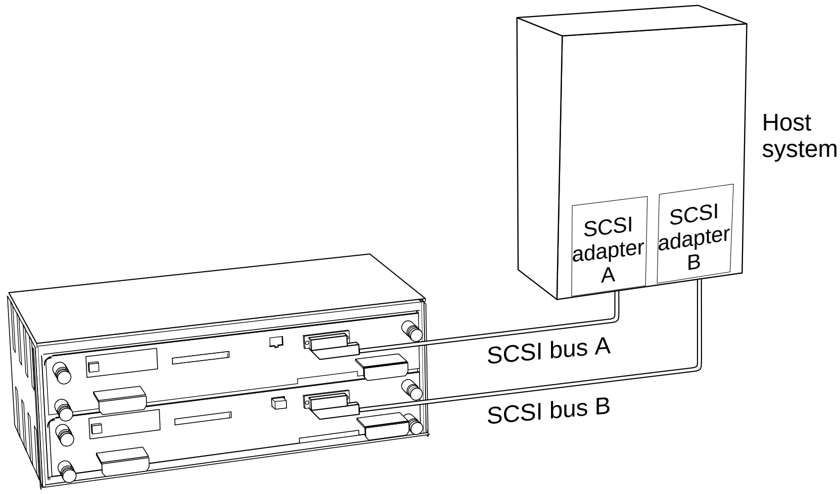
Places THIS_CONTROLLER and the OTHER_CONTROLLER into a dual-redundant (failover) configuration in a multiple bus environment. This allows both controllers to access the storage devices and provide greater throughput. If one of the controllers fails, the devices and cache (if any) attached to the failed controller become available to and accessible through the remaining operational controller.

Note

Partitioned storagesets and single-disk units cannot function in multiple bus failover. You will have to delete your partitions before setting this switch.

This command is intended for use with host systems that have two SCSI adapters, each connected to one controller in the dual-redundant pair, shown in Figure 2–1. The host operating system must be configured to recognize that all units can be accessed through either adapter/controller. If your host operating system does not support such a configuration, use the SET FAILOVER command instead.

Figure 2–1 Controller Connections in Multiple Bus Failover Configuration



Format

```
SET MULTIBUS_FAILOVER COPY=configuration-source
```

Parameters

Caution

The device configuration information from the specified controller will overwrite the information on the target controller. **Make sure you know which controller has the good configuration information *before* entering this command**.

COPY=THIS_CONTROLLER
COPY=OTHER_CONTROLLER
Specifies the controller that contains the source configuration for the copy. The other controller will restart when the command is completed.

Specify THIS_CONTROLLER to copy all the device configuration information from the controller running the CLI session to the OTHER_CONTROLLER.

Specify OTHER_CONTROLLER to copy all the device configuration information from the controller that is *not* running the current CLI session to THIS_CONTROLLER (the controller that *is* running the current CLI session).

Due to the amount of information that must be passed between the two controllers, this command may take up to one minute to complete.

Example

To configure two controllers as dual-redundant in a multiple bus environment:

```
CLI> SET THIS_CONTROLLER ID=(0,1,2,3)
CLI> RESTART THIS_CONTROLLER
CLI> SET MULTIBUS_FAILOVER COPY=THIS_CONTROLLER
```

The configuration on THIS_CONTROLLER is automatically copied to the OTHER_CONTROLLER when you issue the SET MULTIBUS_FAILOVER COPY command. If you want to prefer specific units to specific controllers, use the following command after setting multiple bus failover command:

```
CLI> SET D100 PREFERRED=THIS_CONTROLLER
CLI> SET D101 PREFERRED=OTHER_CONTROLLER
```

See also

SET FAILOVER
SET NOFAILOVER

SET NOFAILOVER (SET NOMULTIBUS_FAILOVER)

Takes THIS_CONTROLLER and OTHER_CONTROLLER (if reachable) out of dual-redundant (failover) mode. Immediately after entering this command, you should remove one controller from the shelf because the sharing of devices is not supported by nonredundant controllers.

The controller on which you enter the command changes to nonfailover mode, even if the other controller is not reachable. All units that were accessed through OTHER_CONTROLLER failover to THIS_CONTROLLER. No configuration information is lost when changing modes.

Format

```
SET NOFAILOVER
```

Switches

NODESTROY_UNFLUSHABLE_DATA (Default)
DESTROY_UNFLUSHABLE_DATA

Note

This switch has no effect if both controllers are operational.

If one of your dual-redundant controllers fails before you can properly shut it down with the SET NOFAILOVER or SHUTDOWN commands, you will have to use this switch.

Under some circumstances, the data in a failed controller's write-back cache may not failover to the operational controller's write-back cache. For example, if the cache battery has failed on the operational controller, failover will not jeopardize the write-cache data of the failed controller by trying to move it to the write-back cache associated with the operational controller. This makes the write-cache data associated with the failed controller unflushable.

The SET NOFAILOVER NODESTROY_UNFLUSHABLE_DATA leaves the unwritten data intact in the failed controller's write-back cache. When you replace the failed controller, it will flush the write-cache data to the appropriate devices when the controller comes online.

Caution

You will not be able to recover or obtain unwritten data once it has been destroyed.

Use the SET NOFAILOVER DESTROY_UNFLUSHABLE_DATA if you want to reconfigure the operational controller before you replace the failed controller. The unwritten data of the failed controller may reference devices that may not be present in your new configuration. If you do not destroy the old configuration data, it could conflict with your new configuration and cause your subsystem to behave unpredictably.

Examples

To remove the two controllers from a dual-redundant configuration:

```
CLI> SET NOFAILOVER
```

See also

SET FAILOVER

SET THIS_CONTROLLER and SET OTHER_CONTROLLER

Changes parameters on the specified controller. THIS_CONTROLLER is the controller that is running the current CLI session. The OTHER_CONTROLLER is the controller that is *not* running the current CLI session.

Format

```
SET THIS_CONTROLLER
SET OTHER_CONTROLLER
```

Switches

Table 2–3 lists all switches for the SET THIS_CONTROLLER and SET OTHER_CONTROLLER commands and shows which switches can be used with each type of controller. Descriptions of the switches follow the table.

Table 2–3 Switches for SET THIS and SET OTHER Command

Switch	**HSZ Values**
CACHE_FLUSH_TIMER	1 t o 65535 sec,default
CACHE_POLICY	a, b
HOST_FUNCTION	A, B, C, D
ID	0-7 (x4)
INITIAL_CONFIGURATION	✓
PREFERRED_ID	0-7 (x4)
PROMPT	1 to 16 characters
TERMINAL_PARITY NOTERMINAL_PARITY	odd, even
TERMINAL_SPEED	4800, 9600, 19200
TIME	dd-mmm-yyy: hh:mm:ss
TRANSFER_RATE_REQUESTED	asynchronous, default, 10mHZ, 5mHZ

CACHE_FLUSH_TIMER=n
CACHE_FLUSH_TIMER=DEFAULT
Specifies how many seconds (1–65535) of idle time may elapse before the write-back cache flushes its entire contents to disk. By default, this timer is set to 10 seconds. You must restart the controller before changes to the CACHE_FLUSH_TIMER take effect.

CACHE_POLICY=A (Default)
CACHE_POLICY=B
Sets the write-back cache battery policy used by the controller. The policy affects the availability of RAIDsets and mirrorsets when the write-back cache battery is low:

- CACHE_POLICY=A makes inoperative all RAIDsets and mirrorsets until the battery has been recharged.
- CACHE_POLICY=B allows write-through access to all RAIDsets and mirrorsets for 10 hours, during which time the battery should be recharging. If the batteries do not recharge within 10 hours, the RAIDsets and mirrorsets become inoperative.

There is some risk in using CACHE_POLICY=B: since the battery’s state is not good, there is no guarantee as to how long they will maintain data in the write cache module should the power fail.

Changing this policy on one controller in a dual-redundant configuration automatically changes it on the other controller. You must restart a controller for a new CACHE_POLICY to take effect.

Regardless of the CACHE_POLICY, a low or bad battery affects other controller operations as follows:

- Stripesets and single-disk unit units with write-back caching enabled are accessed in write-through (read cache) mode, as long as unwritten cache data has not been lost.
- RAIDsets and mirrorsets failover to the companion controller’s cache in a dual-redundant configuration, provided the other cache's batteries are fully charged.

If the batteries are low after controller initialization, unwritten cache data is flushed from the cache and any RAIDset or mirrorset that does not have access to good batteries is made inoperative, regardless of the cache policy.

- If the batteries are bad or missing, RAIDsets and mirrorsets are made inoperative.
- Write-back caching automatically resumes when the cache batteries are fully recharged or replaced.

HOST_FUNCTION=(ID, mode)
Sets the host compatibility mode for a controller target ID. Each controller or pair of dual-redundant controllers can service two host modes simultaneously. You must repeat the

command if you want to set different modes for each target ID, such as to support hosts with different operating systems on a single SCSI bus.

You can specify the following modes:

- A (normal: DIGITALUNIX, OpenVMS, SUN®, and HP-UX®)
- B (IBM® host mode)
- C (HSZ15 compatibility mode)
- D (Windows NT® mode)

ID=n
Specifies from one to four SCSI target IDs (0–7). Multiple target IDs allow the controller to present more units to the host. If you enter two or more target IDs, you must enclose the numbers in parentheses and separate them with commas.

Note

An HSZ controller can present up to eight units to the host with each target ID. Unit numbers identify both the target ID and the unit. For example, D203 is target 2, LUN 3. D5 is target 0, LUN 5.

If two HSZ controllers are in a dual-redundant configuration, both controllers have the same IDs. When you change the IDs on one controller, the other is updated automatically.

INITIAL_CONFIGURATION

Note

You cannot recover the controller configuration after issuing the SET THIS_CONTROLLER INITIAL_CONFIGURATION command. Before issuing the command, be sure that you do not need the information.

Erases the entire configuration on THIS_CONTROLLER and restores the factory default values. Use this command to ensure that all of the old configuration information is removed when moving a controller from one location to another.

You can only use the INITIAL_CONFIGURATION switch on THIS_CONTROLLER.

PREFERRED_ID=(n1[,nN])
NOPREFERRED_ID
In an HSZ dual-redundant configuration, PREFERRED_ID defines which targets (and units) are handled by the specified controller. You must first specify the total set of IDs with the

ID switch. If you enter two or more PREFERRED_IDs, you must enclose the numbers in parentheses and separate them with commas.

For example, if a dual-redundant pair of HSZ controllers is set to IDs 0, 1, and 2, you could set one controller to respond to IDs 0 and 1; the other controller would automatically respond to ID 2. The first controller would present unit numbers 0–7 and 100–107 and the other controller would present unit numbers 200–207.

Except in an HSZ multiple bus failover environment, when you change the PREFERRED_IDs on one controller, the other controller is automatically updated to support the remaining IDs (if any).

Specifying NOPREFERRED_ID for one controller makes it a “hot standby.” The other controller processes all I/O during normal operation. However, if that controller fails, the controller with NOPREFERRED_ID would pick up the targets of the failed controller.

PROMPT="new prompt"
Specifies a 1- to 16-character prompt that is displayed when the controller's CLI prompts for input. Only printable ASCII characters are valid and they must be enclosed in quotes.

When first installed, the CLI prompt is set to the first three letters of the controller's model number (for example, HSZ>).

TERMINAL_PARITY=ODD
TERMINAL_PARITY=EVEN
NOTERMINAL_PARITY (Default)
Specifies the parity transmitted and expected.

TERMINAL_SPEED=baud_rate
Sets the terminal speed to 4800, 9600 (default), or 19200.

TIME=dd-mmm-yyyy:hh:mm:ss
The TIME switch sets the date and time. The time is set on both controllers in a dual-redundant configuration.

TRANSFER_RATE_REQUESTED=ASYNCHRONOUS
TRANSFER_RATE_REQUESTED=DEFAULT (Default: 10MHz)
TRANSFER_RATE_REQUESTED=10MHZ
TRANSFER_RATE_REQUESTED=5MHZ

Note

Set the same transfer rate for both controllers in a dual-redundant configuration.

Specifies the maximum data transfer rate that HSZ controllers use when communicating with a SCSI host. You may need to limit the transfer rate to accommodate long cables between the subsystem and the host system.

Table 2–4 lists the maximum transfer rates for different lengths of fast and slow SCSI buses. These lengths represent cable lengths plus shelf-bus lengths.

Table 2–4 Maximum Data Transfer Rates for Different SCSI-Bus Cable Lengths

Bus type	Bus rate	Data transfer rate	Meters	Feet
8-bit, single ended	5 MHz	5 MB/Sec	6	19.68
8-bit, single ended	10 MHz	10 MB/Sec	3	9.84
16-bit, differential	10 MHz	20 MB/Sec	25	82.02

Examples

To change the other controller's CLI prompt:

```
CLI> SET OTHER_CONTROLLER PROMPT="CONTROLLER B"
```

To set this HSZ controller so it responds to requests for targets 2 and 5:

```
CLI> SET THIS_CONTROLLER ID=(2,5)
```

See also

SHOW THIS_CONTROLLER
SHOW OTHER_CONTROLLER

SET *RAIDset-name*

Changes the characteristics of a RAIDset.

Format

`SET` *`RAIDset-name`*

Parameters

RAIDset-name
The name of the RAIDset that will have its characteristics modified.

Switches

POLICY=BEST_FIT
POLICY=BEST_PERFORMANCE (Default)
NOPOLICY
Specifies the replacement policy to use when a member within the RAIDset fails.

Select BEST_FIT to choose a replacement device from the spareset that most closely matches the capacities of the remaining members. If more than one device in the spareset is the correct size, the controller selects the device that gives the best performance.

Select BEST_PERFORMANCE to choose a replacement device from the spareset that results in the best performance (the device should be on a different port than existing members). If more than one device in the spareset has the best performance, the controller selects the device that most closely matches the size of the remaining members.

Select NOPOLICY to prevent the controller from replacing a failed disk drive. This causes the RAIDset to run in a reduced state until a BEST_FIT or BEST_PERFORMANCE policy is selected, or a member is manually replaced in the RAIDset (see SET *RAIDset-name*).

RECONSTRUCT=NORMAL (Default)
RECONSTRUCT=FAST
The RECONSTRUCT switch allows you to set the speed at which the controller reconstructs the data on a new RAIDset member that has replaced a failed member.

Select NORMAL to balance other controller operations against the reconstruct operation. The controller uses relatively few resources to perform the reconstruct, and there is little impact on performance.

Select FAST when the reconstruct operation must take precedence over other controller operations. The controller uses more resources and the reconstruct takes less time, but overall controller performance is reduced during the reconstruct.

REMOVE=disk-name

Note

You cannot specify any other switches to the SET *RAIDset-name* command when you use the REMOVE switch.

Removes a disk member from a RAIDset and adds it to the failedset. If the RAIDset is already in a reduced state, an error is displayed and the command is not carried out. If a replacement policy is specified, a replacement is taken from the spareset to replace the removed member. If NOPOLICY is specified, the RAIDset continues to operate in a reduced state until a replacement is manually specified (see the REPLACE switch) or a policy is specified (see the POLICY switch).

REPLACE=disk-name

Note

You cannot specify any other switches to the SET *RAIDset-name* command when you use the REPLACE switch.

Adds a disk drive into a reduced RAIDset. A reconstruct operation begins immediately on the new drive. The RAIDset must be reduced (missing one member) and have NOPOLICY set. The disk drive must not be in any configuration, including the spareset.

Examples

To change the replacement policy for RAIDset RAID9 to BEST_FIT:

```
CLI> SET RAID9 POLICY=BEST_FIT
```

To remove member DISK100 from the RAID9 RAIDset:

```
CLI> SET RAID9 REMOVE=DISK100
```

If there is a replacement policy, the controller moves a disk from the spareset to the RAIDset automatically.

To add disk DISK210 to the reduced RAIDset, RAID9:

```
CLI> SET RAID9 REPLACE=DISK210
```

A reconstruct operation begins immediately on DISK210.

See also

ADD RAIDSET
SHOW RAIDSETS

SET *unit-number*

Changes the characteristics of a unit.

Format

`SET` *`unit-number`*

Parameters

unit-number
Specifies the logical unit number to modify. The *unit-number* is the name given the unit when it was created using the ADD UNIT command.

Switches

Table 2–5 lists all switches for the SET *unit-number* command and shows which switches can be used with each type of device and storageset. Descriptions of the switches follow the table.

Table 2–5 Switches for the SET unit-number command

Switch	RAID-set	Stripe-set	Mirror-set	NoTrans Disk	Trans Disk	CD–ROM	Pass-through
MAXIMUM_CACHED_ TRANSFER	✓	✓	✓	✓	✓	✓	
PREFERRED_PATH NOPREFERRED_PATH (Default)	✓	✓	✓	✓	✓	✓	✓
READ_CACHE (Default) NOREAD_CACHE	✓	✓	✓	✓	✓	✓	
RUN (Default) NORUN	✓	✓	✓	✓	✓	✓	
WRITE_PROTECT NOWRITE_PROTECT (Default)	✓	✓	✓	✓	✓		
WRITEBACK_CACHE NOWRITEBACK_CACHE (Default)	✓	✓	✓	✓			

Note

The RUN and NORUN switches cannot be specified for partitioned units.

MAXIMUM_CACHED_TRANSFER=n
MAXIMUM_CACHED_TRANSFER=32 (Default)
The MAXIMUM_CACHED_TRANSFER switch allows you to set the largest transfer (in blocks) to be cached by the controller. The controller will not cache any transfers over this size. Valid values are 1–1024.

PREFERRED_PATH=THIS_CONTROLLER
PREFERRED_PATH=OTHER_CONTROLLER
NOPREFERRED_PATH (Default)

Note

All partitions on a container must be addressed through the same controller. If you set PREFERRED_PATH for one partition, all partitions on that container will inherit the same path.

The PREFERRED_PATH switch allows you to balance I/O load by specifying the controller through which the unit will be accessed. If you set NOPREFERRED_PATH for a unit, it can be accessed through either controller.

The controllers only use the PREFERRED_PATH setting if they are in a dual-redundant configuration. If one controller fails, all the devices are accessed through the remaining controller, ignoring the PREFERRED_PATH setting.

Note

The PREFERRED_PATH and NOPREFERRED_PATH switches are valid only for HSZ controllers in dual-redundant multiple bus-failover configurations. The preferred path for units on dual-redundant HSZ controllers is determined by the first digit in the unit number (the controller target ID) specified in the ADD UNIT command.

When the failed controller is restarted, the drives automatically return to the controller specified by the PREFERRED_PATH switch.

You can specify the PREFERRED_PATH switch on a single controller; however, the switch will not take effect until you add a second controller and configure the two controllers for dual-redundancy.

READ_CACHE (Default)
NOREAD_CACHE
The READ_CACHE switch allows you to enable or disable the use of read cache with a unit.

Read caching improves performance in almost all situations, so it is generally good to leave it enabled. However, under certain workloads, like a backup, there may be a low probability for a cache hit and it may be beneficial to turn read caching off to remove the overhead of caching.

RUN (Default)
NORUN
The RUN switch allows you to enable and disable a unit's availability to the host.

Select NORUN to make a unit unavailable to the host and to cause any user data for that unit to be flushed from the write-back cache to the disk drives. The devices that make up the unit are still spun up.

Select RUN to make a unit available to the host.

Note

The RUN and NORUN switches cannot be specified for partitioned units.

WRITE_PROTECT
NOWRITE_PROTECT (Default)

Note

Write protect disables the writing of any new data by the host. However, the controller may still write to a write-protected RAIDset to satisfy a reconstruct pass or to reconstruct a newly replaced member.

The WRITE_PROTECT switch allows you to ensure that data on a unit is not overwritten.

Select WRITE_PROTECT to prevent write operations to the unit.

Select NOWRITE_PROTECT to allow both reads and writes to the unit.

WRITEBACK_CACHE
NOWRITEBACK_CACHE (Default)

Note

WRITEBACK_CACHE requires that write-back cache modules be installed in the subsystem and that the licensed feature be enabled through FLS.

The WRITEBACK_CACHE switch allows you to enable or disable the use of write-back cache with a unit.

Select WRITEBACK_CACHE for all new RAIDsets and mirrorsets, and for all other units for which you want this feature. The write-back cache improves performance for write operations to the unit.

Note

It may take up to 5 minutes for the controller to flush unwritten data from the cache once you disable write-back caching.

Select NOWRITEBACK_CACHE for those units for which you want all writes to go directly to devices without being cached.

Considerations When Using Write-back Caching
The following list summarizes considerations you must be aware of when using write-back cache.

Two conditions will cause data contained within write-back cache to be lost: if power from the main power supply and the external cache battery is interrupted, or if the cache module is removed before the controller flushes the data from the write-back cache.

- When restarted, the controller attempts to flush any unflushed data within write-back cache to the devices. However, by specifying the IGNORE_ERRORS or IMMEDIATE_SHUTDOWN switch, you allow data to reside in write-back cache when the controller is turned off, regardless of any errors detected.
- RAIDsets and mirrorsets require data to be stored in write-back cache to accomodate the write hole and to increase performance—without regard to the WRITEBACK_CACHE switch setting.

If data is contained within the write-back cache while the subsystem is shut down, do not perform any hardware changes until after the controller flushes the data to the devices.

When restarted, the controller attempts to flush any unflushed data within the write-back cache to the devices. However, by specifying the IGNORE_ERRORS or IMMEDIATE_SHUTDOWN switch, you allow data to reside in write-back cache when the controller is turned off, regardless of any errors detected.

Examples

To enable write protect and turn off the read cache on unit D102:

```
CLI> SET D102 WRITE_PROTECT NOREAD_CACHE
```

To set unit T47 to 1600 bpi:

```
CLI> SET T47 DEFAULT_FORMAT=1600BPI_9TRACK
```

See also

SHOW UNITS
SHOW *unit-number*

SHOW

Displays information about controllers, storagesets, devices, and partitions.

Note

The SHOW command may not display some information for devices accessed through the companion controller in a dual-redundant configuration. When device or parameter information does not appear, enter the same SHOW command from the other controller.

Format

```
SHOW device-type
SHOW device-name
SHOW storageset-type
SHOW storageset-name
SHOW UNITS
SHOW unit-name
SHOW THIS_CONTROLLER
SHOW OTHER_CONTROLLER
```

Parameters

device-type
Specifies the type of devices that you want to display:

DEVICES—All devices attached to the controller

CDROMS—All CD–ROM devices attached to the controller

DISKS—All disk drive devices attached to the controller

PASSTHROUGH—All passthrough containers attached to the controller

device-name
Specifies the name of a particular device that you want to display. For example, SHOW DISK210 displays information about the device named DISK210.

storageset-type
Specifies the type of storageset that you want to display:

STORAGESETS—all storagesets configured in the controller. This includes stripesets, mirrorsets, RAIDsets, the spareset and the failedset.

RAIDSETS—all RAIDsets configured in the controller

STRIPESETS—all stripesets configured in the controller

MIRRORSETS—all mirrorsets configured in the controller

SPARESET—the pool of disk drives available to replace failing members of RAIDsets and mirrorsets

FAILEDSET—all disk drives that have failed out of RAIDsets and mirrorsets

storageset-name
Specifies the name of a particular storageset that you want to display. For example, SHOW STRIPE1 displays information about the stripeset named STRIPE1.

units
Displays information for all units configured to the controller.

unit-name
Specifies the name of a particular unit that you want to display. For example, SHOW D102 displays information about the unit named D102.

THIS_CONTROLLER
OTHER_CONTROLLER
Specifies the controller that you want to display.

Switches

FULL
Displays additional information about each device or controller.

Examples

To show a basic listing of CDROMs:

```
CLI> SHOW CDROM
Name          Type        Port Targ  Lun             Used by
-----------------------------------------------------------
CDROM230      cdrom          2    3    0             D623
CDROM240      cdrom          2    4    0             D624
```

To show a full listing of devices attached to the controller:

```
CLI> SHOW DEVICES FULL
Name          Type                      Port Targ  Lun       Used by
---------------------------------------------------------------------------

DISK100       disk                        1    0    0        D500
          DEC      RZ28     (C) DEC 435E
        Switches:
          NOTRANSPORTABLE
          TRANSFER_RATE_REQUESTED = 10MHZ (synchronous 10 MB/SEC
negotiated)
        Size: 4109470 blocks

DISK120       disk                        1    2    0        S0
          DEC      RZ28     (C) DEC 435E
        Switches:
          NOTRANSPORTABLE
          TRANSFER_RATE_REQUESTED = 10MHZ (synchronous 10 MB/SEC
negotiated)
        Size: 4109470 blocks

DISK140       disk                        1    4    0        S0
          DEC      RZ28M    (C) DEC 0616
        Switches:
          NOTRANSPORTABLE
          TRANSFER_RATE_REQUESTED = 10MHZ (synchronous 10 MB/SEC
negotiated)
        Size: 4109470 blocks

DISK200       disk                        2    0    0        D501
          DEC      RZ28     (C) DEC 435E
        Switches:
          NOTRANSPORTABLE
          TRANSFER_RATE_REQUESTED = 10MHZ (synchronous 10 MB/SEC
negotiated)
        Size: 4109470 blocks

DISK220       disk                        2    2    0        S0
          DEC      RZ28B    (C) DEC 0006
        Switches:
          NOTRANSPORTABLE
          TRANSFER_RATE_REQUESTED = 10MHZ (synchronous 10 MB/SEC
negotiated)
        Size: 4109470 blocks
```

To show a complete listing of the mirrorset named MIRR1:

```
CLI> SHOW MIRR1
Name          Storageset             Uses          Used by
------------------------------------------------------------
```

```
MIRR1          mirrorset              DISK130       D202
        Switches:
          NOPOLICY (for replacement)
          COPY (priority) = NORMAL
          READ_SOURCE = LEAST_BUSY
          MEMBERSHIP = 2, 1 member present
        State:
          DISK130   (member  0) is NORMAL
        Size: 4109470 blocks
```

To show the full information for a controller:

```
CLI> SHOW THIS_CONTROLLER FULL
```

```
Controller:
     HSZ40-CX (C) ZG60300054 Firmware V31Z-0, Hardware 0000
     Not configured for dual-redundancy
     SCSI address 6
     Time: NOT SET
Host port:
     SCSI target(s) (4, 5), Preferred target(s) (5)
     TRANSFER_RATE_REQUESTED = ASYNCHRONOUS
Cache:
     No cache
     Host Functionality Mode = A
Licensing information:
     RAID (RAID Option) is ENABLED, license key is VALID
     WBCA (Writeback Cache Option) is ENABLED, license key is VALID
     MIRR (Disk Mirroring Option) is ENABLED, license key is VALID
Extended information:
        Terminal speed 19200 baud, eight bit, no parity, 1 stop bit
        Operation control: 00000005  Security state code: 24968
        Configuration backup disabled
```

SHUTDOWN THIS_CONTROLLER and SHUTDOWN OTHER_CONTROLLER

Flushes all user data from the specified controller's write-back cache (if present) and shuts down the controller. The controller does not restart. All units accessed through the controller failover to the surviving controller.

THIS CONTROLLER is the controller that is running the current CLI session.
OTHER_CONTROLLER is the controller that is *not* running the current CLI session.

Format

```
SHUTDOWN OTHER_CONTROLLER
SHUTDOWN THIS_CONTROLLER
```

Switches

Caution

The IGNORE_ERRORS and IMMEDIATE switches cause the controller to keep unflushed data in the write-back cache until it restarts and is able to write the data to devices. Do not perform any hardware changes and do not let the cache batteries run down before the controller can flush the cache.

IGNORE_ERRORS
NOIGNORE_ERRORS (Default)
The IGNORE_ERRORS and NOIGNORE_ERRORS switches instruct the the controller how to respond to write-back cache errors.

Caution

The IGNORE_ERRORS switch may cause data to remain in write-back cache.

Specify IGNORE_ERRORS to instruct the controller to ignore any write-back cache errors. Such errors can result from data contained within write-back cache that cannot be written to the devices or lost data errors.

Specify NOIGNORE_ERRORS to instruct the controller not to run the self-test program if write-back cache errors are detected.

IMMEDIATE_SHUTDOWN

NOIMMEDIATE_SHUTDOWN

The IMMEDIATE_SHUTDOWN and NOIMMEDIATE_SHUTDOWN switches are used to instruct the controller when to shutdown when running the self-test program.

Caution

The IMMEDIATE_SHUTDOWN switch instructs the controller to immediately shutdown, without regard to any data contained within write-back cache.

Select IMMEDIATE_SHUTDOWN to instruct the controller to run the self-test program immediately without checking for online devices or without flushing user data from write-back cache to devices.

Select NOIMMEDIATE_SHUTDOWN to instruct the controller to flush data from write-back cache before running the self-test program.

Examples

To shut down THIS_CONTROLLER as long as there are no units online:

```
CLI> SHUTDOWN THIS_CONTROLLER
```

To shut down the other controller even if it cannot write all write-back cache data to units:

```
CLI> SHUTDOWN OTHER_CONTROLLER IGNORE_ERRORS
```

See also

RESTART THIS_CONTROLLER
RESTART OTHER_CONTROLLER
SELFTEST THIS_CONTROLLER
SELFTEST OTHER_CONTROLLER

UNMIRROR

Converts a one-member mirrorset back to a single device and deletes the mirrorset from the list of known mirrorsets. You can use this command on mirrorsets that are already members of higher level containers (stripesets or units).

The UNMIRROR command is not valid for devices that have a capacity greater than the capacity of the mirrorset. If a mirrorset is comprised of devices of different capacities, the mirrorset capacity is limited to the size of the smallest member; larger members will have unused capacity. If a member with unused capacity is the last remaining member of a mirrorset, you cannot use the UNMIRROR command to change the device back to a single-disk unit, because that change would cause a change in reported disk capacity, possibly confusing the operating system.

Format

`UNMIRROR` *`disk-name`*

Parameter

disk-name
The disk-name parameter specifies the name of the mirrorset member to be removed from the mirrorset.

Examples

To convert DISK130 back to a single device:

```
CLI> UNMIRROR DISK130
```

See also

ADD MIRRORSET
MIRROR
REDUCE
RUN CLONE
SET *mirrorset-name*