# Compaq StorageWorks

## HSG80 Array Controller ACS Version 8.5

CLI Reference Guide

# Notice

# Contents

*Chapter 1*

**CLI Command Introduction**

*Chapter  2*

## CLI Command Descriptions

## Glossary

# Tables

# About This Guide

This book describes the Command Line Interpreter (CLI) of the HSG80 Array Controller running Array Controller Software (ACS) 8.5. In the description of the CLI commands, the syntax, parameters, and switches to the command are also described.

This book does not contain information about the operating environments to which the controller may be connected, nor does it contain detailed information about subsystem enclosures or their components. See the documentation that accompanied these peripherals for information about them.

## Conventions

This book uses the following special notices and typographical conventions to help you find what you are looking for.

### Special Notices

This book does not contain detailed descriptions of standard safety procedures. However, it does contain warnings for procedures that could cause personal injury and cautions for procedures that could damage the controller or its related components. Look for these symbols when you are carrying out the procedures in this book.

> **WARNING:** A *Warning* contains information essential to people's safety. It advises users that failure to take or avoid a specific action could result in physical harm to the user or hardware. Use a warning, not a caution, when such damage is possible.

**CAUTION:** A *Caution* contains information that the user needs to know to avoid damaging the software or hardware.

**IMPORTANT:** An *important* note is a type of note that provides information essential to the completion of a task. Users can disregard information in a note and still complete a task, but they should not disregard an important note.

**NOTE:** A *note* indicates neutral or positive information that emphasizes or supplements important points of the main text. A note supplies information that may apply only in special cases—for example, memory limitations, equipment configurations, or details that apply to specific versions of a program.

## Typographical Conventions

| Convention | Meaning |
|---|---|
| **ALLCAPS BOLD** | Command syntax that must be entered exactly as shown, for example: <br><br> **SET FAILOVER COPY=OTHER** |
| ALLCAPS | Command syntax that is discussed within text, for example: <br> "Use the SHOW SPARESET command to show the contents of the spareset." |
| MONOSPACED | ALL SCREEN DISPLAYS ARE IN UPPER-CASED MONOSPACED FONT. |
| *Sans serif italic* *Sans serif italic* | Command variable or numeric value that you supply, for example: SHOW *RAIDset-name* or <br><br> **SET THIS_CONTROLLER ID=***(n,n,n,n,)* |
| *italic* | Reference to other books, for example: "See the *HSJ80 Array Controller Configuration Guide* for details." |

*Typographical Conventions Continued...*

| Convention | Meaning |
|---|---|
| .<br>.<br>. | Indicates that a portion of an example or figure has been omitted. |
| "this controller" | The controller serving your current CLI session through a local or remote terminal. |
| "other controller" | The controller in a dual-redundant pair that's connected to the controller serving your current CLI session. |

## Related Publications

The following table lists some of the documents related to the use of the controller, cache module, and external cache battery.

**Table 1   Relevant Publications**

| Document Title | Part Number |
|---|---|
| *StorageWorks Solutions Software Kit Overview* | EK–SOLSR–AA / 166314-001 |
| *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 Maintenance and Service Guide* | EK–HSG84–SV. B01 / 118620–002 |
| *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 Configuration Guide* | EK–HSG85–CG / 165144–001 |
| *Compaq StorageWorks Command Console (HSG80) User Guide* | AA–RFA2D–TE / 387405–004 |
| *Compaq StorageWorks UltraSCSI RAID Enclosure (BA370-Series) User Guide* | EK–BA370–UG / 387403–001 |
| *Compaq StorageWorks Fibre Channel Storage Switch Service Guide* | AA–RHBZA–TE / 135268–001 |
| *Compaq StorageWorks Fibre Channel Storage Switch User Guide* | AA–RHBYA–TE / 135267–001 |
| *Compaq StorageWorks Fibre Channel Switch Quick Setup Guide* | AA–RHC0A–TE / 135269–001 |
| *Compaq StorageWorks Fibre Channel Arbitrated Loop Hub (DS-DHGGA-CA) User Guide* | EK–DHGGA–UG |
| *Compaq StorageWorks Data Replication Manager HSG80 ACS Version 8.5P Operations Guide* | EK–HSG84–DT / 128519–002 |
| *Compaq StorageWorks KGPSA PCI-to-Fibre Channel Host Adapter User Guide* | EK–KGPSA–UG |

| Table 1  Relevant Publications (Continued) | |
|---|---|
| **Document Title** | **Part Number** |
| *The RAIDBOOK—A Source Book for Disk Array Technology* | RAID Advisory Board: I.S.B.N 1-879936-90-9 |
| *Compaq StorageWorks RA8000/ESA12000 Storage Subsystem User Guide* | EK–SMCPR–UG / 387404–001 |
| Compaq StorageWorks RA8000/ESA12000 HSG80 Solution Software V8.5 Installation Reference Guides: | |
| for OpenVMS | AA–RH4BB–TE / 387401–002 |
| for TRU64 UNIX | AA–RFAUB–TE / 387389–002 |
| for HP-UX | AA–RFBED–TE / 387374–004 |
| for IBM AIX | AA–RJ25B–TE / 152848–002 |
| for Novell NetWare | AA–RFB9B–TE / 387376–002 |
| for SILICON GRAPHICS IRIX | AA–RFBKC–TE / 387399–003 |
| for Sun Solaris | AA–RFBQE–TE / 387384–005 |
| for Windows NT Server - Alpha | AA–RFB4A–TE / 388206–001 |
| for Windows NT Server - Intel | AA–RFA9C–TE / 387387–003 |

# Revision History

First Release: October 1999

*Chapter* **1**

# CLI Command Introduction

This guide contains descriptions of the command line interpreter (CLI) commands you can use to interact with the controller. This interaction can be by way of:

■ A local terminal connected to the maintenance port of the controller

■ A remote connection by way of the host system terminal

■ The CLI window of StorageWorks Command Console (SWCC)

This section is divided into an overview and a command description. The overview provides a general description of the CLI interface and how to use it. The command description contains the description, full syntax, and examples of the use of each command available on the CLI interface.

## CLI Overview

The CLI is the most direct interface to the controller. The CLI commands allow you to manage the subsystem by viewing and modifying the configuration of the controller and its attached storage devices. You can also use the CLI to start controller diagnostic and utility programs.

While the CLI provides the most detailed level of subsystem control, a graphical user interface (GUI) is also available. This GUI is called StorageWorks Command Console, or SWCC. SWCC replicates most of the functions available within the CLI in graphic form and provides a graphical method of configuring and monitoring the controller. In addition to the graphical interface, SWCC has a CLI window through which CLI commands can be entered.

CLI commands for configuring and viewing the controllers use the relative terms "this controller" and "other controller." See "Typographical Conventions," page viii, for an explanation of these terms.

## Using the CLI

You can access the CLI through the following methods:

■ By connecting a local terminal/PC to the maintenance port on the front of the controller.

■ Using the DUP utility from the system terminal you may also enable a remote connection to the controller. After you have initially configured the controller, and made it visible to the host, you can perform other configuration tasks through this remote connection.

■ Using the SWCC by way of a PC setup on the subsystem. The SWCC can be used to perform most of the CLI commands that the local terminal can.

## Command Overview

CLI commands are divided into categories based on what options or structures they control. The following is a list of the command categories:

■ **Controller Commands**—Configure the controller's SCSI ID numbers, maintenance terminal characteristics, CLI prompt, and so forth. Controller commands are also used to shut down and restart the controller.

■ **Device Commands**—Create and configure containers made from physical devices attached to the controller.

■ **Storageset Command**s—Create and configure complex containers made from groups of device containers. There are four basic types of storagesets: stripesets, RAIDsets, striped-mirrorsets, and mirrorsets. Storageset commands group device containers together and allow them to be handled as single units.

■ **Logical Unit Commands**—Create and optimize access to logical units made from any container type.

■ **Failover Commands**—Configure a pair of controllers to operate in either transparent failover mode or multiple-bus failover mode.

■ **Diagnostic and Utility Commands**—Perform general controller support functions.

■ **Data Replication Manager Commands**—Work only in a remote copy environment and require V8.5**P** code.

## Controller Commands

CLI Controller commands fall into two categories:

■ General commands

■ Failover commands

The CLI commands that pertain to the controllers in a general way are as follows:

■ CLEAR_ERRORS

  ❏ CLI

  ❏ INVALID_CACHE

■ Configuration

  ❏ RESET

  ❏ RESTORE

  ❏ SAVE

■ EXIT

■ HELP

■ POWEROFF

■ RESTART

■ SELFTEST

■ SET

■ SHOW

CLI Commands that control the failover mode of a controller pair are:

■ SET FAILOVER

■ SET NOFAILOVER

## Device Commands

Device commands fall into two categories:

■ General commands

■ Spareset/failedset commands

CLI Commands that allow you to add and configure physical devices in general are:

■ ADD DISK

■ CLEAR_ERRORS

- ❏ LOST_DATA
- ❏ UNKNOWN
- ❏ UNWRITEABLE_DATA
- ■ DELETE
- ■ INITIALIZE
- ■ LOCATE
- ■ RENAME
- ■ SET

CLI Commands that allow you to change the sparesets/failedsets are:

- ■ ADD SPARESET
- ■ DELETE FAILEDSET
- ■ DELETE SPARESET
- ■ SET FAILEDSET
- ■ SHOW FAILEDSET

## Storageset Commands

CLI Commands that pertain to storagesets are:

- ■ ADD MIRRORSET
- ■ ADD RAIDSET
- ■ ADD STRIPESET
- ■ DELETE
- ■ INITIALIZE
- ■ LOCATE
- ■ MIRROR
- ■ REDUCE
- ■ RENAME
- ■ SET
- ■ SHOW
- ■ SHOW MIRRORSETS
- ■ SHOW RAIDSETS

- SHOW STRIPESETS

- UNMIRROR

## Partition Commands

CLI Commands that pertain to partitions are:

- CREATE_PARTITION

- DESTROY_PARTITION

## Unit Commands

CLI commands that pertain to units are:

- ADD CONCATSETS

- DELETE

- LOCATE

- RENAME

- RETRY_ERRORS UNWRITEABLE_DATA

- SET

- SHOW

- SHOW UNITS

- ADD SNAPSHOT_UNITS

## Diagnostic and Utility Commands

Commands that pertain to the diagnostics and utilities are:

- DIRECTORY

- RUN

## Data Replication Manager (DRM) commands

> **NOTE:** All DRM commands are hidden and inoperative until the controller pair is put into remote copy mode by specifying the *REMOTE_COPY* switch of the SET *this controller* command.

ADD ASSOCIATIONS

ADD REMOTE_COPY_SET

DELETE association-set_name

DELETE remote-copy-set-name

SET association-set-name

SET remote-copy-set-name

SITE FAILOVER

SHOW

## Getting Help

To get help with using CLI commands, enter HELP at the CLI prompt. The resultant display shows an overview of the CLI help system. To obtain help with a specific command or to determine what switches are available with a command, enter as much of the command syntax as you know followed by a space and a question mark.

For example, to get information on the switches used with the SET *this_controller* command, enter:

SET THIS ?

Another example: to see what is allowed for a prompt, enter:

SET THIS PROMPT=?

## Entering CLI Commands

Use the following tips and techniques to aid you in the entering of CLI commands:

■ Commands are not case-sensitive.

■ For most commands, you only need to enter enough of the command to make the command unique. For example, SHO is the same as entering SHOW.

■ The controller processes each command in sequence. You can continue entering subsequent commands while the controller is processing prior commands. A controller experiencing heavy data I/O may respond slowly to CLI commands.

Specific keys or a combination of keys allow you to recall and edit the last four commands. This feature can save time and help prevent mistakes when you need to enter similar commands during the configuration process. Table 1–1 lists the keys used to recall and edit commands.

**Table 1–1  Recall and Edit Command Keys**

| Key | Function |
| --- | --- |
| Up Arrow or Ctrl/B,  Down Arrow or Ctrl/N | Steps forwards or backward through the four 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 all characters on the same line as the cursor. |
| Ctrl/A or F14 | Toggles between insert and overstrike mode:  ■ The default setting is insert mode allowing you to insert characters at the cursor location (moving the existing characters to the right).  ■ Overstrike mode replaces existing characters. The CLI prompt 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. |

## Command Syntax

The following structure defines how each command to the controller is used:

COMMAND PARAMETER_NAME=*parameter SWITCHES*

■ COMMAND—A word or phrase that instructs the controller what to do. Commands are represented in this manual in capitalized form.

■ PARAMETER NAME—The name of a parameter, followed by an equal sign and the parameter variable. If a specific command specifies parameter names, they must be entered in the command string. Parameter names are represented in this manual in capitalized form.

■ *parameter*—When required in the command, parameters are words or phrases that supply necessary variables to support the action of the COMMAND. Not all CLI commands require parameters.

■ *SWITCHES*—An optional word or phrase that modifies the command. Not all CLI commands use switches. Switches are represented in this manual as capitalized, italicized text.

*Chapter* **2**

# CLI Command Descriptions

This section describes the CLI commands and presents full syntax and examples of the use of each command.

## CLI Command Descriptions

This section contains the descriptions of the available CLI commands used with the ACS V8.5 software. Each command is described using the following format:

- Command name and brief description

- Syntax needed to enable command

- Parameters needed (if any) to further specify the command

- Switches needed (if any) to modify the command

- Example(s) to illustrate the command usage

- A "See Also" section to cross-reference to other commands with similar usage

## ADD ASSOCIATIONS

**NOTE:** This command works only in a Data Replication Manager (DRM) environment and requires V8.5P code. Like all DRM commands, it is heavily restricted in its use. See *Data Replication Manager HSG80 ACS Version 8.5P Operations Guide* for examples of usage.

**NOTE:** This command is valid only on the node on which the Initiator resides (where remote copy sets are configured).

An association set is a group of remote copy sets that shares common attributes. This command adds a one-member association set (that is, an association set consisting of one remote copy set) to the controller pair's configuration. Additional remote copy sets can be added to the association set with the SET *associations* command.

This command is rejected if the specified remote copy set is unknown to the controller pair.

### Syntax

ADD ASSOCIATIONS *association-set-name remote-copy-set-name*

### Parameters

The following parameters are required for the ADD ASSOCIATIONS command:

■ association set name

■ remote copy set name

These parameters are described in the following paragraphs.

*association-set-name*

The name that will be assigned to the association set.

**NOTE:** Association sets cannot be renamed with the RENAME command. If the wrong name is entered, the association set must be deleted and then added again.

The association set name must start with a letter (A through Z) and may consist of a maximum of nine characters including letters A through Z, numbers 0 through 9, periods (.), dashes (-), or underscores (_).

*remote-copy-set-name*

The name of the first member of the association set.

## Switches

There are no switches associated with this command.

## Examples

To create an association set, AS3, from remote copy sets RSC1 and RSC2, enter:

ADD ASSOCIATIONS AS3 RSC1

SET AS3 ADD=RSC2

## See also

ADD REMOTE_COPY_SET

SET *associations*

## ADD CONCATSETS

> ⚠️ **CAUTION:** This command should only be executed with host operating systems that support dynamic volume expansion. If the operating system cannot handle one of its disks increasing in size, use of this command could make data inaccessible.

This command creates a specialized volume, called a concatset (short for concatenation set), from a storageset that has been given a unit number. Another storageset can then be added to the concatset by the SET *concatset* command.

### Syntax

ADD CONCATSETS *concat-name storageset-name*

### Parameters

The following parameters are required for the ADD CONCATSETS command:

■ concatset-name

■ storageset-name

These parameters are described in the following paragraphs.

*concatset-name*

Assigns a name to the concatenation set, or concatset.

The name must start with a letter (A through Z) and may consist of a maximum of nine characters including letters A through Z, numbers 0 through 9, periods (.), dashes (-), or underscores (_).

*storageset-name*

Designates the first storageset to be a member of the concatset.

### Switches

There are no switches associated with this command.

## Examples

To expand the capacity of unit D0, which consists of stripeset Stripe1, by adding another stripeset, Stripe2, enter:

    ADD CONCATSETS C1 STRIPE1

    SET C1 ADD=STRIPE2

## See also

DELETE concatset-name

SET *concatset-name*

SHOW CONCATSETS

SHOW concatset-name

## ADD CONNECTIONS

Each path between a Fibre Channel adapter in a host computer and an active host port on a controller is a connection.

This command adds the specified host connection to the table of known connections. This table is maintained in the controller's memory. The maximum table length is 64 connections; if the table contains 64 entries, new connections cannot be added unless some old ones are deleted.

There are two mechanisms for adding a new connection to the table:

1. Physically connecting a host adapter to a controller host port. During Fibre Channel initialization, the controller becomes aware of the connection and adds it to the table. This physical discovery of connections occurs at the point when a host adapter is plugged in to a controller port and after issuing a RESTART command. New connections discovered through physical connection are assigned a default connection name by the controller. The default connection name is of the form !NEWCON*nn*.

   **NOTE:** Certain host conditions, such as a power cycle, that disturb the state of the switched fabric may cause a connection to reappear in the table. The connection will be assigned a default connection name.

2. Adding a connection through the ADD CONNECTIONS command.

   **NOTE:** ADD CONNECTIONS will add an entry to the table whether the connection physically exists or not. The table can be completely filled up with fictitious connections.

### Syntax

ADD CONNECTIONS *connection_name* HOST_ID=*host_id* ADAPTER_ID=*adapter_id* CONTROLLER=*controller* PORT=*port*

### Parameters

The following parameters are required to the ADD CONNECTIONS command:

■  connection name

■  host ID

■  adapter ID

■  controller

■  port

These parameters are described in the following paragraphs.

*connection-name*

> The name that will be assigned to the host connection. The connection name can be any character string, with one exception: it cannot be in the form of a default connection name. The form of a default connection name is `!NEWCONnn`.

> The default connection name is assigned automatically by the controller when the connection is physically made between a host adapter and a controller port. Default connection names are assigned only by the controller.

HOST_ID=*host_id*

> HOST_ID is the parameter name for the *host_id* variable, which is the worldwide name (WWN) of the host. WWN is a 16-character hexadecimal number. The hyphens aren't necessary, but are recommended to avoid mistakes in entering the number. The HOST_ID parameter name must be entered, followed by an equal sign, followed by the WWN of the host.

> For example, to specify a WWN of AAAA-AAAA-AAAA-AAAA, type:

> `host_id=AAAA-AAAA-AAAA-AAAA`

ADAPTER_ID=*adapter_id*

> ADAPTER_ID is the parameter name for the *adapter_id* variable, which is the worldwide name (WWN) of the host Fibre Channel adapter. WWN is a 16-character hexadecimal number. The hyphens aren't necessary, but are recommended to avoid mistakes in entering the number. The ADAPTER_ID parameter name must be entered, followed by an equal sign, followed by the WWN of the host bus adapter.

> For example, to specify a WWN of FFFF-FFFF-BBBB-9876, type:

> `adapter_id=FFFF-FFFF-BBBB-9876`

> **NOTE:** The worldwide name of the host and adapter are sometimes the same. This is a characteristic of the adapter.

CONTROLLER=*controller*

> CONTROLLER is the parameter name for the *controller* variable, which specifies which controller the host is to connect through. The choices are *this_controller*, which can be shortened to *this*, and o*ther_controlle*r, which can be shortened to *other*.

PORT=*port*

*PORT* is the parameter name for the port variable, which specifies which host port (1 or 2) the connection is on.

## Switches

The following switches are available with the ADD CONNECTION command:

■  unit offset

■  operating system

These switches are described in the following paragraphs.

*UNIT_OFFSET=n*

Offset is a decimal value that establishes the beginning of the range of units that a host connection can access. It defines and restricts host connection access to a contiguous group of unit numbers.

In transparent failover mode, host connections on port 1 default to an offset of 0; port 1 connections can see units 0 through 99. Host connections on port 2 default to an offset of 100; port 2 connections can see units 100 through 199.

In multiple-bus failover mode, the default offset is 0 for all host connections.

> **NOTE:** If a controller pair is switched from transparent failover mode to multiple-bus failover mode, the unit offsets for transparent mode remain in effect.

The LUN number equals the unit number minus the offset:

LUN number = unit number - offset.

■  Logical unit number or LUN number = the logical unit number presented to the host connection.

■  Unit number = the number assigned to the unit in the ADD UNIT command. This is the number by which the unit is known internally to the controllers.

For example, a system has three host connections, each of which can see 8 LUNs. Each connection designates its LUNs as 0 through 7. To define for each of these connections a block of 8 units, set the offset for each connection, as follows:

CLI>SET SERVER1 OFFSET=0

CLI>SET SERVER2 OFFSET=10

CLI>SET SERVER3 OFFSET=120

(It is not necessary to use offsets that are divisible by 10, but it makes things simpler.)

The effect of these offset assignments will be as follows:

■ Server1 will access units D0 through D7, which it will see as LUNs 0 through 7.

■ Server2 will access units D10 through D17, which it will see as LUNs 0 through 7.

■ Server3 will access units D120 through D127, which it will see as LUNs 0 through 7.

There is one additional factor to consider when assigning unit numbers and offsets. If the *SCSI_VERSION* qualifier of the SET THIS/OTHER CONTROLLER command is set to SCSI-3, the command console LUN (CCL) is presented as LUN 0 to every connection, superseding any unit assignments. For more information see the ADD UNIT command.

*OPERATING_SYSTEM=OS_name*

Specifies the operating system of the host. The *OPERATING_SYSTEM* switch tailors the controllers behavior for use with a particular operating system. Enter a question mark after *OPERATING_SYSTEM* to see the supported operating systems, as follows:

ADD CONNECTIONS SERVER1 OPERATING_SYSTEM=?

The response for this command is a list of abbreviations for the supported operating systems.

## Examples

This example shows how to add an entry for a connection named George to the table of known connections, with the indicated host and adapter worldwide names, on port 2 of "this controller."

```
ADD CONNECTIONS GEORGE HOST_ID=1000-0000-C920-1234
ADAPTER_ID=1000-0000-C920-5678 CONTROLLER=THIS PORT=2
```

## See also

ADD UNIT
DELETE *connections*
SET *connection-name*

## ADD DISK

This command names a disk drive and adds it to the controller configuration. This command is typically used when disk drives are physically added to a previously configured subsystem. During initial set-up, the entire set of disk drives is customarily added to the controller's configuration through the CONFIG utility.

### Syntax

ADD DISK *disk_name port target lun*

### Parameters

The following parameters are required for the ADD DISK command:

- disk name
- port
- target
- LUN

These parameters are described in the following paragraphs.

*container_name*

Assigns a name to the disk device. This container name is then used with the ADD UNIT command to create a logical unit.

The disk name must start with a letter (A through Z) and may consist of a maximum of nine characters including letters A through Z, numbers 0 through 9, periods (.), dashes (-), or underscores (_).

It is common to name a disk drive DISK*pttll*, where *pttll* is the disk's Port-Target-LUN address. Although other naming conventions are acceptable, this one presents the user with the type of disk drive and its SCSI location.

*port*

Designates the SCSI device port number, from 1 to 6, on which the disk resides.

*target*

Designates the SCSI target ID of the disk on the port. Valid device target IDs for single controller configurations are 00–15, excluding ID 7. Valid device target IDs for dual controller configurations are 00–15, excluding IDs 4 through 7.

*lun*

Indicates the LUN of the disk drive, and is always zero.

The parameters *port*, *target*, and *lun* must be entered with at least one space between them. Leading zeroes can be excluded.

## Switches

The following switches are available with the ADD DISK command:

■   transfer rate requested

■   transportable/notransportable

These switches are described in the following paragraphs.

*TRANSFER_RATE_REQUESTED=ASYNCHRONOUS*
*TRANSFER_RATE_REQUESTED=20MHZ (Default)*
*TRANSFER_RATE_REQUESTED=10MHZ*
*TRANSFER_RATE_REQUESTED=5MHZ*

Specifies the maximum data transfer rate at which the controller is to communicate with the disk drive. The user might need to limit the transfer rate to accommodate long cables between the controllers and the device.

*TRANSPORTABLE*
*NOTRANSPORTABLE (Default)*

If *NOTRANSPORTABLE* is specified, the controller makes a small portion of each disk inaccessible to the host. It uses this restricted space to store administrative information (metadata) which improves data reliability, error detection, and recovery. This improvement comes at the expense of transportability, because the metadata is specific to StorageWorks products.

Transportable disks do not contain metadata or use restricted areas on the media, so they forfeit any advantages this feature provides. Transportable disks, however, can be moved to a non-StorageWorks environment with their data intact.

If NOTRANSPORTABLE is specified and there is no valid metadata on the unit, the unit must be initialized.

> **NOTE:** Avoid using transportable disks unless there is no other way to move the data.

**Examples**

To add DISK10000 at port 1, target 0, LUN 0, enter:

ADD DISK DISK10000 1 0 0

To add DISK40200 as a transportable drive to port 4, target 2, LUN 0, enter:

ADD DISK DISK40200 4 2 0 TRANSPORTABLE

To add a disk drive named DISK30200 as a non-transportable disk to port 3, target 2, LUN 0 and set the data transfer rate to 10 MHz, enter the following:

ADD DISK DISK30200 3 2 0 NOTRANSPORTABLE TRANSFER_RATE_REQUESTED=10MHZ

**See also**

ADD UNIT
DELETE *container-name*
INITIALIZE
LOCATE
SHOW DISKS
SHOW DEVICES
SET *container-name*

# ADD MIRRORSET

This command creates and names a mirrorset and adds it to the controller configuration.

The capacity of the mirrorset is constrained by its smallest member.

## Syntax

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

## Parameters

The following parameters are required for the ADD MIRRORSET command:

■ mirrorset name

■ disk names

These parameters are described in the paragraphs that follow.

*mirrorset-name*

Assigns a name to the mirrorset container. This is the name used with the ADD UNIT command to identify the mirrorset as a host-addressable unit.

The mirrorset name must start with a letter (A through Z) and may consist of a maximum of nine characters including letters A through Z, numbers 0 through 9, periods (.), dashes (-), or underscores (_).

It is common to name a mirrorset MIRR*n*, where *n* is a sequentially assigned, unique identifier. Other naming conventions are acceptable, but this naming convention presents both the type of container and its unique identifier in an intuitive manner.

*disk-name1 [disk-nameN]*

Identifies the disk drives making up the mirrorset. A mirrorset may contain one to six disk drives.

## Switches

The following switches are available with the ADD DISK command:

■ copy

■ policy/nopolicy

■ read source

These switches are described in the following paragraphs.

*COPY=FAST*
*COPY=NORMAL (Default)*

Sets the speed at which the controller copies data to a new member from normal mirrorset members when data is being mirrored to the disk drives of the storageset:

■  Specify *COPY=FAST* to allow the creation of mirrored data to take precedence over other controller operations. When you specify COPY=FAST, the controller uses more resources to create the mirrored data, and copying takes less time. However, overall controller performance is reduced.

■  Specify *COPY=NORMAL* when operations performed by the controller should take priority over the copy operation. If you specify *COPY=NORMAL,* creating the mirrored data has a minimal impact on performance.

*POLICY=BEST_FIT*
*POLICY=BEST_PERFORMANCE (Default)*
*NOPOLICY*

Sets the selection criteria the controller uses to choose a replacement disk from the spareset when a mirrorset member fails.

■  Specify *POLICY=BEST_FIT* to choose a replacement disk drive from the spareset that equals or exceeds the base member size (smallest disk drive at the time the mirrorset was initialized). If there is more than one disk drive in the spareset that meets the criteria, the controller selects a disk drive with the best performance.

■  Specify *POLICY=BEST_PERFORMANCE* to allow the software to choose a replacement disk drive from the spareset with the best performance. The controller attempts to select a disk on a different port than existing mirrorset members. If there is more than one disk drive in the spareset matching the best performance criteria, the controller selects a disk drive that equals or exceeds the base member size.

■  Specify *NOPOLICY* to prevent the controller from automatically replacing a failed disk device. The mirrorset operates in a reduced state until a *POLICY=BEST_FIT* or *POLICY=BEST_PERFORMANCE* is selected, or a member is manually placed in the mirrorset.

*READ_SOURCE=disk-name*
*READ_SOURCE=LEAST_BUSY (Default)*
*READ_SOURCE=ROUND_ROBIN*

Selects the mirrorset member used by the controller to satisfy a read request.

■ Specify the *READ_SOURCE=disk-name* of a specific member to which you want the controller to direct all read requests. If the member fails out of the mirrorset, the controller selects the first normal member it finds to satisfy its read requests.

■ Specify *READ_SOURCE=LEAST_BUSY* to direct read requests to the mirrorset disk with the least amount of work in its queue. If multiple disks have equally short queues, the controller queries normal disks for each read request as it would when *READ_SOURCE=ROUND_ROBIN* is specified.

■ Specify *READ_SOURCE=ROUND_ROBIN* to sequentially direct read requests to each mirrorset disk. The controller equally queries all normal disks for each read request.

## Examples

To create a mirrorset named MIRR1 consisting of disks DISK10000, DISK20100, and DISK30200, enter:

    ADD MIRRORSET MIRR1 DISK10000 DISK20100 DISK30200

## See also

ADD DISK
ADD UNIT
DELETE *container-name*
INITIALIZE
MIRROR
REDUCE
SHOW *mirrorset-name*
SHOW MIRRORSETS
SHOW STORAGESETS
UNMIRROR

## ADD RAIDSET

This command creates and names a RAIDset and adds the RAIDset to the controller configuration. Compaq RAIDsets are often referred to as RAID level 3/5 storagesets because they use the best characteristics of RAID level 3 and RAID level 5.

The capacity of the RAIDset is constrained by its smallest member.

### Syntax

ADD RAIDSET *RAIDset-name disk-name1 disk-name2 disk-name3 [disk-nameN]*

### Parameters

The following parameters are available to the ADD RAIDSET command:

■ RAIDset name

■ disk names

These parameters are described in the paragraphs that follow.

*RAIDset-name*

Assigns a name to the RAIDset. The RAIDset name must start with a letter (A through Z) and may consist of a maximum of nine characters including letters A through Z, numbers 0 through 9, periods (.), dashes (-), or underscores (_).

It is common to name a RAIDset RAID*n*, where *n* is a sequentially assigned, unique identifier. This naming convention presents the user with the type of container and its unique identifier.

*disk-name1 disk-name2 disk-name3 [disk-nameN]*

Identifies the disks making up the RAIDset. RAIDsets must include at least 3 disk drives, and no more than 14.

### Switches

The following switches are available with the ADD RAIDSET command:

■ policy/nopolicy

■ copy

■ reconstruct

■  reduced/noreduced

These switches are described in the following paragraphs.

*POLICY=BEST_FIT*
*POLICY=BEST_PERFORMANCE (Default)*
*NOPOLICY*

Sets the selection criteria the controller uses to choose a replacement member from the spareset when a RAIDset member fails:

■  Specify *POLICY=BEST_FIT* to choose a replacement disk drive from the spareset that equals or exceeds the base member size of the remaining members of the RAIDset. If more than one disk drive in the spareset is the correct size, the controller selects the disk drive that will provide the best performance.

■  Specify *POLICY=BEST_PERFORMANCE* to choose a replacement disk drive from the spareset resulting in the best performance of the RAIDset. The controller attempts to select a disk on a different port than existing RAIDset members. If there is more than one disk drive in the spareset matching the best performance criteria, the controller selects a disk drive that equals or exceeds the base member size of the RAIDset.

■  Specify *NOPOLICY* to prevent the controller from automatically replacing a failed disk device. This RAIDset operates in a reduced state until you select either *POLICY=BEST_PERFORMANCE* or *POLICY=BEST_FIT*, or manually place a member in the RAIDset.

*RECONSTRUCT=FAST*
*RECONSTRUCT=NORMAL (Default)*

Sets the speed at which the controller reconstructs data to a new RAIDset disk that replaces a failed disk:

■  Specify FAST to allow the reconstruct process to take precedence over other controller operations. When the *RECONSTRUCT=FAST* switch is specified, the controller uses more resources to perform the reconstruction. Reconstruction takes less time, but overall controller performance is reduced during reconstruction.

■  Specify *NORMAL* to balance other controller operations with the reconstruct operation. The controller uses relatively few resources to perform the reconstruct process: there is little impact on performance.

*REDUCED*
*NOREDUCED (Default)*

Permits the addition of a RAIDset missing a member:

■ Specify the *REDUCED* switch when you add a reduced RAIDset (a RAIDset that is missing one member).

**NOTE:** Verify the RAIDset contains all but one of its disks before specifying the *REDUCED* switch.

■ Specify the *NOREDUCED* switch when all the disks making up the RAIDset are present—for instance, when creating a new RAIDset.

## Examples

To create a RAIDset named RAID9 that contains disks DISK10000, DISK20100, and DISK30200, enter:

ADD RAIDSET RAID9 DISK10000 DISK20100 DISK30200

To create a RAIDset named RAID8 that contains disks DISK10000, DISK20100, DISK30200, and uses the *BEST_FIT* switch to indicate the replacement policy, enter:

ADD RAIDSET RAID8 DISK10000 DISK20100 DISK30200 POLICY=BEST_FIT

**NOTE:** The ADD RAIDSET command must be entered on one line.

To create a RAIDset named RAID8 that contains disks DISK10000, DISK20100, DISK30200, then initialize it and make it into a host-addressable unit, enter:

ADD RAIDSET RAID8 DISK10000 DISK20100 DISK30200
INITIALIZE RAID8
ADD UNIT D70 RAID8

To create a three-member RAIDset from the members of a reduced four-member RAIDset, enter the following (do not initialize the RAIDset again):

> ⚠ **CAUTION:** Data contained on the RAIDset will be erased if you reinitialize the RAIDset.

ADD RAIDSET RAID6 DISK10300 DISK20400 DISK30200 REDUCED

**See also**

ADD UNIT
DELETE *container-name*
SET RAIDSET
SHOW RAIDSET
SHOW *RAIDset-name*
SHOW STORAGESETS
INITIALIZE

## ADD REMOTE_COPY_SETS

> **NOTE:** This command works only in a Data Replication Manager (DRM) environment and requires V8.5P code. Like all DRM commands, it is heavily restricted in its use. See *Data Replication Manager HSG80 ACS Version 8.5P Operations Guide* for examples of usage.

> **NOTE:** This command must only be used on the initiator (local) site.

This command creates a remote copy set (RCS), consisting of one unit at the initiator site and one unit at the target site. The units may be a single disk (JBOD) or a stripeset, mirrorset, or RAIDset.

The following restrictions apply to remote copy sets:

■ *MAXIMUM_CACHE_TRANSFER_SIZE* (a switch of the ADD UNIT and SET *unit* command) should be set to 1 on target units. For initiator units, *MAXIMUM_CACHE_TRANSFER_SIZE* should be set to whatever is optimal for the application.

■ There is a maximum of 12 remote copy sets per initiator/target pair.

■ Neither initiator nor target units can be transportable units, concatset units, or snapshot units.

### Syntax

ADD REMOTE_COPY_SETS *remote-copy-set-name initiator-unit-name remote-node-name\target-unit-name*

### Parameters

The parameters required for this command are:

■ remote copy set name

■ initiator unit name

■ remote node name\target unit name (this parameter is not required; it can be added later through the SET command)

These parameters are described in the following paragraphs.

*remote-copy-set-name*

The name by which the remote copy set will be known. This name must be unique across the fabric.

> **NOTE:** Remote copy sets cannot be renamed with the RENAME command. If the wrong name is entered, the remote copy set must be deleted and then added again.

The remote copy set name must start with RCS and may consist of a maximum of 9 characters including letters A through Z, numbers 0 through 9, periods (.), dashes (-), or underscores (_).

*initiator-unit-name*

The name of the initiator unit that is to be the first member of the remote copy set.

*remote-node-name\target-unit- name*

The *remote-node-name* part of the parameter specifies the name of the controller pair—called a *node*—receiving the command. The *target-unit-name* part specifies the unit on the remote node to be added as the target of the remote copy set.

## Switches

*OPERATION_MODE=SYNCHRONOUS* (default)
*OPERATION_MODE=ASYNCHRONOUS*

This switch determines which of two normal operating modes is assigned to the remote copy set.

In synchronous mode, write operations must be completed on the remote units of the remote copy set before the host is informed that the operation is complete. Synchronous operation assures data consistency at all times among the members of a remote copy set.

In asynchronous mode, the write operation is reported as complete to the host before the data is written to the remote units of the remote copy set. Asynchronous mode gives greater performance and faster response time, but the data on all members of the remote copy set cannot be assumed to be always the same.

*OUTSTANDING_IO=* n

This switch sets the number of outstanding I/O operations from the initiator to the target.

n is a decimal number between 1 and 240, with 20 being the default value.

The way this switch operates depends on which operating mode is set by the *OPERATION_MODE* switch:

■ In synchronous mode, *OUTSTANDING_IO* refers to the number of remote writes (write operations from the initiator to the target) that can be outstanding.

■ In asynchronous mode, *OUTSTANDING_IO* refers to the number of write operations that can be reported as completed to the host before they have been written on all the members of the remote copy set.

### Example

To create remote copy set RCS1, consisting of unit D1 on the local controller pair and unit D21 on a remote controller pair (node) named London, enter:

ADD REMOTE_COPY_SET RCS1 D1 LONDON\D21

### See also

ADD ASSOCIATIONS
SET *controller* REMOTE_COPY
SET *remote-copy-set-name*

# ADD SNAPSHOT_UNITS

**NOTE:** This command is operational only in controller software versions V8.5S and V8.5P.

**NOTE:** This commands is operational only if both controllers have 512MB mirrored cache.

This command creates and names a snapshot unit. A snapshot unit is one that reflects the contents of another unit at a particular point in time (the instant the ADD SNAPSHOT_UNITS command is entered). The snapshot unit can be presented to the host. The snapshot unit remains until it is deleted (DELETE command).

## Syntax

ADD SNAPSHOT_UNITS *snapshot-unit storageset source-unit*

## Parameters

The following parameters are required for the ADD SNAPSHOT_UNITS command:

- snapshot unit

- storageset

- source unit

The relationship of the parameters can be summarized as follows:

When the ADD SNAPSHOT_UNITS command is entered, *storageset* becomes *snapshot-unit* and archives the current contents of *source-unit* at that instant.

These parameters are described in the paragraphs that follow.

*snapshot-unit*

The unit number that will be assigned to the snapshot unit. The unit number must start with a letter (A through Z) and may consist of a maximum of nine characters including letters A through Z, numbers 0 through 9, periods (.), dashes (-), or underscores (_).

The snapshot unit is created with all host access disabled by default. Do a SET command to set up host access.

The snapshot unit will be created on the same controller as the source unit, and must remain there.

*storageset*

Identifies the storageset that will become the snapshot unit. The storageset must have the following characteristics:

❏ capacity equal to or greater than the source unit

❏ initialized

❏ not a partition or a concatset

*source-unit*

The unit whose contents will be frozen in time and preserved on the snapshot unit. The source unit must have the following characteristics:

❏ less than 512 GB

❏ write-back cache enabled

❏ non-transportable

## Switches

There are no switches associated with this command.

## Examples

To create unit D4, which consists of storageset RAID4 and which will become a point-in-time snapshot of unit D1, enter:

ADD SNAPSHOT_UNITS D4 raid2 D1

## See also

SET *unit*
DELETE *container-name*

## ADD SPARESET

This command adds a disk drive to the spareset and initializes the metadata on the drive. The spareset is a pool of drives available to the controller to replace failing members of storagesets.

### Syntax

ADD SPARESET *disk-name*

### Parameters

*disk-name*

Indicates the name of the disk drive being added to the spareset. Only one disk drive can be added to the spareset with each ADD SPARESET command.

### Switches

There are no switches associated with this command.

### Example

To add a disk drive named DISK20200 and DISK30300 to a spareset, enter:

```
ADD SPARESET DISK20200
ADD SPARESET DISK30300
```

### See also

DELETE SPARESET
SHOW SPARESET
SHOW STORAGESETS

## ADD STRIPESET

This command creates and names a stripeset and adds it to the controller configuration. Stripesets are also referred to as RAID level 0 storagesets. The number of members in the stripeset is determined by the number of *container-name* parameters specified.

### Syntax

ADD STRIPESET *stripeset-name container-name1 container-name2 [container-nameN]*

### Parameters

The following parameters are required for the ADD STRIPESET command:

- stripeset name

- container names

These parameters are described in the paragraphs that follow.

*stripeset-name*

Assigns a name to the stripeset. This is the name used with the ADD UNIT command to identify the stripeset as a host-addressable unit.

*container-name1 container-name2 [container-nameN]*

Identifies the members (disk drives or mirrorsets) making up the stripeset. Stripesets can contain between 2 and 14 members.

The container name must start with a letter (A through Z) and may consist of a maximum of nine characters including letters A through Z, numbers 0 through 9, periods (.), dashes (-), or underscores (_).

It is common to name a stripeset STRIPE*n*, where n is a sequentially assigned, unique identifier. This naming convention presents to the user both the type of container and its unique identifier.

> **NOTE:** A 256-character limit exists for the command line. If you are configuring a stripeset with many members, you may have to rename the members to shorter names in order to enter the entire command line.

### Switches

There are no switches associated with this command.

**Examples**

To create a stripeset named STRIPE1 with three disks: DISK10000, DISK20100, and DISK30200, enter:

ADD STRIPESET STRIPE1 DISK10000 DISK20100 DISK30200

To create a two-member striped mirrorset (a stripeset whose members are mirrorsets) and create a logical unit from it, enter the following (because you can initialize the stripeset, you do not need to individually initialize the mirrorsets):

ADD MIRRORSET MR1 DISK10000 DISK20100
ADD MIRRORSET MR2 DISK30200 DISK40300
ADD STRIPESET STRIPE1 MR1 MR2
INITIALIZE STRIPE1
ADD UNIT D104 STRIPE1

**See also**

ADD UNIT
ADD MIRRORSET
DELETE *container-name*
INITIALIZE
SHOW STORAGESET
SHOW STRIPESET
SHOW *stripeset-name*

# ADD UNIT

> ⚠ **CAUTION:** Adding a unit immediately makes that unit available to all host connections unless the *DISABLE_ACCESS_PATH* switch is specified. There is more information about this under the description of the *ENABLE_ACCESS_PATH/DISABLE_ACCESS_PATH* switch.

The ADD UNIT command creates a logical unit for the host connection(s) to access. All requests by the host(s) to the unit are mapped as requests to the container specified in the ADD UNIT command. Units can be created from all container types:

■ disk devices

■ stripesets

■ mirrorsets

■ RAIDsets

■ partitions

■ striped mirrorsets

## Syntax

ADD UNIT *unit-number container-name*

## Parameters

The following parameters are required for the ADD UNIT command:

■ unit number

■ container name

These parameters are described in the paragraphs that follow.

*unit-number*

The unit number is the designation by which the controller keeps track of the unit. This number maps to one or more host-accessible LUNs. Unit numbers are prefixed by D and are in the range of 0—199. How units are assigned to ports depends on the failover mode.

■ Transparent failover mode:

  ❏ D0 to D99 are units assigned to port 1

  ❏ D100 to D199are units assigned to port 2

■ Multiple-bus failover mode:

  ❏ D0 to D199 are units visible to the hosts through both ports of both controllers

In transparent failover mode, adding unit D0 creates a logical unit and presents it as D0 to the host on port 1. Adding unit D100 creates a logical unit and presents it as D0 to the host on port 2.

Partitioned units cannot be split across ports.

In multiple-bus failover mode, adding unit D0 creates a logical unit that is presented to both ports on both controllers. What hosts can see the unit is determined by parameters of the ADD CONNECTIONS command and by the *ENABLE/DISABLE_ACCESS_PATH* switch of the ADD UNIT command.

The LUN number a host connection assigns to a unit is a function of the *UNIT_OFFSET* switch in the ADD (or SET) CONNECTIONS command. LUN number equals unit number minus offset:

LUN number = unit number – offset

If no value is specified for the *UNIT_OFFSET* switch in the ADD (or SET) CONNECTIONS command, the offsets are as follows:

❏ In transparent failover mode, host connections on port 1 have an offset of 0 and host connections on port 2 have an offset of 100.

❏ In multiple-bus failover mode, all ports have a default offset of 0.

*container-name*

Specifies the name of the container (disk drive, device, storageset, or partition) that is to become the unit.

A unit can consist of a maximum of 48 devices.

## Switches

Table 2–1 lists all switches for the ADD UNIT command and identifies which switches may be used with each type of device or storageset. Descriptions of each switch follow the table.

| Switch | Container Type | RAIDset | Stripeset | Mirrorset | NOTransportable Disk | Transportable Disk |
|---|---|---|---|---|---|---|
| *ENABLE_ACCESS_PATH*<br>*DISABLE_ACCESS_PATH* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *PARTITION=partition-number* | | ✓ | ✓ | ✓ | ✓ | |
| *MAXIMUM_CACHE_TRANSFER_SIZE* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *PREFERRED_PATH*<br>*NOPREFERRED_PATH* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *READ_CACHE*<br>*NOREAD_CACHE* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *READAHEAD_CACHE*<br>*NOREADAHEAD_CACHE* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *RUN*<br>*NORUN* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *WRITE_PROTECT*<br>*NOWRITE_PROTECT* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *WRITEBACK_CACHE*<br>*NOWRITEBACK_CACHE* | | ✓ | ✓ | ✓ | ✓ | |

**Table 2–1 ADD UNIT Switches for New Containers**

**NOTE:** The *NORUN* switch cannot be specified for partitioned units.

*ENABLE_ACCESS_PATH=*
*DISABLE_ACCESS_PATH=*

Determines which host connections can access the unit. The allowable variables are:

❏ ALL

❏ a string of connection names, separated by commas and enclosed in parentheses.

Setting *ENABLE_ACCESS_PATH=ALL* permits all host connections to access the unit. This is the default.

> **CAUTION:** If the storage subsystem has more than one host connection, access path must be specified carefully to avoid giving undesirable host connections access to the unit. The default condition is that access paths to all host connections are enabled. To restrict host access to a set of host connections, specify *DISABLE_ACCESS_PATH=ALL* when the unit is added, then use the SET *unit* command to specify the set of host connections that are to have access to the unit.
>
> Enabling the access path to a particular host connection does not override previously enabled access paths. All access paths previously enabled are still valid; the new host connection is simply added to the list of connections that can access the unit.
>
> The procedure of restricting access by enabling all access paths then disabling selected paths is particularly <u>not</u> recommended because of the potential data/security breach that occurs when a new host connection is added.

*MAXIMUM_CACHE_TRANSFER_SIZE=32 (Default)*
*MAXIMUM_CACHE_TRANSFER_SIZE=n*

Sets the largest number of write blocks to be cached by the controller. The controller will not cache any transfers over the specified size. Accepted write block sizes are 1 through 2048.

The MAXIMUM_CACHE_TRANSFER_SIZE switch affects both read and write-back cache when set on a controller that has read and write-back caching.

*PARTITION=partition_number*

Identifies the partition-number for a partition on a container. The *partition-number* identifies the partition associated with the unit number being added. Use the SHOW *container-name* command to find the partition numbers used by a storageset or a single-disk unit.

> **NOTE:** Do not split partitioned units across ports.

*PREFERRED_PATH=OTHER_CONTROLLER*
*PREFERRED_PATH=THIS_CONTROLLER*
*NOPREFERRED_PATH (Default)*

This qualifier is effective only in multiple-bus failover mode. The preferred controller is the controller through which the unit is initially on line. This preference is only temporary, and can be changed by the host sending a SCSI Start Unit command.

If *PREFERRED_PATH is set* for one partition, all partitions on that container inherit the same path.

*READ_CACHE (Default)*
*NOREAD_CACHE*

If *READ_CACHE* is selected, when the controller receives a read request from the host, it reads the data from the disk drives, delivers it to the host, and stores the data in its cache module. Subsequent reads for the same data will take the data from cache rather than accessing the data from the disks.

Read caching improves performance in almost all situations. Therefore, it is recommended you leave its default setting, *READ_CACHE*, enabled. However, under certain conditions, such as when performing a backup, read caching may not be necessary since only a small amount of data is cached. In such instances, it may be beneficial to disable the read cache function and remove the processing overhead associated with caching data.

*READAHEAD_CACHE (Default)*
*NOREADAHEAD_CACHE*

Enables the controller to keep track of read I/Os. If the controller detects sequential read I/Os from the host, it will then try to keep ahead of the host by reading the next sequential blocks of data (those the host has not yet requested), and put the data in cache. This process is sometimes referred to as prefetch. The controller can detect multiple sequential I/O requests across multiple units.

Read-ahead caching improves host application performance since the data will be read from the controller cache rather than from the disk array. Read-ahead caching is the default for units.

If you are adding a unit that is not expected to get sequential I/O requests, select *NOREADAHEAD_CACHE* for the unit.

*RUN (Default)*
*NORUN*

Controls the unit's availability to the host:

■ Specify *RUN* to make a unit available to the host.

■ Specify *NORUN* to make a unit unavailable to the host and to cause any data in cache to be flushed to one or more drives. *NORUN* spins down all the disks used in the unit. The drives making up the unit spin down after the data has been completely flushed.

**NOTE:** Do not specify the *RUN* and *NORUN* switches for partitions.

*WRITE_PROTECT (Default)*
*NOWRITE_PROTECT*

Specifies whether data contained on the selected unit can be overwritten:

■ Specify *WRITE_PROTECT* to prevent host write operations to the unit. However, the controller may still write to a write-protected RAIDset to satisfy a reconstruct pass or to reconstruct a newly replaced member. Additionally, metadata, reconstruct, and copy writes are still allowed to RAIDsets and mirrorsets.

■ Specify *NOWRITE_PROTECT* to allow the host to write data to the unit. This allows the controller to overwrite existing data. *NOWRITE_PROTECT* is the default for transportable disks.

*WRITEBACK_CACHE (Default)*
*NOWRITEBACK_CACHE*

Specifying *WRITEBACK_CACHE* allows the controller to declare the write operation "complete" as soon as the data reaches its cache memory. The controller performs the slower operation of writing the data to the disk drives at a later time.

*WRITEBACK_CACHE* cannot be applied to transportable disks.

> **CAUTION:** Though there is built-in redundancy to protect data contained in cache, allowing data to be written to write-back cache may result in the loss of data if the controller fails.

Specifying *NOWRITEBACK_CACHE* enables only write-through caching. In write-through caching, when the controller receives a write request from the host it places the data in its cache module, writes the data to the disk drives, then notifies the host when the write operation is complete. This process is called write-through caching because the data passes through—and is stored in—the cache memory on its way to the disk drives. Write-through caching is only enabled when write-back caching is disabled.

> **NOTE:** If the switch is changed from *WRITEBACK_CACHE* to *NOWRITEBACK_CACHE*, the controller may take up to five minutes to flush data contained within cache memory.

**Examples**

To create unit D102 from a single-disk drive named DISK10000 and set the preference to be "this controller," enter:

ADD UNIT D102 DISK10000 PREFERRED_PATH=THIS_CONTROLLER

■ If the controller pair is in transparent failover mode, this unit is on port 2. It will be presented to each host connection as LUN (2 – offset). Offset is a host connection characteristic, specified in the ADD (or SET) CONNECTIONS command.

■ If the controller pair is in multiple-bus failover mode, this unit can potentially be seen by all four controller ports. It will be presented to each host connection as LUN (2 – offset). Offset is a host connection characteristic, specified in the ADD (or SET) CONNECTIONS command.

To create unit D5 from mirrorset MIRR1, and to let only two host connections (Server1 and Server2) have access to this unit, enter:

ADD UNIT D5 MIRR1 DISABLE_ACCESS_PATH=ALL
SET UNIT D5 ENABLE_ACCESS_PATH=(SERVER1,SERVER2)

■ If the controller pair is in transparent failover mode, this unit is on port 1, and is available only to host connections Server1 and Server2. Server1 and Server2 must be connected to host port 1 to access the unit. The unit will be presented to each host connection as LUN (5 – offset). Offset is a host connection characteristic, specified in the ADD (or SET) CONNECTIONS command.

■ If the controller pair is in multiple-bus failover mode, this unit can potentially be seen by all four controller ports, but is accessible only by host connections Server1 and Server2. It will be presented to each host connection as LUN (5 – offset).

**See also**

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

# CLEAR_ERRORS CLI

This command stops the display of current or previous error messages at the CLI prompt. Errors are displayed (even after the error condition has been corrected) until the controller is restarted or the CLEAR_ERRORS CLI command is entered.

**NOTE:** This command does not clear the error condition, it only stops the CLI from displaying it.

## Syntax

CLEAR_ERRORS CLI

## Parameters

There are no parameters associated with this command.

## Switches

There are no switches associated with this command.

## Example

CLEAR_ERRORS CLI

## See also

CLEAR_ERRORS *controller* INVALID_CACHE
CLEAR_ERRORS *unit-number* LOST_DATA
CLEAR_ERRORS *device-name* UNKNOWN
CLEAR_ERRORS *unit-number* UNWRITEABLE_DATA

## CLEAR_ERRORS *this_controller* INVALID_CACHE
## CLEAR_ERRORS *other_controller*  INVALID_CACHE

This command clears an invalid cache state and allows the controller and cache to resume operation. If the error is due to an correct mirrored-cache configuration, the controller indicates mirrored mode status after the error is cleared.

Use this command for the following situations:

■ When the controller or cache modules have been replaced resulting in mismatched data between the controllers.

■ When the controller or cache module is replaced while data is still in cache and not properly flushed with the SHUTDOWN or SET NOFAILOVER commands.

### Syntax

CLEAR_ERRORS *controller* INVALID_CACHE

### Parameters

*controller*

Identifies which controller is to receive the CLEAR_ERRORS command. You must specify *this_controller* (the one connected to the CLI maintenance terminal) or *other_controller.*

*data retention policy*

> △ **CAUTION:** Specifying the *destroy_unflushed_data* parameter destroys data remaining in cache, which can result in data loss.

Instructs the controller how to handle write-back cached data:

■ *nodestroy_unflushed_data* —the controller synchronizes its metadata with that of the cache module and preserves any unwritten data for eventual flushing to the disk array.

The *nodestroy_unflushed_data* parameter is used in the following situations:

❏ If the controller module has been replaced.

❏ If the controller's nonvolatile memory has lost its contents.

■ *destroy_unflushed_data*—the controller discards unwritten cache data.

The *destroy_unflushed_data* switch is used in the following situations:

❏ If the cache module has been replaced.

❏ Any other reason not listed above.

## Examples

> **NOTE:** Because misuse of this command can cause data loss, both the command and the switch *destroy_unflushed_data* or *nodestroy_unflushed_data* must be entered in their entirety. They cannot be abbreviated.

To clear an invalid cache error on "this controller" after replacing the controller module, enter the following command on one line:

    CLEAR_ERRORS THIS_CONTROLLER INVALID_CACHE NODESTROY_UNFLUSHED_DATA

To clear an invalid cache error on the "other controller" after a cache module has been replaced, enter the following command on one line:

    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 *device-name* UNKNOWN

Use this command to force the controller to recognize a failed device, regardless of the controller's prior evaluation of the device's condition.

A device failure may cause the controller to label the device as unknown. When that happens, the controller does not check the device again to see if it has been repaired or if the error condition has been corrected. You must enter this command so the controller can recognize the device after the cause of the error has been corrected.

### Syntax

CLEAR_ERRORS *device-name* UNKNOWN

Spell out UNKNOWN when using this command.

### Parameters

*device-name*

Identifies the device with the unknown error.

### Switches

There are no switches associated with this command.

### Example

To force the controller to recognize a previously unknown device named DISK30000. enter:

CLEAR_ERRORS DISK30000 UNKNOWN

### See also

CLEAR_ERRORS CLI
CLEAR_ERRORS INVALID_CACHE
CLEAR_ERRORS UNKNOWN
CLEAR_ERRORS UNWRITEABLE_DATA

# CLEAR_ERRORS *unit-number* **LOST_DATA**

This command clears lost data errors on a unit; all partitions on the unit's container are affected.

The controller reports a lost data error on the unit when you remove a write-back cache module or when the cache module contains unflushed data, possibly due to an interruption in the primary power source with no backup power present. The CLEAR_ERRORS *unit_number* LOST_DATA command clears the lost data error but does not recover the lost data.

> **NOTE:** Clearing lost data errors or lost data block errors on a RAIDset causes a reconstruction of all parity blocks. Clearing lost data errors or lost data block errors on a mirrorset causes members to normalize.

## Syntax

CLEAR_ERRORS *unit-number* LOST_DATA

> **NOTE:** Because misuse of this command can cause data loss, the command must be entered in its entirety. It cannot be abbreviated.

> **CAUTION:** This command may cause data loss.

## Parameters

*unit-number*

Identifies the unit on which the lost data error is to be cleared. The *unit-number* is the same name given to the unit when you added it to the controller's configuration with the ADD UNIT command.

## Switches

There are no switches associated with this command.

### Example

To clear the lost data error on disk unit number D103, enter:

```
CLEAR_ERRORS D103 LOST_DATA
```

### See also

CLEAR_ERRORS CLI
CLEAR_ERRORS INVALID_CACHE
CLEAR_ERRORS UNKNOWN
CLEAR_ERRORS UNWRITEABLE_DATA

# CLEAR_ERRORS *unit-number* **UNWRITEABLE_DATA**

This command clears an unwriteable data error on a unit.

This command affects all partitions on the same container.

If a storageset or disk drive fails before its data has been written to it, the controller reports an unwriteable data error. The CLEAR_ERRORS *unit-number* UNWRITEABLE_DATA command removes the data from the cache and clears the unwriteable data error.

> **CAUTION:** This command will cause data loss.

## Syntax

CLEAR_ERRORS *unit-number* UNWRITEABLE_DATA

> **NOTE:** Because misuse of this command will cause data loss, the command must be entered in its entirety. It cannot be abbreviated.

## Parameters

*unit-number*

Identifies the unit having the unwriteable data error. The *unit-number* is the name given to the unit when it was created with the ADD UNIT command.

## Switches

There are no switches associated with this command.

## Example

To clear the unwriteable data error on disk unit D103, enter:

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

## CONFIGURATION RESET

This command erases the entire subsystem configuration data file stored in the NVRAM of "this controller." The command sets the controller to a default configuration and shuts down the controller.

> **NOTE:** Before issuing this command, ensure the subsystem configuration is saved.

- In single-controller subsystems, the configuration is saved on the disk array. If a container has the *SAVE_CONFIGURATION* switch specified when it is initialized (through the INITIALIZE command), the disks that constitute that unit will be able to save a copy of the configuration.
- For dual-redundant controllers, the configuration is stored in the companion controller.

Specify the CONFIGURATION RESET command on "this controller" in nofailover mode only. Enter this command to ensure all of the old configuration information is removed when a controller module is moved from one subsystem to another.

> **NOTE:** This command disables communication between host and controller.

You can also initiate the CONFIGURATION RESET command from the controller operator control panel (OCP) by holding in port button 5 and pressing the RESET button. Hold the Port button until it starts flashing; then release the button.

### Syntax

CONFIGURATION RESET

### Parameters

There are no parameters associated with this command.

### Switches

There are no switches associated with this command.

### Example

To erase the configuration information in "this controller" enter:

CONFIGURATION RESET

**See also**

      CONFIGURATION RESTORE
      CONFIGURATION SAVE
      INITIALIZE

## CONFIGURATION RESTORE

This command copies the subsystem configuration information from the configuration file saved on the disk array into the "this controller's" non-volatile memory. This command locates the most recent configuration file created on disk and restores it. This command causes a reboot and takes effect immediately.

> **NOTE:** Use this command for a single controller configuration only. Do not use it for controllers in a dual-redundant configuration. (The only time this command is used for a dual-controller configuration is for a remote copy environment. See *Data Replication Manager HSG80 ACS Version 8.5P Operations Guide* for examples of appropriate usage.)

You can also initiate the CONFIGURATION RESTORE command from the controller operator control panel (OCP) by holding in port button 6 and pressing the RESET button. Hold the Port button until it starts flashing; then release the button.

### Syntax

CONFIGURATION RESTORE

### Parameters

There are no parameters associated with this command.

### Switches

There are no switches associated with this command.

### Example

To restore the subsystem configuration into the memory of the controller in a single-controller subsystem, enter:

CONFIGURATION RESTORE

### See also

CONFIGURATION RESET
CONFIGURATION SAVE
INITIALIZE

## CONFIGURATION SAVE

> **NOTE:** This command is appropriate only for a single-controller subsystem. (The only time this command is used for a dual-controller configuration is for a remote copy environment. See *Data Replication Manager HSG80 ACS Version 8.5P Operations Guide* for examples of appropriate usage.)

This command forces the controller to copy the subsystem configuration information in its memory onto a configuration file on the disk array. The command takes effect immediately.

The controller periodically updates the configuration file on the disk array. This command forces an immediate update. In single-controller subsystems, the configuration is saved on the disk array. If a container had the *SAVE_CONFIGURATION* switch specified when it was initialized (through the INITIALIZE command), the disks that constitute that unit will be able to save a copy of the configuration.

### Syntax

CONFIGURATION SAVE

### Parameters

There are no parameters associated with this command.

### Switches

There are no switches associated with this command.

### Example

To save the configuration information of a single-controller subsystem, enter:

CONFIGURATION SAVE

### See also

CONFIGURATION RESET
CONFIGURATION RESTORE
INITIALIZE

# CREATE_PARTITION

This command divides a disk drive or storageset into several separately addressable storage units. The command marks a specified percentage of a disk drive or storageset to be used as the partition.

Initialize all disks and storagesets before creating partitions.

## Syntax

CREATE_PARTITION *container-name SIZE=percent*

## Parameters

*container-name*

Identifies the disk or storageset to partition. This is the same name given to the disk or storageset when it was created with the ADD xxxx command (for example, ADD DISK, ADD STRIPESET, and so forth). Any disk, stripeset, mirrorset, striped mirrorset, or RAIDset can be partitioned. A transportable disk cannot be partitioned. You must initialize the container before creating the first partition.

## Switches

The following switches are used with the CREATE_PARTITION command:

■ size

■ geometry switches

  ❏ cylinders

  ❏ heads

  ❏ sectors per track

These switches are described in the following paragraphs.

*SIZE=percent*
*SIZE=LARGEST*

Specifies the size of the partition to be created as a percentage of the total container's storageset size:

■ Specify *SIZE=percent t*o create a partition that is a percentage of the container's total capacity.

> **NOTE:** The resulting partition is slightly smaller than the size specified because metadata also occupies some of partition's allocated space.

■ Specify *SIZE=LARGEST* to:

❏ Have the controller create the largest partition possible from unused space on the disk or storageset.

❏ Create the last partition on a container. Because the remaining space is not equal to an exact percentage value, specifying LARGEST allows you to optimize use of the remaining space.

*CAPACITY=* (1 to the maximum container size in blocks)
*CYLINDERS=* (1 to 16,777,215)
*HEADS=* (1 to 255)
*SECTORS_PER_TRACK=* (1 to 255)

These are used to set the geometry reported to the host.

> **NOTE:** The geometry switches should not be used unless there is a compatibility problem with the existing defaults.
>
> The geometry switches for the INITIALIZE command are ignored when you create partitions. The geometry switches supplied with the CREATE_PARTITION command are used by the unit.

## Example

To create a RAIDset named RAID9, divide it into four equal parts, and create host-addressable units for each partition, enter:

```
ADD RAIDSET RAID9 DISK10000 DISK20100 DISK30200
INITIALIZE RAID9
CREATE_PARTITION RAID9 SIZE=25
CREATE_PARTITION RAID9 SIZE=25
CREATE_PARTITION RAID9 SIZE=25
CREATE_PARTITION RAID9 SIZE=LARGEST
ADD UNIT D101 RAID9 PARTITION=1
ADD UNIT D102 RAID9 PARTITION=2
ADD UNIT D103 RAID9 PARTITION=3
ADD UNIT D104 RAID9 PARTITION=4
```

**See also**

ADD UNIT
DELETE *unit-number*
DESTROY PARTITION
SET *unit-number*
SHOW

## DELETE *associations*

> **NOTE:** This command works only in a Data Replication Manager (DRM) environment and requires V8.5P code. Like all DRM commands, it is heavily restricted in its use. See *Data Replication Manager HSG80 ACS Version 8.5P Operations Guide* for examples of usage.

This command deletes an existing association set.

### Syntax

DELETE *association-set-name*

### Parameters

association-set-name

The name of the association set to be deleted.

### Switches

There are no switches associated with this command.

### Examples

To delete association set AS4, enter:

DELETE AS4

See also

ADD ASSOCIATIONS
SET ASSOCIATIONS

## **DELETE** *concatsets*

This command deletes an existing concatset. If the concatset is associated with a unit, the unit must be deleted before deleting the concatset.

### Syntax

DELETE *concatset-name*

### Parameters

concatset-name

> The name of the concatset to be deleted.

### Switches

There are no switches associated with this command.

### Examples

To delete concatset C1, which is associated with unit D0, enter:

> DELETE D0

> DELETE C1

See also

DELETE *unit-number*

## DELETE *connections*

This command deletes a specified connection from the table of known connections maintained by the controller.

The table of known host connections is maintained in the controller's memory. Once a connection is added to the table, it stays there, even if the physical connection between host adapter and controller port is severed. The only way to remove a connection from the table is through the DELETE *connections* command.

DELETE *connections* removes the connection from the table whether or not the host adapter is still physically connected to a controller port.

> **NOTE:** A connection that has access path <u>explicitly</u> enabled on a unit cannot be deleted. Access path is enabled explicitly through the *ENABLE_ACCESS_PATH* qualifier of the ADD UNIT or SET *unit* commands. If access path is generically enabled for all connections (*ENABLE_ACCESS_PATH= ALL*), then any or all connections can be deleted. A connection with explicit access path must have the access path disabled (through the *DISABLE_ACCESS_PATH* qualifier of the SET *unit* command) before the connection can be deleted.

### Syntax

DELETE *connection-name*

### Parameters

*connection-name*

The name given to the host connection. The connection name is one of the following:

- The default name assigned to the host connection when it was physically connected to the controller port. Default names are of the form !NEWCON*nn*.

- the name given through the RENAME command

- the name given through the ADD CONNECTIONS command

### Switches

There are no switches associated with this command.

**Examples**

To delete host connection Server1 from the table of known connections (unless the access path to Server1 is specifically enabled for one or more unit), enter:

    DELETE SERVER1

To delete host connection Server4 from the table of known connections when Server4 has explicit access path to unit D7, enter:

    SET D7 DISABLE_ACCESS_PATH=SERVER4
    DELETE SERVER4

**See also**

ADD CONNECTIONS
ADD UNIT
SET *connection-name*
SET *unit-number*

## **DELETE** *remote copy sets*

**NOTE:** This command works only in a Data Replication Manager (DRM) environment and requires V8.5P code. Like all DRM commands, it is heavily restricted in its use. See *Data Replication Manager HSG80 ACS Version 8.5P Operations Guide* for examples of usage.

**NOTE:** This command works only on the site acting as initiator. This is the local site except in a failover situation.

This command deletes a specified remote copy set. Upon completion of this command, the target unit(s) will no longer be a member of the remote copy set.

This command is only valid if the error mode of the remote copy set is normal. The error mode is set by the *ERROR_MODE* switch of the SET *remote copy set* command.

If the remote copy set is a member of an association set, the association set must be deleted before the remote copy set is deleted.

### Syntax

DELETE *remote_copy_set_name*

### Parameters

*remote_copy_set_name*

The name given to the remote copy set that is to be deleted.

### Switches

There are no switches associated with this command.

### Examples

To delete RCS1,enter:

DELETE RCS1

### See also

ADD REMOTE COPY SET
SET remote copy set

# DELETE *container-name*

This command deletes a container belonging to the controller's configuration. You cannot delete a container in use by a higher-level container. For example, you cannot delete a disk belonging to a RAIDset, or a RAIDset belonging to a unit; you must first delete the higher-level container or containers. Likewise, you cannot delete a unit that belongs to a remote copy set; the remote copy set must be deleted first. (There is one exception to this rule: a remote (target) member of a remote copy set can be deleted from the copy set.)

> **NOTE:** This command does not delete sparesets or failedsets. See the DELETE FAILEDSET and DELETE SPARESET commands for details about deleting sparesets and failedsets.

When a storageset is deleted, the individual disks are free to be used by another container. If you create the container again with the exact same disk configuration, and none of the disks have been used for anything else, or initialized, then the container can be reassembled using its original disks.

## Syntax

DELETE *container-name*

## Parameters

container-name

Identifies the container to be deleted. This is the name given to the container when it was created using the ADD command (for example, ADD DISK, ADD STRIPESET, and so forth).

## Switches

There are no switches associated with this command.

## Examples

To delete a disk drive named DISK10000, enter:

DELETE DISK10000

To delete a stripeset named STRIPE1, enter:

DELETE STRIPE1

**See also**

DELETE FAILEDSET
DELETE SPARESET
UNMIRROR

# DELETE FAILEDSET

This command removes a disk drive from the failedset. The failedset contains disk drives that were removed from service either by the controller or by the user (for example, through the command SET *mirrorset-name REMOVE=disk-name)*. Only one disk drive can be deleted from the failedset at a time.

Enter the DELETE FAILEDSET command before physically removing failed members from the storage shelf for testing, repair, or replacement.

Disks that the controller placed in the failedset should be considered defective and should be repaired or replaced.

## Syntax

DELETE FAILEDSET *disk-name*

## Parameters

*disk-name*

Identifies the disk you want to delete from the failedset. Only one disk at a time can be removed from the failedset.

## Switches

There are no switches associated with this command.

## Example

To delete DISK20200 from the failedset, enter:

DELETE FAILEDSET DISK20200

## See also

SET FAILEDSET
SHOW FAILEDSET

## DELETE SPARESET

This command removes a disk drive from the spareset.

### Syntax

DELETE SPARESET *disk-name*

### Parameters

*disk-name*

Identifies the disk drive being deleted from the spareset. Remove only one disk at a time from a spareset.

### Switches

There are no switches associated with this command.

### Example

To remove DISK20300 from the spareset, enter:

DELETE SPARESET DISK20300

### See also

ADD SPARESET
SHOW SPARESET

# **DELETE** *unit-number*

This command deletes a unit from the subsystem configuration.

The DELETE command flushes any user data from the write-back cache to the disks and deletes the unit. That is, the command deletes the association of the unit with its underlying device, storageset, RAIDset, or partition. The container and any data in it are left intact.

If any errors occur when trying to flush the user data, the logical unit is not deleted. To delete a unit that has existing errors, you must clear all the errors using a CLEAR_ERRORS command.

### Syntax

DELETE *unit-number*

### Parameters

*unit-number*

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

### Switches

There are no switches associated with this command.

### Example

To delete unit number D103, enter:

DELETE D103

**See also**

ADD UNIT
CLEAR_ERRORS LOST_DATA
CLEAR_ERRORS UNWRITEABLE_DATA
DELETE *container-name*
DELETE FAILEDSETS
DELETE SPARESETS
DESTROY_PARTITION

# DESTROY_PARTITION

This command removes a partition from a container. The freed area is then consolidated with any adjacent free areas.

> ⚠ **CAUTION:** Data contained on a partition is lost when you enter the DESTROY_PARTITION command.

To destroy a partition that has been assigned a unit number, delete the unit (DELETE *unit-number*) then delete the partition.

After a partition is destroyed, the container from which it was created must be initialized.

## Syntax

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

## Parameters

The following parameters are available to the DESTROY_PARTITION command:

- container-name
- partition-number

These parameters are described in the paragraphs that follow.

*container-name*

Identifies the disk or storageset containing the partition you want to eliminate. This is the name given to the container when it was created using the ADD xxxx command (for example, ADD DISK, ADD STRIPESET, and so forth).

*partition-number*

Identifies the partition to be destroyed. Use the SHOW *container-name* command to identify the correct partition before carrying out the DESTROY_PARTITION command.

## Switches

There are no switches associated with this command.

**Example**

D102 is a unit created from partition 2 of raidset RAID9. To delete this partition, enter:

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

If partition 2 of RAID9 has not yet been assigned a unit number, to delete partition 2, enter:

```
DESTROY_PARTITION RAID9 PARTITION=2
```

**See also**

CREATE_PARTITION
DELETE *unit-number*
SHOW

# DIRECTORY

This command lists the diagnostics and utilities available on "this controller."

## Syntax

DIRECTORY

## Parameters

There are no parameters associated with this command.

## Switches

There are no switches associated with this command.

## Example

To display a directory listing, enter:

DIRECTORY

```
HSUTIL    V85G D
FRUTIL    V85G D
CHVSN     V85G D
CLCP      V85G D
CLI       V85G D
CLONE     V85G D
CONFIG    V85G D
DILX      V85G D
DIRECT    V85G D
DSTAT     V85G D
FMU       V85G D
VTDPY     V85G D
```

> **NOTE:** CHVSN and DSTAT are not user utilities. They may be used by Compaq authorized service personnel only.

## See also

RUN

## HELP

This command displays a brief explanation of how to obtain help on any command or CLI function by using the question mark ( ?). A character space must precede the question mark.

### Syntax

HELP

### Parameters

There are no parameters associated with this command.

### Switches

There are no switches associated with this command.

### Example

To display information about how to get CLI help, enter:

HELP

To get more information about the different kind of SET commands, enter:

SET ?

# INITIALIZE

This command initializes the specified container. During initialization, a small amount of disk space is reserved for data that the controller uses to keep track of the array. This data, called metadata, is inaccessible to the host. Disks made transportable (*TRANSPORTABLE* switch of the ADD DISK command) do not contain metadata.

Units are not initialized; a container is initialized before it is assigned to a unit.

## Syntax

INITIALIZE *container-name*

> **CAUTION:** The INITIALIZE command destroys all user data on the container unless you enter the *NODESTROY* switch. The *NODESTROY* switch is only valid on mirrorsets and striped mirrorsets.

> **CAUTION:** If a partitioned container is initialized, all partitions are eliminated.

If you initialize a transportable disk, any metadata contained on the disk is destroyed and the entire disk drive is accessible by the host. The drive does not have the error detection and data security provided by the metadata that is on notransportable disks.

Use the INITIALIZE command when:

- creating a unit from a newly installed disk

- creating a unit from a newly created RAIDset, stripeset, or mirrorset

- initializing the data structure of a previously partitioned container

Do not use the INITIALIZE command when:

- creating a unit from the same disks previously initialized, such as when a RAIDset is moved

- creating a storageset from existing members

- adding a RAIDset with the REDUCED switch

## Parameters

*container-name*

> Specifies the container to initialize. This is the same name given to the disk or storageset when it was created using the ADD xxxx command (for example, ADD DISK, ADD STRIPESET, and so forth).

## Switches

The following switches are used by this command:

- geometry switches

  - ❏ capacity

  - ❏ cylinders

  - ❏ heads

  - ❏ sectors per track

- chunksize

- destroy user data

- save configuration

These switches are described in the following paragraphs.

*CAPACITY=* (1 to the maximum container size in blocks)
*CYLINDERS=* (1 to16,777,215)
*HEADS=* (1 to 255)
*SECTORS_PER_TRACK=* (1 to 255)

These are used to set the geometry switches reported to the host.

> **NOTE:** The geometry switches should not be used unless there is a compatibility problem with the existing defaults.
>
> The geometry switches for the INITIALIZE command are ignored when you create partitions. The geometry switches supplied with the CREATE_PARTITION command are used by the unit.

*CHUNKSIZE=DEFAULT (Default)*
*CHUNKSIZE=n*

> Specifies the block chunk size to be used for RAIDsets and stripesets. You can specify the chunk block size by entering *CHUNKSIZE=n*, or allow the controller to set the chunk size by entering *CHUNKSIZE=DEFAULT*.

The default chunk size for storagesets with less than or equal to nine members is 256 blocks, or 128 kilobytes (K). The default chunk size for storagesets with more than nine members is 128 blocks, or 64K.

---

**IMPORTANT:** The default chunk size should be used with caution. It is imperative that you analyze the I/O transfers being used with your application to determine the proper chunk size. An improper value used as a chunk size may cause serious performance problems with the subsystem.

---

*DESTROY (Default)*
*NODESTROY*

Controls how the metadata on the initialized container is to be handled:

**NOTE:** The *DESTROY* and *NODESTROY* switches are only valid with mirrorsets and striped mirrorsets.

■ Specify *NODESTROY* to preserve forced error metadata during the initialization process. Use the *NODESTROY* switch only when a unit is to be created from disk drives reduced from mirrorsets. This allows the data on the container to be accessed by a mirrorset or striped mirrorset unit. The *NODESTROY* switch is not valid for RAIDsets, simple stripesets, and single-disk configurations.

■ Specify *DESTROY* to overwrite user data and forced error flags during the initialization.

*SAVE_CONFIGURATION*
*NOSAVE_CONFIGURATION (Default)*

This switch is for a single-controller configuration only. This switch reserves an area on each of the disks that constitute the container being initialized. The controller can write subsystem configuration data on this area. If the controller is replaced, the new controller can read the subsystem configuration from the reserved area of disks.

If you specify *SAVE_CONFIGURATION* for a multi-device storageset, such as a stripeset, the complete subsystem configuration is periodically written on each disk in the storageset.

The SHOW DEVICES *FULL* command shows which disks are used to backup configuration information.

## Examples

To initialize RAID4 and to reserve an area for configuration information on all the disks that constitute RAID4, enter:

INITIALIZE RAID4 SAVE_CONFIGURATION

To initialize stripeset STRIPE1 with the default chunk size, enter:

**NOTE:** Note that the chunk size is not specified, so the controller initializes the unit with the default chunk size.

INITIALIZE STRIPE1

To initialize RAIDset RAID9 with a chunk size of 20, enter:

INITIALIZE RAID9 CHUNKSIZE=20

To initialize DISK40400 and preserve the data after it is removed (reduced) from a mirrorset, enter:

REDUCE DISK40400
INITIALIZE DISK40400 NODESTROY

## LOCATE

This command indicates the physical location of configured units, storagesets, and devices by flashing the device fault LED on the front of the storage building block (SBB). The LOCATE command causes the fault LED to flash once per second until turned off with the LOCATE CANCEL command. The LOCATE command can also be used to test the LED itself.

The device fault LED on a failed device stays on continuously. When located, the device fault LED on a good device flashes. The flashing LED helps to distinguish between located devices and failed devices. The device fault LED on failed devices stays on after the LOCATE CANCEL command is entered.

Not all devices have a device fault LED. Therefore, they do not appear to respond to the LOCATE command.

### Syntax

LOCATE *parameter*

### Parameters

The following parameters are available to the LOCATE command:

■ all

■ cancel

■ disks

■ pttll

■ units

■ container

■ unit

These parameters are described in the paragraphs that follow.

> **NOTE:** Only one of the following parameters may be entered with each LOCATE command.

*all*

Causes the device fault LEDs of all configured devices to flash. You can also specify ALL to test all of the LEDs at once. Enter LOCATE CANCEL to turn off the LEDs.

*cancel*

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

*disks*

Causes the device fault LEDs of all configured disks to flash. Use LOCATE CANCEL to turn off the LEDs.

*pttll (SCSI-location)*

Causes the device fault LED on the device at the given SCSI location to flash. Use LOCATE CANCEL to turn off the LEDs.

*units*

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

*container-name*

Causes the device fault LEDs on the devices within the specified container to flash. If a device name is given, the device's fault LED is turned on. If a storageset name is given, the fault LEDs on all of the devices constituting the storageset turns on. Use LOCATE CANCEL to turn off the LEDs.

*unit-number*

Causes the device fault LEDs on the devices making up the unit to flash. Use LOCATE CANCEL to turn off the LEDs.

## Switches

There are no switches associated with this command.

## Examples

To cause the device fault LED on device DISK10000 to flash, enter:

LOCATE DISK10000

To cause the device fault LEDs on all of the devices assigned to unit D102 to flash, enter:

LOCATE D102

To turn off the flashing LEDs that were specified by a previous LOCATE command, enter:

LOCATE CANCEL

# MIRROR

This command creates a one-member mirrorset from a single disk.

After the disk drive is converted to a mirrorset, increase the nominal number of members by entering the SET *mirrorset-name* MEMBERSHIP=*number-of-members* command, then enter the SET *mirrorset-name* REPLACE=*disk-name* command to add more members to the mirrorset.

## Syntax

MIRROR *disk-name mirrorset-name*

## Parameters

The following parameters are available to the MIRROR command:

■ disk name

■ mirrorset name

These parameters are described in the following paragraphs.

*disk-name*

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

*mirrorset-name*

Assigns a name for the mirrorset.

> **IMPORTANT:** It is common to name a mirrorset MIRR*n*, where n is a sequentially assigned, unique identifier. Other naming conventions are acceptable, but this naming convention presents to the user both the type of container and its unique identifier.

## Switches

The following switches are available for use with the MIRROR command:

■ copy

■ policy

These switches are described in the following paragraphs.

*COPY=FAST*
*COPY=NORMAL (Default)*

> Sets the speed at which the controller copies data to a new member from normal mirrorset members when data is being mirrored to the storageset's disk drives:

> ■  Specify *COPY=FAST* to allow the creation of mirrored data to take precedence over other controller operations. When you specify *COPY=FAST*, the controller uses more resources to create the mirrored data, and copying takes less time. However, overall controller performance is reduced.

> ■  Specify *COPY=NORMAL* when operations performed by the controller should take priority over the copy operation. If you specify *COPY=NORMAL,* creating the mirrored data has a minimal impact on performance.

*POLICY=BEST_FIT*
*POLICY=BEST_PERFORMANCE*
*NOPOLICY (Default)*

> Sets the selection criteria the controller uses to choose a replacement disk from the spareset when a mirrorset member fails:

> ■  Specify *POLICY=BEST_FIT* to choose a replacement disk drive from the spareset that equals or exceeds the base member size (smallest disk drive at the time the mirrorset was initialized). If there is more than one disk drive in the spareset that meets the criteria, the controller selects the disk drive with the best performance.

> ■  Specify *POLICY=BEST_PERFORMANCE* to choose a replacement disk drive from the spareset with the best performance. The controller attempts to select a disk on a different port than existing mirrorset members. If there is more than one disk drive in the spareset matching the best performance criteria, the controller selects a disk drive that equals or exceeds the base member size.

> ■  Specify *NOPOLICY* to prevent the controller from automatically replacing a failed disk device. This causes the mirrorset to operate in a reduced state until either *POLICY=BEST_PERFORMANCE* or *POLICY=BEST_FIT* is selected, or a member is manually placed in the mirrorset.

## Example

The following example shows how to create a striped mirrorset with full redundancy:

3.  create a normal stripeset

4.  create a unit from the stripeset

5.  make each member of the normal stripeset a mirrorset containing two members.

The commands used in this script set the nominal number of members in each mirrorset to two and add a second disk to each mirrorset. It is not necessary to initialize the mirrorsets or add them as units; the higher-level structure of the stripeset is carried down to the mirrorsets.

```
ADD STRIPESET STRIPE1 DISK10100 DISK20100 DISK30100
INITIALIZE STRIPE1
ADD UNIT D102 STRIPE1
MIRROR DISK10100 MIRROR1
SET MIRROR1 MEMBERSHIP=2
SET MIRROR1 REPLACE=DISK20200
MIRROR DISK20100 MIRROR2
SET MIRROR2 MEMBERSHIP=2
SET MIRROR2 REPLACE=DISK30200
MIRROR DISK30100 MIRROR3
SET MIRROR3 MEMBERSHIP=2
SET MIRROR3 REPLACE=DISK10200
```

## See also

ADD MIRRORSET
REDUCE
SHOW MIRRORSETS
UNMIRROR

## POWEROFF

This command turns off the power to the subsystem cabinet.

**NOTE:** Stop all I/O operations between the host(s) and controller(s) before issuing a POWEROFF command.

### Syntax

POWEROFF

### Parameters

There are no parameters for this command.

### Switches

The following switches are available for use with the POWEROFF command:

- battery
- override bad flush
- seconds+

These switches are described in the following paragraphs.

*BATTERY _ON*
*BATTERY_OFF (Default)*

Instructs the external cache battery (ECB) charger to turn off or remain on.

- Specify *BATTERY_ON* to keep the ECB charger on after the POWEROFF command is issued.

- Specify *BATTERY_OFF* to turn off the ECB charger after the POWEROFF command is issued.

**NOTE:** The ECB LEDs will continue to flash in both cases, but the cache module LEDs cease flashing when *BATTERY_OFF* is chosen.

*OVERRIDE_BAD_FLUSH*
*NO_OVERRIDE_BAD_FLUSH (Default)*

Instructs the controller to either power off the cabinet or remain on depending on the cache flush results.

■ Specify *OVERRIDE_BAD_FLUSH* to override a failed cache flush and power off the cabinet.

■ Specify *NO_OVERRIDE_BAD_FLUSH* to prevent a power off when the cache flush fails.

*SECONDS=nn*

When the POWEROFF command is entered, all disk units in the cabinet are set to write-through. When the time interval, as represented by nn seconds, has elapsed, an orderly rundown of all units is started. When all units in the cabinet are successfully rundown, the cabinet power is turned off. Table 2–2 shows what action will be taken depending on the switch settings and the results of the attempted flush:

**Table 2–2 POWEROFF Switch Settings**

| Battery Switch | Override Switch | Flush Results | Action |
|---|---|---|---|
| BATTERY_ON | OVERRIDE_BAD_FLUSH | Success | Controller and units in cabinet shutdown, ECB charger remains on. |
| BATTERY_ON | OVERRIDE_BAD_FLUSH | Failure | Controller and units in cabinet shutdown, ECB charger remains on. |
| BATTERY_ON | NO_OVERRIDE_BAD_FLUSH | Success | Controller and units in cabinet shutdown, ECB charger remains on. |
| BATTERY_ON | NO_OVERRIDE_BAD_FLUSH | Failure | Nothing is shutdown, ECB charger remains on, user is notified of a bad flush. |
| BATTERY_OFF | OVERRIDE_BAD_FLUSH | Success | Controller and units in cabinet shutdown, ECB charger turned off. |
| BATTERY_OFF | OVERRIDE_BAD_FLUSH | Failure | Controller and units in cabinet shutdown, ECB charger turned off. |
| BATTERY_OFF | NO_OVERRIDE_BAD_FLUSH | Success | Controller and units in cabinet shutdown, ECB charger turned off. |
| BATTERY_OFF | NO_OVERRIDE_BAD_FLUSH | Failure | Nothing is shutdown, ECB charger remains on, user is notified of a bad flush. |

In dual-redundant mode, if both controllers cannot be shutdown, then both controllers and their batteries' chargers remain on.

## Example

To power off the disk units and the cabinet in 10 seconds (*BATTERY_OFF* and *NO_OVERRIDE_BAD_FLUSH* are the defaults) enter:

■ POWEROFF SECONDS=10

# REDUCE

This command removes member disk drives from mirrorsets and decreases the nominal number of members in the mirrorsets.

Unlike the SET *mirrorset-name* REMOVE=*disk-name* command, the controller does not put reduced members into the failedset. When using the REDUCE command to take a snapshot of a striped mirrorset, you must reduce all mirrorsets with one command. The CLONE utility does this automatically.

The nominal number of members in a mirrorset is determined by the number of members assigned to the mirrorset with the SET *mirrorset-name* MEMBERSHIP=*number-of-members* command or the ADD MIRRORSET *mirrorset-name disk-name1 [disk-nameN]* command—in other words, the number of disks that the mirrorset originally contained before it was reduced. The actual number of members contained in the mirrorset may be less than the nominal number of members if:

■  a disk drive is not added back to the mirrorset.

■  a member remains removed from the mirrorset.

■  the mirrorset replacement policy switch *NOPOLICY* is specified with the SET *mirrorset-name* command.

■  no spare disks exist.

The actual number of members in the mirrorset can never be greater than the nominal number of members. The disks to be removed do not need to be members of the same mirrorset. However, the disks must all be part of the same unit (for example, the same striped mirrorset). When a disk is reduced from a mirrorset, the controller:

■  flushes all of the unit's data from write-back data cache.

■  pauses I/O to the unit.

■  removes the specified disk(s).

■  decreases the nominal number of members of the mirrorset(s) by the number of disk(s) removed from the mirrorset(s).

■  resumes I/O.

For each reduced mirrorset, there must be at least one remaining normal member after the reduction. If this is not true for all of the *disk-names* specified, the mirrorset is not reduced.

Only normal members can be reduced. A normal member is a mirrorset member whose entire contents are the same as all other normal members within the mirrorset.

### Syntax

REDUCE *disk-name1 disk-name2 disk-name3...*

### Parameters

*disk-name*

Specifies the names of the disk or disks to be removed from the mirrorset or mirrorsets. Multiple members can be removed with the REDUCE command.

### Switches

There are no switches associated with this command.

### Example

To remove DISK20100, DISK20200, and DISK40200 from their respective mirrorsets, enter:

```
SHOW STRIPE1

Name      Storageset    Uses                        Used by
-------------------------------------------------------------
STRIPE1   stripeset     MIRR1                          D104
                        MIRR2
                        MIRR3


SHOW MIRRORSETS

Name      Storageset    Uses                        Used by
-------------------------------------------------------------
MIRR1     mirrorset     DISK10100                    STRIPE1
                        DISK20100
MIRR2     mirrorset     DISK10200                    STRIPE1
                        DISK20200
MIRR3     mirrorset     DISK30300                    STRIPE1
                        DISK40200


REDUCE DISK20100 DISK20200 DISK40200

SHOW MIRRORSETS
```

```
Name          Storageset         Uses                                  Used by
----------------------------------------------------------------------------
MIRR1         mirrorset          DISK10100                             STRIPE1
MIRR2         mirrorset          DISK10200                             STRIPE1
MIRR3         mirrorset          DISK30300                             STRIPE1
```

## See also

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

## RENAME

This command renames a specified container or specified host connection.

**NOTE:** Remote copy sets and association sets cannot be renamed.

### Syntax

RENAME *old-name new-name*

### Parameters

The following parameters are available to the RENAME command:

■ old-name

■ new-name

These parameters are described in the paragraphs that follow.

*old-name*

Specifies the existing name of the container or host connection.

*new-name*

Assigns the new name for the container or host connection.

A name of a host connection can be any combination of letters and numbers, with the one restriction that it cannot take the form of the default assigned by the controller (!NEWCON*nn*).

**NOTE:** Units, remote copy sets, and association sets can not be renamed.

### Switches

There are no switches associated with this command.

### Example

To change the name of DISK10000 to MYDISK, enter:

RENAME DISK10000 MYDISK

To change the name of host connection !NEWCON03 to server2, enter:

```
RENAME !NEWCON03 SERVER2
```

## **RESTART** *this_controller*
## **RESTART** *other_controller*

This command flushes all user data from the specified controller's write-back cache and restarts the controller.

### Syntax

RESTART *controller*

### Parameters

*controller*

The *controller* parameter indicates which controller will restart. Specify *other_controller* or *this_controller*.

### Switches

The following switches are available to the RESTART *controller* command:

■ ignore errors

■ immediate shutdown

These switches are described in the following paragraphs.

*IGNORE_ERRORS*
*NOIGNORE_ERRORS (Default)*

Controls the reaction of the controller based on the status of write-back cache:

> ⚠ **CAUTION:** The *IGNORE_ERRORS* switch might 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 until the controller flushes the cache.

■ Specify *IGNORE_ERRORS* to instruct the controller to restart even if the data within write-back cache cannot be written to the devices.

■ Specify *NOIGNORE_ERRORS* to instruct the controller to not restart if the data within write-back cache cannot be written to the devices.

*IMMEDIATE_SHUTDOWN*
*NOIMMEDIATE_SHUTDOWN (Default)*

Instructs the controller when to shutdown and whether to flush the write-back cache or not:

> △ **CAUTION:** The *IMMEDIATE_SHUTDOWN* switch instructs the controller to immediately shutdown, without regard to any data contained within write-back cache. Do not perform any hardware changes until the controller flushes the cache.

■ Specify *IMMEDIATE_SHUTDOWN* to instruct the controller to restart immediately without flushing data from the write-back cache to devices.

■ Specify *NOIMMEDIATE_SHUTDOWN* to instruct the controller not to restart before all data has been flushed from write-back cache to the devices.

## Examples

To restart "this controller," enter:

    RESTART THIS_CONTROLLER

To restart the "other controller," enter:

    RESTART OTHER_CONTROLLER

## See also

SELFTEST *controller*
SHUTDOWN *controller*

## RETRY_ERRORS *unit-number* UNWRITEABLE_DATA

This command causes the controller to attempt to write previously unwriteable data from the write-back cache to the devices.

If a container fails in a way that prevents the data in write-back cache to be written to the container, an unwriteable data error is reported. The RETRY_ERRORS *unit-number* UNWRITEABLE_DATA command attempts to write the unwriteable data. No data is lost if the retry fails.

### Syntax

RETRY_ERRORS *unit-number* UNWRITEABLE_DATA

### Parameter

*unit-number*

Identifies the unit number to which the data contained in write-back cache tries to write. The unit-number is the same name given to the unit when it was created using the ADD UNIT command.

### Switches

There are no switches associated with this command.

### Example

To retry the writing of the cached data previously marked "unwriteable to disk unit D103," enter;

RETRY_ERRORS D103 UNWRITEABLE_DATA

### See also

CLEAR_ERRORS UNWRITEABLE_DATA

# RUN

Runs a specified diagnostic or utility program. The specified diagnostic or utility program only runs on "this controller."

## Syntax

RUN *program-name*

## Parameter

*program-name*

The *program-name* parameter specifies the name of the diagnostic or utility program to be run. The following programs can currently be run:

■ CHVSN—This is not a user utility. This utility may be used by Compaq authorized service personnel only.

■ CLCP—A utility used to load updated software code or patches. See "Upgrading the Subsystem" in the *HSG80 Array Controller ACS Version 8.5 Maintenance and Service Guide* for more information regarding this utility.

■ CLONE—A utility used to automate the process of mirroring units to create a snapshot copy of host unit data. See "Cloning Data for Backup," in the *HSG80 Array Controller ACS Version 8.5 Configuration Guide* for more information regarding this utility.

■ CONFIG—A utility used to locate and add devices to the controller configuration. CONFIG may be run anytime new devices are added to the subsystem. See the *HSG80 Array Controller ACS Version 8.5 Configuration Guide* for more information.

■ DILX—A utility used to test and verify the controller's operation with attached storage devices under a high or low I/O load. Run DILX (disk inline exerciser) only when there is no activity on the controller. The total I/O load is handled by the controller, bypassing the host.

> **CAUTION:** Run the DILX utility in the autoconfigure mode only at initial installations. When write operations are enabled, the DILX utility may overwrite existing data.

- DIRECT—A command that displays a directory of all executable diagnostic or utility programs.

- DSTAT—This is not a user utility. This utility may be used by Compaq authorized service personnel only.

- FMU—A fault management utility used to control several spontaneous errors. FMU also displays information regarding the most recent controller and memory system failure.

- FRUTIL—A utility used when replacing a failed controller, external cache battery, or cache module.

- HSUTIL—A utility used to format a disk device or to download new firmware to a tape or disk device.

- VTDPY—A utility used to display the current controller state, performance data, processor utilization, host post activity and status, device state, logical unit state, cache performance, Data Replication Manager (DRM) activity, and I/O performance.

Refer to the *HSG80 Array Controller ACS Version 8.5 Maintenance and Service Guide* for more information regarding these utilities.

## Switches

There are no switches associated with this command.

## Example

To start the DILX diagnostic program, enter:

    RUN DILX

## See also

DIRECTORY

## **SELFTEST** *this_controller*
## **SELFTEST** *other_controller*

This command flushes the data from the specified controller write-back cache (if present), shuts down the controller, and restarts the controller in looping self-test mode. Press the controller reset (//) button to take the controller out of self-test mode.

### Syntax

SELFTEST *controller*

### Parameters

*controller*

The *controller* parameter indicates which controller is to perform the self-test program. Specify *other_controller* or *this_controller*.

### Switches

The following switches are available to the SELFTEST *controller* command:

■ ignore errors

■ immediate shutdown

These switches are described in the following paragraphs.

*IGNORE_ERRORS*
*NOIGNORE_ERRORS (Default)*

Instruct the controller how to respond to write-back cache errors:

> △ **CAUTION:** The *IGNORE_ERRORS* switch might cause data to remain in write-back cache. Do not perform any hardware changes until the controller flushes the 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 unable to be written to the devices or lost data errors.

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

*IMMEDIATE_SHUTDOWN*
*NOIMMEDIATE_SHUTDOWN (Default)*

Instructs the controller whether to flush the write-back cache:

> ⚠ **CAUTION:** The *IMMEDIATE_SHUTDOWN* switch instructs the controller to immediately shut down, without regard to any data contained within write-back cache. Do not perform any hardware changes until the controller flushes the cache.

■ Select *IMMEDIATE_SHUTDOWN* to instruct the controller to run the self-test program immediately 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 program on "this controller," enter:

SELFTEST THIS_CONTROLLER

To run the self-test program on the "other controller," even if the "other controller" cannot flush all data from the write-back cache, enter:

SELFTEST OTHER_CONTROLLER IGNORE_ERRORS

## See also

RESTART *controller*
SHUTDOWN *controller*

# **SET** *associations*

> **NOTE:** This command works only in a Data Replication Manager (DRM) environment and requires V8.5P code. Like all DRM commands, it is heavily restricted in its use. See *Data Replication Manager HSG80 ACS Version 8.5P Operations Guide* for examples of usage.

This command changes the characteristics of an association set.

## **Syntax**

SET *association-set-name*

> **NOTE:** Only one characteristic of an association set can be changed at any one time; therefore, only one switch at a time can be specified with any SET *association set* command.

## **Parameters**

association-set-name

These name of the association set whose characteristics are to be changed.

## **Switches**

The following switches are available for the SET *association set* command:

- add
- failsafe lock operation
- log unit
- I/O ordering
- remove

*ADD=remote_copy_set_name*

This switch adds the specified remote copy set to the association set.

*FAIL_ALL*
*NOFAIL_ALL*

This switch determines the failsafe lock operation of members of the association set. (Failsafe lock is a condition a remote copy set assumes under certain potentially catastrophic error conditions. The failsafe locked condition prevents further write operations from the host to the remote copy set members.)

When *FAIL_ALL* is specified, all the members of the association set assume the same failsafe lock condition if they are in failsafe error mode. Failsafe lock is enabled by the *ERROR_MODE* switch of the SET *remote-copy-sets* command. If one member of the association set becomes failsafe locked, any association set member that has failsafe lock enabled also becomes failsafe locked.

When *NOFAIL_ALL* is specified, the members of the association set react independently to failsafe lock conditions. One member of the association set becoming failsafe locked has no effect on the other members of the association set.

This switch has no effect if all members of the association set have failsafe lock disabled (normal error mode).

*LOG_UNIT=unit_name*
*NOLOG_UNIT*

Specifying *LOG_UNIT* assigns a dedicated log unit for the association set. Should the initiator become unable to access the target, the unit specified by *unit-name* logs a history of write commands from the host. When the target becomes accessible, the initiator can read the log disk and can send the write commands, in order, to the target, which brings the target into congruency with the initiator.

> △ **CAUTION:** When the command is entered, a header is immediately written to the log unit, which may make it difficult or impossible to recover any user data previously written on the unit. Care should be taken in specifying which unit should be the log unit.

The log unit must be either a mirrorset or a striped mirrorset. It cannot be a partioned unit. The log unit must reside at the initiator site and cannot be moved to the target site. In addition, it must have the following characteristics:

■ write-back caching disabled

■ access disabled

■ other unit settings at default

This switch is valid only if all members of the association set are in normal (not failsafe) error mode. Error mode is determined by the *ERROR_MODE* switch of the SET *remote copy set* command.

Specifying NOLOG_UNIT deassigns the log unit.

*ORDER_ALL*
*NOORDER_ALL*

When *ORDER_ALL* is specified:

■   if one member of the association set starts logging a history of write operations, all members of the association set start logging write operation history

■   the order of all asynchronous write operations across all members of the association set is preserved

When *NOORDER_ALL* is specified

■   members of the association set start to log (or not) independently

■   the order of all asynchronous write operations across all members of the association set is not preserved

This switch cannot be changed when a log unit is assigned to the association set. The log unit must be removed, then the order switch changed, then the log unit must be reassigned. The order switch goes into effect when the log unit is assigned.

## Examples

To add remote copy sets RCS2 and RCS4 to association set AS3, enter:

    SET AS3 ADD=RCS2

    SET AS3 ADD=RCS4

## See also

ADD ASSOCIATIONS

ADD REMOTE COPY SET

SET *remote copy set*

## SET *concatsets*

⚠ **CAUTION:** This command should only be done with host operating systems that can support dynamic volume expansion. If the operating system cannot handle one of its disks getting suddenly bigger, use of this command could make data inaccessible.

This command expands an existing unit by concatenating another storageset of the same type to the storageset that constitutes the unit.

This command is subject to restrictions:

■ Both storagesets must be of the same type.

■ If either member storageset becomes inoperative, the concatset becomes inoperative. However, a reduced RAIDset does not make the concatset inoperative.

■ Mirrorsets cannot be used.

■ Partitions cannot be used.

■ Concatsets cannot be partitioned.

■ The maximum total disks in a concatset is 24.

■ Storagesets must have been initialized before being added to the concatset.

■ The CLONE utility cannot be used on a concatset.

■ Concatsets cannot be used in remote copy sets.

### Syntax

SET *concatset-name*

### Parameters

*concatset-name*

The name of the concatset to be expanded.

### Switches

*ADD=storagesetname*

Concatenates the storageset specified by *storageset-name* to the concatset.

## Examples

To expand the capacity of unit D0, which consists of stripeset Stripe1, by adding another stripeset, Stripe2, enter:

```
ADD CONCATSETS C1 STRIPE1

SET C1 ADD=STRIPE2
```

## See also

ADD CONCATSETS
DELETE *concatsets*
SHOW CONCATSETS

## SET *connection-name*

This command changes the operating characteristics of a host connection. Each path between a Fibre Channel adapter in a host computer and an active host port on a controller is a connection.

The SET *connection-name* command changes the operating parameters of the specified host connection. A host connection is a specific instance of one host connected to one port of one controller through one host adapter.

### Syntax

SET *connection-name*

### Parameters

*connection-name*

This is the name of the host connection. When a new host-adapter-port-controller connection is made, the new connection is given a default connection name. The default connection name is !NEWCON*nn*, where *nn* is a decimal number. The connection name can be changed through the RENAME command.

### Switches

*UNIT_OFFSET*

Unit offset (usually just called offset) is a decimal value that establishes the beginning of the range of units that a host connection can access. It defines and restricts host connection access to a contiguous group of unit numbers.

If no value is specified for the *UNIT_OFFSET* switch, then host connections have default offsets as follows:

- In transparent failover mode, connections on port 1 have an offset of 0 and connections on port 2 have an offset of 100.

- In multiple-bus failover mode, all connections have an offset of 0.

The relationship between LUN number, unit number, and offset is as follows:

- LUN = unit number - offset.

    - Logical unit number or LUN = the logical unit number presented to the host connection.

■ Unit number = the number assigned to the unit in the ADD UNIT command. This is the number by which the unit is known internally to the controllers.

*OPERATING_SYSTEM=OS_name*

Specifies the operating system of the host. After entering "*Operating_system*" type a question mark. The CLI will display the available options.

## See Also

ADD CONNECTIONS
ADD UNIT
DELETE *connections*
RENAME

## **SET** *controller*

See SET *this_controller*/SET *other_controller*

# SET *device-name*

This command changes the characteristics of the specified device.

## Syntax

SET *device-name*

## Parameter

*device-name*

Specifies the name of the device to change. This can be a previously named device, disk, passthrough device, or container.

## Switches

The following switches are available to the SET *device-name* command:

■ transfer rate requested

■ transportable

These switches are described in the following paragraphs.

*TRANSFER_RATE_REQUESTED=ASYNCHRONOUS*
*TRANSFER_RATE_REQUESTED=20MHZ*
*TRANSFER_RATE_REQUESTED=10MHZ*
*TRANSFER_RATE_REQUESTED=5MHZ*

Specifies the maximum data transfer rate for the controller 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.

> **NOTE:** This switch is used by all container types

*TRANSPORTABLE*
*NOTRANSPORTABLE*

Indicates whether a disk drive can be accessed exclusively by StorageWorks controllers (the *TRANSPORTABLE* switch is used for disk drives only).

> **NOTE:** This switch works only if the disk drive is not used by a higher level assembly, such as a storageset or a unit.

Storagesets cannot be made transportable. Specify *NOTRANSPORTABLE* for all disks used in RAIDsets, stripesets, mirrorsets, and sparesets. Transportable disks do not contain any metadata or restricted areas on the disk. Therefore, transportable disks forfeit the advantage metadata provides. Transportable disks can be moved to a non-StorageWorks environment with their data intact.

If you specify the *NOTRANSPORTABLE* switch and there is no metadata on the unit, the unit must be initialized. If you specify *TRANSPORTABLE* for a disk that was originally initialized as a *NOTRANSPORTABLE*, you should initialize the disk.

**NOTE:** Specifying *TRANSPORTABLE* should be avoided unless transportability of the device or media is imperative and there is no other way to accomplish moving the data.

**NOTE:** This switch is only applicable to disk drives.

## Examples

To set the data transfer rate of DISK20000 to 5MHz, enter:

SET DISK20000 TRANSFER_RATE_REQUESTED=5MHZ

To set DISK10300 to transportable, enter:

SET DISK10300 TRANSPORTABLE

## See also

ADD CDROM
ADD DISK
ADD Optical
SHOW DISKS

## SET EMU

This command sets operating parameters for the environmental monitoring unit (EMU).

In subsystems with more than one cabinet, and therefore more than one EMU, the SET EMU command controls both the master and slave EMU settings. The EMU within the primary cabinet (master) instructs the EMUs within the other cabinets to operate at the same settings as the master EMU.

### Syntax

SET EMU

### Parameters

There are no parameters for this command.

### Switches

The following switches are available to the SET EMU command:

■ alarm

■ sensor

■ fanspeed

These switches are described in the following paragraphs.

*ALARM=silent*
*ALARM=audible*

Determines whether or not the audible alarm of the EMU will sound in the following cases:

■ drive failure

■ controller restart

The EMU alarm will still sound in the event of an adverse environmental condition, such as a power failure or an over-temperature condition.

*SENSOR_1_SETPOINT=nn*
*SENSOR_2_SETPOINT=nn*
*SENSOR_3_SETPOINT=nn*
*SENSOR_x_SETPOINT=35*

> Sets the acceptable temperatures (in Celsius) at which the subsystem operates. Sensor 1 and Sensor 2 set the maximum operating temperature for the primary subsystem cabinet. Sensor 3 sets the maximum operating temperature for the EMU unit. The allowable range for the setpoint is 0°C (32°F) to 49°C (120°F). The EMU determines the default setpoint for all three sensors.
>
> Table 2–3 lists the valid EMU set-point temperatures in both Fahrenheit and Celsius.
>
> If any of the setpoints assigned to a slave EMU do not match the corresponding setpoints assigned to the master EMU, the slave EMU settings change to match the corresponding master EMU settings.
>
> Refer to the enclosure documentation for detailed information about setting the EMU temperature set points.

*FANSPEED=HIGH*
*FANSPEED=AUTOMATIC*

> Sets the speed at which the fan operates.
>
> ■ Select *FANSPEED=HIGH* to force the fans in all connected cabinets to operate at high speed continuously.
>
> ■ Select *FANSPEED=AUTOMATIC* to allow the EMU to control the fan speed for the fans in all connected cabinets.

The EMU instructs the fans to operate at high speed when any of the temperature setpoints are exceeded or when one or more fans are not functioning.

| Table 2–3 EMU Set Point Temperatures | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ℃ | ℉ | ℃ | ℉ | ℃ | ℉ | ℃ | ℉ | ℃ | ℉ |
| 0 | 32 | 10 | 50 | 20 | 68 | 30 | 86 | 40 | 104 |
| 1 | 34 | 11 | 52 | 21 | 70 | 31 | 88 | 41 | 106 |
| 2 | 46 | 12 | 54 | 22 | 72 | 32 | 90 | 42 | 108 |
| 3 | 37 | 13 | 55 | 23 | 73 | 33 | 91 | 43 | 109 |
| 4 | 39 | 14 | 57 | 24 | 75 | 34 | 93 | 44 | 111 |
| 5 | 41 | 15 | 59 | 25 | 77 | 35 | 95 | 45 | 113 |
| 6 | 43 | 16 | 61 | 26 | 79 | 36 | 97 | 46 | 115 |
| 7 | 45 | 17 | 63 | 27 | 81 | 37 | 99 | 47 | 117 |
| 8 | 46 | 18 | 64 | 28 | 82 | 38 | 100 | 48 | 118 |
| 9 | 48 | 19 | 66 | 29 | 84 | 39 | 102 | 49 | 120 |

## Examples

This example shows how to set EMU sensor number 2 to 34°C:

SET EMU SENSOR_2_SETPOINT=34

This example shows how to set the EMU fan to operate at high speed:

SET EMU FANSPEED=HIGH

## See also

SHOW

## SET FAILEDSET

This command hanges the automatic replacement policy for the failedset.

### Syntax

SET FAILEDSET

### Parameters

There are no parameters associated with this command.

### Switches

*AUTOSPARE*
*NOAUTOSPARE*

Specifies the policy to be used by the controller when a disk drive is physically replaced in the failedset:

■ Specify *AUTOSPARE* to instruct the controller to automatically move devices physically replaced in the failedset into the spareset.

■ Specify *NOAUTOSPARE* to instruct the controller to leave devices physically replaced in the failedset. The device, though replaced, remains in the failedset until it is manually removed with the DELETE FAILEDSET command.

In most circumstances, a disk physically replaced into the failedset is functional and contains no metadata—that is, a new, initialized device. If you specify the *AUTOSPARE* switch when a disk is physically replaced in the failedset, the controller checks to see if any metadata is present. If the controller detects metadata, the disk remains in the failedset. If the controller does not detect metadata, the controller automatically initializes the disk and moves it from the failedset to the spareset. Now a member of the spareset, the disk is available for any mirrorset or RAIDset requiring a replacement member. If the automatic initialization fails, the disk remains in the failedset.

Disks that you plan to use for *AUTOSPARE* must not have valid metadata on them. If you suspect a disk does have metadata on it (it was used in a stripeset or was initialized as *NOTRANSPORTABLE*) you must use the following procedure to make the disk available as a spareset replacement disk (DISK10000 is used as an example):

1) Delete all containers to which the disk belongs.

2) Make the disk transportable.

> SET DISK10000 TRANSPORTABLE.

3) Initialize the disk.

> INIT DISK10000

4) Delete the disk.

> DELETE DISK10000

5) Remove DISK10000 from the cabinet until it is time to replace a failed disk.

## Example

To enable the automatic spare feature, enter:

> SET FAILEDSET AUTOSPARE

To disable the automatic spare feature, enter:

> SET FAILEDSET NOAUTOSPARE

## See also

DELETE FAILEDSET
SHOW FAILEDSET

## SET FAILOVER COPY=*this_controller*
## SET FAILOVER COPY=*other_controller*

This command configures a controller pair to transparent failover mode. Failover is a process that allows a controller to take over total control of the storage subsystem in the event of the failure of its companion controller.

> **NOTE:** Before putting the controllers in failover mode, remove any previous mode with the SET NOFAILOVER command.

### Syntax

SET FAILOVER COPY=*controller*

### Parameters

*controller*

 *this_controller*
 *other_controller*

Specifies which controller ("this" or "other") contains the source configuration for the copy. The companion controller receiving the configuration information restarts after the command is carried out:

> **CAUTION:** Make sure you know which controller has the good configuration information before entering this command. The device configuration information from the controller specified by the *controller* parameter overwrites the information on the companion controller.

■ Specify *this_controller* to copy the device configuration information from "this controller" to "other controller."

■ Specify *other_controller* to copy the device configuration information from "other controller" to "this controller."

Due to the amount of information being passed from one controller to the other, this command may take up to two minutes to complete.

### Switches

There are no switches associated with this command.

**Example**

To put a controller pair in transparent failover mode and copy the configuration information from "this controller" to "other controller," enter:

    SET FAILOVER COPY=THIS_CONTROLLER

**See also**

SET MULTIBUS_FAILOVER
SET NOFAILOVER
SET NOMULTIBUS_FAILOVER

## SET *mirrorset-name*

This command changes the characteristics of a mirrorset, including the addition and removal of members.

### Syntax

SET *mirrorset-name*

### Parameter

*mirrorset-name*

Specifies the name of the mirrorset to modify. This is the same name given to the mirrorset when it was created with the ADD MIRRORSET command.

### Switches

The following switches are available to the SET *mirrorset-name* command:

■   copy

■   membership

■   policy/nopolicy

■   read source

■   remove

■   replace

These switches are described in the following paragraphs.

*COPY=FAST*
*COPY=NORMAL*

Sets the speed at which the controller copies data to a new member from normal mirrorset members when data is being mirrored to the storageset's disk drives:

■   Specify *COPY=FAST* to allow the creation of mirrored data to take precedence over other controller operations. When you specify *COPY=FAST*, the controller uses more resources to create the mirrored data, and copying takes less time. However, overall controller performance is reduced.

■ Specify *COPY=NORMAL* when operations performed by the controller should take priority over the copy operation. If you specify *COPY=NORMAL,* creating the mirrored data has a minimal impact on performance.

*MEMBERSHIP=number-of-members*

Sets the nominal number of mirrorset members to the number you specify for the *number-of-members* value. A maximum of six members can be specified.

> **NOTE:** No other switches can be set when you specify the *MEMBERSHIP* switch.

If you increase the number of members and there is a replacement policy specified, the controller automatically adds disk drives from the spareset to the mirrorset until 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 the *NOPOLICY* switch is specified, the *REPLACE=disk-name* switch must be specified to bring the mirrorset up to the new nominal number of members.

You cannot set the nominal number of members lower than the actual number of members. Specify the *REMOVE* switch to reduce the number of disk drives from the mirrorset.

*POLICY=BEST_FIT*
*POLICY=BEST_PERFORMANCE*
*NOPOLICY*

Sets the selection criteria the controller uses to choose a replacement disk from the spareset when a mirrorset member fails:

■ Specify *POLICY=BEST_FIT* to choose a replacement disk drive from the spareset that equals or exceeds the base member size (smallest disk drive at the time the mirrorset was initialized). If there is more than one disk drive in the spareset that meet the criteria, the controller selects the disk drive with the best performance.

■ Specify *POLICY=BEST_PERFORMANCE* to choose a replacement disk drive from the spareset with the best performance. The controller attempts to select a disk on a different port than existing mirrorset members. If there is more than one disk drive in the spareset matching the best performance criteria, the controller selects a disk drive that equals or exceeds the base member size of the mirrorset.

■ Specify *NOPOLICY* to prevent the controller from automatically replacing a failed disk device. The mirrorset operates in a reduced state until a *POLICY=BEST_FIT* or *POLICY=BEST_PERFORMANCE* is selected, or a member is manually placed in the mirrorset

*READ_SOURCE=disk-name*
*READ_SOURCE=LEAST_BUSY*
*READ_SOURCE=ROUND_ROBIN*

Selects the mirrorset member used by the controller to satisfy a read request.

■ Specify the *READ_SOURCE=disk-name* of a specific member to which you want the controller to direct all read requests. If the member fails out of the mirrorset, the controller selects the first normal member it finds to satisfy its read requests.

■ Specify *READ_SOURCE=LEAST_BUSY* to direct read requests to the mirrorset member with the least amount of work in its queue. If multiple members have equally short queues, the controller queries these members for each read request as it would when *READ_SOURCE=ROUND_ROBIN* is specified.

■ Specify *READ_SOURCE=ROUND_ROBIN* to sequentially direct read requests to each mirrorset member. The controller equally queries all normal members for each read request.

*REMOVE=disk-name*

Instructs the controller to remove a member 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.

> **NOTE:** No other switches can be set when the *REMOVE=* switch is specified.

For each reduced mirrorset, there must be at least one remaining normal member after the reduction. A normal member is one whose contents are the same as the other members.

Unlike the REDUCE command, 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 to the nominal number of members.

> **NOTE:** Normalizing members exist only when you first create a mirrorset or when you clear lost data on a mirrored unit. The controller recognizes a member as normal, and all other original mirrorset members as "normalizing." New data that is written to the mirrorset is written to all members. The controller copies the normal member's data to the normalizing members. The controller recognizes a normalizing member as normal when all the normalizing member's blocks are the same as the normal member(s).

*REPLACE=disk-name*

Instructs the controller to add a disk member to an existing mirrorset if the following conditions are met:

■ The replacement policy is set to *NOPOLICY*

■ The mirrorset is missing at least one member

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

The disk name used is the name given to a disk when it was added to the configuration with the ADD DISK command.

> **NOTE:** Do not specify any other switches when the *REPLACE=* switch is specified.

## Examples

To change the replacement policy of mirrorset MIRR1 to best fit, enter:

SET MIRR1 POLICY=BEST_FIT

To remove member DISK30000 from mirrorset MIRR1 created above, enter:

ISET MIRR1 REMOVE=DISK30000

> **NOTE:** f 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.

To add disk DISK30200 to the mirrorset MIRR1, enter:

SET MIRR1 REPLACE=DISK30200

A copy operation to DISK30200 begins immediately.

## See also

ADD MIRRORSET
MIRROR
REDUCE
SHOW MIRRORSET
UNMIRROR

## SET MULTIBUS_FAILOVER

This command places "this controller" and the "other controller" into multiple-bus failover mode. Failover is a process that allows a controller to take over total control of the storage subsystem in the event of the failure of its companion controller.

> **NOTE:** Before putting the controllers in multibus_failover mode, remove any previous mode with the SET NOFAILOVER command.

### Syntax

SET MULTIBUS_FAILOVER *copy=controller*

### Parameters

*copy=this_controller*
*copy=other_controller*

Specifies which controller ("this controller" or "other controller") contains the source configuration The companion controller receives the configuration information and restarts.

> △ **CAUTION:** Make sure you know which controller has the good configuration information before entering this command. The device configuration information from the controller specified by the *controller* parameter overwrites the information on the companion controller.

- Specify *this_controller* to copy the device configuration information from the "this controller" to "other controller."

- Specify *other_controller* to copy the device configuration information from the "other controller" to "this controller."

Due to the amount of information being passed from one controller to the other, this command may take up to two minutes to complete.

### Example

This example shows how to configure a controller pair to operate in multiple bus failover mode:

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 command. If you want to prefer specific units to specific controllers, use the following command after setting multiple bus failover:

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

## See also

SET FAILOVER
SET NOFAILOVER
SET NOMULTIBUS_FAILOVER

## SET NOFAILOVER
## SET NOMULTIBUS_FAILOVER

The SET NOFAILOVER and SET NOMULTIBUS_FAILOVER commands remove "this controller" and the "other controller" (if currently reachable) from failover mode. Either command can be used to cancel failover, as they both perform exactly the same actions.

The controller on which the command was entered is always removed from the dual-redundant mode, even if the other controller is not currently reachable. No configuration information is lost. All units that were accessed through the "other controller" fail over to "this controller," and the "other controller" is shut down.

Immediately after entering this command, one controller should be physically removed because the sharing of devices is not supported by single controller configurations.

These commands affect both controllers, regardless of the controller on which the command is carried out.

### Syntax

SET NOFAILOVER

or

SET NOMULTIBUS_FAILOVER

### Parameters

There are no parameters associated with this command.

### Switches

*DESTROY_UNFLUSHABLE_DATA*
*NODESTROY_UNFLUSHABLE_DATA*

Instructs the controller how to handle data contained within write-back cache. These switches have no effect if both controllers are operational. Select one of these switches to indicate how the controller is to handle data contained in cache if one of the controllers fails before it can properly shut down with the SET NOFAILOVER or SHUTDOWN commands.

Under some circumstances, the data in a failed controller's write-back cache may not fail over to the operating controller's write-back cache. For example, cache data will not failover if the operating controller has a failed cache battery because of the risk of data loss if the power is interrupted.

■ Specify *NODESTROY_UNFLUSHABLE_DATA* to leave the unwritten data intact in the failed controller's write-back cache. When the failed controller is replaced and placed into service, the write-back cache data is flushed to the appropriate devices.

■ Specify *DESTROY_UNFLUSHABLE_DATA* to reconfigure the operational controller before replacing the failed controller. The unwritten data of the failed controller may reference devices not present in the new configuration. If you do not destroy the old configuration data, it may conflict with the new configuration and cause the subsystem to behave unpredictably.

> **CAUTION:** Unflushed data cannot be recovered after it is destroyed.

## Example

To terminate failover mode between two controllers in a dual-redundant configuration, enter:

    SET NOFAILOVER

## See also

SET FAILOVER
SET MULTIBUS_FAILOVER

## **SET** *RAIDset-name*

This command hanges the characteristics of a RAIDset.

### Syntax

SET *RAIDset-name*

### Parameters

*RAIDset-name*

Specifies the name of the RAIDset to modify. This is the name used with the ADD UNIT command to identify the RAIDset as a host-addressable unit.

### Switches

The following switches are available to the SET RAIDset-name command:

■ policy/nopolicy

■ reconstruct

■ remove

■ replace

These switches are described in the following paragraphs.

*POLICY=BEST_FIT*
*POLICY=BEST_PERFORMANCE*
*NOPOLICY*

Specifies the replacement policy to use when a member within the RAIDset fails:

■ Specify *BEST_FIT* to choose a replacement disk drive from the spareset that equals or exceeds the base member size (smallest disk drive at the time the RAIDset was initialized). If more than one disk drive in the spareset is the correct size, the controller selects a disk drive having the best performance.

■ Specify *POLICY=BEST_PERFORMANCE* to choose a replacement disk drive from the spareset resulting in the best performance of the RAIDset. The controller attempts to select a disk on a different port than existing members. If more than one disk drive in the spareset matches the best performance criteria, the controller selects a disk drive that equals or exceeds the base member size of the RAIDset.

■ Specify *NOPOLICY* to prevent the controller from automatically replacing a failed disk device. This causes the RAIDset to operate in a reduced state until either *POLICY=BEST_PERFORMANCE* or *POLICY=BEST_FIT* is selected, or a member is manually replaced in the mirrorset.

*RECONSTRUCT=FAST*
*RECONSTRUCT=NORMAL*

Sets the speed at which the controller reconstructs the data on the new RAIDset member replacing a failed member:

■ Specify *RECONSTRUCT=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.

■ Specify *RECONSTRUCT=FAST* when the reconstruct operation must take precedence over other controller operations. The controller uses more resources to perform the reconstruction. Reconstruction takes less time, but overall controller performance is reduced during the reconstruction.

*REMOVE=disk-name*

Instructs the controller to remove a member from an existing RAIDset. The disk drive specified by *disk-name* is removed from the RAIDset specified by *RAIDset-name*. The removed disk drive is added to the failedset.

If a RAIDset is already in a reduced state, an error is displayed and the command is rejected. If a replacement policy is specified, the replacement is taken from the spareset to replace the removed member using the policy specified.

If the *NOPOLICY* switch is specified with the SET RAIDset command, the RAIDset continues to operate in a reduced state until a replacement policy is specified or the *REPLACE* switch is specified. See the *REPLACE=disk-name* switch for information on manually replacing a RAIDset member. See the *POLICY* and *NOPOLICY* switches for information regarding setting a policy for automatic member replacement.

> **NOTE:** Do not specify other switches when you use the *REMOVE=* switch.

*REPLACE=disk-name*

Instructs the controller to add a disk member to an existing RAIDset if the following conditions are met:

■ The replacement policy is set to *NOPOLICY.*

■ The disk member is not in any configuration, including a spareset.

An error is displayed and the command is rejected if the RAIDset is not in a reduced state, if a replacement policy is already specified, or if the disk specified is already being used by a configuration (including a spareset).

**NOTE:** Do not specify other switches when you use the *REPLACE=* switch.

## Examples

To change the replacement policy for RAIDset RAID9 to *BEST_FIT*, enter:

SET RAID9 POLICY=BEST_FIT

To remove member DISK10000 from the RAID9 RAIDset, enter:

SET RAID9 REMOVE=DISK10000

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

To add disk DISK20100 to the reduced RAIDset, RAID9, enter:

SET RAID9 REPLACE=DISK20100

Reconstruction immediately begins on DISK20100.

## See also

ADD RAIDSET
SHOW RAIDSETS

# SET *remote-copy-set-names*

**NOTE:** This command works only in a Data Replication Manager (DRM) environment and requires V8.5P code. Like all DRM commands, it is heavily restricted in its use. See *Data Replication Manager HSG80 ACS Version 8.5P Operations Guide* for examples of usage.

**NOTE:** This command works only on the site that is functioning as initiator.

This command changes the characteristics of a remote copy set.

## Syntax

SET *remote-copy-set-name*

## Parameters

*remote-copy-set-name*

Specifies the name of the remote copy set to modify. This is the name used with the ADD REMOTE_COPY_SETS command to identify the remote copy set as a host-addressable unit.

## Switches

*ADD=remote-node-name/target-unit-name*

Specifies a unit to add into the remote copy set.

*Remote-node-name* is the name of the controller pair that controls the unit that is to be added. This name is assigned through the *REMOTE_COPY* switch of the SET *this controller* command.

*Target-unit-name* is the name of the unit to be added to the remote copy set.

*ERROR_MODE=FAILSAFE*
*ERROR_MODE=NORMAL*

■ *ERROR_MODE=FAILSAFE* enables failsafe lock mode for the members of the remote copy. Failsafe lock is a condition the members of the remote copy set assume under certain potentially catastrophic error conditions. The failsafe lock condition disables access to the remote copy set members.

■ *ERROR_MODE=NORMAL* allows write operations to continue even when an error condition is present.

*INITIATOR=initiator-unit-name*

> **NOTE:** This command is part of the site failover procedure. Use of this switch is restricted to only site failover conditions. See *Data Replication Manager HSG80 ACS Version 8.5P Operations Guide* for examples of usage.

Moves the role of initiator to the specified target member.

*OPERATION_MODE=SYNCHRONOUS*
*OPERATION_MODE=ASYNCHRONOUS*

This switch determines which of two normal operating modes is assigned to the remote copy set.

In synchronous mode, write operations must be written to the target cache before the host is informed that the operation is complete. Synchronous operation assures data consistency among the members of a remote copy set.

In asynchronous mode, the write operation is reported as complete to the host before the data is written to the remote units of the remote copy set. Asynchronous mode provides greater performance and response time, but the data on all members of the remote copy set cannot be assumed to be always the same.

*OUTSTANDING_IO= n*

This switch sets the number of outstanding I/O operations from the initiator to the target.

n is a decimal number between 1 and 240, with 20 being the default value.

The way this switch operates depends on which operating mode is set by the *OPERATION_MODE* switch:

■ In synchronous mode, *OUTSTANDING_IO* refers to the number of remote writes (write operations from the initiator to the target) that can be outstanding.

■ In asynchronous mode, *OUTSTANDING_IO* refers to the number of write operations that can be reported as completed to the host before they have been written on all the members of the remote copy set.

*REMOVE=remote-node-name/target-unit-name*

Removes a unit from an existing remote copy set.

*Remote-node-name* is the name of the controller pair that controls the unit that is to be removed. This name is assigned through the *REMOTE_COPY* switch of the SET *this controller* command.

*Target-unit-name* is the name of the unit to be removed from the remote copy set.

*RESUME*

> This switch starts a recovery procedure called "mini-merge." When the target becomes accessible, this switch enables the initiator to read the log disk and send the write commands, in order, to the target, which brings the target into congruency with the initiator.

> A SET *remote-copy-set-name* command with the *SUSPEND* switch specified must have been entered prior to entering a SET r*emote-copy-set-name* command with the *RESUME* switch specified.

*SUSPEND*

> This switch allows suspension of write operations to the target so that the target can be used for backup, and then remote copy functionality can be resumed.

> This switch starts the logging of write operations. When the target becomes accessible, entering a SET *remote-copy-set* command with the *RESUME* switch specified enables the initiator to read the log disk and send the write commands, in order, to the target, which brings the target into congruency with the initiator.

## See also

ADD REMOTE_COPY_SETS
SET *controller* REMOTE_COPY

## **SET** *this_controller*
## **SET** *other_controller*

This command changes switch values on the specified controller. "This controller" indicates the controller that is connected to the maintenance terminal while executing CLI commands. "Other controller" is the controller not connected to the maintenance terminal in dual-redundant controller configurations.

> **NOTE:** After doing a SET command to either controller, use a SHOW *this_controller* and a SHOW *other_controller* command to verify that the changes went into effect.

### Syntax

SET *this_controller*
SET *other_controller*

### Parameter

*controller*

Indicates which controller is to be set. Specify *other_controller* or *this_controller*.

### Switches

The following switches are available:

- allocation class

- cache flush timer

- command console LUN

- identifier

- mirrored cache

- node ID

- port ALPA

- port topology

- prompt

- remote copy

- SCSI version

- terminal parity

■ terminal speed

■ time

■ UPS

The switches are described in the following paragraphs.

*ALLOCATION_CLASS*

Allocation class is a unique identification number assigned to the controller pair under certain operating systems. The value for an allocation class is 0-4294967295. In the OpenVMS operating system, this is a 2-byte number (do not specify a number over 65535); for Compaq Tru64 UNIX, it is a 4-byte number. It is reported in response to the SCSI inquiry command and is the same for all units connected to one or both controllers. The allocation class value allows the host to identify the controllers that are a matched dual-redundant pair. This number should be unique for every pair of dual-redundant controllers in the cluster.

> **NOTE:** This value must not be zero (default) in dual-redundant configurations in host systems that implement allocation class. A zero value in this configuration causes the operating system to disable failover between the controller pair. Some operating systems do not implement allocation class, in which case the default of zero has no meaning.

*CACHE_FLUSH_TIMER=n*
*CACHE_FLUSH_TIMER=10 (Default)*

Specifies how many seconds (1–65535) of idle time on a unit may elapse before the write-back cache flushes its entire contents to the idle units. The default setting is 10 seconds. When changed, the new value entered for this switch takes effect immediately.

*COMMAND_CONSOLE_LUN*
*NOCOMMAND_CONSOLE_LUN*

Enables or disables the command console LUN (CCL), a communication LUN used to initially setup StorageWorks Command Console. When changed, the new setting for this switch takes effect immediately.

This switch works in SCSI-2 mode only. This switch has no effect in SCSI-3 mode. SCSI mode is set by the *SCSI_MODE* switch of this command.

■ Select *COMMAND_CONSOLE_LUN* to enable the CCL.

■ Select *NOCOMMAND_CONSOLE_LUN* to disable the CCL.

*IDENTIFIER= n*
*NOIDENTIFIER*

> *IDENTIFIER* provides an alternative way (other than worldwide name) for some operating systems to identify the command console LUN (CCL).

> The *IDENTIFIER* switch is used with the *SCSI_VERSION* switch when the SCSI version is set to SCSI-3 mode. The *IDENTIFIER* switch creates a CCL identifier that makes the controller and the CCL visible to the host. This number can be a value between 1–9999.

*MIRRORED_CACHE*
*NOMIRRORED_CACHE*

> Enables the mirrored-write-back-data cache feature on dual-redundant controllers. When changed, both controllers restart for the new switch setting to take effect.

> The following tasks are performed when the *NOMIRRORED_CACHE* switch is specified:

■ Data in write-back cache is flushed when cache is configured in non-mirrored mode.

■ Enables mirrored write-back cache on both controllers.

■ If an invalid cache configuration exists within the cache modules, an error is generated

> **NOTE:** Both controllers must be operational before this command is accepted.

Specify this switch through only one controller. The controllers must have a valid cache configuration before specifying this switch. The controllers automatically restart when this switch is specified.

> **NOTE:** All unwritten write-cached data is automatically flushed from cache before restart when the *MIRRORED_CACHE* switch is specified. Depending on the amount of data to be flushed, this command may take several minutes to complete before the controller is restarted.

The *NOMIRRORED_CACHE* switch disables mirror mode. Data in write-back cache is flushed when this switch is entered from mirrored mode. This switch disables mirrored write-back cache on both controllers. Therefore, this switch is only to be specified through one controller. The controller must contain a valid cache configuration before this switch is assigned. Unlike going from nonmirrored mode to mirrored mode, going from mirrored mode to nonmirrored mode is permitted with a failed cache module. The controller automatically restarts when this switch is specified.

*NODE_ID=nnnn-nnnn-nnnn-nnnn xx*

Sets the subsystem worldwide name (node ID).

■ nnnn-nnnn-nnnn-nnnn is the node id

■ xx is the checksum

> △ **CAUTION:** Each subsystem has its own unique worldwide name (node ID). If you attempt to set the subsystem worldwide name to a name other than the one that came with the subsystem, the data on the subsystem will not be accessible. Never set two subsystems to the same worldwide name; data corruption will occur.

*NODE_ID* is used to set the worldwide name. Worldwide name, which is actually a number, is a Fibre Channel convention. Every port has a unique 64-bit number, consisting of 16 hex digits. Each StorageWorks fibre channel cabinet is assigned a node ID, which the controller pair in the cabinet assumes. The port IDs derive from the node ID. The node ID assigned to a cabinet never changes.

The node ID ends in a zero. An example of a node ID is 5000-1FE1-FF0C-EE00. The derivation of port IDs from the node ID is done automatically by the controller(s). The numerical value of the port IDs depends on what failover mode the controllers are in.

In transparent failover mode, the port ID of Port 1 for both controllers is the node ID plus 1, so in the above example, the port ID for Port 1 would be 5000-1FE1-FF0C-EE01. Both controllers share this port ID for port 1 because in the transparent failover mode, only one controller's Port 1 is active at any time, and the other Port 1 is on standby. Similarly, both controllers share a port ID for Port 2, which is the node ID plus 2 (in the example, 5000-1FE1-FF0C-EE02).

For a single configuration controller, the Port ID is derived the same way (plus 1 for Port 1, plus 2 for Port 2).

In multiple-bus failover, all four ports are independent, and each has its own port ID. The port IDs are assigned as follows:

■ Controller B port 1 = node ID + 1

■ Controller B port 2 = node ID + 2

■ Controller A port 1 = node ID + 3

■ Controller A port 2 = node ID + 4

When one of the pair of controllers in a dual redundant configuration is replaced, the remaining controller remembers the node ID. When the replacement controller is installed, the in-place controller tells the node ID to the replacement controller, and the replacement controller automatically assumes the correct port IDs.

If a situation occurs that requires the node ID to be reset, it can be done through the CLI by specifying the *NODE_ID* parameter in the SET *this_controller/other_controller* command. The node ID **must** be reset to the number on the cabinet sticker. (The sticker calls it the worldwide name.) The sticker also contains a checksum, which verifies the ID number. When the node ID is entered, both controllers will assume the correct port IDs automatically. Figure 2–1 shows the location of the sticker.



CXO6873B

❶   Node ID                                      ❷   Checksum
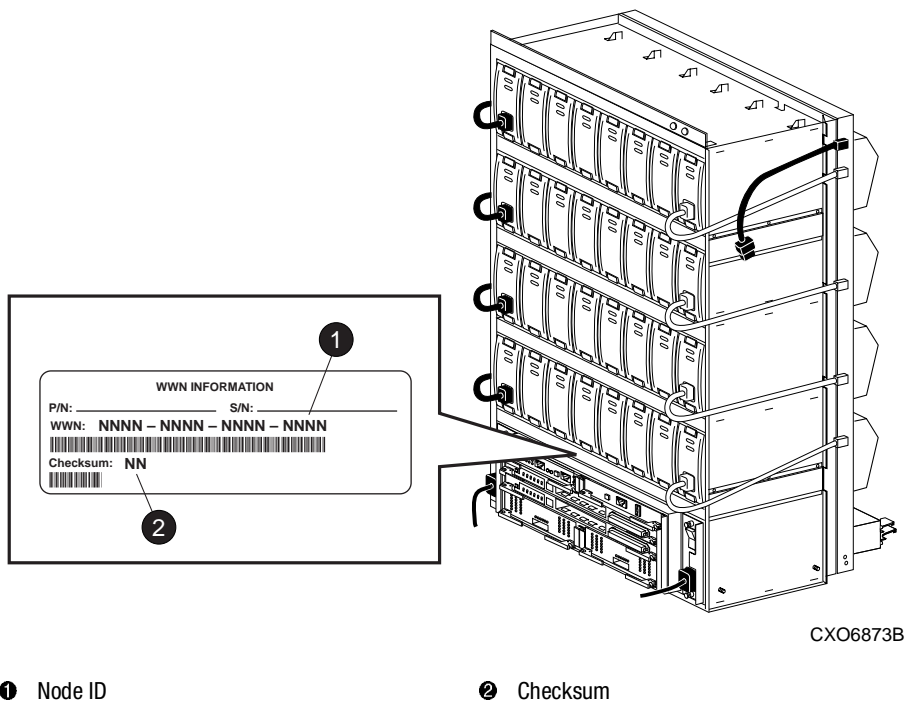
Figure 2–1.  Location of node ID sticker

*PORT_1_AL_PA=*
*PORT_2_AL_PA=*

Specifies the arbitrated loop physical address (ALPA) for the host ports (used only in arbitrated loop (FC-AL) topology). This switch works only when *LOOP_HARD* is specified for the *PORT_1_TOPOLOG*Y or *PORT_2_TOPOLOGY* switches. The range of addresses allowed is 0-EF (hexadecimal).

The factory settings for ALPA are 71 for port 1 and 72 for port 2. If the controller's memory is wiped out, the ALPA for both ports defaults to 69

*PORT_1_TOPOLOGY=LOOP_HARD*
*PORT_1_TOPOLOGY=LOOP_SOFT*
*PORT_1_TOPOLOGY=OFFLINE*
*PORT_2_TOPOLOGY=LOOP_HARD*
*PORT_2_TOPOLOGY=LOOP_SOFT*
*PORT_2_TOPOLOGY=OFFLINE*

Indicates whether the user or controller selects the ALPA for a host port, or whether the port is to be set offline (used only in arbitrated loop (FC-AL) topology). LOOP_HARD allows you to pick the ALPA. LOOP_SOFT requests the controller to pick the ALPA. OFFLINE sets the host port offline. Specify OFFLINE for a port when it will not be used.

**NOTE:** If a topology is already selected, topology must be set to offline before another topology is selected.

*PORT_1_TOPOLOGY=FABRIC*
*PORT_2_TOPOLOGY=FABRIC*

Specifies switch topology for a host port (used only in switch (FC-SW) topology).

*PROMPT="new prompt"*

Specifies a 1- to 16-character prompt displayed when the controller's CLI prompts for input. Only printable ASCII characters and spaces are valid. The new prompt name must be enclosed within quotes. When changed, the new text entered for this switch takes effect immediately.

*REMOTE_COPY = node- name*
*NOREMOTE_COPY*

**IMPORTANT:** This command works in a remote copy environment and requires V8.5P code.

The REMOTE_COPY switch names a controller pair (node) and puts the controller pair into remote copy mode. The node name can be up to 8 characters long. It is suggested that the node name be something meaningful in the context of the DRM setup, such as the location, that distinguishes one controller pair from another.

**NOTE:** Do not use "local" and "remote" as node names. These are reserved as keywords.

All the other DRM commands are hidden until *REMOTE_COP*Y is specified for the controller pair.

The *REMOTE_COPY* switch causes a restart of the controller.

The *NOREMOTE_COPY* switch disables remote copy functionality for the controller pair. It also removes the node name given by the *REMOTE_COPY* switch. The switch does not work if there are any remote copy sets or remote copy connections associated with the controller pair.

**NOTE:** No other switch can be specified in the same command string if NOREMOTE_COPY is specified.

*SCSI_VERSION=SCSI-2*
*SCSI_VERSION=SCSI-3*

Specifies the host protocol to use. SCSI-3 means that the controller uses some SCSI 3 commands. It does not mean that it is a fully-compliant SCSI 3 device.

A side effect of the SCSI mode is how the command console LUN (CCL) is handled.

■ SCSI-2 specifies that the CCL is not fixed at a particular location, but floats depending on the configuration.

■ SCSI-3 specifies that the CCL is fixed at LUN 0.

Changes to this switch take place at the next controller restart.

*TERMINAL_PARITY=ODD*
*TERMINAL_PARITY=EVEN*
*NOTERMINAL_PARITY*

Specifies the parity with which terminal data is transmitted and received. When changed, the new setting for this switch takes effect immediately

*TERMINAL_SPEED=baud_rate*
*TERMINAL_SPEED=9600*

Sets the terminal transmission and reception speed (**baud** rate) to 4800, 9600 (default), or 19200 baud. When changed, the new value entered for this switch takes effect immediately.

*TIME=dd–mmm–yyyy:hh:mm:ss*

Sets the date and time. The time is set on both controllers in a dual-redundant configuration.When changed, the new value entered for this switch takes effect immediately.

*UPS=NODE_ONLY*
*UPS=DATACENTER_WIDE*
*NOUPS*

Specifies whether the controller should perform regular battery condition checks. Setting the *UPS/NOUPS* switch for either controller sets the switch for both controllers. Both controllers must be restarted in order for the new setting to take effect.

■ Specify *UPS=NODE_ONLY* if your storage subsystem power is supplied by a separate uninterruptible power supply (UPS) directly connected to the PVA. If *NODE_ONLY* is specified the controller continues to check the condition of the cache batteries.

> **CAUTION:** Setting *UPS=NODE_ONLY or UPS=DATACENTER_WIDE* without having a UPS or similar backup system in place may result in data loss if power is interrupted.

■ Specify *UPS=DATACENTER_WIDE* if the whole room or building (not just the subsystem) is supported by a UPS. If *DATACENTER_WIDE* is specified, the controller does not check the condition of the cache batteries and ignores the battery's state. This causes RAIDsets and mirrorsets to always be available, regardless of the condition of the cache batteries.

■ Specify *NOUPS* to instruct the controller to perform regular cache battery checks and evaluate the condition of the cache batteries.

## Examples

To change the other controller's CLI prompt, enter:

    SET OTHER_CONTROLLER PROMPT="TOP"

To set the name of the controller pair and put the controller pair in remote copy mode, enter:

    SET THIS_CONTROLLER REMOTE_COPY=LONDON

> **NOTE:** The other controller in the pair automatically receives the same name.

**See also**

ADD REMOTE_COPY_SETS
SET *remote-copy-set-name*
SHOW *this_controller*
SHOW *other_controller*

# SET *unit-number*

This command changes the characteristics of a unit.

### Syntax

SET *unit-number*

### Parameters

*unit-number*

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

### Switches

Table 2–4 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 2–4 SET UNIT Switches for Existing Containers**

| Switch | Container Type | RAIDset | Stripeset | Mirrorset | NO-Transportable Disk | Transportable Disk |
|---|---|---|---|---|---|---|
| *ENABLE_ACCESS_PATH* *DISABLE_ACCESS_PATH* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *MAXIMUM_CACHED_TRANSFER* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *IDENTIFIER* *NOIDENTIFIER* | | ✓ | ✓ | ✓ | ✓ | |
| *PREFERRED_PATH* *NOPREFERRED_PATH* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *READ_CACHE* *NOREAD_CACHE* | | ✓ | ✓ | ✓ | ✓ | ✓ |

**Table 2–4 SET UNIT Switches for Existing Containers  (Continued)**

| Switch | Container Type | RAIDset | Stripeset | Mirrorset | N0- Transportable Disk | Transportable Disk |
|---|---|---|---|---|---|---|
| *READAHEAD_CACHE* *NOREADAHEAD_CACHE* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *RUN* *NORUN* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *WRITE_PROTECT* *NOWRITE_PROTECT* | | ✓ | ✓ | ✓ | ✓ | ✓ |
| *WRITEBACK_CACHE* *NOWRITEBACK_CACHE* | | ✓ | ✓ | ✓ | ✓ | |

**NOTE:** Regardless of storageset type, the *NORUN* switch cannot be specified for partitioned units.

*ENABLE_ACCESS_PATH=*
*DISABLE_ACCESS_PATH*

> △ **CAUTION:**  If the storage subsystem has more than one host connection, access path must be specified carefully to avoid giving undesirable host connections access to the unit. The default condition is that access paths to all host connections are enabled. To restrict host access to a set of host connections, specify
> *DISABLE_ACCESS_PATH=ALL* when the unit is added, then use the SET *unit* command to specify the set of host connections that are to have access to the unit.
>
> Enabling the access path to a particular host connection does not override previously enabled access paths. All access paths previously enabled are still valid; the new host connection is simply added to the list of connections that can access the unit.
>
> The procedure of restricting access by enabling all access paths then disabling selected paths is particularly not recommended because of the potential data/security breach that occurs when a new host connection is added.

Specifies the access path. It can be a single specific host ID, multiple host IDs, or all host IDs (ALL). If you have multiple hosts on the same bus, you can use this switch to restrict hosts from accessing certain units. This switch limits visibility of specific units from certain hosts. For example, if two hosts are on the same bus, you can restrict each host to access only specific units.

*MAXIMUM_CACHED_TRANSFER=32*
*MAXIMUM_CACHED_TRANSFER=n*

Sets the largest number of write blocks to be cached by the controller. The controller will not cache any transfers over the set size. Acceptable write block sizes are 1 through 1024. The *MAXIMUM_CACHED_TRANSFER* switch affects both read and write-back cache when set on a controller that has read and write-back caching.

*IDENTIFIER=n*
*NOIDENTIFIER*

The *IDENTIFIER* switch is used with the SCSI_VERSION switch when the SCSI version is set to SCSI-3 mode. The *IDENTIFIER* switch creates a unique *unit identifier* makes the unit visible to the host. This number can be a value between 1–9999.

Identifier is an alternative way (other than worldwide name) for OpenVMS operating systems to identify the unit.

*PREFERRED_PATH=OTHER_CONTROLLER*
*PREFERRED_PATH=THIS_CONTROLLER*
*NOPREFERRED_PATH*

The *PREFERRED_PATH* switch allows you to balance the 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 (used only in Windows NT operating systems).

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.

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

■ When no preferred path is assigned, the unit is targeted through the controller which detects the unit first after the controllers start.

■ Select *PREFERRED_PATH=THIS_CONTROLLER* to instruct "this controller" to bring the units online.

■ Select *PREFERRED_PATH=OTHER_CONTROLLER* to instruct the "other controller" to bring the units online.

> **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 inherit the same path.

*READ_CACHE*
*NOREAD_CACHE*

If *READ_CACHE* is selected, when the controller receives a read request from the host, it reads the data from the disk drives, delivers it to the host, and stores the data in its cache module. Subsequent reads for the same data will take the data from cache rather than accessing the data from the disks.

Read caching improves performance in almost all situations. Therefore, it is recommended you leave its default setting, *READ_CACHE*, enabled. However, under certain conditions, such as when performing a backup, read caching may not be necessary since only a small amount of data is cached. In such instances, it may be beneficial to disable the read cache function and remove the processing overhead associated with caching data.

*READAHEAD_CACHE*
*NOREADAHEAD_CACHE*

Enables the controller to keep track of read I/Os. If the controller detects sequential read I/Os from the host, it will then try to keep ahead of the host by reading the next sequential blocks of data (those the host has not yet requested) and put the data in cache. This process is sometimes referred to as prefetch. The controller can detect multiple sequential I/O requests across multiple units.

Read ahead caching improves host application performance since the data will be read from the controller cache instead of disk. Read ahead caching is the default for units.

If you are adding a unit that is not expected to get sequential I/O requests, select *NOREADAHEAD_CACHE* for the unit.

*RUN*
*NORUN*

Controls the unit's availability to the host:

■ Specify *RUN* to make a unit available to the host.

■ Specify *NORUN* to make a unit unavailable to the host and to cause any data in cache to be flushed to one or more drives. *NORUN* spins down all the disks used in the unit. The drives making up the unit spin down after the data has been completely flushed.

**NOTE:** Do not specify the *RUN* and *NORUN* switches for partitions.

*WRITE_PROTECT*
*NOWRITE_PROTECT*

Specifies whether data contained on the selected unit can be overwritten:

■ Specify *WRITE_PROTECT* to prevent host write operations to the unit. However, the controller may still write to a write-protected RAIDset to satisfy a reconstruct pass or to reconstruct a newly replaced member. Additionally, metadata, reconstruct, and copy writes are still allowed to RAIDsets and mirrorsets.

■ Specify *NOWRITE_PROTECT* to allow the host to write data to the unit. This allows the controller to overwrite existing data. *NOWRITE_PROTECT* is the default for transportable disks.

*WRITEBACK_CACHE*
*NOWRITEBACK_CACHE*

Specifying *WRITEBACK_CACHE* allows the controller to declare the write operation "complete" as soon as the data reaches its cache memory. The controller performs the slower operation of writing the data to the disk drives at a later time.

*WRITEBACK_CACHE* cannot be applied to transportable disks.

> ⚠ **CAUTION:** Though there is built-in redundancy to protect data contained in cache, allowing data to be written to write-back cache may result in the loss of data if the controller fails.

Specifying *NOWRITEBACK_CACHE* enables only write-through caching. In write-through caching, when the controller receives a write request from the host it places the data in its cache module, writes the data to the disk drives, then notifies the host when the write operation is complete. This process is called write-through caching because the data passes through—and is stored in—the cache memory on its way to the disk drives. Write-through caching is only enabled when write-back caching is disabled.

**NOTE:** If the switch is changed from *WRITEBACK_CACHE* to *NOWRITEBACK_CACHE*, he controller may take up to five minutes to flush data contained within cache memory.

**Example**

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

    SET D102 WRITE_PROTECT NOREAD_CACHE

To allow only a host connection named Roger1 to access unit D0, enter:

    SET D0 DISABLE_ACCESS_PATH=ALL

    SET D0 ENABLE_ACCESS_PATH=ROGER1

**See also**

SHOW UNITS
SHOW *unit-number*

# SHOW

This command displays information about controllers, association sets, remote copy sets, storagesets, devices, partitions, and units.

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

## Syntax

SHOW *parameter*

## Parameters

The following parameters are available to the SHOW command:

■  association-set-name

■  associations

■  concatset-name

■  concatsets

■  connections

■  controller ("this controller" or "other controller")

■  device-name

■  device-type

■  EMU

■  remote copy

■  remote-copy-set-name

■  storageset-name

■  storageset-type

■  unit-number

■  units

These parameters are described in the paragraphs that follow.

*association-set-name*

Specifies the name of the association set.

*associations*

Shows information on all association sets known to the controller pair.

*concatset-name*

Specifies the name of the concatset to be displayed.

concatsets

Shows information on all concatsets.

*connection*

Shows the following connection information: connection name, operating system, controller, controller port, adapter ID address, online or offline status, and unit offset.

*controller*

*this_controller*
*other_controller*
Specifies the controller to be displayed.

*device-name*

Specifies the name of a particular device to be displayed. For example, SHOW DISK20100 displays information about the device named DISK20100.

*device-type*

Specifies the type of devices you want to be displayed. Valid choices are:

■ *devices*—Shows all devices attached to the controller.

■ *disks*—Shows all disks attached to the controller.

*EMU*

Displays information regarding the status of the environmental monitoring unit.

*remote_copy*

Displays information on all remote copy sets known to the controller pair.

*remote-set-name*

Specifies the remote copy set to display.

*storageset-name*

> Specifies the name of a particular storageset to be displayed. For example, SHOW STRIPE1 displays information about the stripeset named STRIPE1.

*storageset-type*

> Specifies the type of storageset to be displayed. Valid types are:

- *concatsets*

- *failedset*

- *mirrorsets*

- *raidsets*

- *sparesets*

- *storagesets*

- *stripesets*

*unit-number*

> Specifies the name of a particular unit to be displayed. For example, SHOW D102 displays information about the unit named D102.

*units*

> Displays information for all units configured to the controller.

## Switches

*FULL*

> Displays additional information.

## Examples

To display a listing of disks, enter:

SHOW DISKS

```
Name  . . . . . . . . . . . . . . . . . . TypePort  Targ  LunUsed by
-------------------------------------------------------------------
DISK20300 . . . . . . . . . . . . . . . . .disk1      0       0D100
DISK10100 . . . . . . . . . . . . . . . . .disk1      1       0D101
```

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

SHOW DEVICES FULL

```
Name      Type        Port      Targ      Lun       Used by
-------------------------------------------------------------------------
DISK100   disk          1         0         0           R0

       Switches:
          NOTRANSPORTABLE
         TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 20.00 MHZ negotiated)

       Size: 8378028 blocks
DISK200   disk          2         0         0           R0

       Switches:
          NOTRANSPORTABLE
         TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 20.00 MHZ negotiated)

       Size: 8378028 blocks
DISK300 disk            3         0         0           R0

       Switches:
           NOTRANSPORTABLE
         TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 20.00 MHZ negotiated)

       Size: 8378028 blocks
DISK400   disk          4         0         0           R0

       Switches:
           NOTRANSPORTABLE
         TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 20.00 MHZ negotiated)

       Size: 8378028 blocks
       Configuration being backed up on this container
```

To display a complete listing of the mirrorset named MIRR1, enter:

**SHOW MIRR1**

```
Name        Storageset       Uses                                Used by
-------------------------------------------------------------------------
MIRR1       mirrorset        DISK50300                              S0
            DISK60300
            Switches:
             POLICY (for replacement) = BEST_PERFORMANCE
             COPY (priority) = NORMAL
```

```
        READ_SOURCE = LEAST_BUSY
        MEMBERSHIP = 2, 2 members present
      State:
        NORMAL
        DISK60300 (member  0) is NORMAL
        DISK50300 (member  1) is NORMAL
      Size: 17769177 blocks
```

To show the full information for a mirrorset, enter:

SHOW MIRRORSET FULL

```
Name            Storageset       Uses                                 Used by
-------------------------------------------------------------------------------
M0 mirrorset    DISK100                                               D0
                DISK200
                DISK300
                DISK400
        Switches:
          COPY (priority) = NORMAL
          READ_SOURCE = LEAST_BUSY
          MEMBERSHIP = 4, 4 MEMBER PRESENT

        State:
          RECONSTRUCTING
  Size: 8378028 blocks
```

To show the information for a controller, enter:

SHOW THIS_CONTROLLER

```
Controller:
HSG80 (c) Compaq ZG81701083 Software xxxxx, Hardware xxxx
  NODE_ID=AAAA-AAAA-AAAA-AAA0
  ALLOCATION_CLASS= 0
  SCSI_VERSION= SCSI-3
  Configured for MULTIBUS_FAILOVER with ZG81701006
  in dual-redundant configuration
  Device Port SCSI address 7
  Time: NOT SET
  Command Console LUN is lun 0 (identifier = 109)

Host Port_1:
  Reported PORT_ID= AAAA-AAAA-AAAA-AAA1
  PORT_1_TOPOLOGY= FABRIC (fabric up)
  Address= 200113
Host Port_2:
```

```
      Reported PORT_ID= AAAA-AAAA-AAAA-AAA2
      PORT_1_TOPOLOGY= FABRIC (fabric up)
      Address= 200313
      REMOTE_COPY= 1709
Cache:
   512 megabyte write cache, version 0022
   Cache is GOOD
   No unflushed data in cache
   CACHE_FLUSH_TIMER = DEFAULT (10 seconds)
   NOCACHE_UPS
Mirrored Cache:
   256 megabyte write cache, version 0022
   Cache is GOOD
   No unflushed data in cache
Battery:
   NOUPS
   FULLY CHARGED
   Expires: 12-MAY-2001
```

## SHUTDOWN *this_controller*
## SHUTDOWN *other_controller*

This command flushes all user data from the specified controller's write-back cache (if present) and shuts down the controller. The controller does not automatically restart. All units accessed through the failed controller fail over to the surviving controller.

### Syntax

SHUTDOWN *controller*

### Parameter

*controller*

> *this_controller*
> *other_controller*
> Indicates which controller is to shut down.

### Switches

The following switches are available to the SHUTDOWN controller command:

■ IGNORE_ERRORS/NOIGNORE_ERRORS

■ IMMEDIATE_SHUTDOWN/NOIMMEDIATE_SHUTDOWN

These switches are described in the following paragraphs.

*IGNORE_ERRORS*
*NOIGNORE_ERRORS (Default)*

Controls the reaction of the controller, based on the status of write-back cache:

> **CAUTION:** The *IGNORE_ERRORS* switch causes 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 until the controller flushes the cache.

■ Specify *IGNORE_ERRORS* to instruct the controller to shut down even if the data within write-back cache cannot be written to the devices.

■ Specify *NOIGNORE_ERRORS* to instruct the controller to stop operation if the data within write-back cache cannot be written to the devices.

*IMMEDIATE_SHUTDOWN*
*NOIMMEDIATE_SHUTDOWN (Default)*

Instructs the controller when to shutdown:

> **CAUTION:** The *IMMEDIATE_SHUTDOWN* switch causes 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 until the controller flushes the cache.

■ Specify *IMMEDIATE_SHUTDOWN* if you want the controller to shutdown immediately without checking for online devices or before flushing data from the write-back cache to devices.

■ Specify *NOIMMEDIATE_SHUTDOWN* if you want the controller not to shutdown without checking for online devices or before all data has been flushed from the write-back cache to devices.

## Examples

To shut down "this controller," enter:

```
SHUTDOWN THIS_CONTROLLER
```

To shut down the other controller, even if it cannot write all of the write-back cached data to the units, enter:

```
SHUTDOWN OTHER_CONTROLLER IGNORE_ERRORS
```

## See also

RESTART *controller*
SELFTEST *controller*

## SITE_FAILOVER

**NOTE:** This command works only in a Data Replication Manager (DRM) environment and requires V8.5P code. <u>Use of this command is heavily restricted.</u> See *Data Replication Manager HSG80 ACS Version 8.5P Operations Guide* for explanation and examples of usage.

This command performs the site failover function in the event the original initiator site is not operational. This command changes the role of a target unit to an initiator unit. A remote copy set is created with two members:

■ the original target (which is the new initiator)

■ the original initiator (which is the new target)

### Syntax

SITE_FAILOVER remote-node-name\remote-copy-set-name

### Parameters

There are no parameters associated with this command.

### Switches

There are no switches associated with this command.

### Examples

See *Data Replication Manager HSG80 ACS Version 8.5P Operations Guide*.

## UNMIRROR

This command converts a one-member mirrorset back to a non-mirrored disk drive and deletes its mirrorset from the list of known mirrorsets. This command can only be used on mirrorsets already members of higher-level containers (stripesets or units).

The UNMIRROR command is not valid for disk drives that have a capacity greater than the capacity of the existing mirrorset. If a mirrorset is comprised of disk drives with different capacities, the mirrorset capacity is limited to the size of the smallest member; larger members contain unused capacity. If a member with unused capacity is the last remaining member of a mirrorset, the UNMIRROR command cannot be used to change the disk drive back to a single-disk unit. This change would cause a change in the reported disk capacity, possibly corrupting user data.

### Syntax

UNMIRROR *disk-name*

### Parameters

*disk-name*

Specifies the name of the normal mirrorset member to be removed from a mirror storageset.

### Switches

There are no switches associated with this command.

### Example

To convert DISK10300 back to a single device, enter:

UNMIRROR DISK10300

### See also

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

# Glossary

This glossary defines terms pertaining to the HSG80 Fibre Channel array controller. It is not a comprehensive glossary of computer terms.

| | |
|---|---|
| **adapter** | A device that converts the protocol and hardware interface of one bus type into another without changing the function of the bus. |
| **ACS** | *See* array controller software. |
| **AL_PA** | *See* arbitrated loop physical address. |
| **ANSI** | Pronounced "ann-see." Acronym for the American National Standards Institute. An organization who develops standards used voluntarily by many manufacturers within the USA. ANSI is not a government agency. |
| **arbitrated loop physical address** | Abbreviated AL_PA. A one-byte value used to identify a port in an Arbitrated Loop topology. |
| **array controller** | *See* controller. |
| **array controller software** | Abbreviated ACS. Software contained on a removable ROM program card that provides the operating system for the array controller. |
| **asynchronous** | Pertaining to events that are scheduled as the result of a signal asking for the event; pertaining to that which is without any specified time relation. *See also* synchronous. |

| | |
|---|---|
| **autospare** | A controller feature that automatically replaces a failed disk drive. To aid the controller in automatically replacing failed disk drives, you can enable the AUTOSPARE switch for the failedset causing physically replaced disk drives to be automatically placed into the spareset. Also called "autonewspare." |
| **bad block** | A data block that contains a physical defect. |
| **bad block replacement** | Abbreviated BBR. A replacement routine that substitutes defect-free disk blocks for those found to have defects. This process takes place in the controller, transparent to the host. |
| **backplane** | The electronic printed circuit board into which you plug subsystem devices—for example, the SBB or power supply. |
| **BBR** | *See* bad block replacement. |
| **BIST** | *See* built-in self-test. |
| **bit** | A single binary digit having a value of either 0 or 1. A bit is the smallest unit of data a computer can process. |
| **block** | Also called a sector. The smallest collection of consecutive bytes addressable on a disk drive. In integrated storage elements, a block contains 512 bytes of data, error codes, flags, and the block's address header. |
| **bootstrapping** | A method used to bring a system or device into a defined state by means of its own action. For example, a machine routine whose first few instructions are enough to bring the rest of the routine into the computer from an input device. |
| **built-in self-test** | A diagnostic test performed by the array controller software on the controller's policy processor. |
| **byte** | A binary character string made up of 8 bits operated on as a unit. |
| **cache memory** | A portion of memory used to accelerate read and write operations. |
| **CDU** | Cable distribution unit. The power entry device for StorageWorks cabinets. The CDU provides the connections necessary to distribute power to the cabinet shelves and fans. |
| **channel** | An interface which allows high speed transfer of large amounts of data. Another term for a SCSI bus. *See also* SCSI. |
| **chunk** | A block of data written by the host. |

| | |
|---|---|
| **chunk size** | The number of data blocks, assigned by a system administrator, written to the primary RAIDset or stripeset member before the remaining data blocks are written to the next RAIDset or stripeset member. |
| **CLCP** | An abbreviation for code-load code-patch utility. |
| **CLI** | *See* command line interpreter. |
| **coax** | A two-conductor wire in which one conductor completely wraps the other with the two separated by insulation. |
| **coaxial cable** | *See* coaxial cable. |
| **cold swap** | A method of device replacement that requires the entire subsystem to be turned off before the device can be replaced. *See also* hot swap and warm swap. |
| **command line interpreter** | The configuration interface to operate the controller software. |
| **configuration file** | A file that contains a representation of a storage subsystem's configuration. |
| **container** | 1) Any entity that is capable of storing data, whether it is a physical device or a group of physical devices. (2) A virtual, internal controller structure representing either a single disk or a group of disk drives linked as a storageset. Stripesets and mirrorsets are examples of storageset containers the controller uses to create units. |
| **controller** | A hardware device that, with proprietary software, facilitates communications between a host and one or more devices organized in an array. HS family controllers are examples of array controllers. |
| **copying** | A state in which data to be copied to the mirrorset is inconsistent with other members of the mirrorset. *See also* normalizing. |
| **copying member** | Any member that joins the mirrorset after the mirrorset is created is regarded as a copying member. Once all the data from the normal member (or members) is copied to a normalizing or copying member, the copying member then becomes a normal member. *See also* normalizing member. |
| **CSR** | An acronym for control and status register. |
| **DAEMON** | Pronounced "demon." A program usually associated with a UNIX systems that performs a utility (housekeeping or maintenance) function without being requested or even known of by the user. A daemon is a diagnostic and execution monitor. |
| **data center cabinet** | A generic reference to large DIGITAL subsystem cabinets, such as the SW600-series and 800-series cabinets in which StorageWorks components can be mounted. |

| | |
|---|---|
| **data striping** | The process of segmenting logically sequential data, such as a single file, so that segments can be written to multiple physical devices (usually disk drives) in a round-robin fashion. This technique is useful if the processor is capable of reading or writing data faster than a single disk can supply or accept the data. While data is being transferred from the first disk, the second disk can locate the next segment. |
| **device** | *See* node and peripheral device. |
| **differential I/O module** | A 16-bit I/O module with SCSI bus converter circuitry for extending a differential SCSI bus. *See also* I/O module. |
| **differential SCSI bus** | A bus in which a signal's level is determined by the potential difference between two wires. A differential bus is more robust and less subject to electrical noise than is a single-ended bus. |
| **DIMM** | Dual inline Memory Module. |
| **dirty data** | The write-back cached data that has not been written to storage media, even though the host operation processing the data has completed. |
| **DMA** | Direct Memory Access. |
| **DOC** | DWZZA-On-a-Chip. An NCR53C120 SCSI bus extender chip used to connect a SCSI bus in an expansion cabinet to the corresponding SCSI bus in another cabinet. |
| **driver** | A hardware device or a program that controls or regulates another device. For example, a device driver is a driver developed for a specific device that allows a computer to operate with the device, such as a printer or a disk drive. |
| **dual-redundant configuration** | A controller configuration consisting of two active controllers operating as a single controller. If one controller fails, the other controller assumes control of the failing controller's devices. |
| **dual-simplex** | A communications protocol that allows simultaneous transmission in both directions in a link, usually with no flow control. |
| **DUART** | Dual universal asynchronous receiver and transmitter. An integrated circuit containing two serial, asynchronous transceiver circuits. |
| **ECB** | External cache battery. The unit that supplies backup power to the cache module in the event the primary power source fails or is interrupted. |
| **ECC** | Error checking and correction. |
| **EDC** | Error detection code. |

**EIA**               The abbreviation for Electronic Industries Association. EIA is a standards organization specializing in the electrical and functional characteristics of interface equipment. Same as Electronic Industries Association.

**EMU**               Environmental monitoring unit. A unit that provides increased protection against catastrophic failures. Some subsystem enclosures include an EMU which works with the controller to detect conditions such as failed power supplies, failed blowers, elevated temperatures, and external air sense faults. The EMU also controls certain cabinet hardware including DOC chips, alarms, and fan speeds.

**ESD**               Electrostatic discharge. The discharge of potentially harmful static electrical voltage as a result of improper grounding.

**extended subsystem**   A subsystem in which two cabinets are connected to the primary cabinet.

**external cache battery**   *See* ECB.

**F_Port**            A port in a fabric where an N_Port or NL_Port may attach.

**fabric**            A group of interconnections between ports that includes a fabric element.

**failedset**         A group of failed mirrorset or RAIDset devices automatically created by the controller.

**failover**          The process that takes place when one controller in a dual-redundant configuration assumes the workload of a failed companion controller. Failover continues until the failed controller is repaired or replaced.

**FC–AL**             The Fibre Channel Arbitrated Loop standard.

**FC–ATM**            ATM AAL5 over Fibre Channel

**FC–FG**             Fibre Channel Fabric Generic Requirements

**FG–FP**             Fibre Channel Framing Protocol (HIPPI on FC)

**FC-GS-1**           Fibre Channel Generic Services-1

**FC–GS-2**           Fibre Channel Generic Services-2

**FC–IG**             Fibre Channel Implementation Guide

**FC–LE**             Fibre Channel Link Encapsulation (ISO 8802.2)

**FC–PH**             The Fibre Channel Physical and Signaling standard.

**FC–SB**              Fibre Channel Single Byte Command Code Set

**FC–SW**              Fibre Channel Switched Topology and Switch Controls

**FCC**                Federal Communications Commission. The federal agency responsible for
                       establishing standards and approving electronic devices within the United States.

**FCC Class A**        This certification label appears on electronic devices that can only be used in a
                       commercial environment within the United States.

**FCC Class B**        This certification label appears on electronic devices that can be used in either a
                       home or a commercial environment within the United States.

**FCP**                The mapping of SCSI-3 operations to Fibre Channel.

**FDDI**               Fiber Distributed Data Interface. An ANSI standard for 100 megabaud transmission
                       over fiber optic cable.

**FD SCSI**            The fast, narrow, differential SCSI bus with an 8-bit data transfer rate of 10 MB/s.
                       *See also* FWD SCSI and SCSI.

**fiber**              A fiber or optical strand. Spelled ***fibre*** in Fibre Channel.

**fiber optic cable**  A transmission medium designed to transmit digital signals in the form of pulses of
                       light. Fiber optic cable is noted for its properties of electrical isolation and resistance
                       to electrostatic contamination.

**FL_Port**            A port in a fabric where N_Port or an NL_Port may be connected.

**flush**              The act of writing dirty data from cache to a storage media.

**FMU**                Fault management utility.

**forced errors**      A data bit indicating a corresponding logical data block contains unrecoverable data.

**frame**              An invisible unit used to transfer information in Fibre Channel.

**FRU**                Field replaceable unit. A hardware component that can be replaced at the customer's
                       location by DIGITAL service personnel or qualified customer service personnel.

**full duplex (n)**    A communications system in which there is a capability for 2-way transmission and
                       acceptance between two sites at the same time.

**full duplex (adj)**  Pertaining to a communications method in which data can be transmitted and
                       received at the same time.

| | |
|---|---|
| **FWD SCSI** | A fast, wide, differential SCSI bus with a maximum 16-bit data transfer rate of 20 MB/s. *See also* SCSI and FD SCSI. |
| **GBIC** | Gigabit Interface Converter. The devices that are inserted into the ports of the Fibre Channel switch and that hold the Fibre Channel cables. |
| **GLM** | Gigabit link module |
| **giga** | A prefix indicating a billion ($10^9$) units, as in gigabaud or gigabyte. |
| **gigabaud** | An encoded bit transmission rate of one billion ($10^9$) bits per second. |
| **gigabyte** | A value normally associated with a disk drives storage capacity, meaning a billion ($10^9$) bytes. The decimal value 1024 is usually used for one thousand. |
| **half-duplex (adj)** | Pertaining to a communications system in which data can be either transmitted or received but only in one direction at one time. |
| **hard address** | The AL_PA which an NL_Port attempts to acquire during loop initialization. |
| **heterogeneous host support** | Also called *noncooperating host support*. |
| **HIPPI–FC** | Fibre Channel over HIPPI |
| **host** | The primary or controlling computer to which a storage subsystem is attached. |
| **host adapter** | A device that connects a host system to a SCSI bus. The host adapter usually performs the lowest layers of the SCSI protocol. This function may be logically and physically integrated into the host system. |
| **hot disks** | A disk containing multiple hot spots. Hot disks occur when the workload is poorly distributed across storage devices which prevents optimum subsystem performance. *See also* hot spots. |
| **hot spots** | A portion of a disk drive frequently accessed by the host. Because the data being accessed is concentrated in one area, rather than spread across an array of disks providing parallel access, I/O performance is significantly reduced. *See als*o hot disks. |
| **hot swap** | A method of device replacement that allows normal I/O activity on a device's bus to remain active during device removal and insertion. The device being removed or inserted is the only device that cannot perform operations during this process. *See also* cold swap and warm swap. |

**IBR**            Initial Boot Record.

**ILF**            Illegal function.

**INIT**           Initialize input and output.

**initiator**      A SCSI device that requests an I/O process to be performed by another SCSI device, namely, the SCSI target. The controller is the initiator on the device bus. The host is the initiator on the host bus.

**instance code**  A four-byte value displayed in most text error messages and issued by the controller when a subsystem error occurs. The instance code indicates when during software processing the error was detected.

**interface**      A set of protocols used between components, such as cables, connectors, and signal levels.

**I/O**            Refers to input and output functions.

**I/O driver**     The set of code in the kernel that handles the physical I/O to a device. This is implemented as a fork process. Same as driver.

**I/O interface**  *See* interface.

**I/O module**     A 16-bit SBB shelf device that integrates the SBB shelf with either an 8-bit single ended, 16-bit single-ended, or 16-bit differential SCSI bus.

**I/O operation**  The process of requesting a transfer of data from a peripheral device to memory (or visa versa), the actual transfer of the data, and the processing and overlaying activity to make both of those happen.

**IPI**            Intelligent Peripheral Interface. An ANSI standard for controlling peripheral devices by a host computer.

**IPI-3 Disk**     Intelligent Peripheral Interface Level 3 for Disk

**IPI-3 Tape**     Intelligent Peripheral Interface Level 3 for Tape

**JBOD**           Just a bunch of disks. A term used to describe a group of single-device logical units.

**kernel**         The most privileged processor access mode.

**LBN**            Logical Block Number.

**L_port**         A node or fabric port capable of performing arbitrated loop functions and protocols. NL_Ports and FL_Ports are loop-capable ports.

| | |
|---|---|
| **LED** | Light Emitting Diode. |
| **link** | A connection between two Fibre Channel ports consisting of a transmit fibre and a receive fibre. |
| **logical block number** | *See* LBN. |
| **local connection** | A connection to the subsystem using either its serial maintenance port or the host's SCSI bus. A local connection enables you to connect to one subsystem controller within the physical range of the serial or host SCSI cable. |
| **local terminal** | A terminal plugged into the EIA-423 maintenance port located on the front bezel of the controller. *See also* maintenance terminal. |
| **logical bus** | A single-ended bus connected to a differential bus by a SCSI bus signal converter. |
| **logical unit** | A physical or virtual device addressable through a target ID number. LUNs use their target's bus connection to communicate on the SCSI bus. |
| **logical unit number** | A value that identifies a specific logical unit belonging to a SCSI target ID number. A number associated with a physical device unit during a task's I/O operations. Each task in the system must establish its own correspondence between logical unit numbers and physical devices. |
| **logon** | Also called login. A procedure whereby a participant, either a person or network connection, is identified as being an authorized network participant. |
| **loop** | *See* arbitrated loop. |
| **loop_ID** | A seven-bit value numbered contiguously from zero to 126-decimal and represent the 127 legal AL_PA values on a loop (not all of the 256 hex values are allowed as AL_PA values per FC-AL. |
| **loop tenancy** | The period of time between the following events: when a port wins loop arbitration and when the port returns to a monitoring state. |
| **L_Port** | A node or fabric port capable of performing Arbitrated Loop functions and protocols. NL_Ports and FL_Ports are loop-capable ports. |
| **LRU** | Least recently used. A cache term used to describe the block replacement policy for read cache. |
| **Mbps** | Approximately one million ($10^6$) bits per second—that is, megabits per second. |

**MBps**              Approximately one million ($10^6$) bytes per second—that is, megabytes per second.

**maintenance**      An EIA-423-compatible terminal used with the controller. This terminal is used to
**terminal**         identify the controller, enable host paths, enter configuration information, and check
                     the controller's status. The maintenance terminal is not required for normal
                     operations.
                     *See also* local terminal.

**member**           A container that is a storage element in a RAID array.

**metadata**         The data written to a disk for the purposes of controller administration. Metadata
                     improves error detection and media defect management for the disk drive. It is also
                     used to support storageset configuration and partitioning. Nontransportable disks
                     also contain metadata to indicate they are uniquely configured for StorageWorks
                     environments. Metadata can be thought of as "data about data."

**mirroring**        The act of creating an exact copy or image of data.

**mirrorset**        *See* RAID level 1.

**MIST**             Module Integrity Self-Test.

**N_port**           A port attached to a node for use with point-to-point topology or fabric topology.

**NL_port**          A port attached to a node for use in all three topologies.

**network**          A data communication, a configuration in which two or more terminals or devices are
                     connected to enable information transfer.

**node**             In data communications, the point at which one or more functional units connect
                     transmission lines.

**Non-L_Port**       A Node of Fabric port that is not capable of performing the Arbitrated Loop functions
                     and protocols. N_Ports and F_Ports loop-capable ports.

**non-participating**  A mode within an L_Port that inhibits the port from participating in loop activities.
**mode**             L_Ports in this mode continue to retransmit received transmission words but are not
                     permitted to arbitrate or originate frames. An L_Port in non-participating mode may
                     or may not have an AL_PA. *See also* participating mode.

**nominal**          The desired number of mirrorset members when the mirrorset is fully populated with
**membership**       active devices. If a member is removed from a mirrorset, the actual number of
                     members may fall below the "nominal" membership.

| | |
|---|---|
| **node** | In data communications, the point at which one or more functional units connect transmission lines. In Fibre Channel, a device that has at least one N_Port or NL_Port. |
| **nonredundant controller configuration** | (1) A single controller configuration. (2) A controller configuration that does not include a second controller. |
| **normal member** | A mirrorset member that, block-for-block, contains the same data as other normal members within the mirrorset. Read requests from the host are always satisfied by normal members. |
| **normalizing** | Normalizing is a state in which, block-for-block, data written by the host to a mirrorset member is consistent with the data on other normal and normalizing members. The normalizing state exists only after a mirrorset is initialized. Therefore, no customer data is on the mirrorset. |
| **normalizing member** | A mirrorset member whose contents is the same as all other normal and normalizing members for data that has been written since the mirrorset was created or lost cache data was cleared. A normalizing member is created by a normal member when either all of the normal members fail or all of the normal members are removed from the mirrorset. *See also* copying member. |
| **NVM** | Non-Volatile Memory. A type of memory where the contents survive power loss. Also sometimes referred to as NVMEM. |
| **OCP** | Operator control panel. The control or indicator panel associated with a device. The OCP is usually mounted on the device and is accessible to the operator. |
| **other controller** | The controller in a dual-redundant pair that is connected to the controller serving your current CLI session. *See also* this controller. |
| **outbound fiber** | One fiber in a link that carries information away from a port. |
| **parallel data transmission** | A data communication technique in which more than one code element (for example, bit) of each byte is sent or received simultaneously. |
| **parity** | A method of checking if binary numbers or characters are correct by counting the ONE bits. In odd parity, the total number of ONE bits must be odd; in even parity, the total number of ONE bits must be even. |
| **parity bit** | A binary digit added to a group of bits that checks to see if errors exist in the transmission. |

| | |
|---|---|
| **parity check** | A method of detecting errors when data is sent over a communications line. With even parity, the number of ones in a set of binary data should be even. With odd parity, the number of ones should be odd. |
| **participating mode** | A mode within an L_Port that allows the port to participate in loop activities. A port must have a valid AL_PA to be in participating mode. |
| **PCM** | Polycenter Console Manager. |
| **PCMCIA** | Personal Computer Memory Card Industry Association. An international association formed to promote a common standard for PC card-based peripherals to be plugged into notebook computers. The card commonly known as a PCMCIA card is about the size of a credit card. |
| **parity** | A method of checking if binary numbers or characters are correct by counting the ONE bits. In odd parity, the total number of ONE bits must be odd; in even parity, the total number of ONE bits must be even. Parity information can be used to correct corrupted data. RAIDsets use parity to improve the availability of data. |
| **parity bit** | A binary digit added to a group of bits that checks to see if there are errors in the transmission. |
| **parity RAID** | *See* RAIDset. |
| **partition** | A logical division of a container, represented to the host as a logical unit. |
| **peripheral device** | Any unit, distinct from the CPU and physical memory, that can provide the system with input or accept any output from it. Terminals, printers, tape drives, and disks are peripheral devices. |
| **point-to-point connection** | A network configuration in which a connection is established between two, and only two, terminal installations. The connection may include switching facilities. |
| **port** | (1) In general terms, a logical channel in a communications system. (2) The hardware and software used to connect a host controller to a communications bus, such as a SCSI bus or serial bus. |
| | Regarding the controller, the port is (1) the logical route for data in and out of a controller that can contain one or more channels, all of which contain the same type of data. (2) The hardware and software that connects a controller to a SCSI device. |
| **port_name** | A 64-bit unique identifier assigned to each Fibre Channel port. The Port_Name is communicated during the logon and port discovery process. |
| **preferred address** | The AL_PA which an NL_Port attempts to acquire first during initialization. |

| | |
|---|---|
| **primary cabinet** | The primary cabinet is the subsystem enclosure that contains the controllers, cache modules, external cache batteries, and the PVA module. |
| **private NL_Port** | An NL_Port which does not attempt login with the fabric and only communicates with NL_Ports on the same loop. |
| **public NL_Port** | An NL_Port that attempts login with the fabric and can observe the rules of either public or private loop behavior. A public NL_Port may communicate with both private and public NL_Ports. |
| **program card** | The PCMCIA card containing the controller's operating software. |
| **protocol** | The conventions or rules for the format and timing of messages sent and received. |
| **PTL** | Port-Target-LUN. The controller's method of locating a device on the controller's device bus. |
| **PVA module** | Power Verification and Addressing module. |
| **RAID** | Redundant Array of Independent Disks. Represents multiple levels of storage access developed to improve performance or availability or both. |
| **RAID level 0** | A RAID storageset that stripes data across an array of disk drives. A single logical disk spans multiple physical disks, allowing parallel data processing for increased I/O performance. While the performance characteristics of RAID level 0 is excellent, this RAID level is the only one that does not provide redundancy. Raid level 0 storagesets are sometimes referred to as stripesets. |
| **RAID level 0+1** | A RAID storageset that stripes data across an array of disks (RAID level 0) and mirrors the striped data (RAID level 1) to provide high I/O performance and high availability. This RAID level is alternatively called a striped mirrorset. Raid level 0+1 storagesets are sometimes referred to as striped mirrorsets. |
| **RAID level 1** | A RAID storageset of two or more physical disks that maintains a complete and independent copy of the entire virtual disk's data. This type of storageset has the advantage of being highly reliable and extremely tolerant of device failure. Raid level 1 storagesets are sometimes referred to as mirrorsets. |
| **RAID level 3** | A RAID storageset that transfers data parallel across the array's disk drives a byte at a time, causing individual blocks of data to be spread over several disks serving as one enormous virtual disk. A separate redundant check disk for the entire array stores parity on a dedicated disk drive within the storageset. *See also* RAID level 5. |
| **RAID level 5** | A RAID storageset that, unlike RAID level 3, stores the parity information across all of the disk drives within the storageset. *See also* RAID level 3. |

**RAID level 3/5**   A DIGITAL-developed RAID storageset that stripes data and parity across three or more members in a disk array. A RAIDset combines the best characteristics of RAID level 3 and RAID level 5. A RAIDset is the best choice for most applications with small to medium I/O requests, unless the application is write intensive. A RAIDset is sometimes called parity RAID. Raid level 3/5 storagesets are sometimes referred to as RAIDsets.

**RAIDset**   *See* RAID level 3/5.

**RAM**   Random access memory.

**read ahead caching**   A caching technique for improving performance of synchronous sequential reads by prefetching data from disk.

**read caching**   A cache management method used to decrease the subsystem's response time to a read request by allowing the controller to satisfy the request from the cache memory rather than from the disk drives.

**reconstruction**   The process of regenerating the contents of a failed member's data. The reconstruct process writes the data to a spareset disk and then incorporates the spareset disk into the mirrorset, striped mirrorset, or RAIDset from which the failed member came. *See also* regeneration.

**reduced**   Indicates that a mirrorset or RAIDset is missing one member because the member has failed or has been physically removed.

**redundancy**   The provision of multiple interchangeable components to perform a single function in order to cope with failures and errors. A RAIDset is considered to be redundant when user data is recorded directly to one member and all of the other members include associated parity information.

**regeneration**   (1) The process of calculating missing data from redundant data. (2) The process of recreating a portion of the data from a failing or failed drive using the data and parity information from the other members within the storageset. The regeneration of an entire RAIDset member is called reconstruction. *See also* reconstruction.

**request rate**   The rate at which requests are arriving at a servicing entity.

**RFI**   Radio frequency interference. The disturbance of a signal by an unwanted radio signal or frequency.

**replacement policy**   The policy specified by a switch with the SET FAILEDSET command indicating whether a failed disk from a mirrorset or RAIDset is to be automatically replaced with a disk from the spareset. The two switch choices are AUTOSPARE and NOAUTOSPARE.

**SBB**                      StorageWorks building block. (1) A modular carrier plus the interface required to mount the carrier into a standard StorageWorks shelf. (2) any device conforming to shelf mechanical and electrical standards installed in a 3.5-inch or 5.25-inch carrier, whether it is a storage device or power supply.

**SCSI**                     Small computer system interface. (1) An ANSI interface standard defining the physical and electrical parameters of a parallel I/O bus used to connect initiators to devices. (2) a processor-independent standard protocol for system-level interfacing between a computer and intelligent devices including hard drives, floppy disks, CD-ROMs, printers, scanners, and others.

**SCSI-A cable**             A 50-conductor (25 twisted-pair) cable generally used for single-ended, SCSI-bus connections.

**SCSI bus signal**          Sometimes referred to as an adapter. (1) A device used to interface between the
**converter**                subsystem and a peripheral device unable to be mounted directly into the SBB shelf of the subsystem. (2) a device used to connect a differential SCSI bus to a single-ended SCSI bus. (3) A device used to extend the length of a differential or single-ended SCSI bus. *See also* I/O module.

**SCSI device**              (1) A host computer adapter, a peripheral controller, or an intelligent peripheral that can be attached to the SCSI bus. (2) Any physical unit that can communicate on a SCSI bus.

**SCSI device ID**           A bit-significant representation of the SCSI address referring to one of the signal
**number**                   lines, numbered 0 through 7 for an 8-bit bus, or 0 through 15 for a 16-bit bus. *See also* target ID number.

**SCSI ID number**           The representation of the SCSI address that refers to one of the signal lines numbered 0 through 15.

**SCSI-P cable**             A 68-conductor (34 twisted-pair) cable generally used for differential bus connections.

**SCSI port**                (1) Software: The channel controlling communications to and from a specific SCSI bus in the system. (2) Hardware: The name of the logical socket at the back of the system unit to which a SCSI device is connected.

**serial transmission**      A method transmission in which each bit of information is sent sequentially on a single channel rather than simultaneously as in parallel transmission.

**service rate**             The rate at which an entity is able to service requests For example, the rate at which an Arbitrated Loop is able to service arbitrated requests.

**signal converter**         *See* SCSI bus signal converter.

| | |
|---|---|
| **single ended I/O module** | A 16-bit I/O module. *See also* I/O module. |
| **single-ended SCSI bus** | An electrical connection where one wire carries the signal and another wire or shield is connected to electrical ground. Each signal's logic level is determined by the voltage of a single wire in relation to ground. This is in contrast to a differential connection where the second wire carries an inverted signal. |
| **spareset** | A collection of disk drives made ready by the controller to replace failed members of a storageset. |
| **storage array** | An integrated set of storage devices. |
| **storage array subsystem** | *See* storage subsystem. |
| **storageset** | (1) A group of devices configured with RAID techniques to operate as a single container. (2) Any collection of containers, such as stripesets, mirrorsets, striped mirrorsets, and RAIDsets. |
| **storage subsystem** | The controllers, storage devices, shelves, cables, and power supplies used to form a mass storage subsystem. |
| **storage unit** | The general term that refers to storagesets, single-disk units, and all other storage devices that are installed in your subsystem and accessed by the host. A storage unit can be any entity that is capable of storing data, whether it is a physical device or a group of physical devices. |
| **StorageWorks** | A family of DIGITAL modular data storage products that allow customers to design and configure their own storage subsystems. Components include power, packaging, cabling, devices, controllers, and software. Customers can integrate devices and array controllers in StorageWorks enclosures to form storage subsystems. |
| | StorageWorks systems include integrated SBBs and array controllers to form storage subsystems. System-level enclosures to house the shelves and standard mounting devices for SBBs are also included. |
| **stripe** | The data divided into blocks and written across two or more member disks in an array. |
| **striped mirrorset** | *See* RAID level 0+1. |
| **stripeset** | *See* RAID level 0. |

**stripe size**         The stripe capacity as determined by $n$–1 times the chunksize, where $n$ is the number of RAIDset members.

**striping**            The technique used to divide data into segments, also called chunks. The segments are striped, or distributed, across members of the stripeset. This technique helps to distribute hot spots across the array of physical devices to prevent hot spots and hot disks.

                        Each stripeset member receives an equal share of the I/O request load, improving performance.

**surviving controller** The controller in a dual-redundant configuration pair that serves its companion's devices when the companion controller fails.

**switch**              A method that controls the flow of functions and operations in software.

**synchronous**         Pertaining to a method of data transmission which allows each event to operate in relation to a timing signal. *See also* asynchronous.

**tape**                A storage device supporting sequential access to variable sized data records.

**target**              (1) A SCSI device that performs an operation requested by an initiator. (2) Designates the target identification (ID) number of the device.

**this controller**     The controller that is serving your current CLI session through a local or remote terminal. *See also* other controller.

**topology**            An interconnection scheme that allows multiple Fibre Channel ports to communicate with each other. For example, point-to-point, Arbitrated Loop, and switched fabric are all Fibre Channel topologies.

**transfer data rate**  The speed at which data may be exchanged with the central processor, expressed in thousands of bytes per second.

**ULP**                 Upper Layer Protocol.

**ULP process**         A function executing within a Fibre Channel node which conforms to the Upper Layer Protocol (ULP) requirements when interacting with other ULP processes.

**Ultra SCSI**          A Fast-20 SCSI bus. *See also* Wide Ultra SCSI.

**unit**                A container made accessible to a host. A unit may be created from a single disk drive or tape drive. A unit may also be created from a more complex container such as a RAIDset. The controller supports a maximum of eight units on each target. *See also* target and target ID number.

| | |
|---|---|
| **unwritten cached data** | Sometimes called unflushed data. *See* dirty data. |
| **UPS** | Uninterruptible power supply. A battery-powered power supply guaranteed to provide power to an electrical device in the event of an unexpected interruption to the primary power supply. Uninterruptible power supplies are usually rated by the amount of voltage supplied and the length of time the voltage is supplied. |
| **VHDCI** | Very high-density-cable interface. A 68-pin interface. Required for Ultra-SCSI connections. |
| **virtual terminal** | A software path from an operator terminal on the host to the controller's CLI interface, sometimes called a host console. The path can be established via the host port on the controller (using HSZterm) or via the maintenance port through an intermediary host. |
| **VTDPY** | An abbreviation for Virtual Terminal Display Utility. |
| **warm swap** | A device replacement method that allows the complete system remains online during device removal or insertion. The system bus may be halted, or quiesced, for a brief period of time during the warm-swap procedure. |
| **Wide Ultra SCSI** | Fast/20 on a Wide SCSI bus. |
| **Worldwide name** | A unique 64-bit number assigned to a subsystem by the Institute of Electrical and Electronics Engineers (IEEE) and set by DIGITAL manufacturing prior to shipping. This name is referred to as the node ID within the CLI. |
| **write-back caching** | A cache management method used to decrease the subsystem's response time to write requests by allowing the controller to declare the write operation "complete" as soon as the data reaches its cache memory. The controller performs the slower operation of writing the data to the disk drives at a later time. |
| **write-through caching** | A cache management method used to decrease the subsystem's response time to a read. This method allows the controller to satisfy the request from the cache memory rather than from the disk drives. |
| **write hole** | The period of time in a RAID level 1 or RAID level 5 write operation when an opportunity emerges for undetectable RAIDset data corruption. Write holes occur under conditions such as power outages, where the writing of multiple members can be abruptly interrupted. A battery backed-up cache design eliminates the write hole because data is preserved in cache and unsuccessful write operations can be retried. |

**write-through cache**
A cache management technique for retaining host write requests in read cache. When the host requests a write operation, the controller writes data directly to the storage device. This technique allows the controller to complete some read requests from the cache, greatly improving the response time to retrieve data. The operation is complete only after the data to be written is received by the target storage device.

This cache management method may update, invalidate, or delete data from the cache memory accordingly, to ensure that the cache contains the most current data.

# Index