COMPAQ
STORAGEWORKS

HSG80 Array Controller ACS Version 8.4

Configuration and CLI Reference Guide

First Edition (April, 1999)
Part Number EK-HSG84-RG. A01/118619-001
Compaq Computer Corporation

While Compaq Computer Corporation believes the information included in this publication is correct as of the date of publication, it is subject to change without notice.

Compaq Computer Corporation makes no representations that the interconnection of its products in the manner described in this document will not infringe existing or future patent rights, nor do the descriptions contained in this document imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

Contents

About This Guide

Chapter 1
General Description

Chapter 2
Controller and Host Concepts

Chapter 3
Controller and Host Addressing

Chapter 4
Planning Storagesets

Chapter 5
Configuring an HSG80 Array Controller

Chapter 6
Configuring Storagesets

Chapter 7
CLI Commands

System Profiles

Controller Specifications

Glossary

Index

Figures

Tables

About This Guide

This book describes the features of the HSG80 array controller and configuration procedures for the controller and storagesets running Array Controller Software (ACS) Version 8.4F.

Getting Help

If you have a problem and have exhausted the information in this guide, you can get further information and other help in the following locations.

Compaq Website

The Compaq Website has information on this product as well as the latest drivers and Flash ROM images. You can access the Compaq website by logging on to the Internet at http://www.compaq.com.

Telephone Numbers

For Compaq support:

In the United States and Canada, call 1-800-354-9000.

For Compaq technical support phone numbers outside the United States and Canada, visit the Compaq Website at:

http://www.compaq.com.

Precautions

Follow these precautions when carrying out the procedures in this book.

Electrostatic Discharge Precautions

Static electricity collects on all nonconducting material, such as paper, cloth, and plastic. An electrostatic discharge (ESD) can easily damage a controller or other subsystem component even though you may not see or feel the discharge. Follow these precautions whenever you're servicing a subsystem or one of its components:

- Always use an ESD wrist strap when servicing the controller or other components in the subsystem. Make sure that the strap contacts bare skin, fits snugly, and that its grounding lead is attached to a bus that is a verified earth ground.
- Before touching any circuit board or component, always touch a verifiable earth ground to discharge any static electricity that may be present in your clothing.
- Always keep circuit boards and components away from nonconducting material.
- Always keep clothing away from circuit boards and components.
- Always use antistatic bags and grounding mats for storing circuit boards or components during replacement procedures.
- Always keep the ESD cover over the program card when the card is in the controller. If you remove the card, put it in its original carrying case. Never touch the contacts or twist or bend the card while you're handling it.
- Never touch the connector pins of a cable when it is attached to a component or host.

Component Precaution

System components referenced in this manual comply to regulatory standards documented herein. Use of other components in their place may violate country standards, negate regulatory compliance, or invalidate the warranty on your product.

Maintenance Port Precautions

The maintenance port generates, uses, and radiates radio-frequency energy through cables that are connected to it. This energy may interfere with radio and television reception. Do not leave a cable connected to this port when you’re not communicating with the controller.

FCC Precautions

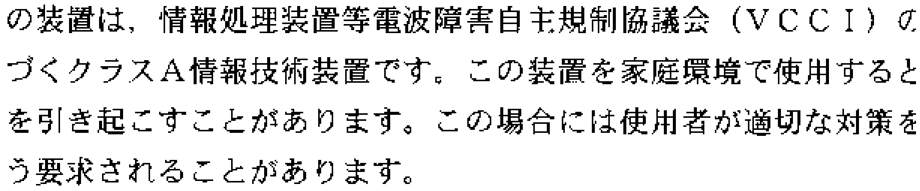
This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the manuals, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense. Restrictions apply to the use of the local-connection port on this series of controllers; failure to observe these restrictions may result in harmful interference. Always disconnect this port as soon as possible after completing the setup operation. Any changes or modifications made to this equipment may void the user's authority to operate the equipment.

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

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

Attention!
Ceci est un produit de Classe A. Dans un environnement domestique, ce produit risque de créer des interférences radioélectriques, il appartiendra alors à l'utilisateur de prendre les mesures spécifiques appropriées.

JAPAN



の装置は，情報処理装置等電波障害自主規制協議会（ＶＣＣＩ）の
づくクラスＡ情報技術装置です。この装置を家庭環境で使用すると
を引き起こすことがあります。この場合には使用者が適切な対策を
う要求されることがあります。

USA

This equipment generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules, which are designed to provide reasonable protection against such radio frequency interference. Operation of this equipment in a residential area may cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference. Any modifications to this device - unless expressly approved by the manufacturer - can void the user's authority to operate this equipment under part 15 of the FCC rules.

Conventions

This book uses the following typographical conventions and special notices to help you find what you’re looking for. See “Typographical Conventions,” page xix for more details.

Typographical Conventions

Convention	Meaning
ALLCAPS	**Command syntax that must be entered exactly as shown and for commands discussed within text, for example:** SET FAILOVER COPY=OTHER_CONTROLLER "Use the SHOW SPARESET command to show the contents of the spareset."
Monospaced	**Screen display.**
Sans serif italic	**Command variable or numeric value that you supply, for example: SHOW** *RAIDset-name* **or** SET THIS_CONTROLLER ID=*(n,n,n,n,)*
italic	**Reference to other books or publications, for example:** "See the *HSG80 Array Controller ACS V8.3 Release Notes* for details."
. . .	**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.**

Special Notices

This book doesn’t 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’re carrying out the procedures in this book:

⚠ **WARNING:** A warning indicates the presence of a hazard that can cause personal injury if you do not observe the precautions in the text.

△ **CAUTION:** A caution indicates the presence of a hazard that might damage hardware, corrupt software, or cause a loss of data.

IMPORTANT: A tip provides alternative methods or procedures that may not be immediately obvious. A tip may also alert customers that the controller’s behavior being discussed is different from prior software or hardware versions.

NOTE: A note provides additional information that’s related to the completion of an instruction or procedure.

Related Publications

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

Document Title	Part Number
Fibre Channel Storage Switch Service Guide	AA-RHBZA-TE / 135268-001
Fibre Channel Storage Switch User’s Guide	AA-RHBYA-TE / 135267-001
Fibre Channel Switch Quick Setup Guide	AA-RHC0A-TE / 135269-001
Compaq StorageWorks Data Replication Manager HSG80 ACS Version 8.4P Operations Guide	EK-HSG84-DT / 128519-001
Fibre Channel Arbitrated Loop Hub (DS-DHGGA-CA) User’s Guide	EK-DHGGA-UG
KGPSA PCI-to-Fibre Channel Host Adapter	EK–KGPSA–UG
HSG80 Illustrated Parts Map	EK-HSG80-MP / 355217-001
Compaq StorageWorks Ultra SCSI RAID Enclosure (BA370-Series) User’s Guide	EK–BA370–UG / 387403-001
The RAIDBOOK—A Source for RAID Technology	RAID Advisory Board
Compaq StorageWorks HSG80 Array Controller ACS Version 8.3 and 8.4 Maintenance and Service Guide	EK-HSG84-SV / 118629-001
Command Console Version 2.1 (HSG80) User’s Guide	AA–RFA2C-TE / 387405-003
Compaq StorageWorks RA8000/ESA12000 Storage Subsystem User’s Guide	EK-SMCPR-UG / 387404-001
Compaq StorageWorks RA8000/ESA12000 HSG80 V8.x Installation Reference Manual	
for OpenVMS	AA-RH4BA-TE / 387401
for DIGITAL UNIX	AA-RFAUA-TE / 387389-001
for Windows NT Server - Intel	AA-RFA9B-TE / 387387-002

Document Title	Part Number
Compaq StorageWorks RA8000/ESA12000 HSG80 V8.x	
Quick Setup Guide	
for OpenVMS	AA-RH49A-TE / 387400-001
for DIGITAL UNIX	AA-RFASA-TE / 387388-001
for Windows NT Server - Intel	AA-RFA7B-TE / 387386-002

Revision History

This is a new document.

Chapter 1
General Description

This chapter illustrates and describes in general terms your subsystem and its major components: the HSG80 array controller, its cache module, and its external cache battery. See the Fibre Channel Switch Documentation that came with the switch kit for specifics about how the switch operates. See the *KGPSA PCI-to-Fibre Channel Host Adapter User Guide* for information about the adapter that connects the subsystem to your host.

The HSG80 Array Controller Subsystem

Take a few moments to familiarize yourself with the major components of the HSG80 array controller subsystem. Figure 1–1 shows the components of a typical installation that includes:

- One BA370 rack-mountable enclosure.
- Two controllers, each supported by a cache module.
- Two cache modules, which support nonvolatile memory and dynamic cache policies to protect the availability of its unwritten (write-back) data.
- Two external cache batteries (ECBs) in one Storage Building Block (SBB), which provide backup power to the cache modules during a primary power failure.
- One environmental monitoring unit (EMU) that monitors the subsystem’s environment, and alerts the controller of equipment failures that could cause an abnormal environment.
- One power verification and addressing (PVA) module that provides a unique address to each enclosure in an extended subsystem.
- Six I/O modules that integrate the SSB shelf with either an 8-bit single-ended, 16-bit single-ended, or 16-bit differential SCSI bus.

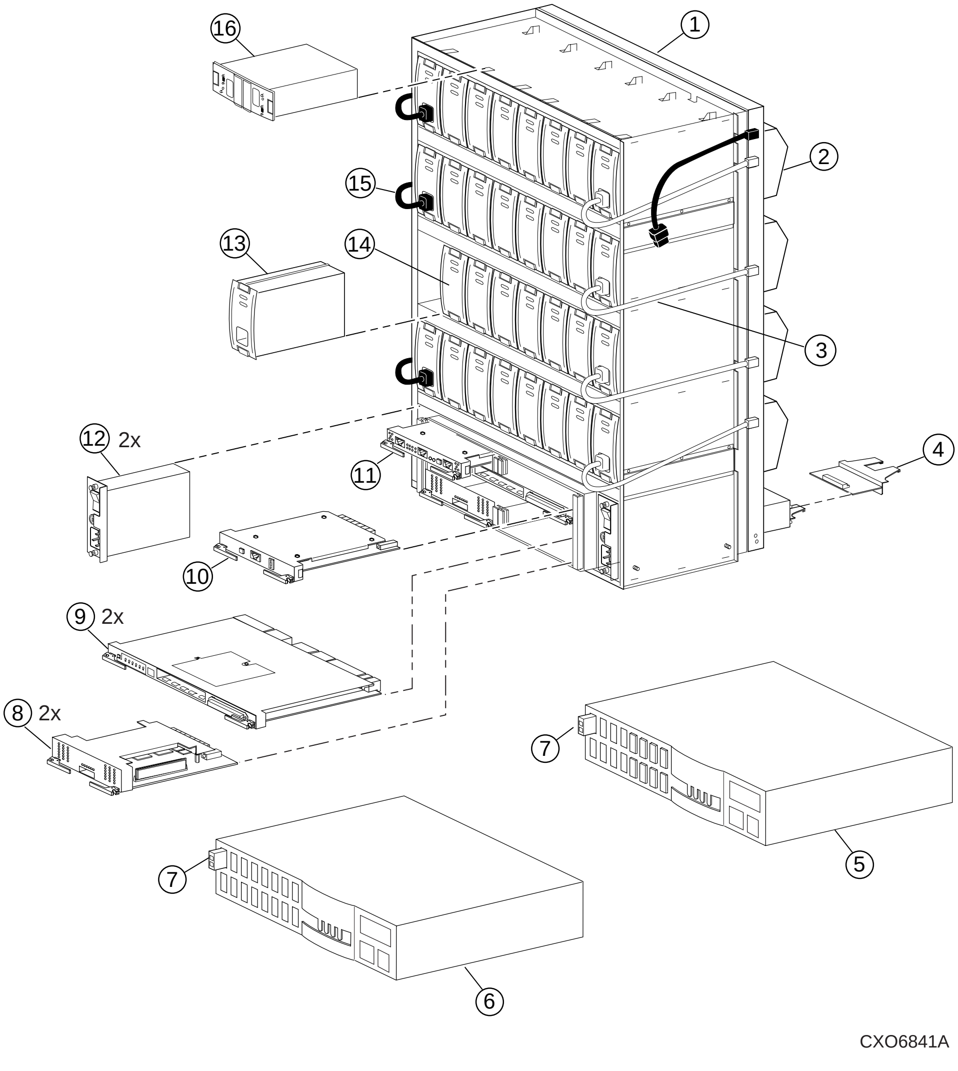


Figure 1–1. The HSG80 Subsystem

Table 1–1. Key to Figure 1–1 The HSG80 Subsystem

Item	Description
1	BA370 rack-mountable enclosure
2	Cooling fan
3	Power cable kit, white
4	I/O module (6)
5	Switch, 8-port
6	Switch, 16-port
7	GBIC
8	Cache module
9	HSG80 controller
10	PVA module
11	EMU
12	AC input module (maximum of 2)
13	180-watt power supply (maximum of 8)
14	Disk drives
15	Power cable, black
16	ECB, dual (shown) ECB, single (not shown)

Summary of HSG80 Features

Table 1–2 summarizes the features of the controller.

Table 1–2. Controller Features

Feature	Supported
Controller Failover	■ Transparent Failover ■ Multiple-Bus Failover
Topology	■ Fibre Channel Switched Fabric (FC–SW) ■ 2 controller subsystems; maximum 4 controllers (2 dual-redundant configurations)
Number of controllers in a Fabric	■ 4
Number of mixed hosts (heterogeneous host support)	■ 2
Number of same hosts	■ 4
Number of host ports (adapters)	■ 2 on each host
Number of host paths for same hosts	■ 8
Number of host paths for mixed hosts	■ 4
Supported Operating Systems	■ Transparent Failover: ❑ WINNT/Intel ❑ WINNT/Alpha ❑ DIGITAL UNIX ■ Multiple-Bus Failover: ❑ WINNT/Intel ❑ WINNT/Alpha ❑ Alpha OpenVMS
Host protocol	■ FC–SW

Table 1–2. Controller Features (Continued)

Feature	Supported
Host bus interconnect	■ Optical: MultiMode 50 Micron (Do not mix media types) ■ Gigabit Link Module (GLM)
Device protocol	■ SCSI–2 ■ SCSI–3 (supported on OpenVMS)
Device bus interconnect	■ Wide Ultra Single-ended
Number of SCSI device I/O ports	■ 6
Number of SCSI device targets per port	■ 12
Maximum number of SCSI devices	■ 72
Disk Drives	■ 4 and 9 GB Ultra SCSI and Wide Ultra SCSI ■ 18 GB Ultra SCSI ■ 9 and 18 GB 10K Ultra SCSI
RAID levels supported	■ 0 ■ 1 ■ 0+1 ■ 3/5
Cache Capacity	■ 64 MB and 128 MB (32 MB DIMMs only) ■ 256 MB and 512 MB (128 MB DIMMs only)
Caching Features	■ Mirrored Cache ■ Sequential Read Ahead
Maximum number of RAID-5 and RAID-1 storagesets	■ 30
Maximum number of RAID-5 storagesets	■ 20

Table 1–2. Controller Features (Continued)

Feature	Supported
Maximum number of RAID-5, RAID-1, and RAID-0 storagesets	■ 45
Maximum number of partitions per storageset or individual disk	■ 8
Maximum number of units presented to each host	■ WINNT—16 (8 on each of 2 host adapters). This is a driver limitation. ■ OpenVMS—127 and one Command Console LUN.
Maximum number of devices per unit	■ 48
Serial interconnect speed	■ 1 Gigabit/second
Maximum device, storageset, or unit size	■ 512 GB LUN capacity
Configuration Save	■ Transfer configuration from HSZ70 subsystem to HSG80 controller ■ Transfer configuration from ACS V8.0 to ACS V 8.4
General Features	■ Host modes and access privileges ■ Persistent reserves ■ Graceful powerdown policy ■ Program card updates ■ Device warm swap ■ Utilities to test disks

Controller Components

Figure 1–2 details the HSG80 Array Controller and its Fibre Channel optical components.

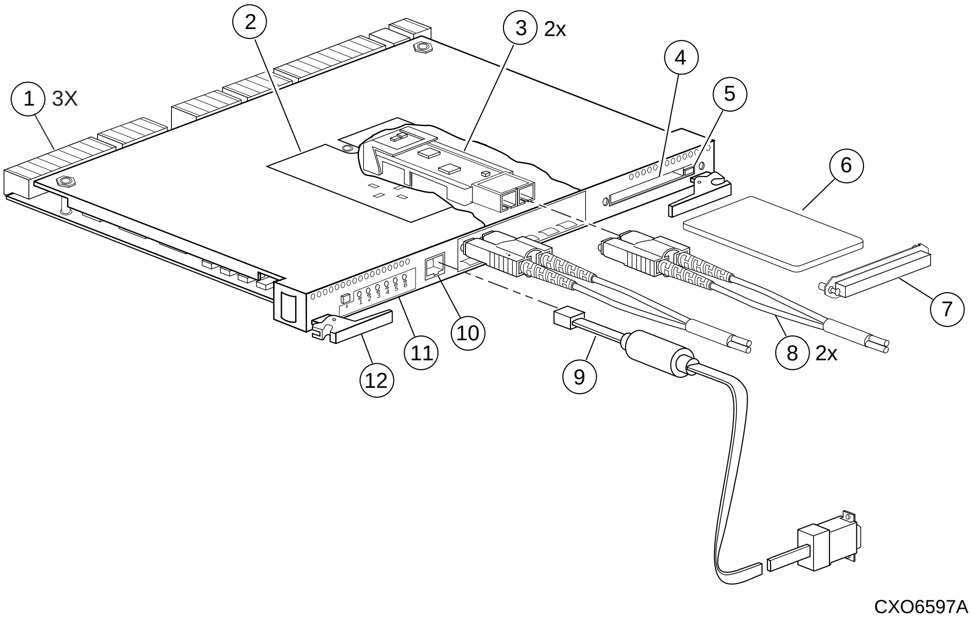


Figure 1–2. HSG80 Array Controller Fibre Channel Optical Cabling

Table 1–3. Key to Figure 1–2 HSG80 Array Controller Fibre Channel Optical Cabling

Item	Description
1	**Backplane connectors**
2	**GLM access door**
3	**GLM**

Table 1–3. Key to Figure 1–2 HSG80 Array Controller Fibre Channel Optical Cabling (Continued)

Item	Description
4	Program card slot
5	Program card ejection button
6	PCMCIA Program card
7	ESD/PCMCIA program card cover
8	Fibre Channel optical cable
9	Maintenance port cable
10	Maintenance port
11	Operator control panel (OCP)
12	Lever for removing, installing, and retaining controller module (2)

NOTE: To view an optional maintenance port cable and its components for a terminal connection, see the *HSG80 Array Controller ACS Version 8.3 and 8.4 Maintenance and Service Guide*.

△ **CAUTION:** If the Fibre Channel optical cable is not properly connected to the controller, controller failure may result. In addition, if the cable is not regularly maintained, its performance and lifespan will be affected. Before proceeding, it is important to administer the precautionary measures detailed below and in the *HSG80 Array Controller ACS Version 8.3 and 8.4 Maintenance and Service Guide*.

Fibre Channel Optical Cable Precaution

Prior to connecting the Fibre Channel cable to the controller, look for the white stripe on each side of the coupling. After the cable is seated into the controller, be sure that the white stripes are hidden. Also, when connecting the Fibre Channel cable to the controller, listen for a distinctive “snap” sound. This will indicate that the cable is properly inserted into the controller.

Controller Components

The HSG80 Array Controller components that you will use most often, such as the maintenance port and the OCP, are conveniently located on the controller's front panel. The host port and program-card slot are also located on the front panel, making it easy to update the controller's software or to connect the controller to a different host.

Each controller is supported by its own cache module. Figure 1–3 shows which cache module supports which controller in a dual-redundant configuration in a BA370 rack-mountable enclosure.

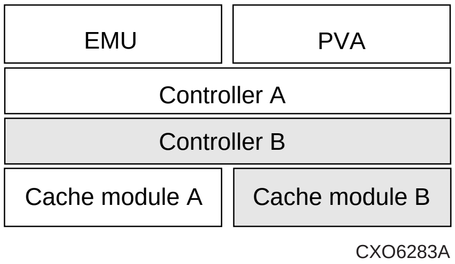


Figure 1–3. Location of Controllers and Cache Modules

IMPORTANT: Compaq recommends that you use the slots for controller A and cache module A for single configurations. Slot A responds to SCSI target ID number 7 on device buses; slot B responds to SCSI target ID number 6 on the device buses.

Operator Control Panel

The operator control panel (OCP) contains a reset button and six port LED buttons, as shown in Figure 1–4. The reset button flashes about once per second to indicate that the controller is operating normally. The port button LEDs correspond to the controller's device ports and remain off during normal operation. If an error occurs, the reset button and device fault LEDs will illuminate in a solid or flashing pattern to help you diagnose the problem. See the *HSG80 Array Controller ACS Version 8.3 and 8.4 Maintenance and Service Guide* for further explanation on these codes.

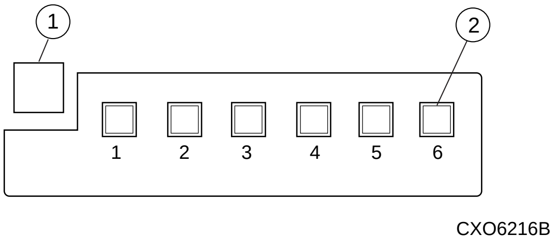


Figure 1–4. HSG80 Controller Operator Control Panel (OCP)

Table 1–4. Key to Figure 1–4 HSG80 Controller Operator Control Panel (OCP)

Item	Description
①	**Reset button/LED**
②	**Port button/LED**

To identify the exact location of the OCP, refer to Figure 1–2.

Under normal circumstances, you will not need to remove the controller from its enclosure. For this reason, the components that you will use most often are conveniently located on the front panel. For example, the maintenance port provides a convenient way to connect a PC or terminal to your controller so that you can interact with it.

After you configure your controller, you should periodically check its control panel. If an error occurs, one or more of the device fault LEDs on the control panel will flash in a pattern that will help you to diagnose the problem. See the *HSG80 Array Controller ACS Version 8.3 and 8.4 Maintenance and Service Guide* for details about troubleshooting your controller.

Maintenance Port

You can access the controller through a PC or a local terminal via the maintenance port, or through a remote terminal—sometimes called a virtual terminal or host console—via the host. Compaq recommends that you use a PC or a local terminal to carry out the troubleshooting and servicing procedures in this manual. See “Establishing a Local Connection to the Controller,” page 5–33, for more information on connecting the controller with a maintenance port cable.

Utilities and Exercisers

The controller’s software includes the utilities and exercisers outlined below to assist in troubleshooting and maintaining the controller and the other modules that support its operation. For more information on these utilities and exercisers, see the *HSG80 Array Controller ACS Version 8.3 and 8.4 Maintenance and Service Guide*.

Fault Management Utility

The Fault Management Utility (FMU) provides a limited interface to the controller’s fault-management system. As a troubleshooting tool, you can use FMU to display last-failure and memory-system-failure entries, translate many of the code values contained in event messages, and set the display characteristics of significant events and failures.

Virtual Terminal Display

Use the virtual terminal display (VTDPY) utility to troubleshoot communication between the controller and its host, communication between the controller and the devices in the subsystem, and the state and I/O activity of the logical units, devices, and device ports in the subsystem.

Disk Inline Exerciser

Use the disk inline exerciser (DILX) to investigate the data-transfer capabilities of disk drives. DILX tests and verifies operation of the controller and the SCSI–2 disk drives attached to it. DILX generates intense read and write loads to the disk drive while monitoring the drive's performance and status.

IMPORTANT: DILX cannot be run on remote copy sets.

Configuration Utility

Use the configuration (CONFIG) utility to add one or more storage devices to the subsystem. This utility checks the device ports for new disk drives then adds them to the controller's configuration and automatically names them.

HSUTIL

Use HSUTIL to upgrade the firmware on disk drives in the subsystem and to format disk drives. Some details about formatting are located in "Formatting Disk Drives," page 6–3.

Code Load and Code Patch Utility

Use Code Load/Code Patch (CLCP) utility to upgrade the controller software and the EMU software. You can also use it to patch the controller software. When you install a new controller, you must have the correct software version and patch number.

NOTE: Only Compaq field service personnel are authorized to upload EMU microcode updates. Contact the Customer Service Center (CSC) for directions in obtaining the appropriate EMU microcode and installation guide.

Clone Utility

Use the Clone utility to duplicate the data on any unpartitioned single-disk unit, stripeset, or mirrorset. Back up the cloned data while the actual storageset remains online.

Field Replacement Utility

Use the field replacement utility (FRUTIL) to replace a failed controller (in a dual-redundant configuration) without shutting down the subsystem.You can also use this menu-driven utility to replace cache modules and external cache batteries.

IMPORTANT: FRUTIL cannot be run in remote copy set environments on the target side specifically when I/O is in progress.

Change Volume Serial Number Utility

Only Compaq authorized service personnel may use this utility.

The Change Volume Serial Number (CHVSN) utility generates a new volume serial number (called *VSN*) for the specified device and writes it on the media. It is a way to eliminate duplicate volume serial numbers and to rename duplicates with different volume serial numbers.

Device Statistics Utility

The Device Statistics (DSTAT) utility allows you to log I/O activity on a controller over an extended period of time. Later, you can analyze that log to determine where the bottlenecks are and how to tune the controller for optimum performance.

Cache Module

Each controller requires a companion cache module as shown in Figure 1–5. Figure 1–3 on page 1–10, shows the location of a controller's companion cache module. The cache module, which can contain up to 512 MB of memory, increases the subsystem's I/O performance by providing read, read-ahead, write-through, and write-back caching.

The size of the memory contained in the cache module depends on the configuration of the DIMMs, with the supported combinations shown in Table 1–5. For placement of the DIMMs, see the *HSG80 Array Controller ACS Version 8.3 and 8.4 Maintenance and Service Guide*.

Table 1–5. Cache Module Memory Configurations

DIMMs	Quantity	Memory
32 MB	2	64 MB
32 MB	4	128 MB
128 MB	2	256 MB
128 MB	4	512 MB

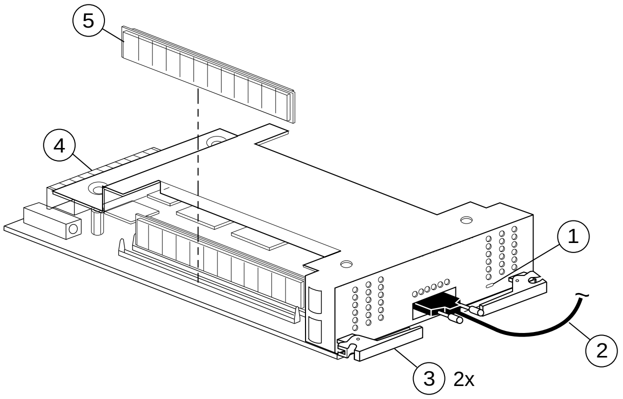


Figure 1–5. Cache Module

Table 1–6. Key to Figure 1–5 Cache Module

Item	Description
①	Cache-memory power LED button
②	ECB Y cable for the BA370 Enclosure ECB Y cable for the Data Center Cabinet
③	Retaining lever
④	Backplane connector
⑤	64 MB cache upgrade 256 MB cache upgrade

External Cache Battery

To preserve the write-back cache data in the event of a primary power failure, a cache module must be connected to an external cache battery (ECB) or a UPS.

Compaq supplies two versions of ECBs: a single-battery ECB for single controller configurations, and a dual-battery ECB for dual-redundant controller configurations, which is shown in Figure 1–6.

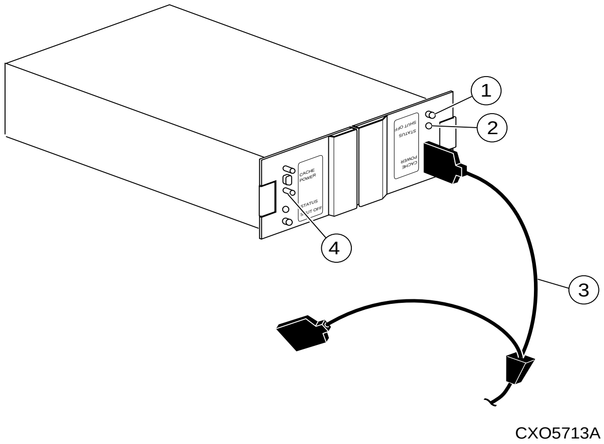


Figure 1–6. Dual-battery ECB SBB

Table 1–7. Key to Figure 1–6 Dual-Battery ECB SBB

Item	Description
①	**Shut off button**
②	**Status LED**
③	**ECB Y cable for the BA370 Enclosure** **ECB Y cable for the Data Center Cabinet**
④	**Micro-D port for second battery**

When the batteries are fully charged, an ECB can preserve 512 MB of cache memory for 24 hours. However, the battery capacity depends upon the size of memory contained in the cache module, as defined in the Table 1–8.

Table 1–8. ECB Capacity Based on Memory Size

Size	DIMM Combinations	Capacity
64 MB	Two, 32 MB	96 hours
128 MB	Four, 32 MB	48 hours
256 MB	Two, 128 MB	48 hours
512 MB	Four, 128 MB	24 hours

Charging Diagnostics

Whenever you restart the controller, its diagnostic routines automatically check the charge in the ECB's batteries. If the batteries are fully charged, the controller reports them as good and rechecks them every 24 hours. If the batteries are charging, the controller rechecks them every four minutes. Batteries are reported as being either above or below 50 percent in capacity. Batteries below 50 percent in capacity are referred to as being low.

This four-minute polling continues for up to 10 hours—the maximum time it should take to recharge the batteries. If the batteries have not been charged sufficiently after 10 hours, the controller declares them to be failed.

Battery Hysteresis

When charging a battery, write-back caching will be allowed as long as a previous down time has not drained more than 50 percent of a battery’s capacity. When a battery is operating below 50 percent capacity, the battery is considered to be low, and write-back caching is disabled.

⚠ **CAUTION:** Compaq recommends that you replace the ECB every two years to prevent battery failure.

NOTE: If a UPS is used for backup power, the controller does not check the battery. See “SET controller,” page 7–91 for information about the CACHE_UPS and NOCACHE_UPS switches.

Chapter 2
Controller and Host Concepts

This chapter provides concepts for understanding how the controller works in general, the difference between failover modes and how failover affects configurations, the basics of caching techniques, and an understanding of cache policies.

Terminology

When configuring the subsystem you will encounter the following terms and concepts that you must understand:

- Controller A and controller B
- This controller and other controller
- Unresponsive and surviving controller

Controller A and Controller B

Controllers and cache modules are designated either A or B depending on their location within the storage enclosure as shown in.

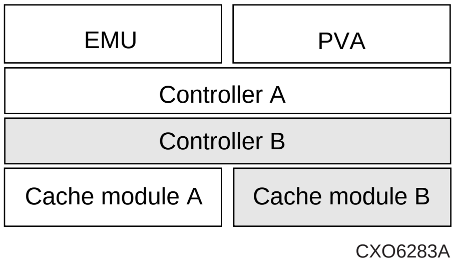


Figure 2–1. Controller A and Controller B Locations

This Controller and Other Controller

Some CLI commands use the terms “this” and “other” to identify one controller or the other in a dual-redundant pair. These designations are defined as follows:

- This controller—the controller that you are currently connected, regardless of whether it is controller A or controller B. It is the controller through which the CLI commands are being entered. The maintenance terminal is connected to the maintenance port of “this” controller.

- Other controller—the controller that you are *not* currently connected to or communicating through a terminal program, regardless of whether it is controller A or controller B. The maintenance terminal is not connected to the maintenance port of the “other” controller.

Figure 2–2 shows an example of when controller B can be called “this” controller when the maintenance port cable is connected to it. To make controller A into “this” controller, simply move the maintenance port cable to it.

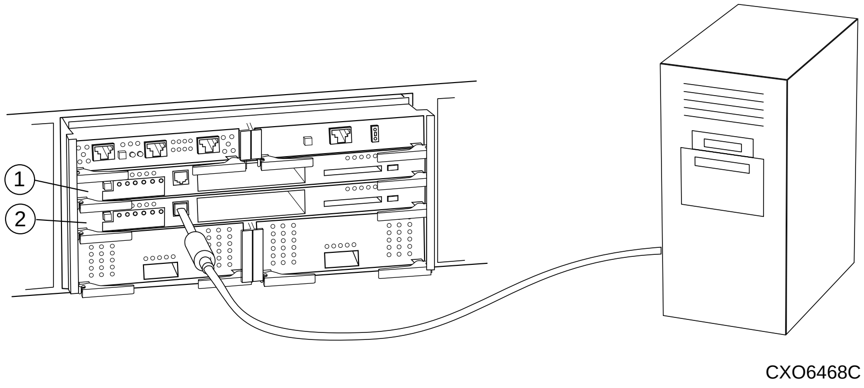


Figure 2–2. This Controller and Other Controller

Table 2–1. Key to Figure 2–2 This Controller and Other Controller

Item	Description
①	**Other controller**
②	**This controller**

Unresponsive and Surviving Controllers

When a failover occurs in a dual-redundant pair of controllers, the controller that ceases I/O processing is sometimes referred to as the *unresponsive* controller. The controller that is failed over to is frequently called the *surviving* controller. The surviving controller then has access to the unresponsive controller's logical units, so that it can continue to present them to the host, in addition to presenting host access to its own logical units.

After failover has occurred, the host sends all commands through the surviving controller. Once the unresponsive controller is fixed or replaced, the surviving controller *fails back* control to the previously unresponsive controller.

See "Selecting a Failover Mode," page 2–5 on the differences between failover modes and how they work.

See "SET FAILOVER," page 7–108 or "SET MULTIBUS_FAILOVER," page 7–115 on how to reset the controller pair once the unresponsive controller has been fixed or replaced.

See the *HSG80 Array Controller ACS Version 8.3 and 8.4 Maintenance and Service Guide* on how to replace a controller.

Selecting a Failover Mode

When you select a failover mode, your selection determines the way in which your controller will be configured in the system. This configuration will take into account how logical units are associated with the controller host ports and how they are accessed by the host. The logical units are said to *fail over to* the surviving controller, essentially granting it access to the logical units.

Failover is the process where a controller loses communication—called the unresponsive controller—and the surviving controller rapidly takes control of the system. In transparent failover mode, failover occurs when the controller fails. In multiple-bus failover mode, failover occurs when the host driver detects it cannot communicate over a particular path to its LUNs.

Use transparent failover if you want the failover to occur without any intervention from the host. Use multiple bus failover if the drivers in the host have the capacity to view multiple instances of a unit. Present one instance to the host and present a different instance when a path fails. See the documentation for your host operating system for more details about the drivers.

Once the controller has been fixed or is replaced, then the surviving controller can transfer back (or *fail back)* the logical units to it.

Using Transparent Failover Mode

Transparent failover occurs when a controller fails, loss of communication occurs, or when you press the reset button on one of the controllers.

In transparent failover, two of the four host ports are active and two are standby. This way, only two host ports at a time allow host access to the logical units. If a controller failure occurs, then the available standby port automatically becomes active, which then maintains host access to the logical units. See “Shared Port IDs,” page 2–6 for more information about the host ports and logical units.

To configure controllers for transparent failover, follow the steps in “Configuring in Transparent Failover Mode,” page 5–12.

Shared Port IDs

Figure 2–3 shows one example of a Fibre Channel Switched Fabric (FC-SW) configuration in transparent failover mode. In this configuration, the active port attaches to a FC-SW node. The standby port on each controller will assume the worldwide name if the active port fails, so that the standby port will be able to give host access to the units.

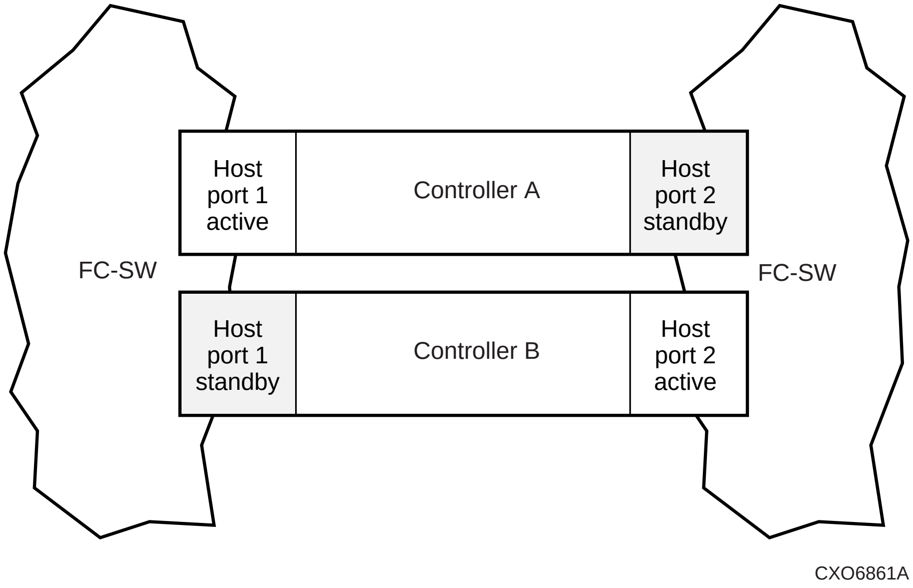


Figure 2–3. Example of Host Port Shared IDs in Transparent Failover

The port IDs are shared between ports so that when a controller loses communication and becomes unresponsive, the surviving controller seamlessly assumes the unresponsive controller’s function. LUNs are distributed between ports as follows:

- LUNs 0–99, port 1
- LUNs 100–199, port 2

As Figure 2–3 shows, only port 1 on controller A normally is active, which means it presents LUNs 0–99 to the host. Similarly, port 2 on controller B normally is active, which means it also presents LUNs, 100–199 to the host. If one controller fails, the surviving controller automatically changes the standby port to an active port presents the LUNs to the host.

Using Multiple-Bus Failover Mode

Multiple-bus (or host-assisted) failover is a dual-redundant controller configuration in which each host port is active, having its own connection or path to the host. Thus, if one of the host paths to a controller fails, the host can cause the units that became inaccessible to failover to a remaining viable path. All units, 0–199, are presented through all host ports, meaning that both controllers can present the same storage units to the host if the other controller fails or loss of communication occurs.

To configure controllers for multiple-bus failover, follow the steps in “Configuring in Multiple-Bus Failover Mode,” page 5–20.

Separate Port IDs

Figure 2–4 shows one example of a basic Fibre Channel Switched Fabric (FC-SW) configuration in multiple-bus failover mode. In this example, all ports are active and all ports have equal access to all LUNs, 0–199.

IMPORTANT: A unit is accessible by all four host ports. Because a unit is assigned to one controller, the unit is considered *online* and available to that controller.

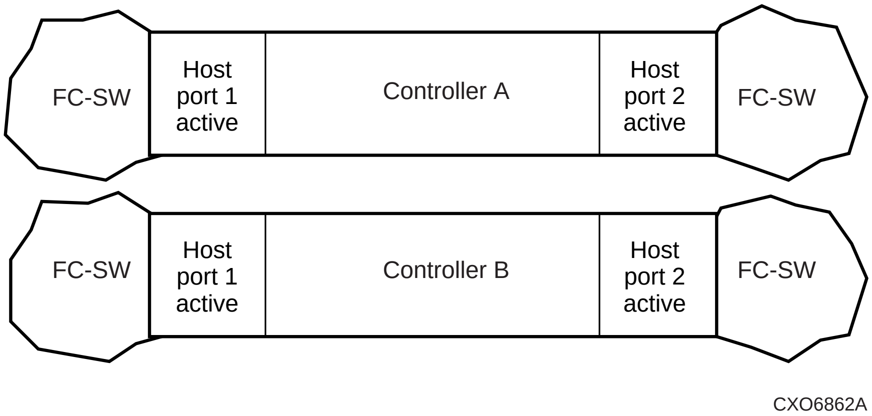


Figure 2–4. Example of Separate Host Port IDs in Multiple-Bus Failover

As Figure 2–4 shows, all ports on both controllers are active and have the ability to present all LUNs, 0–199, to the host. If one pot or one path loses communication, the drivers in the host can access the units through another port or another path.

Preferring the Host Path

In multiple-bus failover, the host can designate which controller initially accesses the units. This is called *preferring host paths*. See "ADD UNIT," page 7–27 and the PREFERRED_PATH switch for details.

Caching Techniques

The cache module supports the following caching techniques to increase the subsystem's read and write performance:

- Read caching
- Read-ahead caching
- Write-through caching
- Write-back caching

Read Caching

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. This process is called *read caching*.

Read caching can decrease the subsystem's response time to many of the host's read requests. If the host requests some or all of the cached data, the controller satisfies the request from its cache module rather than from the disk drives. By default, read caching is enabled for all storage units.

See "ADD UNIT," page 7–27, for information on the switch, MAXIMUM_CACHED_TRANSFER.

Read-Ahead Caching

Read-ahead caching begins when the controller has already processed a read request, and it receives a subsequent read request from the host. If the controller does not find the data in the cache memory, it reads the data from the disks and sends it to the cache memory.

During read-ahead caching, the controller anticipates subsequent read requests and begins to prefetch the next blocks of data from the disks as it sends the requested read data to the host. This is a parallel action. The controller notifies the host of the read completion, and subsequent sequential read requests are satisfied from the cache memory. By default, read-ahead caching is enabled for all disk units.

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 actually passes through—and is stored in—the cache memory on its way to the disk drives.

Write-Back Caching

Write-back caching improves 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.

By default, write-back caching is enabled for all units. In either case, the controller will not provide write-back caching to a unit unless the cache memory is non-volatile, as described in the next section.

Fault-Tolerance for Write-Back Caching

The cache module supports nonvolatile memory and dynamic cache policies to protect the availability of its unwritten (write-back) data:

Nonvolatile Memory

The controller can provide write-back caching for any storage unit as long as the controller's cache memory is nonvolatile. In other words, to enable write-back caching, you must provide a backup power source to the cache module to preserve the unwritten cache data in the event of a power failure. If the cache memory were volatile—that is, if it didn't have a backup power supply—the unwritten cache data would be lost during a power failure.

By default, the controller expects to use an ECB as the backup power source for its cache module. See "External Cache Battery," page 1–17, for more information about the ECB.

However, if your subsystem is backed up by a UPS (uninterruptable power supply), you can tell the controller to use the UPS as the backup power source with the CACHE_UPS switch in the command, "SET controller," page 7–91.

Cache Policies Resulting from Cache Module Failures

If the controller detects a full or partial failure of its cache module or ECB, it automatically reacts to preserve the unwritten data in its cache module. Depending upon the severity of the failure, the controller chooses an interim caching technique (also called the cache policy) which it uses until you repair or replace the cache module or ECB.

Table 2–2 shows the cache policies resulting from a full or partial failure of cache module A in a dual-redundant controller configuration. The consequences shown in this table are the same for cache module B.

Table 2–2. Cache Policies and Cache Module Status

Cache Module Status		Cache Policy	
Cache A	Cache B	Unmirrored Cache	Mirrored Cache
Good	Good	Data loss: No Cache policy: Both controllers support write-back caching. Failover: No	Data loss: No Cache policy: Both controllers support write-back caching. Failover: No
Multibit cache memory failure	Good	Data loss: Yes. Forced error and loss of write-back data for which multibit error occurred. Controller A detects and reports the lost blocks. Cache policy: Both controllers support write-back caching. Failover: No	Data loss: No. Controller A recovers its lost write-back data from the mirrored copy on cache B. Cache policy: Both controllers support write-back caching. Failover: No

Table 2–2. Cache Policies and Cache Module Status (Continued)

Cache Module Status		Cache Policy	
Cache A	Cache B	Unmirrored Cache	Mirrored Cache
DIMM or cache memory controller chip failure	Good	Data integrity: Write-back data that was not written to media when failure occurred was not recovered. Cache policy: Controller A supports write-through caching only; controller B supports write-back caching. Failover: In transparent failover, all units failover to controller B. In multiple-bus failover with host-assist, only those units that use write-back caching, such as RAIDsets and mirrorsets, failover to controller B. All units with lost data become inoperative until you clear them with the CLEAR LOST_DATA command. Units that didn't lose data operate normally on controller B. In single controller configurations, RAIDsets, mirrorsets, and all units with lost data become inoperative. Although you can clear the lost data errors on some units, RAIDsets and mirrorsets remain inoperative until you repair or replace the nonvolatile memory on cache A.	Data integrity: Controller A recovers all of its write-back data from the mirrored copy on cache B. Cache policy: Controller A supports write-through caching only; controller B supports write-back caching. Failover: In transparent failover, all units failover to controller B and operate normally. In multiple-bus failover with host-assist, only those units that use write-back caching, such as RAIDsets and mirrorsets, failover to controller B.

Table 2–2. Cache Policies and Cache Module Status (Continued)

Cache Module Status		Cache Policy	
Cache A	**Cache B**	**Unmirrored Cache**	**Mirrored Cache**
Cache Board Failure	Good	Same as for DIMM failure.	Data integrity: Controller A recovers all of its write-back data from the mirrored copy on cache B. Cache policy: Both controllers support write-through caching only. Controller B cannot execute mirrored writes because cache module A cannot mirror controller B's unwritten data. Failover: No

Table 2–3 shows the cache policies resulting from full or partial failure of cache module A's ECB in a dual-redundant configuration. Note that when cache module A is at least 50% charged, the ECB is still operable and charging. When it is less than 50% charged, the ECB is low but still charging. The consequences shown in this table are opposite for a failure of cache module B's ECB.

Table 2–3. Resulting Cache Policies and ECB Status

ECB Status		Cache Policy	
Cache A	Cache B	Unmirrored Cache	Mirrored Cache
At least 50% charged	At least 50% charged	Data loss: No Cache policy: Both controllers continue to support write-back caching. Failover: No	Data loss: No Cache policy: Both controllers continue to support write-back caching. Failover: No
Less than 50% charged	At least 50% charged	Data loss: No Cache policy: Controller A supports write-through caching only; controller B supports write-back caching. Failover: In transparent failover, all units failover to controller B. In multiple-bus failover with host-assist, only those units that use write-back caching, such as RAIDsets and mirrorsets, failover to controller B. In single configurations, the controller only provides write-through caching to its units.	Data loss: No Cache policy: Both controllers continue to support write-back caching. Failover: No

Table 2–3. Resulting Cache Policies and ECB Status (Continued)

ECB Status		Cache Policy	
Cache A	Cache B	Unmirrored Cache	Mirrored Cache
Failed	At least 50% charged	Data loss: No Cache policy: Controller A supports write-through caching only; controller B supports write-back caching. Failover: In transparent failover, all units failover to controller B and operate normally. In multiple-bus failover with host-assist, only those units that use write-back caching, such as RAIDsets and mirrorsets, failover to controller B. In single configurations, the controller only provides write-through caching to its units.	Data loss: No Cache policy: Both controllers continue to support write-back caching. Failover: No
Less than 50% charged	Less than 50% charged	Data loss: No Cache policy: Both controllers support write-through caching only. Failover: No	Data loss: No Cache policy: Both controllers support write-through caching only. Failover: No

Table 2–3. Resulting Cache Policies and ECB Status (Continued)

ECB Status		Cache Policy	
Cache A	Cache B	Unmirrored Cache	Mirrored Cache
Failed	Less than 50% charged	Data loss: No Cache policy: Both controllers support write-through caching only. Failover: In transparent failover, all units failover to controller B and operate normally. In multiple-bus failover with host-assist, only those units that use write-back caching, such as RAIDsets and mirrorsets, failover to controller B. In single configurations, the controller only provides write-through caching to its units.	Data loss: No Cache policy: Both controllers support write-through caching only. Failover: No
Failed	Failed	Data loss: No Cache policy: Both controllers support write-through caching only. Failover: No. RAIDsets and mirrorsets become inoperative. Other units that use write-back caching operate with write-through caching only.	Data loss: No Cache policy: Both controllers support write-through caching only. Failover: No. RAIDsets and mirrorsets become inoperative. Other units that use write-back caching operate with write-through caching only.

Enabling Mirrored Write-Back Cache

Before configuring dual-redundant controllers and enabling mirroring, ensure the following conditions are met:

- Both controllers are configured with the same size cache, 64 MB, 128 MB, 256 MB, or 512 MB.
- Diagnostics indicates that both caches are good.
- Both caches have a battery present, if you have not enabled the CACHE_UPS switch. A battery does not have to be present for either cache if you enable the CACHE_UPS switch.
- No unit errors are outstanding, for example, lost data or data that cannot be written to devices.
- Both controllers are started and configured in failover mode.

For important considerations when adding or replacing DIMMs in a mirrored cache configuration, refer to the *HSG80 Array Controller ACS Version 8.3 and 8.4 Maintenance and Service Guide*.

Chapter 3
Controller and Host Addressing

This chapter provides a basis for understanding how the controller talks to physical devices (on the controller bus) and how the host talks to logical units (on the host bus), which are created from the physical devices. Also described are storagesets, mapping devices and units, setting unit offsets, assigning preferred host paths, and understanding worldwide names and port IDs.

Bus Dynamics

Your controller is the intelligent bridge between your host and the devices and the logical units comprising your subsystem. The controller is an integral part of any storage subsystem because it provides a host with high-performance and high-availability access to the storage devices, sometimes called disk drive Storage Building Blocks (SBBs).

As Figure 3–1 shows, the controller communicates through two buses:

- Controller device bus
- Host bus

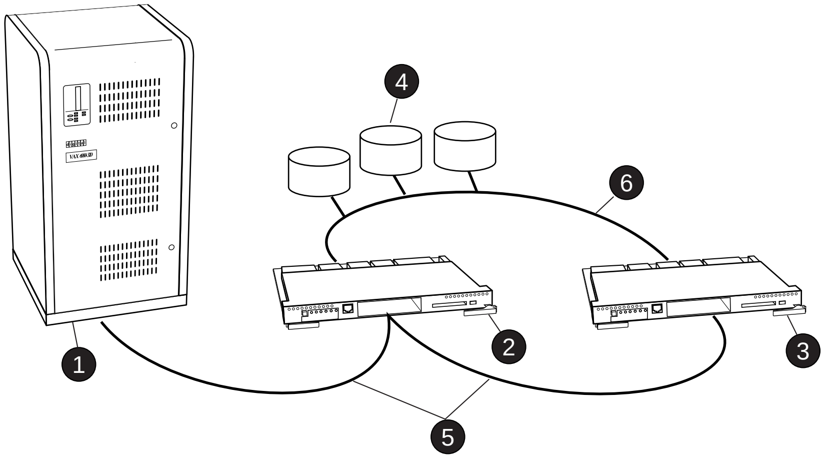


Figure 3–1. Host Bus and Controller Device SCSI Bus

Table 3–1. Key to Figure 3–1 Host Bus and Controller Device SCSI Bus

Item	Description
①	Host
②	Controller A
③	Controller B
④	Logical units made from storagesets
⑤	Host bus
⑥	Controller device bus

Host Bus

When the host communicates to the controller through the host bus, the controller maps that communication to the logical units on the device bus. Even though the physical devices exist, the host accesses only the logical units that you have created from making containers and storagesets. This way, the controller helps the host to be more efficient by doing the following:

- combines several devices into a storageset, which allows the host to spread its I/O load across more devices.
- assists with certain low-level I/O requests, which frees up the host for processing more intensive applications.

Controller Device Bus

A controller communicates between the physical devices and the host, connecting to the devices directly over one of the six available device I/O buses. The controller has access to physical devices, whereas the host has access to only the logical units.

Containers and Storagesets

Containers and storagesets are comprised of physical devices. The logical units are created after the devices and storagesets have been added to the storage subsystem. Figure 3–2 shows the configuration for containers, storagesets, and single devices. Frequently, containers are called by the more generic term, storagesets.

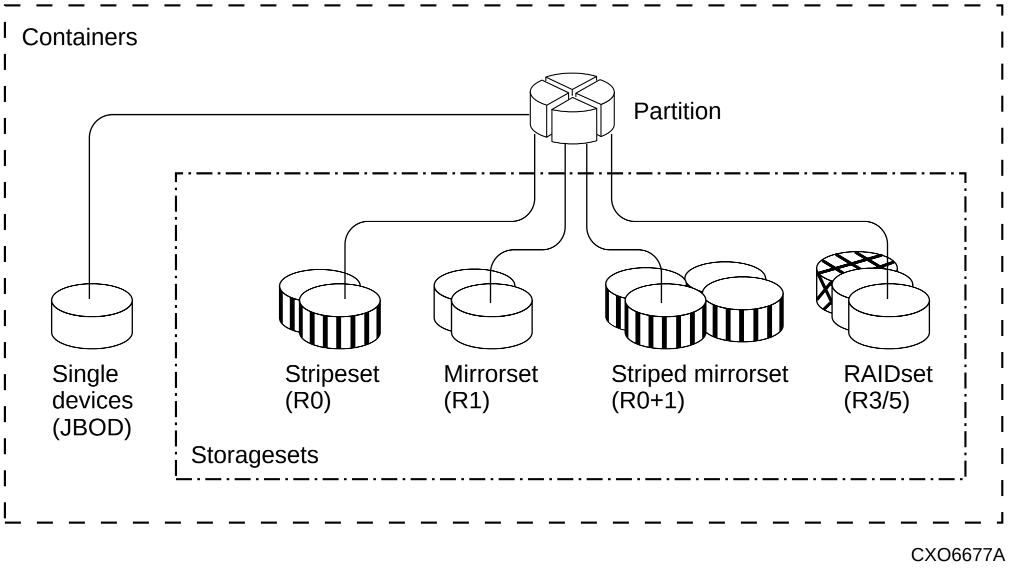


Figure 3–2. Containers and Storagesets

Storagesets

Storagesets are implementations of RAID technology. RAID technology ensures that every unpartitioned storageset, regardless of the number of disk drives, appears to the host as a single storage unit.

The following are the supported storagesets for the HSG80 array controller:

- RAID 0 (Stripesets)—physical devices combined in serial to increase transfer or request rates.
- RAID 1 (Mirrorsets)—physical devices combined in parallel to provide a highly reliable storage unit.
- RAID 0 + 1 (Striped Mirrorsets)—combined mirrorsets in serial and parallel providing the highest throughput and availability.
- RAID 3/5 (RAIDsets)—physical devices combined in serial (as RAID 0), but also stored parity data ensuring high reliability.

See the *The RAIDBook–A Source for RAID Technology* for more information about RAID technology.

Containers

Containers are comprised of storagesets, disk drives, and partitions:

- Storagesets—Any combination of RAID storagesets.
- Single disk drives (also called JBODs)—Any combination of single disk drives that contain data with a distinct unit number and that are not part of RAID storagesets.
- Partitions—A configuration of single disk drives or a storageset where the data storageset area is partitioned.

IMPORTANT: Any storageset can be partitioned in transparent failover mode. However, storagesets are not allowed to be partitioned in multiple-bus failover mode.

Logically Connecting the Storage Array to the Host

The combination of storage devices is called a *storage array*. The controller uses a two-step mapping process to logically connect the host to the storage array:

- Step1—The controller maps the physical devices on its six device buses to storage containers that you have created.
- Step 2—The controller maps its internal containers to user-created logical units that are directly accessible by the host.

Mapping the Physical Devices with Device PTL Addressing

Your controller has six SCSI–2 device ports. Each device port connects to an enclosure that supports 1 to 4 devices called *targets*. Every device uses LUN 0.

The controller identifies the location of devices based on a Port-Target-LUN (PTL) numbering scheme. The controller uses the PTL address to locate physical devices.

- P—Designates the controller's SCSI device port number (1 through 6).
- T—Designates the target identification (ID) number of the device. Valid target ID numbers for a single-controller configuration and dual-redundant controller configuration are 0 through 15, excluding ID numbers 4 through 7. ID 6 and ID 7 are reserved for the controller pair, and ID numbers 4 and 5 are never used.
- L—Designates the logical unit (LUN) of the device.

NOTE: The controller operates with BA370 rack-mountable enclosures that are assigned ID numbers 0, 2, and 3. These ID numbers are set through the PVA module. Enclosure ID number 1, which houses devices at targets 4 through 7, is not supported. Do not use device target ID numbers 4 through 7 in a storage subsystem.

Place one space between the port number, target number, and the two-digit LUN number when entering the PTL address. An example of a PTL address is shown in Figure 3–3.

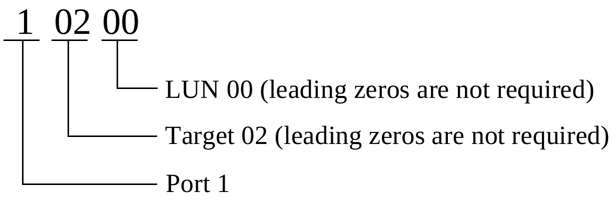
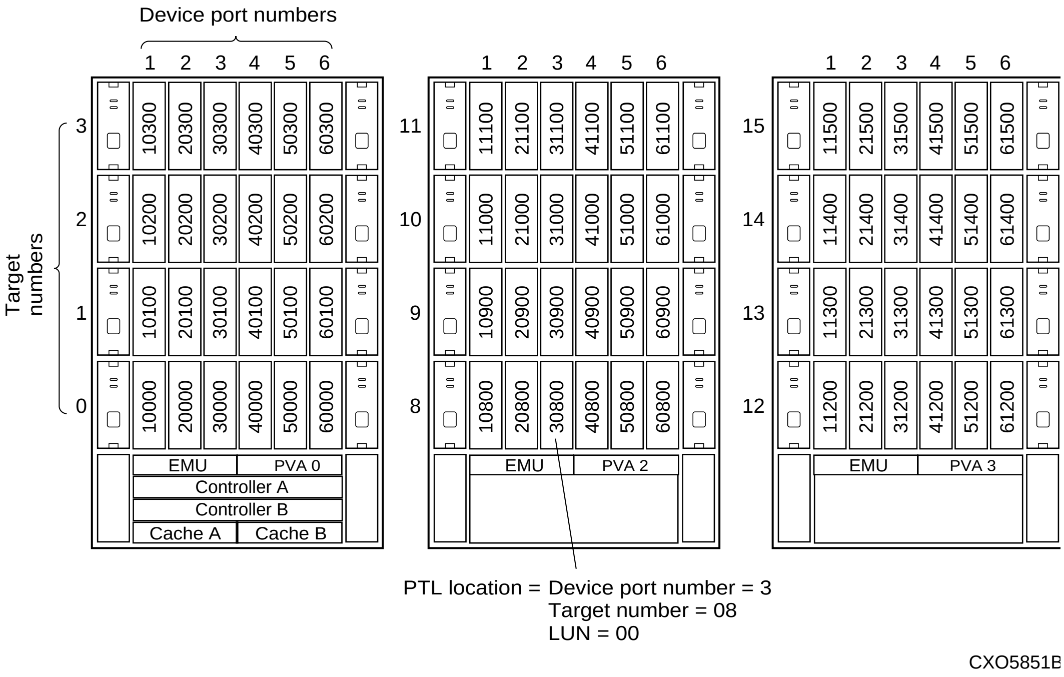

1 02 00
LUN 00 (leading zeros are not required)
Target 02 (leading zeros are not required)
Port 1

Figure 3–3 PTL Naming Convention

Figure 3–4 shows the addresses for each device in a typical, extended configuration. Use this figure along with the “Configuration Rules,” page 5–3 to help you work with the devices in your configuration.



PTL location = Device port number = 3
Target number = 08
LUN = 00

Figure 3–4 PTL Addressing in a Configuration

In Figure 3–4, the controller addresses DISK30800 through device port 3, target 08, LUN 00. This PTL location indicates the pathway the controller uses to address a disk drive (device) in the subsystem. It also indicates the device name.

The controller uses the PTL location to name each device that you add to your subsystem with StorageWorks Command Console or the CONFIG utility. Factory-installed devices are added with the CONFIG utility. Thus, their names derive from their PTL locations. For example, if the controller finds a disk in PTL 10200, the name will be DISK10200.

When your controller receives an I/O request, it identifies the host logical unit number, listing all the storageset names in the logical unit. Using the unit number, the controller locates the appropriate device for the I/O request. For example in Figure 3–5, a RAIDset called RAID1 might contain DISK10000, DISK20000, and DISK30000. The controller generates the read or write request to the appropriate device using the PTL addressing convention.

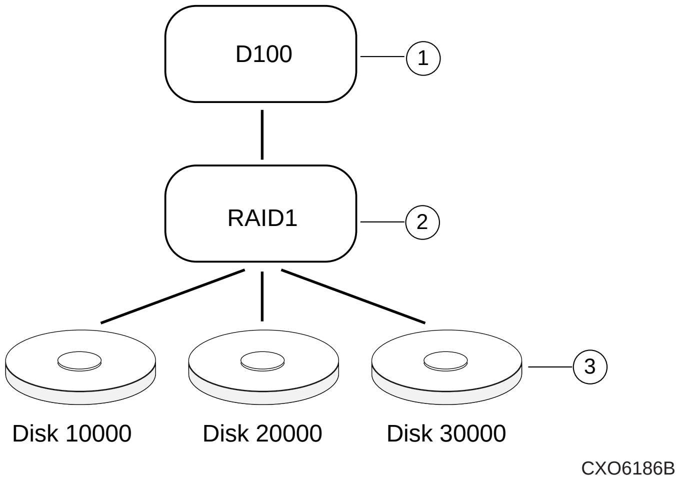


Figure 3–5 Locating Devices using PTLs

Table 3–2 Key to Figure 3–5 Locating Devices using PTLs

Item	Description
①	**Host unit number**
②	**Storageset name**
③	**Physical devices using controller PTL addresses**

Mapping the Containers and Storagesets to the Host with Logical Units

In a subsystem, the host communicates to the physical devices through host-addressable *logical units*, also called *units*. These units are created and given a logical unit number (LUN) after adding physical devices, adding containers, and initializing the containers. The host uses these numbers to indicate the source or target for every I/O request it sends to a controller.

Each unit number contains the following:

- A letter that indicates the kind of devices in the storage unit. For example, D for disk drives.
- A number from 0–199, depending on which failover mode you use.

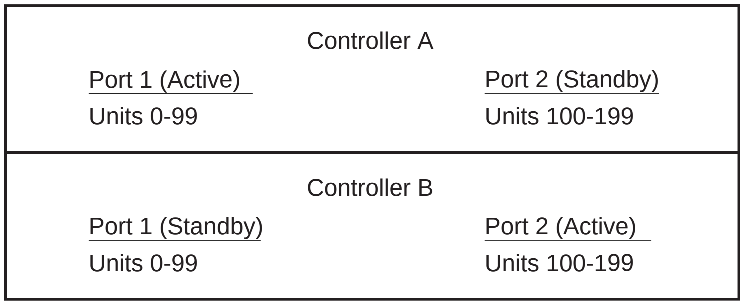
IMPORTANT: For OpenVMS hosts, you must set a *unit identifier* just after creating the logical unit. Without a unit identifier, the OpenVMS host will never recognize the logical unit. For information on when to add the identifier, see Chapter 6, “Configuring Storagesets.” For information on the *IDENTIFIER* switch, see “SET controller,” page 7–91.

Assigning Unit Numbers in Transparent Failover Mode

Each controller has two ports, port 1 and port 2, as shown in Figure 3–6. A set number of units are accessible, depending on the host operating system.

In transparent failover mode, the range of assignable units is 0–99 on port 1, and 100-199 on port 2, regardless of what unit offset is set on the host. Do not split partitioned storagesets across ports; they must be on the same port.

See “ADD UNIT,” page 7–27 for more details.



Controller A

Port 1 (Active)	Port 2 (Standby)
Units 0-99	Units 100-199

Controller B

Port 1 (Standby)	Port 2 (Active)
Units 0-99	Units 100-199

CXO6187B

Figure 3–6 Controller Port ID and Unit Numbers in Transparent Failover Mode

Assigning Unit Numbers in Multiple-Bus Failover Mode

In multiple-bus failover mode, the range of assignable units that are accessible from any port on the subsystem is 0–199. Hosts obtain units by reserving the unit for sole access. This process is performed on a first-available basis or by assigning host access. Figure 3–7 illustrates the controller port ID and unit numbers in multiple-bus failover mode.

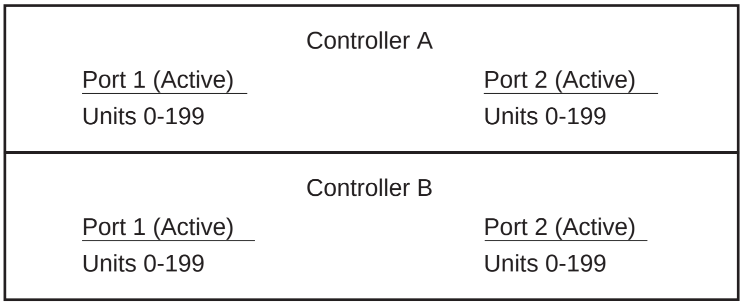


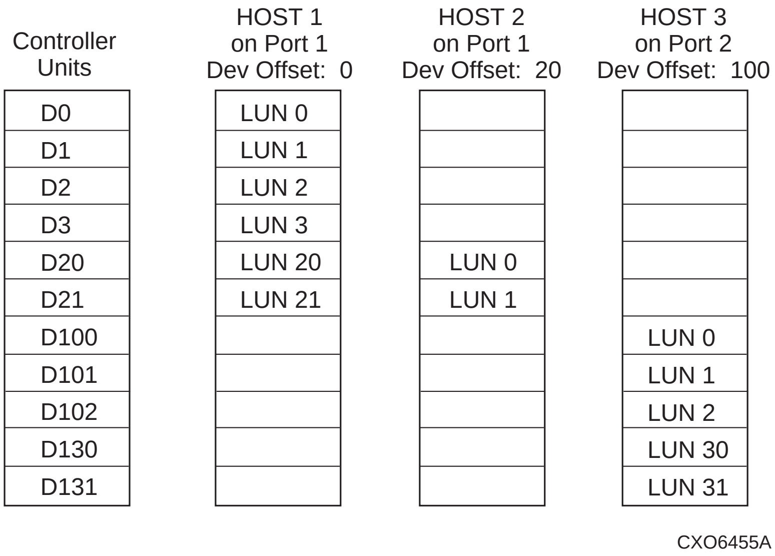
Figure 3–7 Controller Port ID Numbers and Unit Numbers in Multiple-Bus Failover Mode

Assigning Unit Offsets

Unit offsets are used to define the first unit number that a host connection is able to access on the controller. Many operating systems have a limited addressing range, such as 0–7. Using unit offsets allows hosts to see controller LUNs at 0–7.

When assigning unit offsets, keep in mind that the unit offset is set on a host connection basis. Thus, one host connection can have a unit offset of 0 and see units 0–99 as LUNs 0–99, and another host connection can have a unit offset of 10 and see units 10–99 as LUNs 0–89 (controller unit number 10 is seen at the host as LUN 0). This process is useful for hosts that have a limited LUN addressing range.

Figure 3–8 depicts the use of unit offsets on host connection basis. Note that controller units 0–99 are presented only on port 1, and units 100–199 are presented only on port 2. This implies that port 1 host connections can have a unit offset range of 0–99, and port 2 host connections can have a unit offset range of 100–199.



Controller Units	HOST 1 on Port 1 Dev Offset: 0	HOST 2 on Port 1 Dev Offset: 20	HOST 3 on Port 2 Dev Offset: 100
D0	LUN 0		
D1	LUN 1		
D2	LUN 2		
D3	LUN 3		
D20	LUN 20	LUN 0	
D21	LUN 21	LUN 1	
D100			LUN 0
D101			LUN 1
D102			LUN 2
D130			LUN 30
D131			LUN 31

Figure 3–8 LUN Presentation Using Unit Offset on a Per-Host Basis

Assigning Access Paths

Different operating systems may not tolerate other operating systems' access to their units. To limit or restrict host access to certain units, the HSG80 array controller allows you to specify unit access privileges. You may specify unit access on a unit-by-unit basis.

You can enable host access to a specific unit by mapping it as accessible through the specified host paths. A path is the logical connection between the host and the controller and the path is referenced by the name attached to the logical connection. Once the unit is set for access through a path, the access privilege is defined.

By default, host access is set to ALL. If you wish to have the unit access limited, you must first disable access to ALL, then set it to a specified host access path.

Access paths may be enabled with the ENABLE_ACCESS_PATH switch. You can define a unit's access privileges with "ADD UNIT," page 7–27. These access privileges can be changed using "SET unit-number," page 7–126.

IMPORTANT: Before enabling the access path, make sure that the host connections have been added with the command, "SET connection-name," page 7–89.

.See "Configuring in Multiple-Bus Failover Mode," page 5–20 on working in a multiple-host environment.

Controller and Port Worldwide Names (Node IDs)

A worldwide name—also called a NODE_ID—is a unique 64-bit number assigned to a subsystem by the Institute of Electrical and Electronics Engineers (IEEE) and set by Compaq manufacturing prior to shipping. As shown in Figure 3–9, it refers to the shelf the controllers sit on, and the worldwide name is stored in the controllers' memory for the benefit of the shelf. The worldwide name assigned to a subsystem never changes.

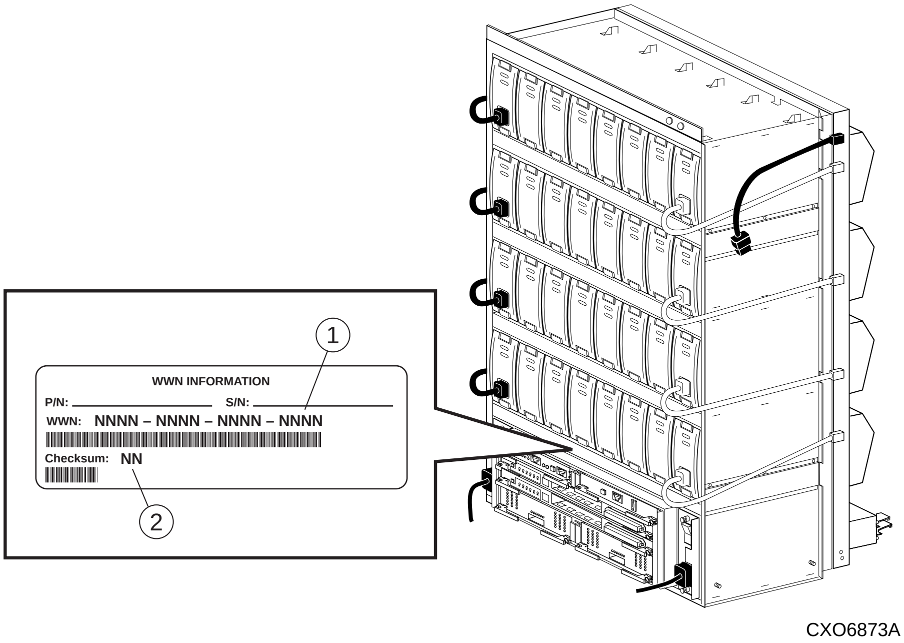


Figure 3–9. Placement of the Worldwide Name Label in a Subsystem

Each subsystem's worldwide name ends in zero, for example 5000-1FE1-FF0C-EE00. The controller port IDs are derived from the worldwide name.

In a subsystem with two controllers in transparent failover mode, the controller port IDs increment as follows:

- Controller A and controller B, port 1—worldwide name + 1
- Controller A and controller B, port 2—worldwide name + 2

For example, using the worldwide name of 5000-1FE1-FF0C-EE00, the following port IDs are automatically assigned and shared between the ports as a REPORTED PORT_ID on each port:

- Controller A and controller B, port 1—5000-1FE1-FF0C-EE01
- Controller A and controller B, port 2—5000-1FE1-FF0C-EE02

In a subsystem with two controllers in multiple-bus failover mode, the port IDs are separate from each other and are not shared, incrementing as follows:

- Controller B, port 1—worldwide name + 1
- Controller B, port 2—worldwide name + 2
- Controller A, port 1—worldwide name + 3
- Controller A, port 2—worldwide name + 4

For example, using the worldwide name of 5000-1FE1-FF0C-EE00, the following port IDs are automatically assigned and shared between the ports as a REPORTED PORT_ID on each port:

- Controller B, port 1—5000-1FE1-FF0C-EE01
- Controller B, port 2—5000-1FE1-FF0C-EE02
- Controller A, port 1—5000-1FE1-FF0C-EE03
- Controller A, port 2—5000-1FE1-FF0C-EE04

Use the CLI command, SHOW *controller*, to display the subsystem’s worldwide name. The CLI uses the term node ID for worldwide names. When you enter the SHOW command, the subsystem worldwide name (node ID) displays as the NODE_ID and will look like the following:

```
5000-1FE1-FF0C-EE00
```

Restoring Worldwide Names (Node IDs)

CAUTION: Each subsystem has its own unique worldwide name (node ID). This ID can be found on a sticker, which is located on top of the frame that houses the controllers, the EMU, the PVA, and cache modules. 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 or data corruption will occur.

When you remove a controller to replace it in a dual-redundant configuration, the remaining controller remembers the subsystem worldwide name (node ID). When you install the replacement controller, the remaining controller tells the new controller the worldwide name; the replacement controller assumes the correct port ID numbers.

If you have a single controller configuration, you must have a save configuration disk if you want to be able to automatically restore the worldwide name in the event of a failure. In this case the controller could read the worldwide name from the save configuration disk.

If a situation occurs that requires you to restore the worldwide name, you can restore it using the worldwide name and check sum printed on the sticker on the frame into which your controller is inserted. See the command, “SET controller,” page 7–91 for details about setting the worldwide name (node ID).

Unit World Wide Names (LUN IDs)

In addition, each unit has its own world wide name, or LUN ID. This is a unique, 128-bit value that the controller assigns at the time of unit initialization. It cannot be altered by the user but does change when the unit is reinitialized. Use the SHOW command to list the LUN ID.

Chapter 4
Planning Storagesets

This chapter takes you through the planning steps and procedures for planning and creating storagesets for your subsystem.

Introduction

Storagesets are implementations of RAID technology, also known as a “Redundant Array of Independent Disks.” Every storageset shares one important feature: each one looks like a single storage unit to the host, regardless of the number of drives it uses.

You can create storage units by combining disk drives into storagesets, such as stripesets, RAIDsets, and mirrorsets, or by presenting them to the host as single-disk units, as shown in Figure 4–1.

- Stripesets (RAID 0) combine disk drives in serial to increase transfer or request rates.
- Mirrorsets (RAID 1) combine disk drives in parallel to provide a highly-reliable storage unit.
- RAIDsets (RAID 3/5) combine disk drives in serial—as do stripesets—but also store parity data to ensure high reliability.
- Striped mirrorsets (RAID 0+1) combine mirrorsets in serial to provide the highest throughput and availability of any storage unit.

Controllers can support the number of RAIDsets as listed in Table 3-1.

For a complete discussion of RAID, refer to *The RAIDBOOK—A Source Book for Disk Array Technology.*

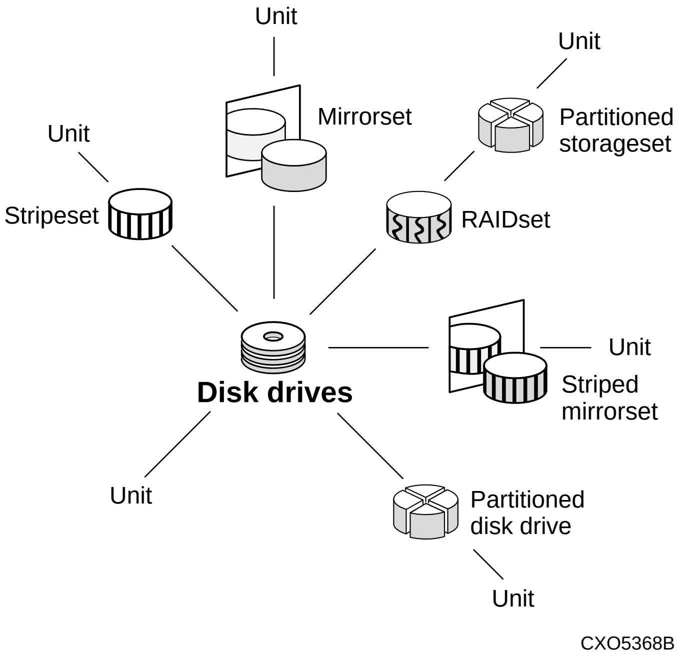


Figure 4–1. Units Created from Storagesets, Partitions, and Drives

Table 4–1. Controller Limitations for RAIDsets

RAIDset Type	Limit
Total number of RAID5	20
Total number of RAID5 + RAID1	30
Total number of RAID5 + RAID1 + RAID0	45

Planning and Configuring Storagesets

Use this procedure to plan and configure the storagesets for your subsystem. Use the references in each step to locate details about specific commands and concepts.

1. Create a storageset and device profile. See “Creating a Storageset and Device Profile,” page 4–5, for suggestions about creating a profile.
2. Determine your storage requirements. Use the questions in “Determining Storage Requirements,” page 4–7, to help you.
3. Choose the type of storagesets you need to use in your subsystem. See “Choosing a Storageset Type,” page 4–8, for a comparison and description of each type of storageset.
4. Select names for your storagesets and units. See “Creating a Storageset Map,” page 4–30, for details about selecting names.
5. Assign unit numbers to storagesets so the host can access the units. See“Creating a Storageset Map,” page 4–30, for information about how to assign units numbers to storagesets.
6. Create a storageset map to help you configure your subsystem. See “Creating a Storageset Map,” page 4–30, for suggestions about creating a storageset map.
7. If you are going to partition your storagesets, plan the partitions. See “Planning Partitions,” page 4–34, for information about partitions and how to plan for them.
8. Choose the switches that you will want to set for your storagesets and devices. See“Choosing Switches for Storagesets and Devices,” page 4–36, for a description of the switches you can select for storagesets.
9. Configure the storagesets you have planned using one of these methods:

- Use StorageWorks Command Console (SWCC), a Graphical User Interface (GUI), to set up and manage RAID storage subsystems. See the *SWCC Getting Started* guide for details about using SWCC to configure your storagesets.
- Use CLI commands. This method allows you flexibility in defining and naming storagesets. See “Configuring Storagesets with CLI Commands,” page 6–2, for information about configuring each type of storageset using CLI commands.

Creating a Storageset and Device Profile

Creating a profile for your storagesets and devices can help simplify the configuration process. This chapter helps you to choose the storagesets that best suit your needs and make informed decisions about the switches that you can enable for each storageset or storage device that you configure in your subsystem.

Familiarize yourself with the kinds of information contained in a storageset profile, as shown in Figure 4–2.

Appendix A, “System Profiles,”contains blank profiles that you can use to record the details for your devices and storagesets. Use the Device Profile to record your devices in the subsystem and use the Storageset Profile when to record storagesets, which are created from the devices. These templates, along with the information in this chapter, will help you plan how devices and storagesets get created and initialized, and ultimately, how they are made into logical units recognized by the host.

TYPE OF STORAGESET

_____ Mirrorset	_✓_ RAIDset	_____ Stripeset	_____ Striped Mirrorset

Storageset Name	*accept default values*
Disk Drives	*DISK10300, DISK20300, DIS30300*
Unit Number	*accept default*

Partitions

Unit #	Unit #	Unit #	Unit #	Unit #	Unit #	Unit #	Unit #
%	%	%	%	%	%	%	%

RAIDset Switches

Reconstruction Policy	Reduced Membership	Replacement Policy
_✓_ Normal (default)	_✓_ No (default)	_✓_ Best performance (default)
___ Fast	___ Yes, missing:	___ Best fit
		___ None

Mirrorset Switches

Replacement Policy	Copy Policy	Read Source
___ Best performance (default)	___ Normal (default)	___ Least busy (default)
___ Best fit	___ Fast	___ Round robin
___ None		___ Disk drive:

Initialize Switches

Chunk size	Save Configuration	Metadata
___ Automatic (default)	___ No (default)	_✓_ Destroy (default)
___ 64 blocks	_✓_ Yes	___ Retain
___ 128 blocks		
___ 256 blocks		
___ Other:		

Unit Switches

Read Cache	Read-Ahead Cache	Maximum Cache Transfer
_✓_ Yes (default)	_✓_ Yes (default)	_✓_ 32 blocks (default)
___ No	___ No	___ Other:
Write-Back Cache	**Write Protection**	**Availability**
___ Yes (default)	_✓_ No (default)	_✓_ Run (default)
_✓_ No	___ Yes	___ NoRun

Figure 4–2 A Typical Storageset Profile

Determining Storage Requirements

Start the planning process by determining your storage requirements. Here are a few of the questions you should ask yourself:

- What applications or user groups will access the subsystem? How much capacity do they need?
- What are the I/O requirements? If an application is data-transfer intensive, what is the required transfer rate? If it is I/O-request intensive, what is the required response time? What is the read/write ratio for a typical request?
- Are most I/O requests directed to a small percentage of the disk drives? Do you want to keep it that way or balance the I/O load?
- Do you store mission-critical data? Is availability the highest priority, or would standard backup procedures suffice?

Use your responses to these questions along with Figure 4–2 to determine the types of storagesets you should create to satisfy your organization’s requirements.

Choosing a Storageset Type

Different applications may have different storage requirements, so you will probably want to configure more than one kind of storageset in your subsystem.

All of the storagesets described in this book implement RAID (Redundant Array of Independent Disks) technology. Consequently, they all share one important feature: each storageset, whether it contains two disk drives or ten, looks like one large, virtual disk drive to the host.

Table 4–2 compares different kinds of storagesets to help you determine which ones satisfy your requirements.

Table 4–2. A Comparison of Different Kinds of Storagesets

Storageset	Relative Availability	Request Rate (Read/ Write) I/O per second	Transfer Rate (Read/ Write) MB per second	Applications
Array of disk drives (JBOD)	Equivalent to a single disk drive	Identical to single disk drive	Identical to single disk drive	
Stripeset (RAID 0)	Proportionate to number of disk drives; worse than single disk drive	Excellent if used with large chunk size	Excellent if used with small chunk size	High performance for non-critical data
Mirrorset (RAID1)	Excellent	Good/Fair	Good/Fair	System drives; critical files
RAIDset (RAID 3/5)	Excellent	Excellent/Fair	Good/Poor	High request rates, read-intensive, data lookup
Striped Mirrorset (RAID 0+1)	Excellent	Excellent if used with large chunk size	Excellent if used with small chunk size	Any critical response-time application

For a comprehensive discussion of RAID, refer to *The RAIDBOOK—A Source Book for Disk Array Technology.*

Using Stripesets to Increase I/O Performance

Stripesets enhance I/O performance by spreading the data across multiple disk drives. Each I/O request is broken into small segments called “chunks.” These chunks are then “striped” across the disk drives in the storageset, thereby allowing several disk drives to participate in one I/O request to handle several I/O requests simultaneously.

For example, in a three-member stripeset that contains disk drives 10000, 20000, and 30000, the first chunk of an I/O request is written to 10000, the second to 20000, the third to 30000, the fourth to 10000, and so forth until all of the data has been written to the drives.

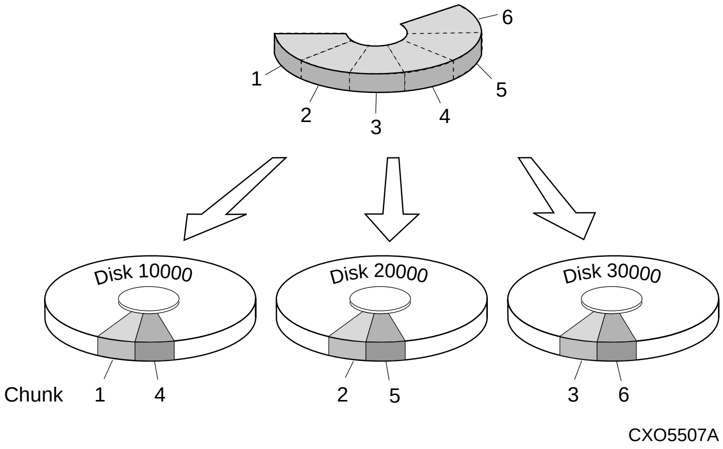


Figure 4–3. Striping Lets Several Disk Drives Participate in Each I/O Request

The relationship between the chunk size and the average request size determines if striping maximizes the request rate or the data-transfer rate. You can set the chunk size or let the controller set it automatically. See “Chunk Size,” page 4–43, for information about setting the chunk size.

A major benefit of striping is that it balances the I/O load across all of the disk drives in the storageset. This can increase the subsystem’s performance by eliminating the hot spots, or high localities of reference, that occur when frequently-accessed data becomes concentrated on a single disk drive.

Considerations for Planning a Stripeset

Keep the following points in mind as you plan your stripesets:

- A controller can support up to 45 storagesets, consisting of stripesets, mirrorsets and RAIDsets (refer to Table 4–1).
- Reporting methods and size limitations prevent certain operating systems from working with large stripesets. See the *HSG80 Array Controller ACS Version 8.X Release Notes* or the *Quick Setup Guide* that came with your platform kit for details about these restrictions.
- A storageset should only contain disk drives of the same capacity. The controller limits the capacity of each member to the capacity of the smallest member in the storageset when the storageset is initialized (the base member size). Thus, if you combine 9 GB disk drives with 4 GB disk drives in the same storageset, the 4 GB disk drive will be the base member size, and you will waste 5 GB of capacity on each 9 GB member.
- Striping does not protect against data loss. In fact, because the failure of one member is equivalent to the failure of the entire stripeset, the likelihood of losing data is higher for a stripeset than for a single disk drive.

For example, if the mean time between failures (MTBF) for a single disk is one hour, then the MTBF for a stripeset that comprises N such disks is l/N hours. As another example, if a single disk’s MTBF is 150,000 hours (about 17 years), a stripeset comprising four of these disks would only have an MTBF of slightly more than four years.

For this reason, you should avoid using a stripeset to store critical data. Stripesets are more suitable for storing data that can be reproduced easily or whose loss does not prevent the system from supporting its critical mission.

- Evenly distribute the members across the device ports to balance load and provide multiple paths as shown in Figure 4–4.

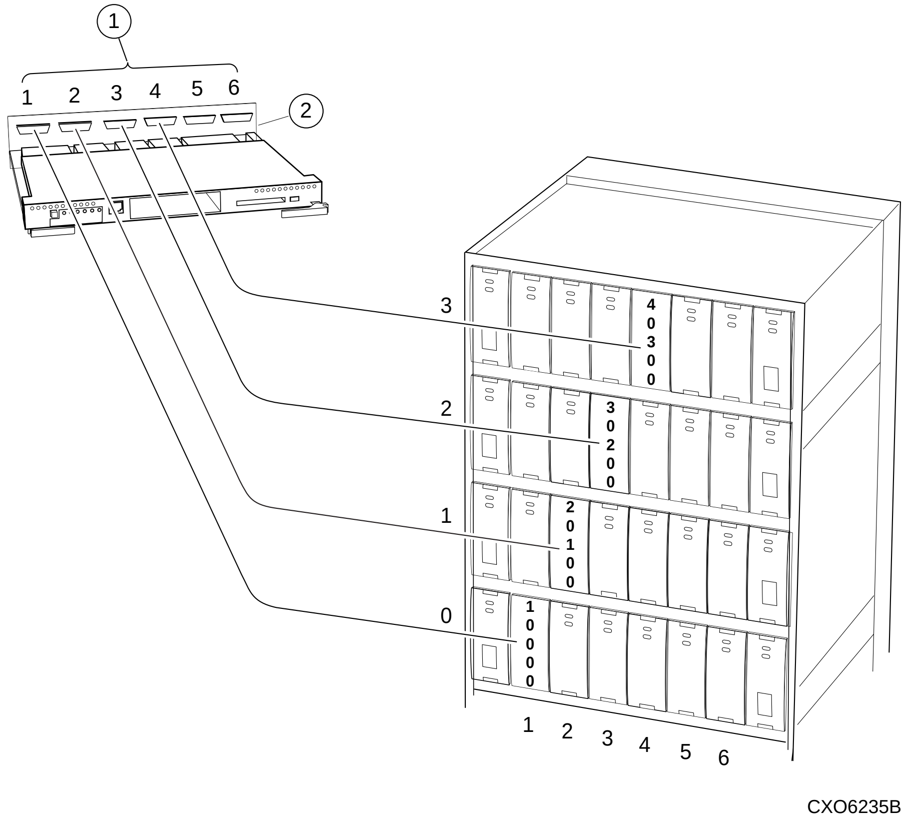


Figure 4–4. Distribute Members across Device Ports

Table 4–3. Key to Figure 4–4 Distribute Members across Device Ports

Item	Description
①	**Device ports**
②	**Backplane**

- Stripesets contain between 2 and 24 members.
- Stripesets are well-suited for the following applications:
  - ❏ Storing program image libraries or run-time libraries for rapid loading
  - ❏ Storing large tables or other structures of read-only data for rapid application access
  - ❏ Collecting data from external sources at very high data transfer rates
- Stripesets are not well-suited for the following applications:
  - ❏ A storage solution for data that cannot be easily reproduced or for data that must be available for system operation
  - ❏ Applications that make requests for small amounts of sequentially-located data
  - ❏ Applications that make synchronous random requests for small amounts of data

By spreading the traffic evenly across the buses, you will ensure that no bus handles the majority of data to the storageset.

Using Mirrorsets to Ensure Availability

Mirrorsets use redundancy to ensure availability, as illustrated in Figure 4–5. For each primary disk drive, there is at least one mirror disk drive. Thus, if a primary disk drive fails, its mirror drive immediately provides an exact copy of the data.

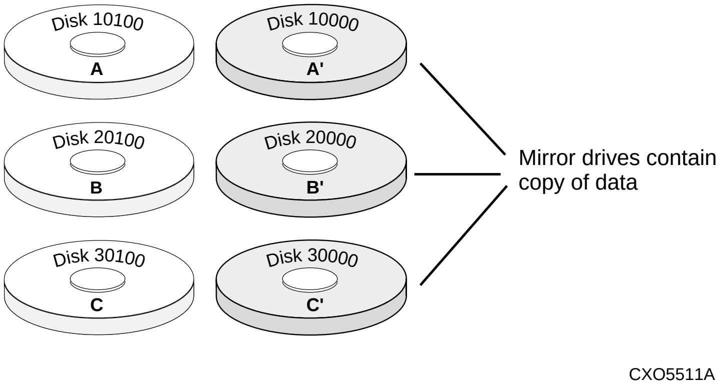


Figure 4–5. Mirrorsets Maintain Two Copies of the Same Data

Considerations for Planning a Mirrorset

Keep these points in mind as you plan your mirrorsets:

- A controller can support up to 30 storagesets, consisting of mirrorsets and RAIDsets. Mirrorsets that are members of a stripeset count against this limitation (refer to Table 4–1).
- Data availability with a mirrorset is excellent but costly—you need twice as many disk drives to satisfy a given capacity requirement. If availability is your top priority, consider using redundant power supplies and dual-redundant controllers.
- You can configure up to 30 mirrorsets per controller or pair of dual-redundant controllers. Each mirrorset contains a minimum of one and a maximum of six members.
- A write-back cache module is required for mirrorsets, but write-back cache need not be enabled for the mirrorset to function properly.
- Both write-back cache modules must be the same size.

- If you’re using more than one mirrorset in your subsystem, you should put the first member of each mirrorset on different buses as shown in Figure 4–6. (The first member of a mirrorset is the first disk drive you add.)

  When a controller receives a request to read data from a mirrorset, it typically accesses the first member of the mirrorset. Read access depends upon the read source switches, as described in “Read Source,” page 4–40. If you have several mirrorsets in your subsystem and their first members are on the same bus, that bus will be forced to handle the majority of traffic to your mirrorsets. When a controller receives a request to write data to a mirrorset, it accesses and writes to all members.

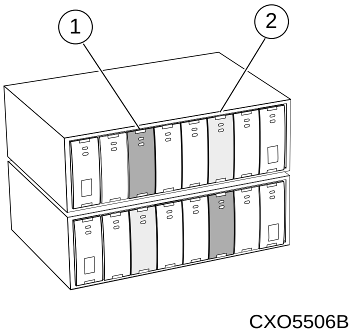


Figure 4–6. First Mirrorset Members on Different Buses

Table 4–4. Key to Figure 4–6 First Mirrorset Members on Different Buses

Item	Description
①	First member of Mirrorset 1
②	First member of Mirrorset 2

To avoid an I/O bottleneck on one bus, you can simply put the first members on different buses. Additionally, you can set the read-source switch to Round Robin. See “Read Source,” page 4–40, for more information about this switch.

- Place mirrorsets and RAIDsets on different ports to minimize risk in the event of a single port bus failure.
- Mirrorset units are set to WRITEBACK_CACHE by default which increases a unit's performance.
- A storageset should only contain disk drives of the same capacity. The controller limits the capacity of each member to the capacity of the smallest member in the storageset. Thus, if you combine 9 GB disk drives with 4 GB disk drives in the same storageset, the 4 GB disk drive will be the base member size, and you waste 5 GB of capacity on each 9 GB member.
- Evenly distribute the members across the device ports to balance load and provide multiple paths as shown in Figure 4–4 on page 4–12.
- Mirrorsets are well-suited for the following:
  - ❏ Any data for which reliability requirements are extremely high
  - ❏ Data to which high-performance access is required
  - ❏ Applications for which cost is a secondary issue
- Mirrorsets are not well-suited for the following applications:

  ❏ Write-intensive applications (JBODs are better for this type of application, but mirrorsets are preferred over Raid5 RAIDsets.)
  ❏ Applications for which cost is a primary issue

Using RAIDsets to Increase Performance and Availability

RAIDsets are enhanced stripesets—they use striping to increase I/O performance and distributed-parity data to ensure data availability. Figure 4–7 illustrates the concept of RAIDsets and parity data.

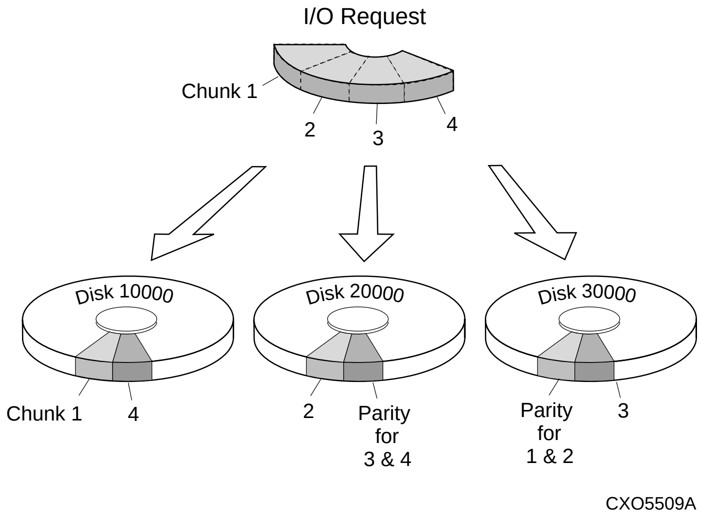


Figure 4–7. Parity Ensures Availability; Striping Provides Good Performance

Just as with stripesets, the I/O requests are broken into smaller “chunks” and striped across the disk drives until the request is read or written. But, in addition to the I/O data, chunks of parity data—derived mathematically from the I/O data—are also striped across the disk drives. This parity data enables the controller to reconstruct the I/O data if a disk drive fails. Thus, it becomes possible to lose a disk drive without losing access to the data it contained. (Data could be lost if a second disk drive fails before the controller replaces the first failed disk drive.)

For example, in a three-member RAIDset that contains disk drives 10000, 20000, and 30000, the first chunk of an I/O request is written to 10000, the second to 20000, then parity is calculated and written to 30000; the third chunk is written to 30000, the fourth to 10000, and so on until all of the data is saved.

The relationship between the chunk size and the average request size determines if striping maximizes the request rate or the data-transfer rates. You can set the chunk size or let the controller set it automatically. See "Chunk Size," page 4–43, for information about setting the chunk size.

Considerations for Planning a RAIDset

Keep these points in mind as you plan your RAIDsets:

- A controller can support up to 20 storagesets, consisting of RAIDsets (refer to Table 4–1).
- Reporting methods and size limitations prevent certain operating systems from working with large RAIDsets. See the *HSG80 Array Controller ACS Version 8.X Release Notes* or the *Quick Setup Guide* that came with your platform kit for details about these restrictions.
- A cache module is required for RAIDsets, but write-back cache need not be enabled for the RAIDset to function properly.
- Both cache modules must be the same size.
- A RAIDset must include at least 3 disk drives, but no more than 14.
- Evenly distribute the members across the device ports to balance load and provide multiple paths as shown in Figure 4–4 on page 4–12.
- A storageset should only contain disk drives of the same capacity. The controller limits the capacity of each member to the capacity of the smallest member in the storageset when the storageset is initialized (the base member size). Thus, if you combine 9-GB disk drives with 4-GB disk drives in the same storageset, you will waste 5 GB of capacity on each 9-GB member.
- RAIDset units are set to WRITEBACK_CACHE by default which increases a unit's performance.
- RAIDsets and mirrorsets on different ports to minimize risk in the event of a single port bus failure.

- RAIDsets are particularly well-suited for the following:
  - ❑ Small to medium I/O requests
  - ❑ Applications requiring high availability
  - ❑ High read request rates
  - ❑ Inquiry-type transaction processing
- RAIDsets are not particularly well-suited for the following:
  - ❑ Write-intensive applications
  - ❑ Applications that require high data transfer capacity
  - ❑ High-speed data collection
  - ❑ Database applications in which fields are continually updated
  - ❑ Transaction processing

Using Striped Mirrorsets for Highest Performance and Availability

As illustrated in Figure 4–8, striped mirrorsets are simply stripesets whose members are mirrorsets. Consequently, this kind of storageset combines the performance of striping with the reliability of mirroring. The result is a storageset with very high I/O performance and high data availability.

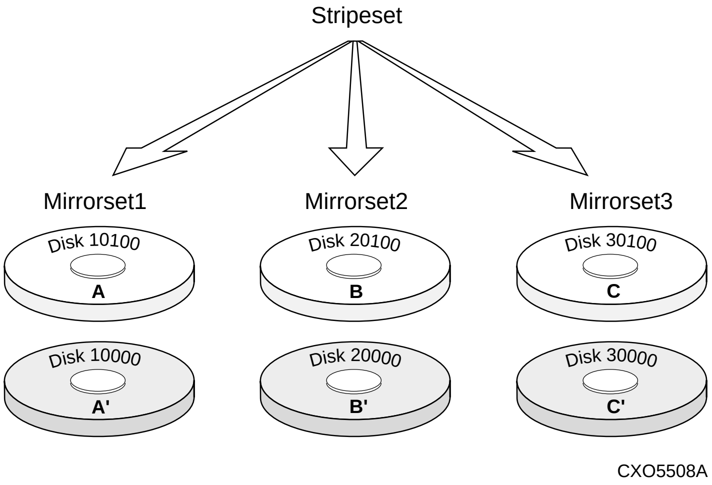


Figure 4–8. Striping and Mirroring in the Same Storageset

The failure of a single disk drive has no effect on this storageset's ability to deliver data to the host and, under normal circumstances, it has very little effect on performance. Because striped mirrorsets do not require any more disk drives than mirrorsets, this storageset is an excellent choice for data that warrants mirroring.

Considerations for Planning a Striped Mirrorset

Plan the mirrorset members, then plan the stripeset that will contain them. Review the recommendations in "Considerations for Planning a Stripeset," page 4–10, and "Considerations for Planning a Mirrorset," page 4–14.

There are the following limitations for a striped mirrorset:

- A maximum of 24 mirrorsets in a stripeset.
- A maximum of 6 disks in each mirrorset.
- A maximum of 48 disks in the entire striped mirrorset.

Cloning Data for Backup

Use the CLONE utility to duplicate the data on any unpartitioned single-disk unit, stripeset, mirrorset, or striped mirrorset in preparation for backup. When the cloning operation is done, you can back up the clones rather than the storageset or single-disk unit, which can continue to service its I/O load. When you are cloning a mirrorset, CLONE does not need to create a temporary mirrorset. Instead, it adds a temporary member to the mirrorset and copies the data onto this new member.

The CLONE utility creates a temporary, two-member mirrorset for each member in a single-disk unit or stripeset. Each temporary mirrorset contains one disk drive from the unit you are cloning and one disk drive onto which CLONE copies the data. During the copy operation, the unit remains online and active so the clones contain the most up-to-date data.

After the CLONE utility copies the data from the members to the clones, it restores the unit to its original configuration and creates a clone unit you can backup. The CLONE utility uses steps shown in Figure 4–9 to duplicate each member of a unit.

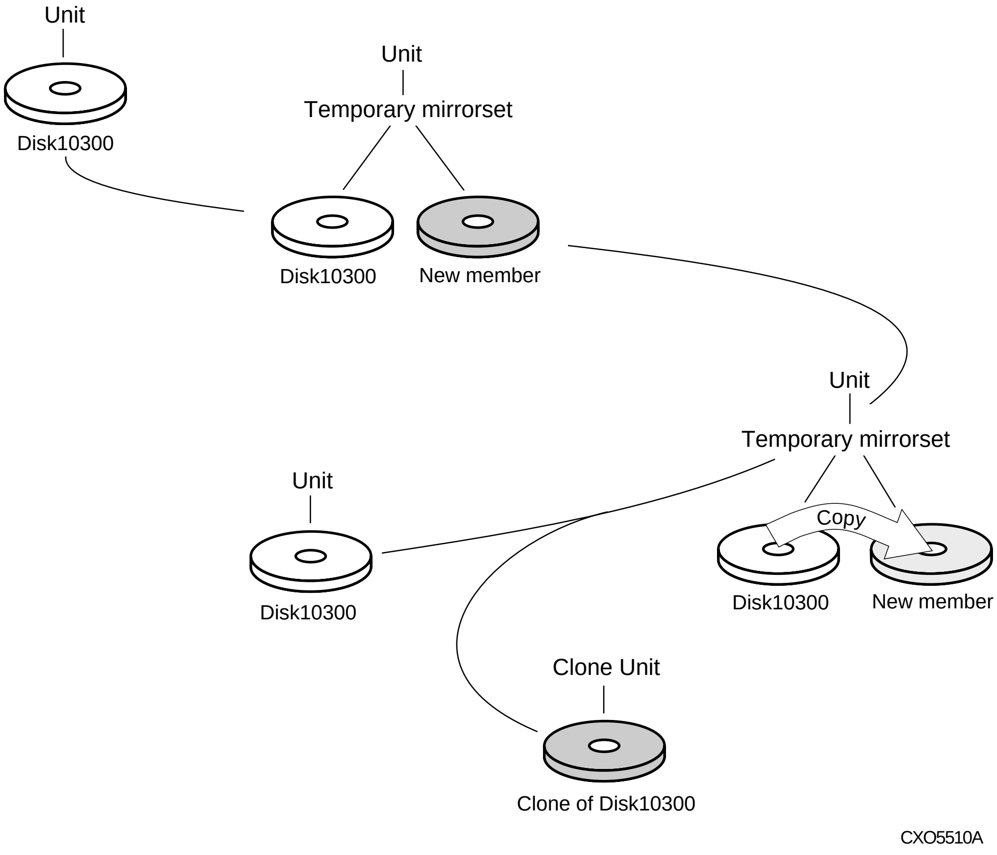


Figure 4–9. CLONE Steps for Duplicating Unit Members

To clone a single-disk unit, stripeset, or mirrorset:

1. Establish a connection to the controller that accesses the unit you want to clone.
2. Start CLONE using the following syntax:

```
RUN CLONE
```

3. When prompted, enter the unit number of the unit you want to clone.
4. When prompted, enter a unit number for the clone unit that CLONE will create.
5. When prompted, indicate how you would like the clone unit to be brought online: either automatically or only after your approval.
6. When prompted, enter the disk drives you want to use for the clone units.
7. Back up the clone unit.

The following example shows the commands you would use to clone storage unit D204. The clone command terminates after it creates storage unit D205, a clone or copy of D204.

```
                RUN CLONE
CLONE LOCAL PROGRAM INVOKED
UNITS AVAILABLE FOR CLONING:101
                             204
                Enter unit to clone ? 204
CLONE WILL CREATE A NEW UNIT WHICH IS A COPY OF UNIT 204.
                Enter the unit number which you want assigned to the new unit ? 205
THE NEW UNIT MAY BE ADDED USING ONE OF THE FOLLOWING METHODS:
1. CLONE WILL PAUSE AFTER ALL MEMBERS HAVE BEEN COPIED. THE USER MUST
THEN PRESS RETURN TO CAUSE THE NEW UNIT TO BE ADDED.
2. AFTER ALL MEMBERS HAVE BEEN COPIED, THE UNIT WILL BE ADDED
AUTOMATICALLY.
                Under which above method should the new unit be added[]?1
DEVICES AVAILABLE FOR CLONE TARGETS:
DISK20200 (SIZE=832317)
DISK20400 (SIZE=832317)
DISK30100 (SIZE=832317)
                Use available device DISK20200(size=832317) for member
```

```
                    DISK10300(size=832317) (y,n) [y] ? y
MIRROR DISK10300 C_MA
SET C_MA NOPOLICY
SET C_MA MEMBERS=2
SET C_MA REPLACE=DISK220
DEVICES AVAILABLE FOR CLONE TARGETS:
DISK20400 (SIZE=832317)
DISK30100 (SIZE=832317)
                    Use available device DISK20400(size=832317) for member
                    DISK20000(size=832317) (y,n) [y] ? y
MIRROR DISK20000 C_MB
SET C_MB NOPOLICY
SET C_MB MEMBERS=2
SET C_MB REPLACE=DISK20400
COPY IN PROGRESS FOR EACH NEW MEMBER. PLEASE BE PATIENT...
.
.
COPY FROM DISK10300 TO DISK20200 IS 100% COMPLETE
COPY FROM DISK20000 TO DISK20400 IS 100% COMPLETE

                    Press RETURN when you want the new unit to be created
REDUCE  DISK20200 DISK20400
UNMIRROR DISK10300
UNMIRROR DISK20000
ADD MIRRORSET C_MA       DISK20200
ADD MIRRORSET C_MB       DISK20400
ADD STRIPESET C_ST1  C_MA C_MB
INIT C_ST1     NODESTROY CHUNK=128
ADD UNIT D205 C_ST1
D205 HAS BEEN CREATED. IT IS A CLONE OF D204.
CLONE - NORMAL TERMINATION
```

Backing Up Your Subsystem Configuration

Your controller stores information about your subsystem configuration in its nonvolatile memory. This information could be lost if the controller fails or when you replace a module in your subsystem.

You can avoid reconfiguring your subsystem manually by saving configuration information on one or more of your subsystem disks using the INITIALIZE SAVE_CONFIGURATION command. The controller updates the configuration information saved to disk whenever it changes. If the controller fails or you replace a module, you can easily restore your subsystem configuration from this information on the disks. Storing the configuration information uses a small amount of space on each device.

You do not need to store the configuration on all devices in the subsystem. You can use the INITIALIZE command without the SAVE_CONFIGURATION option for any devices on which you do not want to save the configuration.

You cannot use the SAVE_CONFIGURATION switch on TRANSPORTABLE disks.

Saving Subsystem Configuration Information to a Single Disk

You can choose to save your subsystem configuration information on a single disk.

Choose a disk on which to save the information by using the SAVE_CONFIGURATION switch when you initialize the disk with the INITIALIZE command. Use the following syntax:

```
INITIALIZE DISKnnn SAVE_CONFIGURATION
```

Saving Subsystem Configuration Information to Multiple Disks

You can save your subsystem configuration information to as many individual disks as you would like, but you must initialize each using the SAVE_CONFIGURATION switch. Use the following syntax for each:

```
INITIALIZE DISKnnn SAVE_CONFIGURATION
```

Saving Subsystem Configuration Information to a Storageset

You can save your subsystem configuration information to a storageset. The configuration information is duplicated on every disk that is a member of the storageset. Use the following syntax:

```
INITIALIZE storageset-name SAVE_CONFIGURATION
```

Displaying the Status of the Save Configuration Feature

You can use the SHOW THIS_CONTROLLER FULL command to find out if the save configuration feature is active and which devices are being used to store the configuration. The display includes a line indicating status and how many devices have copies of the configuration, as shown in the following example.

```
           SHOW THIS_CONTROLLER FULL
Controller:
    HSG80    (C) DEC CX00001905 Software R024F-0, Hardware 0000
    NODE_ID = XXXX-XXXX-XXXX-XXXX
    ALLOCATION_CLASS = 767
    SCSI_VERSION = SCSI-2
    Configured for dual-redundancy with ZG83301871
           In dual-redundant configuration
    Device Port SCSI address 7
    Time: NOT SET
    Command Console LUN is lun 0 (IDENTIFIER = 8888)
Host PORT_1:
    Reported PORT_ID = XXXX-XXXX-XXXX-XXXX
    PORT_1_PROFILE = PLDA
    PORT_1_TOPOLOGY = FABRIC (FABRIC UP)
    Address = 200113
    NOREMOTE_COPY
Host PORT_2:
    Reported PORT_ID = XXXX-XXXX-XXXX-XXXX
    PORT_2_PROFILE = PLDA
    PORT_2_TOPOLOGY = FABRIC (standby)
    Address = 200413
Cache
    256 megabyte write cache, version 0012
    Cache is GOOD
    Unflushed data in cache
    CACHE_FLUSH_TIMER = DEFAULT (10 seconds)
Mirrored Cache:
    Not enabled
Battery:
    FULLY CHARGED
    Expires:       WARNING: UNKNOWN EXPIRATION DATE!
    WARNING: AN UNKNOWN NUMBER OF DEEP DISCHARGES HAVE OCCURRED!
    NOCCACHE_UPS
Extended information:
    Terminal speed 19200 baud, eight bit, no parity, 1 stop bit
    Operation control: 00000000 Security state code: 6985
    Configuration backup enabled on 1 devices
```

The following example shows sample devices with the SAVE_CONFIGURATION switch enabled:

```
                                   $ SHOW DEVICES FULL
Name          Type                      Port Targ  Lun        Used by
------------------------------------------------------------------------------
DISK10000     disk                         1    0    0        S2
          DEC      RZ28M    (C) DEC 1003
        Switches:
          NOTRANSPORTABLE
          TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 10.00 MHZ negotiated)
        Size: 4108970 blocks
        Configuration being backed up on this container
DISK30300     disk                         3    3    0        S2
          DEC      RZ28M    (C) DEC 1003
        Switches:
          NOTRANSPORTABLE
          TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 10.00 MHZ negotiated)
        Size: 4108970 blocks
        Configuration being backed up on this container
```

Creating a Storageset Map

Configuring your subsystem will be easier if you know how the storagesets correspond to the disk drives in your subsystem. You can see this relationship by creating a storageset map like the one shown in Figure 4–10. This storageset map is for a subsystem that contains two RAIDsets, two mirrorsets, and three disk drives in the spareset. Each enclosure also has redundant power supplies.

Power Supply	***Spare***	***spare***	***Spare***	***R1***	**R1**	**R1**	**Power Supply**
Power Supply	***R2***	***R2***	***R2***	***R2***	**R2**	**R2**	**Power Supply**
Power Supply	***Spare***	***Spare***	***Spare***	***Spare***	**Spare**	**Spare**	**Power Supply**
Power Supply	***M1***	***M1***	***M2***	***M2***	**Spare**	**Spare**	**Power Supply**

Figure 4–10. Storageset Map

To create a storageset map:

1. Copy the template from Figure 4–11.
2. Establish a local or remote connection to one of the controllers in your subsystem.
3. Show the devices that are assigned to the controller. Use the following command:

   ```
   SHOW DEVICES
   ```

4. Locate each device assigned to the controller and record its location on your copy of the cabinet template. Use the following command:

   LOCATE *device_name*

   The LOCATE command causes the device's LED to flash continuously.

5. Turn off the LED, using the following command:

   ```
   LOCATE CANCEL
   ```

   The controller names each device based on its Port-Target-LUN (PTL) location. See the next section for details about the controller's PTL addressing convention. Repeat step 2 through step 4 for each controller or dual-redundant pair of controllers.

6. After you have mapped the devices to your cabinet template, create the storageset map by circling each group of disk drives that you want to combine into a storageset or put into the spareset. Label each group with a storageset name, for example: RAID1 for a RAIDset; Mirr1 for a mirrorset; and Stripe1 for a stripeset.

Power Supply							Power Supply
Power Supply							Power Supply
Power Supply							Power Supply
Power Supply							Power Supply

Figure 4–11. Storageset Map Template

Planning Partitions

Use partitions to divide a storageset or disk drive into smaller pieces, which can each be presented to the host as its own storage unit. Figure 4–12 shows the conceptual effects of partitioning a single-disk unit.

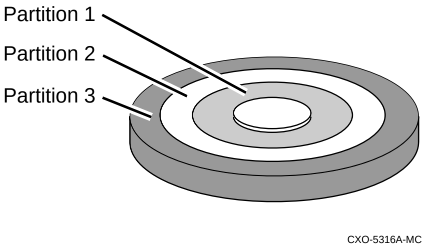


Figure 4–12. Partitioning a Single-Disk Unit

You can create up to eight partitions per disk drive, RAIDset, mirrorset, stripeset, or striped mirrorset. Each partition has its own unit number so that the host can send I/O requests to the partition just as it would to any unpartitioned storageset or device. Because partitions are separately-addressable storage units, you can partition a single storageset to service more than one user group or application.

Defining a Partition

Partitions are expressed as a percentage of the storageset or single disk unit that contains them. For mirrorsets and single disk units, the controller allocates the largest whole number of blocks that are equal to or less than the percentage you specify. For RAIDsets and stripesets, the controller allocates the largest whole number of stripes that are less than or equal to the percentage you specify. For stripesets, the stripe size = chunk size x number of members. For RAIDsets, the stripe size = chunk size x (number of members-1).

An unpartitioned storage unit has more capacity than a partition that uses the whole unit because each partition requires five blocks of administrative metadata. Thus, a single disk unit that contains one partition can store *n*-5 blocks of user or application data.

See “Guidelines for Partitioning Storagesets and Disk Drives,” page 4–35, for information on manually partitioning a storageset or single-disk unit.

Guidelines for Partitioning Storagesets and Disk Drives

Keep these points in mind as you plan your partitions:

- You can create up to eight partitions per storageset or disk drive.
- All of the partitions on the same storageset or disk drive must be addressed through the same controller port. This ensures a transparent failover of devices should one of the dual-redundant controllers fail.
- Partitions cannot be combined into storagesets. For example, you cannot divide a disk drive into three partitions, then combine those partitions into a RAIDset.
- Partitioned storagesets cannot function in multiple-bus failover dual-redundant configurations. Because they are not supported, you must delete your partitions before configuring the controllers for multiple-bus failover.
- Once you partition a container, you cannot unpartition it without reinitializing the container.
- Just as with storagesets, you do not have to assign unit numbers to partitions until you are ready to use them.

Choosing Switches for Storagesets and Devices

Depending upon the kind of storageset or device you are configuring, you can enable the following options or “switches”:

- RAIDset and mirrorset switches
- Initialize switches
- Unit switches
- Device switches

Enabling Switches

If you use StorageWorks Command Console to configure the device or storageset, you can set switches from the command console screens during the configuration process. The Command Console automatically applies them to the storageset or device. See *Getting Started with Command Console* for information about using the Command Console.

If you use CLI commands to configure the storageset or device manually, the procedures in “Configuring Storagesets with CLI Commands,” page 6–2, indicate when and how to enable each switch.

Changing Switches

You can change the RAIDset, mirrorset, device, and unit switches at any time. See “Changing Switches,” page 4–36, for information about changing switches for a storageset or device.

You cannot change the initialize switches without destroying the data on the storageset or device. These switches are integral to the formatting and can only be changed by re-initializing the storageset. (Initializing a storageset is similar to formatting a disk drive; all of the data is destroyed during this procedure.)

RAIDset Switches

You can enable the following switches to control how a RAIDset behaves to ensure data availability:

- Replacement policy
- Reconstruction policy
- Membership

Replacement Policy

Specify a replacement policy to determine how the controller replaces a failed disk drive:

- POLICY=BEST_PERFORMANCE (default) puts the failed disk drive in the failedset then tries to find a replacement (from the spareset) that is on a different device port than the remaining operational disk drives. If more than one disk drive meets this criterion, this switch selects the drive that also provides the best fit.
- POLICY=BEST_FIT puts the failed disk drive in the failedset then tries to find a replacement (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 meets this criterion, this switch selects one that also provides the best performance.
- NOPOLICY puts the failed disk drive in the failedset and does not replace it. The storageset operates with less than the nominal number of members until you specify a replacement policy or manually replace the failed disk drive.

Reconstruction Policy

Specify the speed with which the controller reconstructs the data from the remaining operational disk drives and writes it to a replacement disk drive:

- RECONSTRUCT=NORMAL (default) balances the overall performance of the subsystem against the need for reconstructing the replacement disk drive.
- RECONSTRUCT=FAST gives more resources to reconstructing the replacement disk drive, which may reduce the subsystem's overall performance during the reconstruction task.

Membership

Indicate to the controller that the RAIDset you are adding is either complete or reduced, which means it is missing one of its members:

- NOREDUCED (default) indicates to the controller that all of the disk drives are present for a RAIDset.
- REDUCED lets you add a RAIDset that is missing one of its members. For example, if you dropped or destroyed a disk drive while moving a RAIDset, you could still add it to the subsystem by using this switch.

Mirrorset Switches

You can enable the following switches to control how a mirrorset behaves to ensure data availability:

- Replacement policy
- Copy speed
- Read source

Replacement Policy

Specify a replacement policy to determine how the controller replaces a failed disk drive:

- POLICY=BEST_PERFORMANCE (default) puts the failed disk drive in the failedset then tries to find a replacement (from the spareset) that is on a different device port than the remaining operational disk drives. If more than one disk drive meets this criterion, this switch selects the drive that also provides the best fit.
- POLICY=BEST_FIT puts the failed disk drive in the failedset then tries to find a replacement (from the spareset) that most closely matches the size of the remaining, operational disk drives. If more than one disk drive meets this criterion, this switch selects the one that also provides the best performance.
- NOPOLICY puts the failed disk drive in the failedset and does not replace it. The storageset operates with less than the nominal number of members until you specify a replacement policy or manually replace the failed disk drive.

Copy Speed

Specify a copy speed to determine the speed with which the controller copies the data from an operational disk drive to a replacement disk drive:

- COPY=NORMAL (default) balances the overall performance of the subsystem against the need for reconstructing the replacement disk drive.

- COPY=FAST allocates more resources to reconstructing the replacement disk drive, which may reduce the subsystem’s overall performance during the reconstruction task.

Read Source

Specify the read source to determine how the controller reads data from the members of a mirrorset:

- READ_SOURCE=LEAST_BUSY (default) forces the controller to read data from the “normal” or operational member that has the least-busy work queue. If multiple disks have equally short queues, the controller queries normal local disks for each read request as it would when READ_SOURCE=ROUND_ROBIN is specified. If no normal local disk exists, then the controller will query any remote disks if available.
- READ_SOURCE=ROUND_ROBIN forces the controller to read data sequentially from all “normal” or operational members in a mirrorset. For example, in a four-member mirrorset (A, B, C, and D), the controller reads from A, then B, then C, then D, then A, then B, and so forth. No preference is given to any member. If no normal local disk exists, then the controller will query any remote disks if available.
- READ_SOURCE=DISK*nnnnn* forces the controller to always read data from a particular “normal” or operational member. If the specified member fails, the controller reads from the least busy member.

Device Switches

When you add a disk drive or other storage device to your subsystem, you can enable the following switches:

- Transportability
- Transfer rate

Transportability

Indicate whether a disk drive is transportable when you add it to your subsystem:

- NOTRANSPORTABLE disk drives (default) are marked with StorageWorks-exclusive metadata. This metadata supports the error-detection and recovery methods that the controller uses to ensure data availability. Disk drives that contain this metadata cannot be used in non-StorageWorks subsystems.

  Consider these points when using the NOTRANSPORTABLE switch:

  ❏ When you bring non-transportable devices from another subsystem to your controller subsystem, add the device to your configuration using the ADD command. Do not initialize the device, or you will reset and destroy any forced error information contained on the device.

  ❏ When you add units, the controller software verifies that the disks or storagesets within the units contain metadata. To determine whether a disk or storageset contains metadata, try to create a unit from it. This causes the controller to check for metadata. If no metadata is present, the controller displays a message; initialize the disk or storageset before adding it.

- TRANSPORTABLE disk drives can be used in non-StorageWorks subsystems. Transportable disk drives can be used as single-disk units in StorageWorks subsystems as well as disk drives in other systems. They cannot be combined into storagesets in a StorageWorks subsystem.

  TRANSPORTABLE is especially useful for moving a disk drive from a workstation into your StorageWorks subsystem. When you add a disk drive as transportable, you can configure it as a single-disk unit and access the data that was previously saved on it.

Transportable devices have these characteristics:

❏ Can be interchanged with any SCSI interface that does not use the device metadata, for example, a PC.

❏ Cannot have write-back caching enabled.

❏ Cannot be members of a shadowset, storageset, or spareset.

❏ Do not support forced errors.

Consider these points when using the TRANSPORTABLE switch:

❏ Before you move devices from the subsystem to a foreign subsystem, delete the units and storagesets associated with the device and set the device as transportable. Initialize the device to remove any metadata.

❏ When you bring foreign devices into the subsystem with customer data follow this procedure:

a.Add the disk as a transportable device. Do not initialize it.

b.Copy the data the device contains to another nontransportable unit.

c.Initialize the device again after resetting it as nontransportable. Initializing it now places metadata on the device.

❏ Storagesets cannot be made transportable. Specify NOTRANSPORTABLE for all disks used in RAIDsets, stripesets, and mirrorsets.

❏ Do not keep a device set as transportable on a subsystem. The unit attached to the device loses forced error support which is mandatory for data integrity on the entire array.

Device Transfer Rate

Specify a transfer rate that the controller uses to communicate with the device. Use one of these switches to limit the transfer rate to accommodate long cables between the controller and a device, such as a tape library. Use one of the following values:

- TRANSFER_RATE_REQUESTED=20MHZ (default)
- TRANSFER_RATE_REQUESTED=10MHZ
- TRANSFER_RATE_REQUESTED=5MHZ
- TRANSFER_RATE_REQUESTED=ASYNCHRONOUS

Initialize Switches

You can enable the following kinds of switches to affect the format of a disk drive or storageset:

- Chunk size (for stripesets and RAIDsets only)
- Save configuration
- Destroy/Nodestroy

After you initialize the storageset or disk drive, you cannot change these switches without reinitializing the storageset or disk drive.

Chunk Size

Specify a chunk size to control the stripesize used for RAIDsets and stripesets:

- CHUNKSIZE=DEFAULT lets the controller set the chunk size based on the number of disk drives (d) in a stripeset or RAIDset. If d $\leq$ 9 then chunk size = 256. If d > 9 then chunk size = 128.
- CHUNKSIZE=*n* lets you specify a chunk size in blocks. The relationship between chunk size and request size determines whether striping increases the request rate or the data-transfer rate.

See “INITIALIZE,” page 7–62 for more information on setting the Chunk Size.

IMPORTANT: While a storageset may be initialized with a user-selected chunk size, it is recommended that only the default value be used. The default value is chosen to produce optimal performance for a wide variety of loads. The use of a chunk size less than 128 blocks (64K) is **strongly discouraged**. There are almost no customer loads for which small chunk sizes are of value and, in almost all cases, selecting a small chunk size will **severely** degrade the performance of the storageset **and** the controller as a whole. Use of a small chunk size on any storageset can result in **severe** degradation of overall system performance.

Increasing the Request Rate

A large chunk size (relative to the average request size) increases the request rate by allowing multiple disk drives to respond to multiple requests. If one disk drive contains all of the data for one request, then the other disk drives in the storageset are available to handle other requests. Thus, in principle, separate I/O requests can be handled in parallel, thereby increasing the request rate. This concept is shown in Figure 4–13.

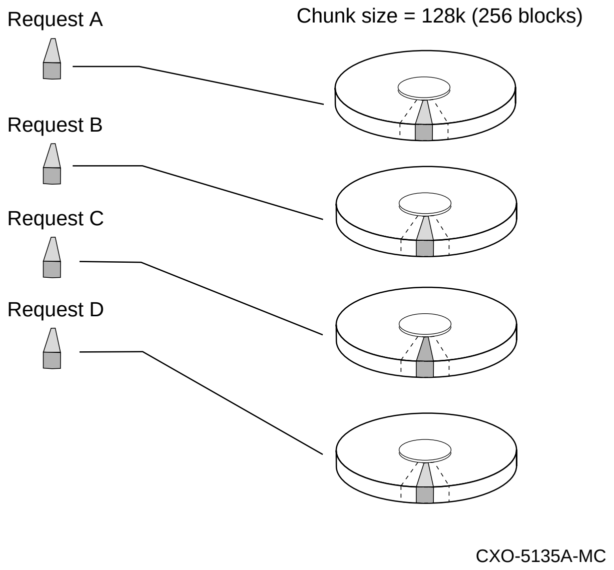


Figure 4–13. Chunk Size Larger than the Request Size

Applications such as interactive transaction processing, office automation, and file services for general timesharing tend to require high I/O request rates.

Large chunk sizes also tend to increase the performance of random reads and writes. It is recommended that you use a chunk size of 10 to 20 times the average request size, rounded up to the nearest multiple of 64. In general, a chunk size of 256 works well for UNIX® systems; 128 works well for OpenVMS™ systems.

Increasing the Data Transfer Rate

A small chunk size relative to the average request size increases the data transfer rate by allowing multiple disk drives to participate in one I/O request. This concept is shown in Figure 4–14.

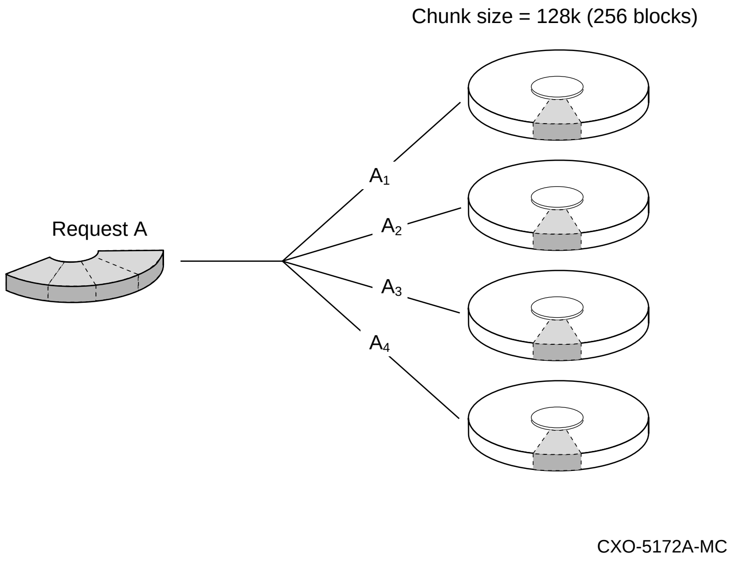


Figure 4–14. Chunk Size Smaller than the Request Size

Applications such as CAD, image processing, data collection and reduction, and sequential file processing tend to require high data-transfer rates.

Increasing Sequential Write Performance

For stripesets (or striped mirrorsets), use a large chunk size relative to the I/O size to increase the sequential write performance. A chunk size of 256 generally works well.

Chunk size does not significantly affect sequential read performance.

Maximum Chunk Size for RAIDsets

Do not exceed the chunk sizes shown in Table 4–5 for a RAIDset. (The maximum chunk size is derived by 2048/(*d* – 1) where *d* is the number of disk drives in the RAIDset.)

Table 4–5. Maximum Chunk Sizes for a RAIDset

RAIDset Size	Maximum Chunk Size
3 members	1024 blocks
4 members	682 blocks
5 members	512 blocks
6 members	409 blocks
7 members	341 blocks
8 members	292 blocks
9 members	256 blocks
10 members	227 blocks
11 members	204 blocks
12 members	186 blocks
13 members	170 blocks
14 members	157 blocks

Save Configuration

Indicate whether to save the subsystem’s configuration on the storage unit when you initialize it:

- NOSAVE_CONFIGURATION (default) means that the controller stores the subsystem’s configuration in its nonvolatile memory only. Although this is generally secure, the configuration could be jeopardized if the controller fails. For this reason, you should initialize at least one of your storage units with the SAVE_CONFIGURATION switch enabled.

- SAVE_CONFIGURATION allows the controller to use 256K of each device in a storage unit to save the subsystem’s configuration. The controller saves the configuration every time you change it or add a patch to your controller. If the controller should fail, you can recover your latest configuration from the storage unit rather than rebuild it from scratch.

  The save configuration option saves the following information:

  ❏ All configuration information normally saved when you restart your controller except, the controller serial number, product ID number, vendor ID number, and any manufacturing fault information.
  ❏ Patch information

  The save configuration option does not save the following information:

  ❏ Software or hardware upgrades
  ❏ Inter-platform conversions

Considerations for Saving the Configuration

- Use the SET FAILOVER COPY= command to restore configuration information in a replacement controller. See “Saving Configuration Information in Dual-Redundant Configurations,” page 4–48 for details.
- Do not remove and replace disk devices between the time you save and restore your configuration. This is particularly important for devices that you migrate from another system. The controller could recover and use the wrong configuration information on your subsystem.
- Save your subsystem configuration as soon as possible after removing and replacing any disk devices in your subsystem. This ensures that the devices always contain the latest, valid information for your system.
- When you incorporate a spare into a storageset that you initialized with the INITIALIZE SAVE_CONFIGURATION command, the controller reserves space on the spare for configuration information. The controller updates this information when the configuration changes.
- You cannot use a storageset that contains user data to save your subsystem configuration unless you backup and restore the user data.

- If you previously configured storagesets with the SAVE_CONFIGURATION option, you do not need to initialize them again after you reconfigure your devices with a new controller.
- When you replace a controller, make sure the replacement controller does not contain any configuration data. If the controller is not new, use the CONFIGURATION RESET command to purge any existing configuration. If you do not take this precaution, you can lose configuration data if non-volatile memory changes.

Saving Configuration Information in Dual-Redundant Configurations

If you decide to use the SAVE_CONFIGURATION switch in a dual-redundant configuration, keep these points in mind:

- The controller-unique data for both controllers is saved.
- Saved configuration data from a dual controller configuration can be used to restore the configuration to a replacement controller. However, if one controller in a dual configuration is replaced, use the SET FAILOVER COPY= command to restore the configuration.
- When replacing both controllers, you can replace the first and restart it alone by holding in port button 6 and simultaneously pressing the reset button on the controller's operator control panel. (This controller picks up any previously saved configuration data on disk and uses it to set up the subsystem configuration.) Replace the second controller using the SET FAILOVER COPY= command to copy the configuration information from the operating controller.
- Both controllers update the saved data; each writes to only those devices currently preferred to it. This prevents conflicting data transfer.

See "INITIALIZE," page 7–62 for more information on saving the configuration.

Destroy/Nodestroy

Specify whether to destroy or retain the user data and metadata when you initialize a disk drive that has been previously used in a mirrorset or as a single-disk unit.

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

- DESTROY (default) overwrites the user data and forced-error metadata on a disk drive when it is initialized.
- NODESTROY preserves the user data and forced-error metadata when a disk drive is initialized. Use NODESTROY to create a single-disk unit from any disk drive that has been used as a member of a mirrorset. The NODESTROY switch is not valid for RAIDsets and single-disk configurations.
- See “REDUCE,” page 7–77 for information on removing disk drives from a mirrorset.

See “INITIALIZE,” page 7–62 for more information on saving the configuration.

Chapter 5
Configuring an HSG80 Array Controller

This chapter explains how to configure an HSG80 array controller and the modules that support its operation in a StorageWorks subsystem.

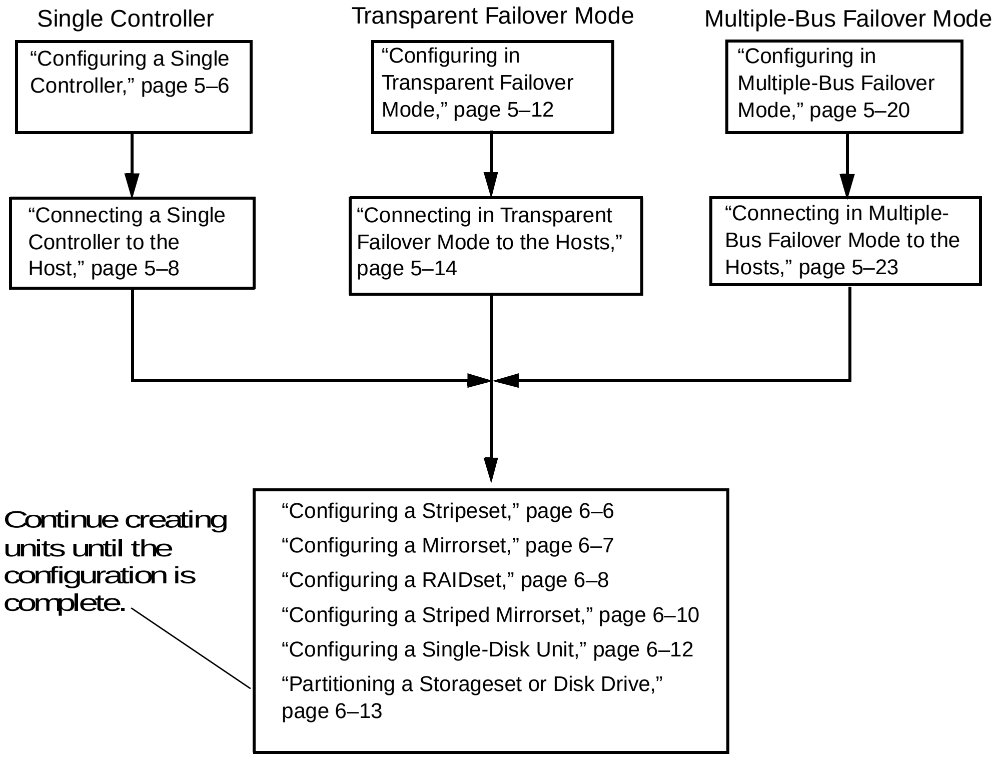

Single Controller
Transparent Failover Mode
Multiple-Bus Failover Mode
"Configuring a Single Controller," page 5–6
"Configuring in Transparent Failover Mode," page 5–12
"Configuring in Multiple-Bus Failover Mode," page 5–20
"Connecting a Single Controller to the Host," page 5–8
"Connecting in Transparent Failover Mode to the Hosts," page 5–14
"Connecting in Multiple-Bus Failover Mode to the Hosts," page 5–23
Continue creating units until the configuration is complete.
"Configuring a Stripeset," page 6–6
"Configuring a Mirrorset," page 6–7
"Configuring a RAIDset," page 6–8
"Configuring a Striped Mirrorset," page 6–10
"Configuring a Single-Disk Unit," page 6–12
"Partitioning a Storageset or Disk Drive," page 6–13

Figure 5–1. Configuration Process

Introduction

Unless you specifically requested a preconfigured subsystem, you will have to configure your controller and its subsystem before you can use them. Figure 5–1 shows the configuration process of how to configure the controller followed by configuring storagesets and units. To help you in configuring, you can use the following books in your kit:

- *Quick Setup Guide*
- *Hardware Configuration Poster for HSG80*
- *RA8000/ESA12000 Installation Reference Guide* for your operating system

Configuration Rules

Before you configure your controller, review these configuration rules and ensure your planned configuration meets the requirements and conditions.

- Maximum 127 visible LUNs and one Command Console LUN (CCL)
- Maximum 512-GB LUN capacity
- Maximum 72 physical devices
- Maximum 20 RAID-5 storagesets
- Maximum 30 RAID-5 and RAID-1 storagesets
- Maximum 45 RAID-5, RAID-1, and RAID-0 storagesets
- Maximum 8 partitions of a storageset or individual disk
- Maximum 6 members per mirrorset
- Maximum 14 members per RAID-5 storageset
- Maximum 24 members per Stripeset
- Maximum 48 physical devices per striped mirrorset

Configuring and Cabling an HSG80 Array Controller

You can use these procedures to configure and cable your controller in one of three configurations:

- "Configuring a Single Controller," page 5–6
  - ❏ "Connecting a Single Controller to the Host," page 5–8
- "Configuring in Transparent Failover Mode," page 5–12
  - ❏ "Connecting in Transparent Failover Mode to the Hosts," page 5–14
- "Configuring in Multiple-Bus Failover Mode," page 5–20
  - ❏ "Connecting in Multiple-Bus Failover Mode to the Hosts," page 5–23

When configuring, you will use the CLI commands in Chapter 7, "CLI Commands."

Tips

As you configure the controller, keep these points in mind:

- Cabling—You can either configure the controller with a maintenance port cable and no optical host bus cables connected to the host, or you can configure with the optical host bus cables connected:

  ❏ SWCC—You can configure the controller using the SWCC graphical user interface. See the *Command Console for HSG80 Solutions Getting Started Guide* for more details.

  ❏ CLI Commands—You can configure the controller with CLI commands using a terminal program.

  NOTE: Eventually, you must connect the optical host bus cables between the controllers and the host. You can connect the optical host bus cables after configuring the controllers using a maintenance port cable.

- EMU Alarm Control Switch—If the EMU alarm goes off, simply press it to stop it. Check the error message to verify the type of error condition.
- CLI Commands—You can use the following CLI commands frequently while configuring the controller:

  ❏ **CLEAR_ERRORS CLI**—Clears all error messages so you can type without the messages being repeated. Make sure you read each error message before clearing it.

  ❏ **SHOW THIS_CONTROLLER FULL** and **SHOW OTHER_CONTROLLER FULL**—Lists the full information about the controller.

Configuring a Single Controller

Follow these steps to configure an HSG80 array controller:

NOTE: This procedure has been written for first-time configuring. However, you can adapt the procedure when you reconfigure the controllers. For replacing and upgrading the controllers, see the *HSG80 Array Controller ACS Version 8.3 and 8.4 Maintenance and Service Guide*.

1. Use the power verification and addressing (PVA) module ID switch to set the SCSI ID number for the BA370 rack-mountable enclosure.

   See “Setting the PVA Module ID Switch,” page 5–29, for details about PVA switch settings.

2. Remove the program card ESD cover, and insert the controller’s program card. Replace the ESD cover.

3. Turn on the power to the enclosure by turning on the ac input boxes.

4. Establish a local connection to the controller.

   See “Establishing a Local Connection to the Controller,” page 5–33, for details about creating a local connection.

5. If the controller reports a node ID of all zeros (0000-0000-0000-0000), set the subsystem worldwide name (node ID) to the worldwide name that came with your subsystem. See “Restoring Worldwide Names (Node IDs),” page 3–17.

6. Set the port topology for each port.

   ```
   SET THIS_CONTROLLER PORT_1_TOPOLOGY=FABRIC
   ```

   ```
   SET THIS_CONTROLLER PORT_2_TOPOLOGY=FABRIC
   ```

7. Set the time on the controller, which provides a baseline for replacing the external battery, using the following command:

   ```
   SET THIS_CONTROLLER TIME=DD-MMM-YYYY:HH:MM:SS
   ```

   NOTE: All values are numbers, except for MMM, which uses the first three letters of the month. When setting hours, minutes, and seconds—HH, MM, and SS—you must use 24-hour time.

8. Set up the battery discharge timer:

   a. Start Frutil using the following command:

      ```
      RUN FRUTIL
      ```

   Frutil displays the following:

   ```
   Do you intend to replace this controller’s cache battery? Y/N
   ```

   b. Enter Y(es).

   Frutil displays a three-step procedure and prompts you to press return.

   c. Press return.

9. Enter any other optional CLI commands for your configuration.

   See “Optional Steps,” page 5–29 and Chapter 7, “CLI Commands.”

10. Display details about the controller you configured. Use the following command:

    ```
    SHOW THIS_CONTROLLER FULL
    ```

See “SHOW,” page 7–133.

IMPORTANT: If you are using a PC or terminal with the maintenance port cable during configuring and have not yet cabled the configuration, you must connect the optical host bus cables in the desired configuration as shown in the next section.

Connecting a Single Controller to the Host

For a detailed description of the controller and cables, see Chapter 1, “General Description.”

Using Two Switches

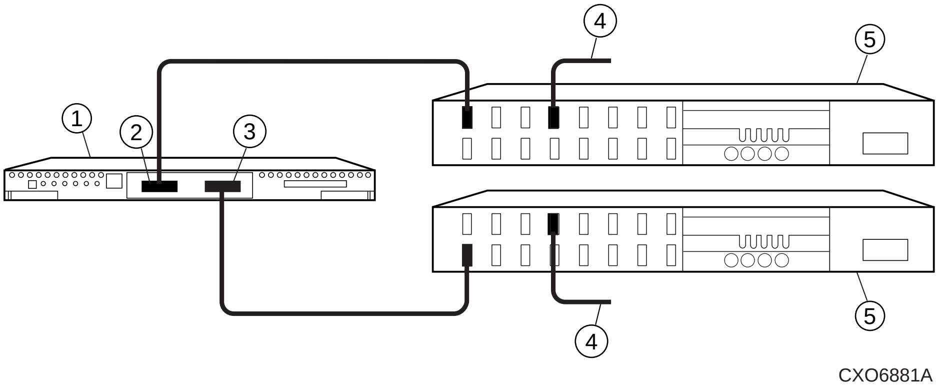


Figure 5–2. Example Cabling for a Single Configuration using Two Switches

Table 5–1. Key to Figure 5–2 Example Cabling for a Single Configuration using Two Switches

Item	Description
①	HSG80 controller
②	Host port 1 connection to switch port 0
③	Host port 2 connection to switch port 1
④	Host bus connection from the switch ports to the host adapters
⑤	Switches

Before cabling, note the following:

- Ports
  - ❏ The active ports are port 1 and port 2, controller A.
  - ❏ Standby ports do not exist in single-controller configurations when a failover mode has not been selected.
  - ❏ Once logical units are added, the active ports will have access to the units.
  - ❏ For a more detailed explanation of active versus standby ports and how units are assigned on each port, see “Assigning Unit Numbers in Transparent Failover Mode,” page 3–10 and “Assigning Unit Numbers in Multiple-Bus Failover Mode,” page 3–11.
- Cables connecting into the switch
  - ❏ The order of how you connect cables into the switch does not matter, which differs from multiple-bus failover connections. However, to limit confusion, start with switch port 0 and increment the cables into the next switch port as you make connections.

Use the following steps to connect a single, nonredundant controller to the host using two switches:

1. Configure the controller. See “Configuring a Single Controller,” page 5–6.
2. Ensure the host and adapter are also configured. See the host user’s guide for details.
3. Connect the cable between host port 1 on the controller and the switch port on the switch.
4. Connect the cable between host port 2 on the controller and the switch port on the switch.
5. Connect the host bus cable between the switch and the host adapter.
6. Route and tie the cables as desired.
7. Turn on the subsystem.

   Some operating systems may require you to start the host to see the devices attached to the new controller. See the *HSG80 Array Controller ACS Version 8.X Release Notes* for your operating system.

Using One Switch

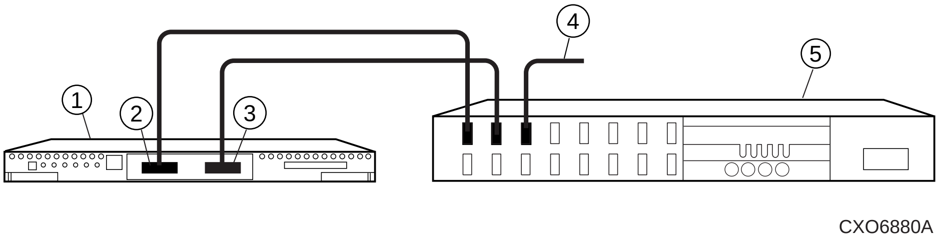


Figure 5–3. Example Cabling for a Single Configuration using One Switch

Table 5–2. Key to Figure 5–3 Example Cabling for a Single Configuration using One Switch

Item	Description
①	HSG80 controller
②	Host port 1 connection to switch port 0
③	Host port 2 connection to switch port 2
④	Host bus connection from the switch port to the host adapter
⑤	Switch

Before cabling, note the following:

- Ports
  - ❏ The active ports are port 1 and port 2, controller A.
  - ❏ Standby ports do not exist in single-controller configurations when a failover mode has not been selected.
  - ❏ Once logical units are added, the active ports will have access to the units.
  - ❏ For a more detailed explanation of active versus standby ports and how units are assigned on each port, see “Assigning Unit Numbers in Transparent Failover Mode,” page 3–10 and “Assigning Unit Numbers in Multiple-Bus Failover Mode,” page 3–11.
- Cables connecting into the switch
  - ❏ The order of how you connect cables into the switch does not matter, which differs from multiple-bus failover connections. However, to limit confusion, start with switch port 0 and increment the cables into the next switch port as you make connections.

Use the following steps to connect a single, nonredundant controller to the host using one switch:

1. Configure the controller. See “Configuring a Single Controller,” page 5–6.
2. Ensure the host and adapter are also configured. See the host user’s guide for details.
3. Connect the cable between host port 1 on the controller and the switch port on the switch.
4. Connect the cable between host port 2 on the controller and the switch port on the switch.
5. Connect the host bus cable between the switch and the host adapter.
6. Route and tie the cables as desired.
7. Turn on the subsystem.

   Some operating systems may require you to start the host to see the devices attached to the new controller. See the *HSG80 Array Controller ACS Version 8.X Release Notes* for your operating system.

Configuring in Transparent Failover Mode

You can configure dual-redundant controllers to your hosts with one or more switches.

Follow these steps to configure an HSG80 array controller:

> **NOTE:** This procedure has been written for first-time configuring. However, you can adapt the procedure when you reconfigure the controllers. For replacing and upgrading the controllers, see the *HSG80 Array Controller ACS Version 8.3 and 8.4 Maintenance and Service Guide*.

1. Use the power verification and addressing (PVA) module ID switch to set the SCSI ID number for the BA370 rack-mountable enclosure.

   See “Setting the PVA Module ID Switch,” page 5–29, for details about PVA switch settings.

2. Remove the program card ESD cover, and insert the controller’s program card. Replace the ESD cover.
3. Turn on the power to the enclosure by turning on the ac input boxes.
4. Establish a local connection to the controller.

   See “Establishing a Local Connection to the Controller,” page 5–33, for details about creating a local connection.

5. If the controller reports a node ID of all zeros (0000-0000-0000-0000), set the subsystem worldwide name (node ID) to the worldwide name that came with your subsystem. See “Restoring Worldwide Names (Node IDs),” page 3–17.
6. Set the port topology for each port.

   ```
   SET THIS_CONTROLLER PORT_1_TOPOLOGY=FABRIC
   ```

   ```
   SET THIS_CONTROLLER PORT_2_TOPOLOGY=FABRIC
   ```

7. Put “this controller” into transparent failover mode. Use the following syntax:

   ```
   SET FAILOVER COPY = THIS_CONTROLLER
   ```

   The “other controller” inherits “this controller’s” configuration, then restarts.Wait for it to return to normal operation before continuing.

   See details about failover modes in “Selecting a Failover Mode,” page 2–5.

8. If you want to enable mirrored write-back cache in a failover configuration, enter this command:

   ```
   SET THIS_CONTROLLER MIRRORED_CACHE
   ```

   NOTE: Both controllers restart when you set mirror mode. This process can take up to 5 minutes, depending on the amount of data that must be flushed from the cache.

   For details about caching techniques, see CACHING TECHNIQUES, chapter 2.

9. Set the time on the controller, which provides a baseline for replacing the external battery, using the following command:

   ```
   SET THIS_CONTROLLER TIME=DD-MMM-YYYY:HH:MM:SS
   ```

   NOTE: All values are numbers, except for MMM, which uses the first three letters of the month. When setting hours, minutes, and seconds—HH, MM, and SS—you must use 24-hour time.

10. Set up the battery discharge timer:

    a. Start Frutil using the following command:

       ```
       RUN FRUTIL
       ```

    Frutil displays the following:

    ```
    Do you intend to replace this controller’s cache battery? Y/N
    ```

    b. Enter Y(es).

    Frutil displays a three-step procedure and prompts you to press return.

    c. Press return.

11. Enter any other optional CLI commands for your configuration.

    See “Optional Steps,” page 5–29 and Chapter 7, “CLI Commands.”

12. Display details about the controller you configured. Use the following command:

    ```
    SHOW THIS_CONTROLLER FULL
    ```

    See “SHOW,” page 7–133.

IMPORTANT: If you are using a PC or terminal with the maintenance port cable during configuring and have not yet cabled the configuration, you must connect the optical host bus cables in the desired configuration as shown in the next section.

Connecting in Transparent Failover Mode to the Hosts

You can connect dual-redundant controllers to your hosts with one or more switches.

Using Two Switches

For a detailed description of the controller and cables, see Chapter 1, “General Description.”

Figure 5–4 shows how to connect dual-redundant controllers to the host using two switches.

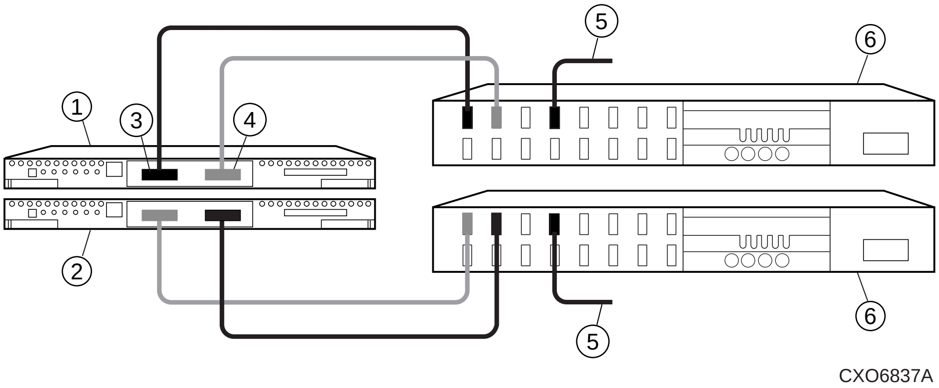


Figure 5–4. Example Cabling in Transparent Failover using Two Switches

Table 5–3. Key to Figure 5–4 Example Cabling in Transparent Failover using Two Switches

Item	Description
①	HSG80 controller A
②	HSG80 controller B
③	Host port 1 (active) connection from controller A to switch port 0 Host port 1 (standby) connection from controller B to switch port 0
④	Host port 2 (standby) connection from controller A to switch port 2 Host port 2 (active) connection from controller B to switch port 2
⑤	Host bus connections to the host adapters
⑥	Switches

The following steps show how to connect a pair of dual-redundant controllers to the host using two switches.

Before cabling, note the following:

- Ports
  - ❑ The active ports are port 1, controller A and port 2, controller B.
  - ❑ The standby ports are port 2, controller A and port 1, controller B.
  - ❑ Once logical units are added, the active ports will have access to the units.
  - ❑ For a more detailed explanation of active vs. standby ports and how units are assigned on each port, see “Assigning Unit Numbers in Transparent Failover Mode,” page 3–10.
- Cables connecting into the switch
  - ❑ The cables going from port 1 on both controllers must go into different switches. Similarly, the cables from port 2 on both controllers must go into different switches.
  - ❑ To limit confusion on the switch port connections, start with switch port 0 and increment the cables into the next switch port as you make connections.

1. Configure the controller. See “Configuring in Transparent Failover Mode,” page 5–12.
2. Ensure the host and adapter are also configured. See the host user’s guide for details.
3. Connect the cables between host port 1 on both controllers and the switch ports on both switches.

   Make sure you connect the host port 1 cables into different switches.
4. Connect the cables between host port 2 on both controllers and the switch ports on both switches.

   Make sure you connect the host port 2 cables into different switches.

5. Connect the host bus cable between the switches and the host adapters.

IMPORTANT: Figure 5–4 shows two cables for a host connection. The cables can go either to the same host or they can go to separate hosts.

6. Route and tie the cables as desired.
7. Turn on the subsystem.

   Some operating systems may require you to start the host to see the devices attached to the new controller. See the *HSG80 Array Controller ACS Version 8.X Release Notes* for your operating system.

Using One Switch

For a detailed description of the controller and cables, see Chapter 1, “General Description.”

Figure 5–5 shows how to connect dual-redundant controllers to the host using one switch.

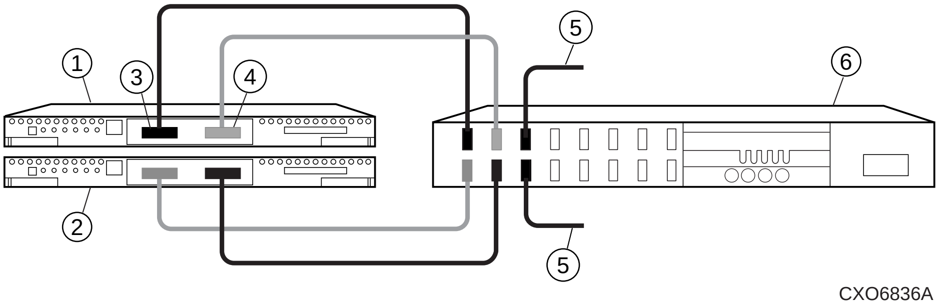


Figure 5–5. Example Cabling for Transparent Failover with One Switch

Table 5–4. Figure 5–5 Example Cabling for Transparent Failover with One Switch

Item	Description
①	HSG80 controller A
②	HSG80 controller B
③	Host port 1 (active) connection from controller A to switch port 0 Host port 1 (standby) connection from controller B to switch port 1
④	Host port 2 (standby) connection from controller A to switch port 2 Host port 2 (active) connection from controller B to switch port 3
⑤	Host bus connections to the host adapters
⑥	Switch

The following steps show how to connect a pair of dual-redundant controllers to the host using one switch.

Before cabling, note the following:

- Ports
  - ❏ The active ports are port 1, controller A and port 2, controller B.
  - ❏ The standby ports are port 2, controller A and port 1, controller B.
  - ❏ Once logical units are added, the active ports will have access to the units.
  - ❏ For a more detailed explanation of active vs. standby ports and how units are assigned on each port, see “Assigning Unit Numbers in Transparent Failover Mode,” page 3–10.
- Cables connecting into the switch
  - ❏ To limit confusion on the switch port connections, start with switch port 0 and increment the cables into the next switch port as you make connections.

1. Configure the controller. See “Configuring in Transparent Failover Mode,” page 5–12.
2. Ensure the host and adapter are also configured. See the host user’s guide for details.
3. Connect the cables between host port 1 on both controllers and the switch ports on the switch.
4. Connect the cables between host port 2 on both controllers and the switch ports on the switch.
5. Connect the host bus cable between the switch and the host adapters.

   IMPORTANT: Figure 5–4 shows two cables for a host connection. The cables can go either to the same host or they can go into separate hosts.

6. Route and tie the cables as desired.
7. Turn on the subsystem.

   Some operating systems may require you to start the host to see the devices attached to the new controller. See the *HSG80 Array Controller ACS Version 8.X Release Notes* for your operating system.

Configuring in Multiple-Bus Failover Mode

IMPORTANT: If you are using HSG80 ACS V8.4P code, see the *Data Replication Manager HSG80 ACS Version 8.4P Guide to Operations* for cabling yourcontroller into the subsystem.

Follow these steps to configure an HSG80 array controller:

NOTE: This procedure has been written for first-time configuring. However, you can adapt the procedure when you reconfigure the controllers. For replacing and upgrading the controllers, see the *HSG80 Array Controller ACS Version 8.3 and 8.4 Maintenance and Service Guide*.

1. Use the power verification and addressing (PVA) module ID switch to set the SCSI ID number for the BA370 rack-mountable enclosure.

   See “Setting the PVA Module ID Switch,” page 5–29, for details about PVA switch settings.

2. Remove the program card ESD cover, and insert the controller’s program card. Replace the ESD cover.

3. Turn on the power to the enclosure by turning on the ac input boxes.

4. Establish a local connection to the controller.

   See “Establishing a Local Connection to the Controller,” page 5–33, for details about creating a local connection.

5. If the controller reports a node ID of all zeros (0000-0000-0000-0000), set the subsystem worldwide name (node ID) to the worldwide name that came with your subsystem. See “Restoring Worldwide Names (Node IDs),” page 3–17.

6. On OpenVMS hosts only, change the SCSI version to SCSI-3 mode:

   ```
   SET THIS_CONTROLLER SCSI_VERSION=SCSI-3
   ```

   IMPORTANT: Once you change to the SCSI-3 version in OpenVMS, you must use a special number called a *controller identifier*, which makes the controller and the command console LUN visible to the host. See the *IDENTIFIER* switch with the command "SET controller," page 7–91.

7. Set the port topology for each port.

   ```
   SET THIS_CONTROLLER PORT_1_TOPOLOGY=FABRIC
   ```

   ```
   SET THIS_CONTROLLER PORT_2_TOPOLOGY=FABRIC
   ```

8. Put "this controller" into multiple-bus failover mode. Use the following syntax:

   ```
   SET MULTIBUS_FAILOVER COPY = THIS_CONTROLLER
   ```

   The "other controller" inherits "this controller's" configuration, then restarts.Wait for it to return to normal operation before continuing.

   See details about failover modes in "Selecting a Failover Mode," page 2–5.

9. If you want to enable mirrored write-back cache in a failover configuration, enter this command:

   ```
   SET THIS_CONTROLLER MIRRORED_CACHE
   ```

   NOTE: Both controllers restart when you set mirror mode. This process can take up to 5 minutes, depending on the amount of data that must be flushed from the cache.

   For details about caching techniques, see CACHING TECHNIQUES, chapter 2.

10. Set the time on the controller, which provides a baseline for replacing the external battery, using the following command:

    ```
    SET THIS_CONTROLLER TIME=DD-MMM-YYYY:HH:MM:SS
    ```

    NOTE: All values are numbers, except for MMM, which uses the first three letters of the month. When setting hours, minutes, and seconds—HH, MM, and SS—you must use 24-hour time.

11. Set up the battery discharge timer:

    a. Start Frutil using the following command:

    ```
    RUN FRUTIL
    ```

    Frutil displays the following:

    ```
    Do you intend to replace this controller’s cache battery? Y/N
    ```

    b. Enter Y(es).

    Frutil displays a three-step procedure and prompts you to press return.

    c. Press return.

12. Enter any other optional CLI commands for your configuration.

    See “Optional Steps,” page 5–29 and Chapter 7, “CLI Commands.”

13. Display details about the controller you configured. Use the following command:

    ```
    SHOW THIS_CONTROLLER FULL
    ```

    See “SHOW,” page 7–133.

IMPORTANT: If you are using a PC or terminal with the maintenance port cable during configuring and have not yet cabled the configuration, you must connect the optical host bus cables in the desired configuration as shown in the next section.

Connecting in Multiple-Bus Failover Mode to the Hosts

IMPORTANT: If you are using HSG80 ACS V8.4P code, see the *Data Replication Manager HSG80 ACS Version 8.4P Guide to Operations* for cabling yourcontroller into the subsystem.

You can connect each controller pair to your hosts with two switches.

Using Two Switches

Follow the procedures below to properly connect your dual-redundant controllers to your host. Figure 5–6 and Figure 5–7 illustrate two different choices on how your set-up should appear.

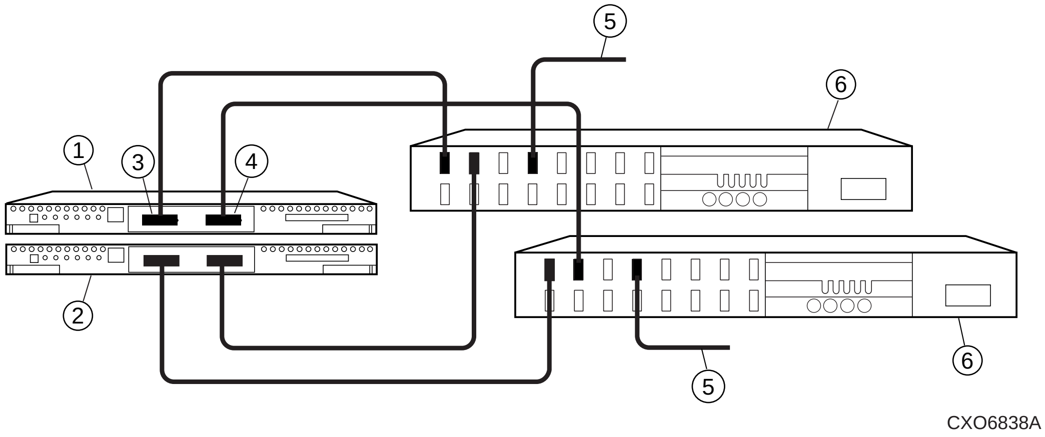


Figure 5–6. Example Cabling in Multiple-Bus Failover: Configuration 1

Table 5–5. Key to Figure 5–6 Example Cabling in Multiple-Bus Failover: Configuration 1

Item	Description
①	HSG80 controller A
②	HSG80 controller B
③	Host port 1 (active) connection from controller A to switch port 0 Host port 1 (active) connection from controller B to switch port 0
④	Host port 2 (active) connection from controller A to switch port 2 Host port 2 (active) connection from controller B to switch port 2
⑤	Host bus connections to the host adapters
⑥	Switches

Before cabling, note the following:

- Ports
  - ❑ All ports are active. Once units are added, all ports will have access to the same units.
  - ❑ Port 1, controller A and port 2, controller B connect into the same switch.
  - ❑ Port 1 controller B and port 2, controller A connect into the same switch.
  - ❑ Port 1 on both controllers connect into the same switch port on different switches.
  - ❑ Port 2 on both controllers connects into the same switch port on different controllers.
- Hosts
  - ❑ Host operating systems may be mixed together or one host may be used for all host connections, for example, two Windows NT hosts.

1. Configure each controller. See “Configuring in Multiple-Bus Failover Mode,” page 5–20.
2. Ensure the host and adapter are also configured. See the host user’s guide for details.
3. Connect the cables between host port 1 on both controllers and the switch ports on the switches.

   IMPORTANT: Each host port 1 cable must go to a different switch, but into the same switch port. Do not cable them into the same switch.

4. Connect the cables between host port 2 and the switch ports.

   IMPORTANT: Each host port 2 cable must go to a different switch, but into the same switch port. Do not cable them into the same switch.

5. Connect the host bus cable between the switches and the host adapters.

   IMPORTANT: Cables going into the same host must connect into the same switch port.

6. Route and tie the cables as desired.

7. Turn on the subsystem.

   Some operating systems may require you to start the host to see the devices attached to the new controller. See the *HSG80 Array Controller ACS Version 8.X Release Notes* for your operating system.

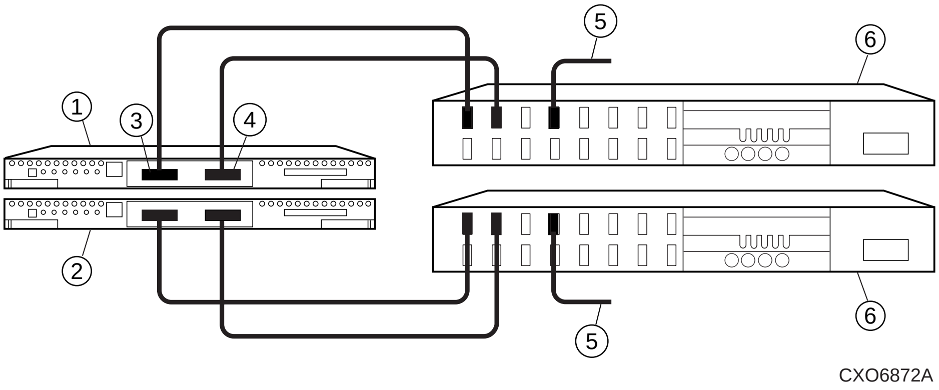


Figure 5–7. Example Cabling in Multiple-Bus Failover: Configuration 2

Table 5–6. Key to Figure 5–7 Example Cabling in Multiple-Bus Failover: Configuration 2

Item	Description
①	HSG80 controller A
②	HSG80 controller B
③	Host port 1 (active) connection from controller A to switch port 0 Host port 1 (active) connection from controller B to switch port 0
④	Host port 2 (active) connection from controller A to switch port 2 Host port 2 (active) connection from controller B to switch port 2
⑤	Host bus connections to the host adapters
⑥	Switches

Before cabling, note the following:

- Ports
  - ❑ All ports are active. Once units are added, all ports will have access to the same units.
  - ❑ Port 1, controller A and port 2, controller A connect into the same switch.
  - ❑ Port 1 controller B and port 2, controller B connect into the same switch.
  - ❑ Port 1 on both controllers connect into the same switch port on different switches.
  - ❑ Port 2 on both controllers connects into the same switch port on different controllers.
- Hosts
  - ❑ Host operating systems may be mixed together or one host may be used for all host connections, for example, two Windows NT hosts.

1. Configure each controller. See "Configuring in Multiple-Bus Failover Mode," page 5–20.
2. Ensure the host and adapter are also configured. See the host user's guide for details.
3. Connect the cables between host port 1 on both controllers and the switch ports on the switches.

   IMPORTANT: Each host port 1 cable must go to a different switch, but into the same switch port. Do not cable them into the same switch.

4. Connect the cables between host port 2 and the switch ports.

   IMPORTANT: Each host port 2 cable must go to a different switch, but into the same switch port. Do not cable them into the same switch.

5. Connect the host bus cable between the switches and the host adapters.

IMPORTANT: Cables going into the same host must connect into the same switch port.

6. Route and tie the cables as desired.
7. Turn on the subsystem.

   Some operating systems may require you to start the host to see the devices attached to the new controller. See the *HSG80 Array Controller ACS Version 8.X Release Notes* for your operating system.

Optional Steps

Following are some of the more common steps that can be performed anytime during configuring.

Changing the CLI Prompt

Enter a 1- to 16-character string as the new prompt. For example, you could use the prompt to indicate the controller’s name, such as “HSG1.”

Type the following command:

```
SET THIS_CONTROLLER PROMPT = “new prompt”
```

If you are configuring dual-redundant controllers, you can also change the CLI prompt on it. Use the following command:

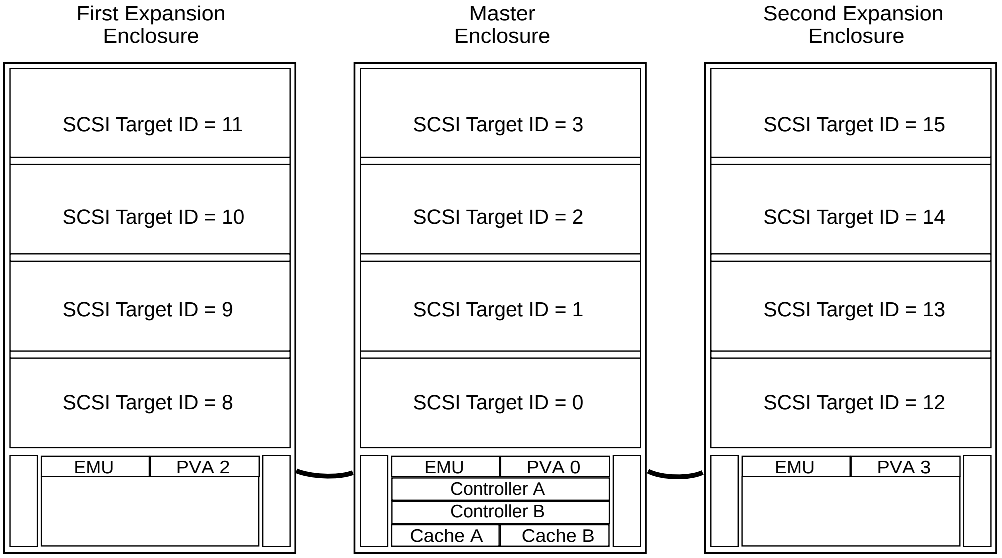
```
SET OTHER_CONTROLLER PROMPT = “new prompt”
```

See “SET controller,” page 7–91.

Setting the PVA Module ID Switch

The Power, Verification, and Addressing (PVA) module provides unique addresses to extended subsystems. Each BA370 rack-mountable enclosure in an extended subsystem must have its own PVA ID. Use PVA ID 0 for the enclosure that contains the array controllers. Use PVA IDs 2 and 3 for the additional enclosures. Figure 5–8 illustrates the PVA settings in an extended subsystem.

See the documentation that accompanied your enclosure for more details about the PVA and its settings.



NOTE: SCSI target IDs 4 and 5 are reserved. IDs 6 and 7 are used by the controllers.

CXO5806B

Figure 5–8. SCSI Target ID Numbers on the Controller Device Bus and PVA Settings in an Extended Subsystem

Setting a Port Offline

NOTE: By default, all ports are online and able to access targets and logical units, even when they are designated as standby ports in transparent failover mode.

To take a port offline and turn off its access to targets and logical units, use the following command:

```
SET THIS_CONTROLLER PORT_2_TOPOLOGY=OFFLINE
```

See “SET controller,” page 7–91 on how to use the offline switch.

Backing Up Power with a UPS

NOTE: By default, the controller expects to use an external cache battery (ECB) as backup power to the cache module. You can instead choose to use an uninterruptable power supply (UPS) to provide backup power in the event of a primary power failure.

Use the following command:

```
SET THIS_CONTROLLER CACHE_UPS
```

If you are configuring dual-redundant controllers, also indicate that the other controller’s power is supported by a UPS. Use the following command:

```
SET OTHER_CONTROLLER CACHE_UPS
```

The other controller inherits the cache UPS setting for this controller.

See “SET controller,” page 7–91 for details about setting the UPS.

See Table 2–2 on page 2–12 and Table 2–3 on page 2–15 for information about cache policies.

Changing the Host Default Value

By default, the host is Windows NT. However, when two or more of the same host exists, the default will be that host. For example, if a system is configured using three Tru64 UNIX hosts, then the default host will be the Tru64 UNIX operating system.

To view the current host connection, use the following command:

```
SHOW connection-name
```

To change the default host to a different host, use the following command:

```
SET connection-name
```

Establishing a Local Connection to the Controller

You can communicate with a controller locally or remotely. Use a local connection to configure the controller for the first time. Use a remote connection to your host system for all subsequent configuration tasks. See the *Quick Setup Guide* that came with your platform kit for details.

The maintenance port provides a convenient way to connect a PC or terminal to the controller so that you can troubleshoot and configure it. This port accepts a standard RS-232 jack from any EIA-423 compatible terminal or a PC with a terminal-emulation program. The maintenance port supports serial communications with default values of 9600 baud using 8 data bits, 1 stop bit, and no parity.

NOTE: The maintenance port cable shown in Figure 5–9 has a 9-pin connector molded onto its end for a PC connection. If you need a terminal connection, or for any information on part numbers, see the *HSG80 Array Controller ACS Version 8.3 and 8.4 Maintenance and Service Guide* for information on optional cabling.

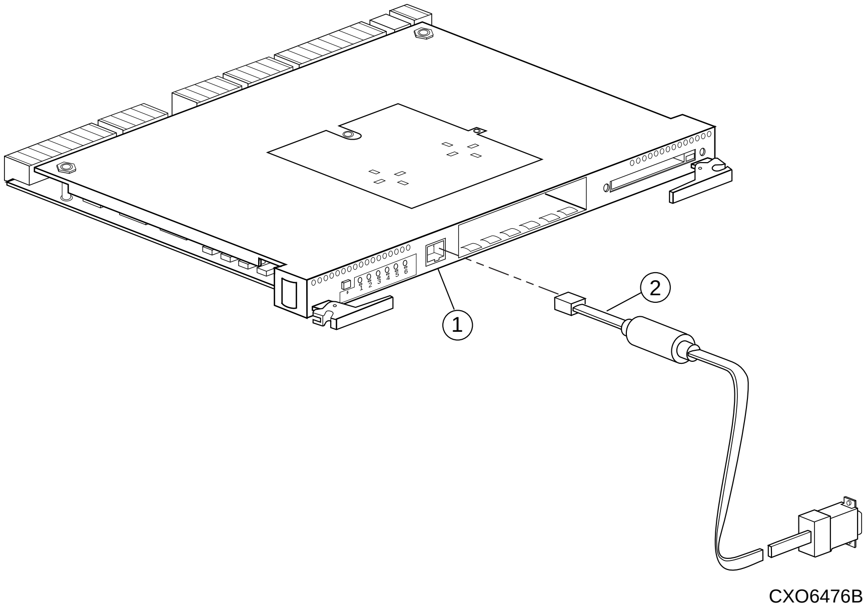


Figure 5–9. Terminal to Local-Connection Port Connection

Table 5–7. Key to Figure 5–9 Terminal to Local-Connection Port Connection

Item	Description
①	Maintenance port
②	Maintenance port cable kit, which includes the following PC Serial Port Adapters: • 9 Pin D-sub to 25 Pin SKT D-sub for a PC • 9 Pin D-sub to 25 pin D-sub for Sun Operating System • 9 Pin D-sub to 25 pin D-sub, mod for an HP 800 Operating System

CAUTION: The local-connection port described in this book generates, uses, and can radiate radio-frequency energy through cables that are connected to it. This energy may interfere with radio and television reception. Do not leave any cables connected to it when you are not communicating with the controller.

Follow these steps to establish a local connection for setting the controller’s initial configuration:

1. Turn off the PC or terminal, and connect it to the controller, as shown in Figure 5–9.
    a. For a PC connection, plug one end of the maintenance port cable into the terminal; plug the other end into the controller’s maintenance port.
2. Turn on the PC or terminal.
3. Configure the terminal for 9600 baud, 8 data bits, 1 stop bit, and no parity.
4. Press the Enter or Return key. A copyright notice and the CLI prompt appear, indicating that you established a local connection with the controller.
5. Optional: to increase the data transfer rate to 19200 baud:
    a. Set the controller to 19200 baud with one of the following commands:

    ```
    SET THIS_CONTROLLER TERMINAL SPEED=19200
    ```

    ```
    SET OTHER_CONTROLLER TERMINAL SPEED=19200
    ```

    b. Configure the PC or terminal for 19200 baud.

When you are entering CLI commands in a dual-redundant controller configuration, remember that the controller to which you’re connected is “this controller” and the remaining controller is the “other controller.” See “Terminology,” page 2–2, on the definitions for “this controller” and “other controller.”

Shutting Down Your Subsystem

Follow these steps to shut down your StorageWorks subsystem:

1. On the host, dismount the storage units in your subsystem.
2. Connect a maintenance terminal to one of the controllers in your subsystem.
3. Shut down the controllers. If you have dual-redundant controllers, shut down the “other controller” first, then shut down “this controller.” Use the following syntax:

   ```
   SHUTDOWN OTHER_CONTROLLER
   ```

   ```
   SHUTDOWN THIS_CONTROLLER
   ```

   NOTE: This process can take up to five minutes to complete depending on the amount of data to be flushed from cache.
4. Turn off the power to the subsystem.
5. Unplug the subsystem’s power cord.
6. Disable the ECB by pressing its shut-off button until its status light stops blinking—about two seconds.

Restarting Your Subsystem

Follow these steps to restart your subsystem:

1. Plug in the subsystem’s power cord.
2. Turn on the subsystem.
3. Press and hold the reset button on the controller for three seconds, then release it.
4. Check the status of the write-back cache module’s backup battery. If your subsystem has been off for an extended period of time, the battery may be drained. Use the following syntax to check the battery’s status:

   ```
   SHOW THIS_CONTROLLER
   ```

Chapter 6
Configuring Storagesets

This chapter shows you how to configure storagesets with CLI Commands and the Command Console LUN. It outlines the procedures for partitioning storagesets and disk drives and how to enable autospare. You will also learn how to configure stripesets, mirrorsets, and striped mirrorsets.

Configuring Storagesets with CLI Commands

One method of configuring storagesets is manual configuration. This method allows you the most flexibility in defining and naming storagesets. See Chapter 7, “CLI Commands.” for complete information about the CLI commands shown in this chapter.

Unit Switches

You can enable the Unit switches listed in Table 6–1 for the listed storagesets and devices.

Table 6–1 Unit Switches

Switch Container Type	ENABLE_ACCESS_PATH DISABLE_ACCESS_PATH	MAXIMUM_CACHED_ TRANSFER	IDENTIFIER NOIDENTIFIER	PREFERRED_PATH NOPREFERRED_PATH	READ_CACHE NOREAD_CACHE	READAHEAD_CACHE NOREADAHEAD_CACHE	WRITE_PROTECT NOWRITE_PROTECT	WRITEBACK_CACHE NOWRITEBACK_CACHE	RUN NORUN
RAIDset	✔	✔	✔	✔	✔	✔	✔	✔	✔
Stripeset	✔	✔	✔	✔	✔	✔	✔	✔	✔
Mirrorset	✔	✔	✔	✔	✔	✔	✔	✔	✔
NoTransportable Disk	✔	✔	✔	✔	✔	✔	✔	✔	✔
Transportable Disk	✔	✔	✔	✔	✔	✔	✔		✔

See “ADD UNIT,” page 7–27 and “SET unit-number,” page 7–126 for a complete list and further explanation of the unit switches.

Adding Disk Drives

The factory-installed devices in your StorageWorks subsystem have already been added to the controller's list of eligible devices. If you want to add new devices to your subsystem, you must issue one of the following CLI commands before you can use them in any kind of storageset, single disk unit, or spareset:

Adding One Disk Drive at a Time

To add one new disk drive to your controller's list of eligible devices, enter the following command at the prompt:

```
ADD DISK disknnn ptl-location switch_value
```

Adding Several Disk Drives at a Time

To add several new disk drives to your controller's list of eligible devices, enter the following command at the prompt:

```
RUN CONFIG
```

Formatting Disk Drives

Use HSUTIL's FORMAT option to simultaneously format up to seven disk drives attached to a single controller or up to six disk drives attached to a dual-redundant pair of controllers. Refer to the *HSG80 Array Controller ACS Version 8.3 and 8.4 Maintenance and Service Guide* for more information on HSUTIL.

Consider the following points before formatting disk drives with HSUTIL:

- HSUTIL cannot format disk drives configured as single-disk drive units or as members of a storageset, spareset, or failedset. If you want to format a disk drive that was previously configured as one of these things, you must first delete the unit number and storageset name associated with it.

- If the power fails or the bus is reset while HSUTIL is formatting a disk drive, the drive may become unusable. To minimize this possibility, Compaq recommends you use a reliable power source and suspend all non-HSUTIL activity to the bus that services the target disk drive.
- HSUTIL cannot control or affect the defect management for a disk drive.
- Do not invoke any CLI command or run any local program that might reference the target disk drive while HSUTIL is active. Also, do not reinitialize either controller in the dual-redundant configuration while using HSUTIL.

To format one or more disk drives complete the following steps:

> △ **CAUTION:** Suspend all I/O to the buses that service the target disk drives before using HSUTIL to format the drives.

1. Start HSUTIL by issuing the following command:

   ```
   RUN HSUTIL
   ```

2. Enter **1** to select the FORMAT function.

   HSUTIL finds and displays all of the unattached disk drives configured on the controller.

3. Type the name of a disk drive you want to format.
4. Enter **Y** to enter another disk drive name or **N** to begin the formatting operation.
5. Read the cautionary information that HSUTIL displays, then confirm or cancel the formatting operation.

The formatting operation will complete in approximately the estimated time.

The following example shows the sequence of steps as they would appear on the terminal when you format a disk drive:

```
CLI> RUN HSUTIL

*** AVAILABLE FUNCTIONS ARE:
0. EXIT
1. FORMAT
```

```
2. DEVICE_CODE_LOAD_DISK
3. DEVICE_CODE_LOAD_TAPE
ENTER FUNCTION NUMBER (0:3) [0] ? 1

UNATTACHED DEVICES ON THIS CONTROLLER INCLUDE:
DEVICE       SCSI PRODUCT ID    CURRENT DEVICE REV
DISK10000       RZ26   (C) DEC     T386
DISK20000       RZ26   (C) DEC     T386
DISK20100       RZ29B  (C) DEC      0006
DISK30100       RZ25   (C) DEC     0900
DISK30200       RZ26L  (C) DEC      X442

ENTER A DEVICE TO FORMAT ? DISK10000
FORMAT DISK10000 MAY TAKE UP TO 40 MINUTES TO FORMAT
SELECT ANOTHER DEVICE (Y/N) [N] Y

ENTER A DEVICE TO FORMAT ? DISK20000
FORMAT DISK20000 MAY TAKE UP TO 35 MINUTES TO FORMAT
SELECT ANOTHER DEVICE (Y/N) [N] Y

ENTER A DEVICE TO FORMAT ? DISK20100
FORMAT DISK20100 MAY TAKE UP TO 15 MINUTES TO FORMAT
SELECT ANOTHER DEVICE (Y/N) [N] N

^Y AND ^C WILL BE DISABLED WHILE THE FORMAT OPERATION IS IN PROGRESS.

CAUTION:
   WHEN YOU FORMAT A DEVICE, IT WILL DESTROY THE DATA ON
   THE DEVICE. A BACKUP OF THE DEVICE SHOULD HAVE BEEN
   DONE IF THE DATA IS IMPORTANT.

NOTE:
   IN ORDER TO MINIMIZE THE POSSIBILITY OF A SCSI BUS
   RESET, IT IS RECOMMENDED THAT YOU PREVENT NON-HSUTIL IO
   OPERATIONS TO ALL OTHER DEVICES ON THE SAME PORT AS THE
   DESTINATION DEVICE(S). IF A SCSI BUS RESET OCCURS, THE
   FORMAT MAY BE INCOMPLETE AND YOU MAY HAVE TO RE-INVOKE
   HSUTIL.

AFTER YOU ANSWER THE NEXT QUESTION, THE FORMAT WILL START. DO YOU WANT
TO CONTINUE (Y/N) [N] ? Y

HSUTIL STARTED AT: 14-JAN-1997 15:00:31
FORMAT OF DISK10000 FINISHED AT 14-JAN-1997 15:25:12
FORMAT OF DISK20000 FINISHED AT 14-JAN-1997 15:30:31
FORMAT OF DISK20100 FINISHED AT 14-JAN-1997 15:30:43

HSUTIL - NORMAL TERMINATION AT 14-JAN-1997 15:31:09
```

Configuring a Stripeset

See “Using Striped Mirrorsets for Highest Performance and Availability,” page 4–19, for information about creating a profile and understanding the switches you can set for this kind of storage unit.

To configure a stripeset:

1. Create the stripeset by adding its name to the controller’s list of storagesets and specifying the disk drives it contains. Use the following command:

   ADD STRIPESET *stripeset-name* disk*nnnnn* disk*nnnnn*

2. Initialize the stripeset. If you want to set any Initialize switches, you must do so in this step. Use the following command:

   INITIALIZE *stripeset-name switch*

3. Present the stripeset to the host by giving it a unit number the host can recognize. Optionally, you can append Unit switch values. If you do not specify switch values, the default values are applied.

   ADD UNIT *unit-number stripeset-name switch*

   a. For OpenVMS hosts, set the unit number to a decimal value, ranging from 1–9999. Usually, this number is the same as the unit number set in step 3.

      SET *unit-number identifier=nnnn*

4. Verify the stripeset configuration and switches. Use the following command:

   SHOW *stripeset-name*

5. Verify the unit configuration and switches. Use the following command:

   SHOW *unit-number*

The following example shows the commands you would use to create Stripe1, a three-member stripeset:

```
ADD STRIPESET STRIPE1 disk10000 disk20000 disk30000

INITIALIZE STRIPE1 CHUNKSIZE=128

ADD UNIT D100 STRIPE1 MAXIMUM_CACHED_TRANSFER=16

SHOW STRIPE1

SHOW D100
```

See Chapter 7, “CLI Commands,” for more information on stripeset switches and values.

Configuring a Mirrorset

See Chapter 4, “Planning Storagesets,” for information about creating a profile and understanding the switches you can set for this kind of storage unit.

To configure a mirrorset:

1. Create the mirrorset by adding its name to the controller’s list of storagesets and specifying the disk drives it contains. Optionally, you can append Mirrorset switch values. If you do not specify switch values, the default values are applied.

   Use the following command to create a mirrorset:

   ADD MIRRORSET *mirrorset-name* DISK*nnnnn* DISK*nnnnn switch*

2. Initialize the mirrorset. If you want to set any Initialize switches, you must do so in this step. Use the following command:

   INITIALIZE *mirrorset-name switch*

3. Present the mirrorset to the host by giving it a unit number the host can recognize. Optionally, you can append Unit switch values. If you do not specify switch values, the default values are applied. Use the following command:

   ADD UNIT *unit-number mirrorset-name switch*

   a. For OpenVMS hosts, set the unit number to a decimal value, ranging from

1–9999. Usually, this number is the same as the unit number set in step 3.

```
SET unit-number identifier=nnnn
```

4. Verify the mirrorset configuration and switches. Use the following command:

   ```
   SHOW mirrorset-name
   ```

5. Verify the unit configuration and switches. Use the following command:

   ```
   SHOW unit-number
   ```

Example

The following example shows the commands you would use to create Mirr1, a two-member stripeset:

```
ADD MIRRORSET Mirr1 disk10000 disk20000

INITIALIZE Mirr1

ADD UNIT D110 Mirr1

SHOW Mirr1

SHOW D110
```

Refer to Chapter 7, “CLI Commands,” for more information on the ADD_MIRRORSET, INITIALIZE, ADD_UNIT, and SHOW commands for creating a mirrrorset.

Configuring a RAIDset

To configure a RAIDset:

1. Create the RAIDset by adding its name to the controller’s list of storagesets and specifying the disk drives it contains. Optionally, you can append RAIDset switch values. If you do not specify switch values, the default values are applied.

   Use the following command to create a RAIDset:

   ```
   ADD RAIDSET RAIDset-name DISKnnnnn DISKnnnnn
   DISKnnnnn switch
   ```

2. Initialize the RAIDset. Optional: If you want to set the Initialize switches, you must do so in this step. Use the following command:

   INITIALIZE *RAIDset-name switch*

   NOTE: It is recommended that you allow initial reconstruct to complete before allowing I/O to the RAIDset. Not doing so may generate forced errors at the host level. To determine whether initial reconstruct has completed, enter SHOW S or SHOW RAIDSET FULL.

3. Present the RAIDset to the host by giving it a unit number the host can recognize. Optionally, you can append Unit switch values. If you do not specify switch values, the default values are applied.

   Use the following command to present the RAIDset to the host:

   ADD UNIT *unit-number RAIDset-name switch*

   a. For OpenVMS hosts, set the unit number to a decimal value, ranging from 1–9999. Usually, this number is the same as the unit number set in step 3.

      SET *unit-number identifier=nnnn*

4. Verify the RAIDset configuration and switches. Use the following command:

   SHOW *RAIDset-name*

5. Verify the unit configuration and switches. Use the following command:

   SHOW *unit-number*

Example

The following example shows the commands you would use to create Raid1, a three-member RAIDset:

```
ADD RAIDSET Raid1 disk10000 disk20000 disk30000

INITIALIZE Raid1

ADD UNIT D99 Raid1

SHOW Raid1

SHOW D99
```

Chapter 7, “CLI Commands,” contains more information on valid switches and values for configuring a RAIDset.

Configuring a Striped Mirrorset

See Chapter 4, “Planning Storagesets,” for information about creating a profile and understanding the switches you can set for this kind of storage unit.

To configure a striped mirrorset:

1. Create—but do not initialize—at least two mirrorsets.
2. Create a stripeset and specify the mirrorsets it contains. Use the following command:

   ```
   ADD STRIPESET mirrorset_1 mirrorset_2
   ```

3. Initialize the stripeset. If you want to set any Initialize switches, you must do so in this step. Use the following command:

   ```
   INITIALIZE stripeset-name switch
   ```

   See “SET mirrorset-name,” page 7–110 for a complete list of valid mirrorset switches and values.

4. Present the stripeset to the host by giving it a unit number the host can recognize. Optionally, you can append Unit switch values. If you do not specify switch values, the default values are applied.

   Use the following command to present the stripeset to the host:

   ```
   ADD UNIT unit-number stripeset-name switch
   ```

   a. For OpenVMS hosts, set the unit number to a decimal value, ranging from 1–9999. Usually, this number is the same as the unit number set in step 4.

      ```
      SET unit-number identifier=nnnn
      ```

5. Verify the striped mirrorset configuration and switches. Use the following command:

   ```
   SHOW stripeset-name
   ```

6. Verify the unit configuration and switches. Use the following command:

   ```
   SHOW unit-number
   ```

Example

The following example shows the commands you would use to create Stripe1, a three-member striped mirrorset that comprises Mirr1, Mirr2, and Mirr3, each of which is a two-member mirrorset:

```
ADD MIRRORSET Mirr1 disk10000 disk20000

ADD MIRRORSET Mirr2 disk30000 disk40000

ADD MIRRORSET Mirr3 disk50000 disk60000

ADD STRIPESET Stripe1 Mirr1 Mirr2 Mirr3

INITIALIZE Stripe1 CHUNKSIZE=default

ADD UNIT D101 Stripe1

SHOW Stripe1

SHOW D101
```

For more detailed information on configuring a striped mirrorset, refer to Chapter 7, “CLI Commands.”

Configuring a Single-Disk Unit

Follow these steps to use a single disk drive as a single-disk unit in your subsystem:

1. Add the disk drive by following the steps in “Adding Disk Drives,” page 6–3. Optionally, you can append Device switch values. If you do not specify switch values, the default values are applied.
2. Initialize the disk drive using the following command:

   INITIALIZE DISK*nnn switch*
3. Present the disk drive to the host by giving it a unit number the host can recognize. Optionally, you can append Unit switch values. If you do not specify switch values, the default values are applied. Use the following command:

   ADD UNIT *unit-number* DISK*nnn switch_value*

   NOTE: If you make a disk transportable, you cannot specify WRITEBACK_CACHE for that disk.

   a. For OpenVMS hosts, set the unit number to a decimal value, ranging from 1–9999. Usually, this number is the same as the unit number set in step 3.

      SET *unit-number identifier=nnnn*
4. Verify the configuration using the following command:

   SHOW DEVICES

Example

The following example shows the commands you would use to configure DISK10000 as a single-disk unit.

```
ADD DISK DISK10000 1 0 0

ADD UNIT D101 disk10000

SHOW DEVICES
```

See Chapter 7, “CLI Commands” for further information on these switches and values.

Partitioning a Storageset or Disk Drive

See “Planning Partitions,” page 4–34, for details about partitioning a storage unit.

To partition a storageset or disk drive:

1. Add the storageset or disk drive to the controller’s list of storagesets and specify the disk drives it contains. Use the following command:

   ```
   ADD storageset-name DISKnnnnn DISKnnnnn
   ```

   or

   ```
   ADD DISK DIsknnnnn ptl-location
   ```

Do not split partitioned units across ports. They must be on a single port. The subsystem assigns units 0-99 to Port 1; units 100-199 are assigned to Port 2.

2. Initialize the storageset or disk drive. If you want to set any Initialize switches, you must do so in this step. Use the following command:

   ```
   INITIALIZE storageset-name switch
   ```

3. Create each partition in the storageset or disk drive by indicating the partition’s size. Use the following command:

   ```
   CREATE_PARTITION storageset-name SIZE=n
   ```

   where n is the percentage of the disk drive or storageset that will be assigned to the partition. Enter SIZE=LARGEST to let the controller assign the largest free space available to the partition.

4. Verify the partitions, using the following command:

   ```
   SHOW storageset-name
   ```

   The partition number appears in the first column, followed by the size and starting block of each partition.

5. Present each partition to the host by giving it a unit number the host can recognize. (You can skip this step until you are ready to put the partitions online.) Optionally, you can append Unit switch values. If you do not specify switch values, the default values are applied.

Use the following command to present partitions to the host:

ADD UNIT *unit-number storageset-name* PARTITION=*partition-number switch*

6. Verify the unit numbers for the partitions using the following command:

   SHOW *storageset-name*

Example

The following example shows the commands you would use to create Raid1, a three-member RAIDset, then partition it into four storage units:

```
ADD RAIDSET Raid1 disk10000 disk20000 disk30000

INITIALIZE Raid1

CREATE_PARTITION Raid1 SIZE=25

CREATE_PARTITION Raid1 SIZE=25

CREATE_PARTITION Raid1 SIZE=25

CREATE_PARTITION Raid1 SIZE=LARGEST

SHOW Raid1
```

```
Partition number     Size       Starting Block  Used by
       1        1915 (0.98 MB)        0          Raid1
       2        1915 (0.98 MB)       1920        Raid1
       3        1915 (0.98 MB)       3840        Raid1
       4        2371 (1.21 MB)       5760        Raid1
       .
       .
       .
```

```
ADD UNIT D1 Raid1 PARTITION=1

ADD UNIT D2 Raid1 PARTITION=2

ADD UNIT D3 Raid1 PARTITION=3

ADD UNIT D4 Raid1 PARTITION=4

SHOW Raid1

.

.

.
```

```
Partition number      Size      Starting Block  Used by
       1        1915 (0.98 MB)        0           D1
       2        1915 (0.98 MB)      1920          D2
       3        1915 (0.98 MB)      3840          D3
       4        2371 (1.21 MB)      5760          D4
       .
       .
       .
```

Chapter 7, “CLI Commands,” contains more information on partitioning a storageset or disk drive.

Adding a Disk Drive to the Spareset

The spareset is a collection of hot spares that are available to the controller should it need to replace a failed member of a RAIDset or mirrorset.

Use the following steps to add a disk drive to the spareset:

> **NOTE:** This procedure assumes that the disks that you are adding to the spareset have already been added to the controller’s list of eligible devices.

1. Add the disk drive to the controller’s spareset list. Use the following command:

   ```
   ADD SPARESET disknnnnn
   ```

   Repeat this step for each disk drive you want to add to the spareset:

2. Verify the contents of the spareset using the following command:

   ```
   SHOW SPARESET
   ```

Example

The following example shows the commands you would use to add DISK60000 and DISK60100 to the spareset.

```
ADD SPARESET disk60000
ADD SPARESET disk60100
SHOW SPARESET
```

Removing a Disk Drive from the Spareset

You cannot delete the spareset—it always exists whether or not it contains disk drives. However, you can delete disks in the spareset if you need to use them elsewhere in your StorageWorks subsystem.

To remove a disk drive from the spareset:

1. Show the contents of the spareset using the following command:

   ```
   SHOW SPARESET
   ```

2. Delete the desired disk drive using the following command:

   ```
   DELETE SPARESET disknnnnn
   ```

   Verify the contents of the spareset using the following command:

   ```
   SHOW SPARESET
   ```

Example

The following example shows the commands you would use to remove DISK60000 from the spareset.

```
SHOW SPARESET
```

```
   Name    Storageset     Uses      Used by
-------------------------------------------
 SPARESET   spareset   disk60000
                       disk60100
```

```
DELETE SPARESET disk60000
```

```
SHOW SPARESET
```

```
   Name    Storageset     Uses      Used by
-------------------------------------------
 SPARESET   spareset   disk60100
```

Enabling Autospare

With AUTOSPARE enabled on the failedset, any new disk drive that is inserted into the PTL location of a failed disk drive is automatically initialized and placed into the spareset. If initialization fails, the disk drive remains in the failedset until you manually delete it from the failedset.

To enable autospare use the following command:

```
SET FAILEDSET AUTOSPARE
```

To disable autospare use the following command:

```
SET FAILEDSET NOAUTOSPARE
```

During initialization, AUTOSPARE checks to see if the new disk drive contains metadata—the information that indicates it belongs to, or has been used by, a known storageset. If the disk drive contains metadata, initialization stops. (A new disk drive will not contain metadata but a repaired or re-used disk drive might. To erase metadata from a disk drive, add it to the controller’s list of devices, then set it to be TRANSPORTABLE and initialize it.)

Deleting a Storageset

If the storageset you are deleting is partitioned, you must delete each partitioned unit before you can delete the storageset. Use the following steps to delete a storageset:

1. Show the configuration using the following command:

   ```
   SHOW STORAGESETS
   ```

2. Delete the unit number shown in the “Used by” column. Use the following command:

   DELETE *unit-number*

3. Delete the name shown in the “Name” column. Use the following command:

   DELETE *storageset-name*

4. Verify the configuration using the following command:

   ```
   SHOW STORAGESETS
   ```

Example

The following example shows the commands you would use to delete Stripe1, a three-member stripeset that is comprised of DISK10000, DISK20000, and DISK30000.

```
SHOW STORAGESETS
```

```
   Name     Storageset      Uses      Used by
 STRIPE1    stripeset    DISK10000     D100
                         DISK20000
                         DISK30000
```

```
DELETE D100
DELETE Stripe1
SHOW STORAGESETS
```

Changing Switches for a Storageset or Device

You can optimize a storageset or device at any time by changing the switches that are associated with it. See “Choosing Switches for Storagesets and Devices,” page 4–36, for an explanation of the switches. Remember to update the storageset’s profile when you change its switches.

Displaying the Current Switches

To display the current switches for a storageset or single-disk unit, enter the following command at a CLI prompt:

```
SHOW storageset-name or device-name FULL
```

Changing RAIDset and Mirrorset Switches

Use the SET *storageset-name* command to change the RAIDset and Mirrorset switches associated with an existing storageset. For example, the following command changes the replacement policy for RAIDset Raid1 to BEST_FIT:

```
SET RAID1 POLICY=BEST_FIT
```

Changing Device Switches

Use the SET command to change the device switches. For example, the following command enables DISK10000 to be used in a non-StorageWorks environment:

```
SET DISK10000 TRANSPORTABLE
```

The TRANSPORTABLE switch cannot be changed for a disk if the disk is part of an upper-level container. Additionally, the disk cannot be configured as a unit if it is to be used as indicated in this example.

Changing Initialize Switches

The Initialize switches cannot be changed without destroying the data on the storageset or device. These switches are integral to the formatting and can only be changed by reinitializing the storageset. Initializing a storageset is similar to formatting a disk drive; all data is destroyed during this procedure.

Changing Unit Switches

Use the SET command to change Unit switches that are associated with a unit. For example, the following command enables write protection for unit D100:

```
SET D100 WRITE_PROTECT
```

Configuring with the Command Console LUN

The Command Console LUN (CCL) is a type of LUN that allows you to communicate with the controller from the host using StorageWorks Command Console (SWCC) or CLI commands instead of using the maintenance port cable.

The most common tasks performed with the CCL include:

- Configuring storage units
- Preparing the subsystem for use
- Checking a failed set
- Checking performance with VTDPY
- Troubleshooting with FMU

NOTE: Do not use the CCL with HSUTIL, FRUTIL, or DILX. The host and controller must communicate through the maintenance port cable when using these utilities and exercisers. Also, do not use the CCL to troubleshoot and maintain the controller. Instead, run the utilities and exercisers with the local connection that uses the maintenance port cable. See “Establishing a Local Connection to the Controller,” page 5–33.

Enabling and Disabling the CCL

If you have not configured any units and have not yet enabled the CCL, you must first establish a local connection through the maintenance port cable to provide a means of enabling CCL. Once the CCL is enabled, you can communicate with the controller over the host port connection instead of through the maintenance port connection. See “Establishing a Local Connection to the Controller,” page 5–33.

If the CCL is not automatically enabled on your controller, use the following command:

```
SET THIS_CONTROLLER COMMAND_CONSOLE_LUN
```

To turn it off, use the following command:

```
SET THIS_CONTROLLER NOCOMMAND_CONSOLE_LUN
```

CAUTION: Disabling the CCL while SWCC is running may result in loss of connection for the StorageWorks Command Console. Turn off SWCC before issuing the command.

Finding the CCL Location

To see where each CCL is located, use the following commands:

```
SHOW THIS_CONTROLLER
```

or

```
SHOW OTHER_CONTROLLER
```

Look under host port to find the Command Console LUN location. Because the CCL is not an actual device or unit, the SHOW UNITS command will display only unit information and no CCL locations.

Multiple-Bus and Transparent Failover in SCSI-2 and SCSI-3 Modes

The way the host sees the CCL varies, depending on whether you've enabled transparent or multiple-bus failover modes and whether you are in SCSI-2 or SCSI-3 mode.

SCSI-2 Mode

In SCSI-2 mode, the CCL is located at the first unassigned LUN. As units are added, it automatically floats—or moves—to the next available LUN.

Multiple-Bus Failover

If SET MULTIBUS_FAILOVER is enabled, all ports will be able to see and access the CCL. In addition, the CCL appears to the host as a direct access device.

IMPORTANT: A port will not be able to see the CCL if a host communication has a unit offset higher than the location of the CCL.

Transparent Failover

If you are in SCSI-2 mode and have enabled the SET FAILOVER command, only one CCL will be enabled. Either host port 1 or host port 2 will see it, but not both.

SCSI-3 Mode

In SCSI-3 mode, the CCL is automatically enabled and cannot be disabled. The CCL will appear at unit 0 of each unit offset.

Multiple-Bus Failover

If you are in multiple-bus failover mode, all ports will be able to see and access the CCLs. As a result, all hosts will have access to each CCL, and they will appear to the host as array controllers.

IMPORTANT: A host outside of the CCL range will not be able to see the CCL.

Transparent Failover

If you are in transparent failover mode, each CCL will be accessible from the port that has the unit offset enabled. Ports with multiple unit offsets assigned will have multiple CCLs, and they will appear to the host as an array controller.

CAUTION: Selecting SCSI-3 mode enables access to the CCL by all hosts. If the hosts access the CCL simultaneously, unpredictable consequences can occur. In cases where the CCL can be accessed through multiple paths and LUNs, system administrators of each host must not attempt to access the CCL simultaneously.

Adding Storage Units Using the CCL

To start configuring storage units, you must first assign unit offsets. See "Assigning Unit Offsets," page 3–13.

IMPORTANT: Unit 0 is the CCL location. Any hosts connected to host port 2 will not see the CCL.

SCSI-2 Mode

As you add storage units and assign unit offsets over the CCL or above its location, the CCL immediately and automatically moves into the next available free space, which would be the lowest available LUN.

NOTE: If you delete a unit at a setting below the CCL setting, the CCL does not automatically move. Instead, it only moves to the lowest deleted unit's setting when you reboot the controller.

SCSI-3 Mode

The CCL will appear at the default unit offset of each port. Because the default unit offset is 0 for Port 1 and 100 for Port 2, the CCL will be at LUN 0 on Port 1 and LUN 100 on Port 2.

NOTE: LUN 100 on Port 2 appears as LUN 0 to its hosts.

Moving Storagesets

You can move a storageset from one subsystem to another without destroying its data as shown in Figure 6–1. You also can follow the steps in this section to move a storageset to a new location within the same subsystem.

CAUTION: Move only normal storagesets. Do not move storagesets that are reconstructing or reduced, or data corruption will result.

You can use the procedure in this section to migrate wide devices from an HSZ70 controller in a BA370 rack-mountable enclosure to an HSG80 environment. However, if you have an HSZ40 or HSZ50 subsystem, you cannot migrate to an HSG80 in a BA370 rack-mountable enclosure. Refer to the *StorageWorks Array Controller HSG80 Array Controller Operating Software ACS Version 8.3 Release Notes* for drives that can be supported.

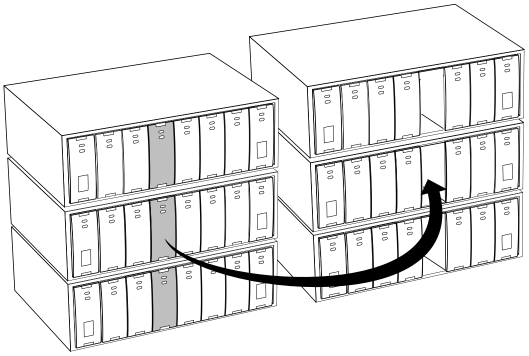


Figure 6–1 Moving a Storageset from one Subsystem to Another

CAUTION: Never initialize any container or this procedure will not protect data.

Use the following procedure to move a storageset while maintaining the data it contains:

1. Show the details for the storageset you want to move. Use the following command:

   SHOW *storageset-name*

2. Label each member with its name and PTL location.

   If you do not have a storageset map for your subsystem, you can enter the LOCATE command for each member to find its PTL location. Use the following command:

   LOCATE *disk-name*

   To cancel the locate command, enter the following:

   LOCATE CANCEL

3. Delete the unit-number shown in the “Used by” column of the SHOW storageset-name command. Use the following command:

   DELETE *unit-number*

4. Delete the storageset shown in the “Name” column of the SHOW storageset-name command. Use the following command:

   DELETE *storageset-name*

5. Delete each disk drive—one at a time—that the storageset contained. Use the following command:

   DELETE *disk-name*

   DELETE *disk-name*

   DELETE *disk-name*

6. Remove the disk drives and move them to their new PTL locations.

7. Add again each disk drive to the controller’s list of valid devices. Use the following command:

   ADD DISK *disk-name PTL-location*

   ADD DISK *disk-name PTL-location*

   ADD DISK *disk-name PTL-location*

8. Recreate the storageset by adding its name to the controller’s list of valid storagesets and specifying the disk drives it contains. (Although you have to recreate the storageset from its original disks, you do not have to add them in their original order.) Use the following command:

   ADD *storageset-name disk-name disk-name*

9. Represent the storageset to the host by giving it a unit number the host can recognize. You can use the original unit number or create a new one. Use the following command:

   ADD UNIT *unit-number storageset-name*

Example

The following example moves unit D100 to another cabinet. D100 is the RAIDset RAID99 that is comprised of members DISK10000, DISK20000, and DISK30000.

SHOW Raid99

```
   Name      Storageset       Uses          Used by
-------------------------------------------------------
  RAID99      raidset       disk10000         D100
                            disk20000
                            disk30000
```

DELETE D100

DELETE RAID99

DELETE DISK10000

DELETE DISK20000

DELETE DISK30000

(...move the disk drives to their new location...)

```
ADD DISK DISK20000 2 0 0
ADD DISK DISK30000 3 0 0
ADD DISK DISK40000 4 0 0
ADD RAIDSET RAID99 DISK20000 DISK30000 DISK40000
ADD UNIT D100 RAID99
```

Example

The following example moves the reduced RAIDset, R3, to another cabinet. (R3 used to contain DISK20000, which failed before the RAIDset was moved. R3 contained DISK10000, DISK30000, and DISK40000 at the beginning of this example.)

```
DELETE D100
DELETE R3
DELETE DISK10000
DELETE DISK30000
DELETE DISK40000
```

(...move disk drives to their new location...)

```
ADD DISK DISK10000 1 0 0
ADD DISK DISK30000 3 0 0
ADD DISK DISK40000 4 0 0
ADD RAIDSET R3 DISK10000 DISK30000 DISK40000 REDUCED
ADD UNIT D100 R3
```

Chapter 7

CLI Commands

Command Line Interpreter (CLI) commands allow you to interact with your controller. In the overview, you will learn about CLI and how to use it. The rest of the chapter details the syntax of each command and shows examples of how they are used.

CLI Overview

The Command Line Interpreter (CLI) is one of the user interfaces through which you control your StorageWorks array controller in the StorageWorks subsystem. The CLI commands allow you to manage the subsystem by viewing and modifying the configuration of the controller and the devices attached to them. 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 available for use with the CLI. The GUI, StorageWorks Command Console (SWCC), replicates most of the functions available within the CLI in graphic form and provides a user-friendly method of executing CLI commands.

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

Using the CLI

You can access the CLI by connecting a maintenance terminal to the port in the front bezel of the controller (local connection) or by using HSZterm software (remote connection). See "Establishing a Local Connection to the Controller," page 5–33, for instructions explaining how to connect a local terminal to the controller. After you have initially configured the controller, making it visible to the host, you can perform all other configuration tasks through a remote connection.

The section entitled "Maintenance Port Precautions," page xvii, explains precautions you should observe when operating the CLI through a maintenance port.

Command Overview

The CLI consists of six basic command types:

- 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 Commands—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 the controllers to operate in transparent failover while also providing support for dual-redundant configurations.
- Diagnostic and Utility Commands—Perform general controller support functions.

Getting Help

Help on using the CLI is at your fingertips. For an overview of the CLI help system, enter help at the prompt. For help on a specific command or to determine what switches are available with a command, enter as much of the command 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_CONTROLLER ?
```

See "HELP," page 7–61 for further information.

Entering CLI Commands

Use the following tips and techniques when entering 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 7–1 lists the keys used to recall and edit commands.

Table 7–1. Recall and Edit Command Keys

Key	Function
Up Arrow or Ctrl/B, Down Arrow or Ctrl/N	Steps backward and forward 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/A or F14	Toggles between insert mode and overstrike mode. The default setting is insert mode, which allows you to insert characters at the cursor location, moving the existing characters to the right. Overstrike mode replaces existing characters. The CLI returns to insert mode at the beginning of each line.
Ctrl/E	Moves the cursor to the end of the line.
Ctrl/H or Backspace or F12	Moves the cursor to the beginning of the line.
Ctrl/J or Linefeed or F13	Deletes the word to the left of the cursor.

Table 7–1. Recall and Edit Command Keys (Continued)

Key	Function
Ctrl/R	Recalls the contents of the command line. This is especially helpful if the system issues a message that interrupts your typing.
Ctrl/U	Deletes all characters on the same line as the cursor.

Changing the CLI Prompt

You can change the CLI prompt that displays. Use the SET THIS_CONTROLLER PROMPT= command. Enter a 1- to 16- character string as the new prompt. For example, you could use the prompt to indicate the array controller’s name, such as “HSG>.”

Command Syntax

Commands to the controller must use the following command structure:

COMMAND *parameter SWITCHES*

- **Command.** A word or phrase expressed as a verb that is used to instruct the controller what to do. Every CLI command begins with a command. Commands are represented in this manual in capitalized form.
- **Parameter.** When required in the command, one or more words or phrases that supply necessary information to support the action of the command. Not all CLI commands require parameters. The parts of parameters that have to be entered as predefined text are in uppercase italics and the variables are in lower-case italicized text.
- **Switches.** An optional word or phrase that modifies the command. Not all CLI commands require switches. Switches are represented in this manual as capitalized, italicized text.

ADD CONNECTIONS

Adds the specified host connection to the table of known connections. This table is maintained in NVRAM. The maximum table length is 32 connections; if the table contains 32 entries, new connections cannot be added unless some old ones are deleted.

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

- Physically connecting a host adapter to a controller port. During 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.

- 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=*n* ADAPTER_ID=*n* CONTROLLER=*controller* PORT=*n*

Parameters

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 !NEWCON*nn*.

> **NOTE:** 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=nnnn-nnnn-nnnn-nnnn

Host_ID is the worldwide name of the host. It is a 16-character hexadecimal number. The hyphens aren't necessary, but are recommended to avoid mistakes in entering the number.

ADAPTER_ID=nnnn-nnnn-nnnn-nnnn

Adapter_ID is the worldwide name of the adapter. It is a 16-character hexadecimal number. The hyphens aren't necessary, but are recommended to avoid mistakes in entering the number. Adapter_ID maps to the Fibre Channel convention port name.

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

CONTROLLER=THIS_CONTROLLER
CONTROLLER=OTHER_CONTROLLER

The controller parameter specifies whether the connection is to “this controller” or “other controller.”

PORT=1
PORT=2

The port parameter specifies which port, 1 or 2, the connection is on.

Switches

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 and normal mode, host connections on controller port 1 have an offset of 0 and host connections on controller port 2 have an offset of 100. These are the default offset values.

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

- 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.

> **NOTE:** In SET *controller* SCSI_VERSION=SCSI-3, the command console LUN (CCL) is presented as LUN 0 to every connection, superseding any unit assignments. See “ADD UNIT,” page 7–27, for more information.

OPERATING_SYSTEM=OS_name

Specifies the operating system of the host. The choices are:

- DIGITAL_UNIX
- IBM
- SNI
- SUN
- VMS
- WINNT
- SGI
- HP

Examples

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

```
CLI>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

Names a disk drive and adds it to the controller’s configuration.

NOTE: The controller supports a maximum of 72 storage devices, even though more than 72 target IDs are available. Do not exceed the maximum number of devices in the subsystem.

Syntax

ADD DISK *container-name scsi-port-target-lun*

Parameters

container-name

Assigns a name to the disk device. This is the name used with the ADD UNIT command to create a single-disk unit.

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

IMPORTANT: 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.

scsi-port-target-lun

Indicates the SCSI device PTL address. Place one space between the port number, target number, and the two-digit LUN number when entering the PTL address. See “Mapping the Physical Devices with Device PTL Addressing,” page 3–6, for an explanation of the PTL addressing naming format.

NOTE: See the *HSG80 Array Controller ACS V8.4 Release Notes* to determine whether the disk drive you are planning to use is compatible with the controller.

Switches

NOTRANSPORTABLE (Default)
TRANSPORTABLE

Indicates whether a disk drive can be accessed exclusively by StorageWorks controllers.

If the NOTRANSPORTABLE switch is specified, the controller makes a small portion of the disk inaccessible to the host. This restricted space is used to store information (metadata) that is used to improve data reliability, error detection, and the ability to recover data. Because of this metadata, only StorageWorks controllers can retrieve data from non-transportable devices.

Transportable disk drives do not contain any metadata or restricted areas. Therefore, transportable disks forfeit the advantage metadata provides but can be moved to a non-StorageWorks environment with their data intact. Disks that are to be used in storagesets cannot be set as transportable.

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:** Compaq recommends you avoid using transportable disks unless there is no other way to move the data.

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.

Examples

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

```
ADD DISK DISK10000 1 0 0
```

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

```
ADD DISK DISK40200 4 2 0 TRANSPORTABLE
```

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

```
ADD DISK DISK30200 3 2 0 NOTRANSPORTABLE
TRANSFER_RATE_REQUESTED=10MHZ
```

This example creates a host-addressable unit after the disk is added:

```
INITIALIZE DISK20000

ADD UNIT D199 DISK20000
```

See also

ADD MIRRORSET

ADD UNIT

DELETE *container-name*

LOCATE

SHOW DISKS

SHOW DEVICES

SET *container-name*

ADD MIRRORSET

Names a mirrorset and adds it to the controller configuration.

Syntax

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

Parameters

mirrorset-name

Assigns a name to the mirrorset. 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 9 characters including letters A through Z, numbers 0 through 9, periods (.), dashes (-), or underscores (_).

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 both the type of container and its unique identifier.

disk-name1 [disk-nameN]

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

Switches

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 during copying.

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 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 (see “SET mirrorset-name,” page 7–110).

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 members 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 add DISK10000, DISK20100, and DISK30200 as a mirrorset with the name MIRR1, type:

```
ADD DISK DISK10000 1 0 0
ADD DISK DISK20100 2 1 0
ADD DISK DISK30200 3 2 0
ADD MIRRORSET MIRR1 DISK10000 DISK20100 DISK30200
```

The following example shows how to create a host-addressable unit after the mirrorset MIRR1 has been created:

```
INITIALIZE MIRR1
ADD UNIT D104 MIRR1
```

See also

ADD DISK

ADD UNIT

DELETE *container-name*

INITIALIZE

MIRROR

REDUCE

SHOW *mirrorset-name*

SHOW MIRRORSETS

SHOW STORAGESETS

UNMIRROR

ADD RAIDSET

Names a RAIDset and adds the RAIDset to the controller's 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 number of members in the storageset is determined by the number of containers specified by the *container-name* parameter in the command. The data capacity of the RAIDset is determined by the storage size of the smallest member.

Syntax

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

Parameters

RAIDset-name

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

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

IMPORTANT: 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.

container-name1 container-name2 [container-nameN]

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

Switches

POLICY=BEST_FIT
POLICY=BEST_PERFORMANCE (Default)
NOPOLICY

Set 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 (smallest disk drive at the time the RAIDset was initialized) of the remaining members of the RAIDset. If more than one disk drive in the spareset is the correct size, the controller selects a disk drive giving 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 the 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. See “SET RAIDset-name,” page 7–121, for more information regarding this procedure.

RECONSTRUCT=FAST
RECONSTRUCT=NORMAL (Default)

Sets the speed at which the controller reconstructs data to a new RAIDset disk that replaces the 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; therefore, 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 a member).

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

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

Examples

To create a RAIDset named RAID9 that contains disks DISK10000, DISK20100, and DISK30200, use the following commands:

```
ADD DISK DISK10000 1 0 0
ADD DISK DISK20100 2 1 0
ADD DISK DISK30200 3 2 0
ADD RAIDSET RAID9 DISK10000 DISK20100 DISK30200
```

This example shows how to create a RAIDset named RAID8 that contains disks DISK10000, DISK20100, and DISK30200 and uses the BEST_FIT switch to indicate the replacement policy. Enter the ADD RAIDSET command on one line.

```
ADD DISK DISK10000 1 0 0
ADD DISK DISK20100 2 1 0
ADD DISK DISK30200 3 2 0
ADD RAIDSET RAID8 DISK10000 DISK20100 DISK30200 POL-
ICY=BEST_FIT
```

This example creates RAIDset RAID8 and then creates a host-addressable unit.

```
INITIALIZE RAID8
```

```
ADD UNIT D70 RAID8
```

This example shows how you can create a three-member RAIDset from the members of a reduced four-member RAIDset. Do not initialize the RAIDset again.

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

```
ADD DISK DISK10300 1 3 0
ADD DISK DISK20400 2 4 0
ADD DISK DISK30200 3 2 0
ADD RAIDSET RAID6 DISK10300 DISK20400 DISK30200 REDUCED
```

See also

ADD UNIT

DELETE *container-name*

SET RAIDSET

SHOW RAIDSET

SHOW *RAIDset-name*

SHOW STORAGESETS

INITITALIZE

ADD REMOTE_COPY_SETS

IMPORTANT: This command works in a remote copy environment only using HSG80 ACS V8.4P code, where local (initiator) and remote (target) sites coexist.

Creates a remote copy set (RCS) in multiple-bus failover that comprises one unit at the initiator site and one unit at the target site. The units may be a single disk or part of a storageset, mirrorset, or RAIDset. Up to 2 units may be assigned to each remote copy set for a total of 8 copy sets.

Syntax

ADD REMOTE_COPY_SETS *remote-copy-set-name initiator unit-name target path-name*

Parameters

remote-copy-set-name

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 (_).

IMPORTANT: It is common to name a remote copy set RCS*n*, where *n* is a sequentially-assigned, unique identifier. This naming convention presents the user with the type of container and its unique identifier.

Switches

initiator unit-name target path-name

Specifies the name of the controller pair—called a *node*—receiving the command.

The initiator unit name is the logical unit name specified with the ADD UNIT command. The target path-name consists of two parts: 1) the controller pair name specified with the SET *controller* REMOTE_COPY= command and 2) the remote unit name, which you create. Both parts are separated with a back slash.

The combined *initiator unit-name target path-name* must be typed as follows:

- initiator unit name controller pair name\remote unit name

Examples

Create a host-addressable unit D1 on the local controller:

```
ADD DISK DISK10000 1 0 0
ADD DISK DISK20100 2 1 0
ADD STRIPESET STRIPE1 DISK10000 DISK20100
INITIALIZE STRIPE1
ADD UNIT D1 STRIPE1
```

Create a host-addressable unit D21 on the remote controller:

```
ADD DISK DISK30200 3 2 0
ADD DISK DISK40300 4 3 0
ADD STRIPESET STRIPE1 DISK30200 DISK40300
INITIALIZE STRIPE1
ADD UNIT D21 STRIPE1
```

Name the controller pair LONDON, which consists of the local controller and the remote controller:

```
SET THIS_CONTROLLER REMOTE_COPY=LONDON
```

Add the remote copy set RCS1:

```
ADD REMOTE_COPY_SET RCS1 D1 LONDON\D21
```

See also

ADD REMOTE_COPY_SETS

SET *controller* REMOTE_COPY

SET *remote-copy-set-name*

ADD SPARESET

Adds a disk drive to the spareset.

Syntax

ADD SPARESET *disk-name*

Parameter

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.

Example

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

```
ADD DISK DISK20200 2 2 0
ADD DISK DISK30300 3 3 0
ADD SPARESET DISK20200
ADD SPARESET DISK30300
```

See also

DELETE SPARESET

SET FAILEDSET

SHOW SPARESET

SHOW STORAGESETS

ADD STRIPESET

Names a stripeset and adds it to the controller configuration. Stripesets are sometimes 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

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. Stripeset 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 9 characters including letters A through Z, numbers 0 through 9, periods (.), dashes (-), or underscores (_).

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

NOTE: There is a 240 character limit for the command line. If you are configuring a stripeset with multiple members (for example, more than 20) you will have to rename the members in order to execute the command.

Examples

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

```
ADD DISK DISK10000 1 0 0
ADD DISK DISK20100 2 1 0
ADD DISK DISK30200 3 2 0
ADD STRIPESET STRIPE1 DISK10000 DISK20100 DISK30200
```

To create a stripeset named STRIPE1 and then create a logical unit from it, type:

```
INITIALIZE STRIPE1
ADD UNIT D103 STRIPE1
```

This example shows how to create a two-member striped mirrorset (a stripeset whose members are mirrorsets), and how to create a logical unit from it. Because you can initialize the stripeset, you do not need to individually initialize the mirrorsets.

```
ADD DISK DISK10000 1 0 0
ADD DISK DISK20100 2 1 0
ADD DISK DISK30200 3 2 0
ADD DISK DISK40300 4 3 0
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

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

If you add a newly-created storageset or disk to your subsystem, you must initialize it before it can be added as a logical unit. If you are adding a storageset or disk that has data on it that you want to maintain, do not initialize it; it will be added as logical unit.

Syntax

ADD UNIT *unit-number container-name*

Parameters

unit-number

IMPORTANT: Once you change to the SCSI-3 version in OpenVMS operating systems for multiple-bus failover, you must use a special number called a *unit identifier*, which makes the unit visible to the host. See the *IDENTIFIER* switch with the command "SET unit-number," page 7–126 for more information.

Assigns a number to the unit being created from a device, container, or partition in the subsystem. The host uses this number to indicate the source or target for every I/O request it sends to the controller. The *unit-number* is a host-addressable LUN. The *unit-number* is assigned to one of the host ports.

Unit numbers are 0-199 and are prefixed by one of the following:

- Transparent failover mode:
  - ❏ D0 to D99—assigns units to Port 1
  - ❏ D100 to D199—assigns units to Port 2
- Multiple-bus failover mode:
  - ❏ D0 to D199—assigns units on both ports

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 D100 to the host on port 2. Units must be on a single port. Do not split partitioned units across ports.

In multiple-bus failover mode, adding unit D0 creates a logical unit that is presented to both ports on that controller. However, the other controller's ports can also access that unit if given permission by the host.

The LUN number a host connection assigns to a LUN is a function of the UNIT_OFFSET qualifier in the ADD (or SET) connections command: LUN number = unit number - offset

If no value is specified for the UNIT_OFFSET qualifier in the ADD (or SET) CONNECTIONS command, then host connections on controller port 1 have an offset of 0 and host connections on controller port 2 have an offset of 100. These are the default offset values.

container-name

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

A maximum of 48 devices can make up one unit.

Switches

Table 7–2 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.

Table 7–2. ADD UNIT Switches for Storagesets

Switch Container Type	ENABLE_ACCESS_PATH DISABLE_ACCESS_PATH	PARTITION	MAXIMUM_CACHED_ TRANSFER_SIZE	PREFERRED_PATH NOPREFERRED_PATH	READ_CACHE NOREAD_CACHE	READAHEAD_CACHE NOREADAHEAD_CACHE	WRITE_PROTECT NOWRITE_PROTECT	WRITEBACK_CACHE NOWRITEBACK_CACHE	RUN NORUN
RAIDset	✔	✔	✔	✔	✔	✔	✔	✔	✔
Stripeset	✔	✔	✔	✔	✔	✔	✔	✔	✔
Mirrorset	✔	✔	✔	✔	✔	✔	✔	✔	✔
NoTransportable Disk	✔	✔	✔	✔	✔	✔	✔	✔	✔
Transportable Disk	✔		✔	✔	✔	✔	✔		✔

NOTE: Regardless of the storageset type, you cannot specify RUN and NORUN for partitioned units.

ENABLE_ACCESS_PATH=
DISABLE_ACCESS_PATH=

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.

If you enable another host ID, previously enabled hosts are not disabled. The new IDs are added. If you wish to enable only certain IDs, disable all access paths (DISABLE_ACCESS_PATH=ALL), then enable the desired IDs. The system will display the following message:

```
Warning 1000: Access IDs in addition to the one(s) specified are still
enabled. If you wish to enable ONLY the id(s) listed, disable all
access paths (DISABLE_ACCESS_PATH=ALL), then enable the ones
previously listed.
```

NOTE: To enable access by more than one host connection, list the connection names separated by commas and enclosed in parenthesis. Enabling access by more than one host connection can also be done by sequential commands.

PARTITION=partition-number

Identifies the unit 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. The subsystem assigns units 0-99 to Port 1; units 100-199 are assigned to Port 2. Partitioned units must be on a single port.

Transportable units cannot be partitioned.

MAXIMUM_CACHED_TRANSFER_SIZE=32 (Default)
MAXIMUM_CACHED_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_CACHED_TRANSFER_SIZE switch affects both read and write-back cache when set on a controller that has read and write-back caching.

PREFERRED_PATH=OTHER_CONTROLLER
PREFERRED_PATH=THIS_CONTROLLER
NOPREFERRED_PATH (Default)

Used only in Windows NT operating systems.

May be set only when dual-redundant controllers are operating in a multiple-bus failover configuration. In a multiple bus failover configuration, the host determines which controller the units are accessed through. The host's unit-to-controller settings always take precedence over the preferred path assigned to units with this switch.

> **NOTE:** If your controllers are configured to operate in transparent-failover mode, do not set the PREFERRED_PATH switch with the ADD UNIT or SET *unit-number* command—otherwise, an error message is displayed. The error message indicates the assignment of a preferred controller path at the unit level is valid only when operating in multiple bus failover mode.

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.

> **IMPORTANT:** Subsystem performance is better if target ID numbers are balanced across the dual-redundant pair.

READ_CACHE (Default)
NOREAD_CACHE

Sets the controller's cache read policy function. 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 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 (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
NOWRITE_PROTECT (Default)

Tells the controller whether data contained on the unit can be overwritten.

Specify WRITE_PROTECT to prevent the host from writing data to the unit. However, the controller may still write to a write-protected RAIDset to complete a reconstruct operation and metadata, reconstruct data, and copy data may still be written 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

Enable or disable the write-back data caching function of the controller. The controller's write-back caching feature improves write performance. WRITEBACK_CACHE is the default on transportable disks.

Specify WRITEBACK_CACHE for all new RAIDsets, mirrorsets, and units you want to take advantage of the controller's write-back caching feature.

Specify NOWRITEBACK_CACHE for units you want to receive data directly from the host without being cached.

⚠ **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 a catastrophic subsystem failure occurs.

NOTE: The controller may take up to five minutes to flush data contained within the write-back cache when you specify the NOWRITEBACK_CACHE switch.

Examples

This example shows how to create unit D102 from a single-disk drive named DISK10000 and sets the host’s access to the unit through “this controller.”

```
ADD DISK DISK10000 1 0 0
INITIALIZE DISK10000
ADD UNIT D102 DISK10000
PREFERRED_PATH=THIS_CONTROLLER
```

This example shows how to create unit D107 from a RAIDset named RAID9 and instructs the unit to take advantage of the controller’s write-back caching feature.

```
ADD DISK DISK10100 1 1 0
ADD DISK DISK20100 2 1 0
ADD DISK DISK30100 3 1 0
ADD DISK DISK40100 4 1 0
ADD RAIDSET RAID9 DISK10100 DISK20100 DISK30100 DISK40100
INITIALIZE RAID9
ADD UNIT D107 RAID9 WRITEBACK_CACHE
```

See also

CREATE_PARTITION

DELETE *unit-number*

SET *unit-number*

SHOW UNITS

CLEAR_ERRORS CLI

Stops the display of current or previous error messages at the CLI prompt. This command does not clear the error conditions, it only stops the display of errors at the CLI prompt.

After the cause of the error condition has been corrected, issue the CLEAR_ERRORS CLI command to clear the error message.

> **NOTE:** There are three message types: info—general information; warning—user may want to examine, but command will be executed; and error—command will not execute.

Syntax

CLEAR_ERRORS CLI

Example

To clear the message "All NVPM components initialized to their default settings" from the CLI prompt, type:

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

See also

CLEAR_ERRORS INVALID_CACHE

CLEAR_ERRORS LOST_DATA

CLEAR_ERRORS UNKNOWN

CLEAR_ERRORS UNWRITEABLE_DATA

CLEAR_ERRORS *controller* INVALID_CACHE

Clears an invalid cache error and allows the controller and cache to resume operation. If the error is due to an incorrectly-mirrored 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 COPY= commands.

Syntax

CLEAR_ERRORS *controller* INVALID_CACHE

Spell out INVALID_CACHE when using this command.

Parameters

controller

Identifies which controller is to receive the CLEAR_ERRORS command. You must specify THIS_CONTROLLER or OTHER_CONTROLLER.

data-retention-policy
DESTROY_UNFLUSHED_DATA
NODESTROY_UNFLUSHED_DATA (Default)

Instructs the controller on how to handle write-back cached data.

Specify NODESTROY_UNFLUSHED_DATA (default) to retain the cached data and discard controller information.

Specify DESTROY_UNFLUSHED_DATA to retain the controller information and discard the cached data.

Specify NODESTROY_UNFLUSHED_DATA in the following situations:

- If the controller module has been replaced
- If the controller's nonvolatile memory (NVMEM) has lost its contents.

Specify DESTROY_UNFLUSHED_DATA in the following situations:

- If the cache module has been replaced
- Any other reason not listed above

CAUTION: Specifying the DESTROY_UNFLUSHED_DATA switch destroys data remaining in cache, which can result in data loss.

Examples

This example shows how to clear an invalid cache error on "this controller" after you have replaced a controller module. Enter the command on one line.

```
CLEAR_ERRORS THIS_CONTROLLER INVALID_CACHE
NODESTROY_UNFLUSHED_DATA
```

This example shows how to clear an invalid cache error on the "other controller" after a cache module has been replaced. Enter this command on the same line. Enter the 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

If a device failure causes the controller to label the device as unknown, 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.

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

Syntax

CLEAR_ERRORS *device-name* UNKNOWN

Spell out UNKNOWN when using this command.

Parameters

device-name

Identifies the device with the unknown error.

Example

To force the controller to recognize a previously unknown device named DISK30000, enter this command:

```
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

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 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

Spell out LOST_DATA when using this command.

> **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.

Example

The following command will clear the lost data error on disk unit number D103:

```
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

Clears an unwriteable data error on a unit.

It 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 UNWRITEABLE_DATA command removes the data from the cache and clears the unwriteable data error.

△ **CAUTION:** This command causes data loss.

Syntax

CLEAR_ERRORS *unit-number* UNWRITEABLE_DATA

Spell out UNWRITEABLE_DATA when using this command.

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.

Example

Use the following command to clear the unwriteable data error on disk unit D103:

```
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

Erases the entire configuration on “this controller,” restores the controller’s default configuration, and shuts down the controller.

> **NOTE:** If you plan to use this feature, SAVE_CONFIGURATION must be set when you initialize the container. See “INITIALIZE,” page 7–62.

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 is moved from one subsystem to another.

This command disables communication between host and controller. Enter new configuration information through the SET THIS_CONTROLLER command or the CONFIGURATION RESTORE command to make the controller operational.

You can also initiate the CONFIGURATION RESET command from the controller’s operator control panel (OCP) by holding in port button 5 and pressing the reset button.

Syntax

CONFIGURATION RESET

See also

CONFIGURATION RESTORE

CONFIGURATION SAVE

INITIALIZE

CONFIGURATION RESTORE

Copies a controller’s configuration from the disk configuration file into the 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.

Use this command for a single controller configuration only. Do not use it for controllers in a dual-redundant configuration.

You can also initiate the CONFIGURATION RESTORE command from the controller’s operator control panel (OCP) by holding in port button 6 and pressing the reset button.

NOTE: The controller must not have devices configured prior to issuing this command. Use “CONFIGURATION RESET,” page 7–44, instead.

If the controller you’re installing was previously used in another subsystem, it will restart with the configuration that resides in its nonvolatile memory. If this differs from the subsystem’s current configuration, you can purge the controller’s old configuration with the following command:

```
configuration RESET
```

This will erase the entire configuration on the controller, restore the controller’s default configuration, and shut down the controller. Press its reset button to restart the controller after the controller has been configured. See Chapter 5, “Configuring an HSG80 Array Controller.”

NOTE: The INITIALIZE *container-name* SAVE_CONFIGURATION must be used to save the controller’s configuration to a disk (see “SAVE_CONFIGURATION,” page 7–64), in order to reset the configuration (see “CONFIGURATION RESET,” page 7–44) or to restore the configuration (see “CONFIGURATION RESTORE,” page 7–45).

Syntax

CONFIGURATION RESTORE

See also

CONFIGURATION RESET

CONFIGURATION SAVE

INITIALIZE

CONFIGURATION SAVE

Forces a current copy of configuration information in a controller’s non-volatile memory into a configuration file on a disk. This allows the user to determine when a copy of the configuration is saved. Use this command to explicitly save a single controller’s configuration. The command takes effect immediately. In a dual-redundant configuration, issue this command to both controllers.

Use the INITIALIZE *container-name* SAVE_CONFIGURATION command to set up the location of the configuration file on disk.

Syntax

CONFIGURATION SAVE

See also

CONFIGURATION RESET

CONFIGURATION RESTORE

INITIALIZE

CREATE_PARTITION

Divides a non-transportable disk drive storageset into several, separately-addressable storage units. The command marks a specified percentage of a disk drive or storageset to be used as a separately addressable unit. You can divide any nontransportable disk or storageset into a maximum of eight partitions. Each partition can be separately presented to the host. Partitions are not supported in multiple bus failover mode. Initialize disks and storagesets before creating partitions.

> **NOTE:** Partitioned units cannot function in multiple bus failover dual-redundant configurations. Because they are not supported, you must delete your partitions before configuring the controllers for multiple bus failover.

After you partition a container, you must initialize it in order to destroy the 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 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.

SIZE=percent
SIZE=LARGEST

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

To create a partition, specify a percentage of the container's total capacity. The entire container is then divided into segments equal to the percentage specified. For example, if SIZE=20, the container is divided into five (1.0/0.2=5) equal segments. The resulting partition is slightly smaller than the size specified because metadata also occupies some of the partition's allocated space.

Specify LARGEST in the following situations:

- To have the controller create the largest partition possible from unused space on the disk or storageset.
- To 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=
CYLINDERS=
HEADS=
SECTORS_PER_TRACK=

CAPACITY may be specified 1 to the maximum container size (in blocks); CYLINDERS may be specified 1 to16,777,215; HEADS may be specified 1 to 255; and SECTORS_PER_TRACK may be specified 1 to 255.

> **NOTE:** These are used to set the SCSI parameters reported to the host. They should not be used unless there is a compatibility problem with the existing defaults.
>
> The geometry parameter switches for the INITIALIZE command are ignored when you create partitions. The parameters supplied with the CREATE_PARTITION command are used by the unit.

Example

The next example shows how to create a RAIDset named RAID9 and divide it into four equal parts. It also creates host-addressable units for each partition.

```
ADD DISK DISK10000 1 0 0
ADD DISK DISK20100 2 1 0
ADD DISK DISK30200 3 2 0
ADD RAIDSET RAID9 DISK10000 DISK20100 DISK30200
INITIALIZE RAID9
CREATE_PARTITION RAID9 SIZE=25
CREATE_PARTITION RAID9 SIZE=25
CREATE_PARTITION RAID9 SIZE=25
CREATE_PARTITOIN 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

SHOW

DELETE *connections*

Deletes a host connection entry from the table of known 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 controllers NVRAM. 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 controlled port.

> **NOTE:** A connection that has access path explicitly enabled on a unit cannot be deleted. Access path is enabled through 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.

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.

Examples

Deletes the host connection Server1 from the table of known connections (unless the access path to Server1 is specifically enabled for one or more unit.)

```
CLI> DELETE SERVER1
```

See also

ADD CONNECTIONS

ADD UNIT

SET *connection-name*

SET *unit-number*

DELETE *container-name*

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.

> **NOTE:** This command does not delete sparesets or failedsets. You cannot delete spareset and failedset containers. See the DELETE FAILEDSET and DELETE SPARESET commands for details.

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, 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).

Examples

To delete a disk drive named DISK10000, type:

```
DELETE DISK10000
```

To delete a stripeset named STRIPE1, enter:

```
DELETE STRIPE1
```

To delete a RAIDset named RAID9, use:

```
DELETE RAID9
```

See also

DELETE FAILEDSET

DELETE SPARESET

UNMIRROR

DELETE FAILEDSET

Removes a disk drive from the failedset. The failedset contains disk drives removed by the controller from RAIDsets and mirrorsets because they failed or were manually removed using the SET command. Enter the DELETE FAILEDSET command before physically removing failed disks from the storage shelf for testing, repair, or replacement.

You should consider all disk drives in the failedset defective. Repair or replace disks found in the failedset.

Syntax

DELETE FAILEDSET *disk-name*

Parameter

disk-name

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

Example

To delete DISK20200 from the failedset, use the following command:

```
DELETE FAILEDSET DISK20200
```

See also

DELETE *container-name*

DELETE SPARESET

SET FAILEDSET

SHOW FAILEDSET

DELETE SPARESET

Removes a disk drive from the spareset.

Syntax

DELETE SPARESET *disk-name*

Parameter

disk-name

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

Example

This command will remove DISK20300 from the spareset:

```
DELETE SPARESET DISK20300
```

See also

DELETE *container-name*

DELETE FAILEDSET

ADD SPARESET

SHOW SPARESET

DELETE *unit-number*

Deletes a logical unit from the controller configuration. The host cannot address deleted units. If the unit's write-back caching feature is enabled, the controller flushes the cached data to the unit's devices before deleting the unit.

Before using the DELETE *unit-number* command, clear any errors with the CLEAR_ERRORS UNWRITEABLE_DATA or CLEAR_ERRORS LOST_DATA commands.

Syntax

DELETE *unit-number*

Parameter

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.

Example

To delete disk unit number D103, enter:

```
DELETE D103
```

See also

ADD UNIT

CLEAR_ERRORS LOST_DATA

CLEAR_ERRORS UNWRITEABLE_DATA

DESTROY_PARTITION

DESTROY_PARTITION

Marks the area reserved for a partition as available. 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.

You cannot destroy a partition that has been assigned a unit number. First enter the DELETE *unit-number* command to delete the unit using the partition.

After you partition a container, you must initialize it in order to destroy the partitions.

Syntax

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

Parameters

container-name

Identifies the disk or storageset containing the partition to be destroyed. 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).

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.

Example

The following example shows how to delete the unit for partition 2 on RAIDset RAID9 and destroy the partition:

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

See also

ADD DISK

ADD STORAGESET

ADD STRIPESET

CREATE_PARTITION

DELETE *unit-number*

SHOW

DIRECTORY

Lists the diagnostics and utilities available on “this controller.”

Syntax

DIRECTORY

Example

The example below shows how to display a directory listing:

```
           DIRECTORY
HSUTIL  V84G   D
CHVSN   V84G   D
CLCP    V84G   D
CLONE   V84G   D
CONFIG  V84G   D
DILX    V84G   D
DIRECT  V84G   D
DSTAT   V84G   D
FRUTIL  V84G   D
FMU     V84G   D
VTDPY    V84G   D
```

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

See also

RUN

HELP

Displays a brief explanation of how to use the question mark (?) to obtain help on any command or CLI function. You must precede the question mark with a space.

Syntax

HELP

Example

To display information regarding the HELP command, type:

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

SET THIS_CONTROLLER ?

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

The following example shows how to get help on the SET command using the question mark (?):

```
               SET ?
Your options are:

    EMU
    FAILEDSET
    FAILOVER
    NOFAILOVER
    OTHER_CONTROLLER
    THIS_CONTROLLER
    Unit number or mirrorset or raidset or device name
```

INITIALIZE

Initializes or destroys metadata on a container. During initialization, a small amount of disk space is reserved for controller metadata and is made inaccessible to the host. Disks made transportable do not contain controller metadata.

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.

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 before:

- 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 command (for example, ADD DISK, ADD STRIPESET, and so forth).

Switches

CAPACITY=
CYLINDERS=
HEADS=
SECTORS_PER_TRACK=

CAPACITY may be specified 1 to the maximum container size (in blocks); CYLINDERS may be specified 1 to16,777,215; HEADS may be specified 1 to 255; and SECTORS_PER_TRACK may be specified 1 to 255.

> **NOTE:** These are used to set the SCSI parameters reported to the host. They should not be used unless there is a compatibility problem with the existing defaults.
>
> The geometry parameter switches for the INITIALIZE command are ignored when you create partitions. The parameters 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 determine the optimal chunk block size by entering CHUNKSIZE=DEFAULT.

> **NOTE:** The CHUNKSIZE switch is only valid with stripesets and raidsets.

The default chunk size for storagesets with less than 9 members is 256 blocks, or 128 kilobytes (K). The default chunk size for storagesets with more than 9 members is 128 blocks, or 64K. The default values provide optimal storageset performance for a wide variety of applications. A chunk size less than 128 blocks (64K) is not recommended.

IMPORTANT: Accept the default chunk size setting for most applications. Do not change the default setting unless you are fully aware of the impact to the storageset's performance.

See "Chunk Size," page 4–43, for information regarding recommended chunk size settings for your application.

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 and single-disk configurations.

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

SAVE_CONFIGURATION
NOSAVE_CONFIGURATION (Default)

Instructs the controller whether to save the controller's configuration to the container being initialized.

The SAVE_CONFIGURATION switch requires only one disk to be initialized with this option. However, more disks may be used, if desired, for redundancy.

Specify SAVE_CONFIGURATION to store a copy of the controller configuration on the container being initialized. A new controller can receive information from a container containing configuration information saved with the SAVE_CONFIGURATION switch. If you specify SAVE_CONFIGURATION for a multi-device storageset, such as a stripeset, the complete controller configuration information is stored on each disk drive in the storageset.

A disk drive initialized with the SAVE_CONFIGURATION switch specified has slightly less storage space available for user data.

Specify NOSAVE_CONFIGURATION if you do not want to store a copy of the controller configuration on a container.

See "Backing Up Your Subsystem Configuration," page 4–26, for more information on SAVE_CONFIGURATION.

Examples

To initialize container DISK10000 and save a copy of the controller configuration on it, enter the following commands:

```
ADD DISK DISK10000 1 0 0
INITIALIZE DISK10000 SAVE_CONFIGURATION
```

The following example shows sample devices with the SAVE_CONFIGURATION switch enabled:

```
SHOW DEVICES FULL
```

```
Name          Type                    Port Targ  Lun        Used by
------------------------------------------------------------------------------
DISK10000     disk                       1    0    0        S2
          DEC      RZ28M    (C) DEC 1003
        Switches:
          NOTRANSPORTABLE
          TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 10.00 MHZ negotiated)
        Size: 4108970 blocks
        Configuration being backed up on this container
DISK30300     disk                       3    3    0        S2
          DEC      RZ28M    (C) DEC 1003
        Switches:
          NOTRANSPORTABLE
          TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 10.00 MHZ negotiated)
        Size: 4108970 blocks
        Configuration being backed up on this container
```

This example shows how to initialize stripeset STRIPE1 with the default chunk size. The chunk size is not specified, so the controller initializes the unit with the default chunk size.

```
ADD DISK DISK10100 1 1 0
ADD DISK DISK20100 2 1 0
ADD DISK DISK30100 3 1 0
ADD STRIPESET STRIPE1 DISK10100 DISK20100 DISK30100
INITIALIZE STRIPE1
```

This example shows how to initialize RAIDset RAID9 with a chunk size of 20:

```
ADD DISK DISK10200 1 2 0
ADD DISK DISK20200 2 2 0
ADD DISK DISK30200 3 2 0
ADD RAIDSET RAID9 DISK10200 DISK20200 DISK30200
INITIALIZE RAID9 CHUNKSIZE=20
```

This example shows how to initialize DISK40400 and preserve the data after it is removed (reduced) from a mirrorset:

```
REDUCE DISK40400
INITIALIZE DISK40400 NODESTROY
```

LOCATE

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 device fault LED flashes 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.

Syntax

LOCATE *parameter*

Parameters

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 turned on with the LOCATE command.

DISKS

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

PTL (SCSI-location)

Causes the device fault LED on the device at the given SCSI location to flash. See “Mapping the Physical Devices with Device PTL Addressing,” page 3–6 for an explanation of the PTL addressing naming format.

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

UNITS

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

container-name

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

Examples

This example shows how to cause the device fault LED on device DISK10000 to flash:

```
LOCATE DISK10000
LOCATE CANCEL
```

This example shows how to cause the device fault LEDs on all of the devices assigned to disk unit number D102 to flash:

```
LOCATE D102
```

This example shows how to cause the device fault LEDs on all configured disk devices to flash:

```
LOCATE DISKS
```

This example shows how to turn off the flashing device fault LEDs on all devices:

```
LOCATE CANCEL
```

MIRROR

Creates a one-member mirrorset from a single disk. This command is used only on disks configured as units or members of a stripeset, then enter the ADD MIRRORSET command to create a mirrorset from disk drives not already members of higher level containers.

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

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

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 during copying.

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 member 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 that has the best performance.

Specify POLICY=BEST_PERFORMANCE to choose a replacement disk drive from the spareset resulting in the best performance. The controller attempts to select a disk on a different port than existing 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 replaced in the mirrorset. See “SET mirrorset-name,” page 7–110.

Example

This example shows how to create a one-member mirrorset from each member of a stripeset. These commands set the nominal number of members in each mirrorset to two and adds 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 DISK DISK10100 1 1 0
ADD DISK DISK20100 2 1 0
ADD DISK DISK30100 3 1 0
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

Powers off all disk units in a cabinet and turns off the cabinet power.

Syntax

POWEROFF

Switches

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 poweroff the cabinet or remain on depending on the cache flush results.

Specify OVERRIDE_BAD_FLUSH to override a failed cache flush and poweroff the cabinet.

Specify NO_OVERRIDE_BAD_FLUSH to prevent a poweroff when the cache flush fails.

SECONDS=nn

As soon as 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 7–3 shows what action will be taken depending on the switch settings and the results of the attempted flush:

Table 7–3. 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 can't be shutdown, then both controllers and their batteries' chargers remain on.

Example

This example shows how to power off the disk units and the cabinet in 10 seconds (BATTERY_OFF and NO_OVERRIDE_BAD_FLUSH are the defaults):

```
POWEROFF SECONDS=10
```

REDUCE

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:

- Pauses I/O to the unit
- Flushes all of the unit’s data from write-back data cache
- Removes the specified disks
- Decreases the nominal number of members of the mirrorsets by the number of disks removed from the mirrorsets.

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.

> **NOTE:** An error is displayed if you attempt to reduce a mirrorset so that there would not be any normal member remaining.

Syntax

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

Parameters

disk-name1 disk-name2 disk-name3...

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.

Example

This example shows how to remove DISK20100, DISK20200, and DISK40200 from their respective mirrorsets:

```
                    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 DISK20500 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

Renames a specified container or a specified host connection.

Syntax

RENAME *old-name new-name*

Parameters

old-name

Specifies the existing name of the container or host connection.

new-name

Assigns the new name for the container or the host connection.

See “Command Syntax,” page 7–5, for information regarding container naming rules.

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 may not be renamed.

Example

This example shows how to rename DISK10000 to MYDISK:

```
                    SHOW DISKS
Name            Type          Port Targ Lun          Used by
-------------------------------------------------------------
DISK10000       disk            1    0    0            D100
DISK10100       disk            1    1    0            D101


                    RENAME DISK10000 MYDISK
                    SHOW DISKS
Name            Type           Port Targ Lun         Used by
-------------------------------------------------------------
MYDISK           disk           1    0    0            D100
DISK10100        disk           1    1    0            D101RESTART controller
```

RESTART *controller*

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 is to be restarted. Specify OTHER_CONTROLLER or THIS_CONTROLLER.

Switches

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 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. See “Fault-Tolerance for Write-Back Caching,” page 2–11 for considerations when implementing 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 without checking for online devices or before all data has been flushed from write-back cache to the devices.

Examples

This example shows how to restart “this controller”:

```
RESTART THIS_CONTROLLER
```

This example shows how to restart the “other controller”:

```
RESTART OTHER_CONTROLLER
```

See also

SELFTEST *controller*

SHUTDOWN *controller*

RETRY_ERRORS UNWRITEABLE_DATA

Causes the controller to attempt to write previously unwriteable data from the write-back cache to the devices. If a container fails, preventing the data in write-back cache to be written to the container, an unwriteable data error is reported. If possible, correct the condition that caused the unwriteable data and try the write operation again. 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.

Example

This example shows how to retry writing the cached data previously marked unwriteable to disk unit D103:

```
RETRY_ERRORS D103 UNWRITEABLE_DATA
```

See also

CLEAR_ERRORS *unit-number* UNWRITEABLE_DATA

RUN

Runs a diagnostic or utility program on “this controller.” Diagnostic and utility programs only run 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 the *HSG80 Array Controller ACS Version 8.3 and 8.4 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,” page 4–21, 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 “Adding Disk Drives,” page 6–3, for more information regarding this utility.
- 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.

  The DILX utility has two modes, an autoconfigure mode, and a standard mode.

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.

The autoconfigure mode is the most thorough mode and allows you to:

- ❑ Automatically test all of the disk units configured
- ❑ Automatically perform thorough tests on all units with writes enabled

The standard mode is more flexible and allows you to:

- ❑ Test disks you select
- ❑ Perform tests in read-only mode or write-only mode
- ❑ Provide run time and performance summary options
- ❑ Can be run in read-only mode

- ■ DIRECT—A command used to display a list 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 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, and I/O performance.

See the *HSG80 Array Controller ACS Version 8.3 and 8.4 Maintenance and Service Guide* for more information regarding the above utilities.

Example

This example shows how to start the DILX diagnostic program:

```
                    RUN DILX
     .
     .
     .
```

See also

DIRECTORY

SELFTEST *controller*

Flushes the data from the specified controller's write-back cache (if present) and shuts down the controller. It then restarts the controller in 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

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. See "Fault-Tolerance for Write-Back Caching," page 2–11, for considerations when implementing 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 or not.

CAUTION: The IMMEDIATE_SHUTDOWN switch instructs the controller to immediately shut down, without regard to any data contained within write-back cache. See "Fault-Tolerance for Write-Back Caching," page 2–11, for considerations when implementing 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 checking for online devices or without flushing user data from write-back cache to devices.

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

Examples

This example shows how to start the self-test program on "this controller:"

```
SELFTEST THIS_CONTROLLER
```

This example shows how to run the self-test program on the "other controller," even if the "other controller" cannot flush all data from the write-back cache:

```
SELFTEST OTHER_CONTROLLER IGNORE_ERRORS
```

See also

RESTART *controller*
SHUTDOWN *controller*

SET *connection-name*

Changes the operating characteristics of a host 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 an decimal number. The connection name can be changed through the RENAME command.

Switches

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.

If no value is specified for the UNIT_OFFSET switch, then host connections on controller port 1 have an offset of 0 and host connections on controller port 2 have an offset of 100. These are the default offset values.

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

- 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.

OPERATING_SYSTEM=OS_name

Specifies the operating system of the host. The choices are:

- DIGITAL_UNIX
- IBM
- SNI
- SUN
- VMS
- WINNT
- SGI
- HP

See also

ADD CONNECTIONS

ADD UNIT

DELETE *connections*

RENAME

SET *controller*

Changes parameters on the specified controller.

> **NOTE:** After doing a SET command to either controller, Compaq recommends using a SHOW THIS_CONTROLLER and a SHOW OTHER_CONTROLLER command to verify that the changes went into effect.

Syntax

SET *controller*

Parameter

controller

Indicates which controller is to be set. Specify OTHER_CONTROLLER or THIS_CONTROLLER.

Switches

Table 7–4 lists the switches available with this command. Descriptions of the switches follow the table.

Table 7–4. SET *controller* Switches

Switch	Values
ALLOCATION_CLASS	0-4294967295 In DIGITAL Open VMS, do not specify a number over 65535
CACHE_FLUSH_TIMER	1–65535 sec, 10 (default)
CACHE_UPS NOCACHE_UPS	None
COMMAND_CONSOLE_LUN NOCOMMAND_CONSOLE_LUN	None
IDENTIFIER NOIDENTIFIER	decimal number
MIRRORED_CACHE NOMIRRORED_CACHE	None
NODE_ID	assigned during manufacturing
PORT_1_AL_PA PORT_2_AL_PA	0-EF (hexadecimal value)
PORT_1_TOPOLOGY PORT_2_TOPOLOGY	FABRIC LOOP_HARD LOOP_SOFT OFFLINE
PROMPT	1–16 characters
REMOTE_COPY NOREMOTE_COPY	None

Table 7–4. SET *controller* Switches (Continued)

Switch	Values
SCSI_VERSION	SCSI-2 (default) SCSI-3
TERMINAL_PARITY NOTERMINAL_PARITY	odd, even
TERMINAL_SPEED	4800, 9600, 19200
TIME	dd–mmm–yyy:hh:mm:ss

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 DIGITAL Open VMS, this is a 2-byte number (do not specify a number over 65535); for DIGITAL 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. It allows the user to place a unique number in the allocation class value (n). 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 may elapse before the write-back cache flushes its entire contents to a given device or RAIDset. The default setting is 10 seconds. When changed, the new value entered for this switch takes effect immediately.

CACHE_UPS
NOCACHE_UPS (Default)

Specifies whether the controller should perform regular battery condition checks. When changed, you must restart both controllers in order for the new setting to take effect.

Specify CACHE_UPS if your storage subsystem power is supported by an uninterruptable power supply (UPS). 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.

CAUTION: Setting CACHE_UPS without having a UPS or similar backup system in place may result in data loss if power is interrupted.

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

Setting the CACHE_UPS switch for either controller sets the CACHE_UPS switch for both controllers.

COMMAND_CONSOLE_LUN
NOCOMMAND_CONSOLE_LUN (Default)

Enables or disables the virtual LUN used with the StorageWorks Command Console. When changed, the new setting for this switch takes effect immediately.

> **NOTE:** This switch enables (COMMAND_CONSOLE_LUN) and disables (NOCOMMAND_CONSOLE_LUN) the CCL in SCSI-2 mode only. This switch has no effect in SCSI-3 mode.

Select COMMAND_CONSOLE_LUN to enable the virtual LUN. Select NOCOMMAND_CONSOLE_LUN to disable the virtual LUN.

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 *CCL identifier* that makes the controller and the CCL visible to the host. This number can be a value between 1–9999.

From the controller perspective, the IDENTIFIER switch provides an alternative way (other than worldwide name) for OpenVMS operating systems to identify the CCL.

MIRRORED_CACHE
NOMIRRORED_CACHE (Default)

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. Both controllers must be operational before this command is accepted.

- 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.

Issue this switch through only one controller. The controller must contain a valid cache configuration before specifying this switch. See Chapter 2 for rules regarding valid cache configurations. 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 issued 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 checksum

Sets the subsystem worldwide name (node ID). If a situation occurs that requires you to reset the subsystem worldwide ID (node ID), use the name and checksum that appear on the sticker on the frame into which your controller is inserted.

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.

PORT_1_AL_PA=
PORT_2_AL_PA=

Used only in arbitrated loop (FC-AL) topology.

Specifies the hexadecimal arbitrated loop physical address (ALPA) for the host ports. Use this switch only when LOOP_HARD is specified for PORT_1_TOPOLOGY or PORT_2_TOPOLOGY. The range of addresses allowed is 0-EF (hexadecimal). The default value is 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

Used only in arbitrated loop (FC-AL) topology.

Indicates whether the user or controller selects the ALPA for a host port, or whether the port is to be set offline. 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.

PORT_1_TOPOLOGY=FABRIC
PORT_2_TOPOLOGY=FABRIC

Used only in switch (FC-SW) topology.

Specifies switch topology for a host port.

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 = "controller name"
NOREMOTE_COPY

IMPORTANT: This command works in a remote copy environment only using HSG80 ACS V8.4P code, where local (initiator) and remote (target) sites coexist.

Enables or disables remote copy at each site by naming the controller pair—called a *node*—at each site. When the name is set on one controller in the pair, the other controller automatically receives that same name, too. Once the name is set, remote copy becomes enabled at that site.

The controller name must be a meaningful name up to 8 characters long.

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

SCSI_VERSION=SCSI-2 (Default)
SCSI_VERSION=SCSI-3

Specifies the host protocol to use; requires operating system support. SCSI-3 is limited SCSI-3, meaning that only OpenVMS operating systems use SCSI-3 mode in multiple-bus failover mode. Once set, It also specifies how the command console LUN is handled. The command console LUN (CCL) presents to the GUI a virtual LUN through which it communicates with the controller.

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. The SCSI device-type returned to the host is *array controller.*

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

TERMINAL_PARITY=ODD
TERMINAL_PARITY=EVEN
NOTERMINAL_PARITY (Default)

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

TERMINAL_SPEED=baud_rate
TERMINAL_SPEED=9600 (Default)

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.

Examples

This example shows how to change the other controller’s CLI prompt:

```
SET OTHER_CONTROLLER PROMPT=CONTROLLER “B”
```

This example shows how to set the name of the controller pair at the initiator site:

```
SET THIS_CONTROLLER REMOTE_COPY=NEW_YORK
```

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

This example shows how to set the name of the controller pair at the target site:

```
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*

ADD REMOTE_COPY_SETS

SHOW THIS_CONTROLLER

SHOW OTHER_CONTROLLER

SET *device-name*

Changes the transportable characteristics and the maximum data transfer rate between the controller and 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

TRANSFER_RATE_REQUESTED=ASYNCHRONOUS
TRANSFER_RATE_REQUESTED=20MHZ (Default)
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.

TRANSPORTABLE
NOTRANSPORTABLE (Default)

Indicates whether a disk can be accessed exclusively by StorageWorks controllers.

Set the TRANSPORTABLE switch for disks only.

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:** Compaq recommends you avoid specifying TRANSPORTABLE unless transportability of the device or media is imperative and there is no other way to accomplish moving the data.

Examples

This example shows how to set the data transfer rate of DISK20000 to 5MHz:

```
SET DISK20000 TRANSFER_RATE_REQUESTED=5MHZ
```

This example shows how to set DISK10300 to transportable:

```
SET DISK10300 TRANSPORTABLE
```

See also

ADD DISK

SHOW DISKS

SET EMU

Sets operating parameters for the environmental monitoring unit (EMU).

Syntax

SET EMU

Switches

The SENSOR and FANSPEED switches control 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 SENSOR and FANSPEED settings to which the master EMU is set.

SENSOR_1_SETPOINT=nn
SENSOR_2_SETPOINT=nn
SENSOR_3_SETPOINT=nn
SENSOR_x_SETPOINT=35 (Default)

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 7–5 lists the valid EMU set-point temperatures in both Fahrenheit and Celsius.

Table 7–5. EMU Set Point Temperatures

ºC	ºF	ºC	ºF	ºC	ºF	ºC	ºF	ºC	ºF
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

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 (Default)

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.

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

Changes the automatic replacement policy for the failedset.

Syntax

SET FAILEDSET

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 moves the disk 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 steps to make the disk available as a spareset replacement disk:

These steps use DISK10000 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. Move DISK10000 to the failedset’s vacant slot.

Example

This example shows how to enable the automatic spare feature:

```
SET FAILEDSET AUTOSPARE
```

This example shows how to disable the automatic spare feature:

```
SET FAILEDSET NOAUTOSPARE
```

See also

SHOW FAILEDSET

SET FAILOVER

Configures both controllers to operate in a dual-redundant, transparent failover, configuration. This allows both controllers to access the storage devices, providing controller fault-tolerant data processing. If one of the two controllers fails, the devices and any cache attached to the failed controller become available to and accessible through the other controller.

> **NOTE:** The controllers must be present and placed in non-failover mode by entering the SET NOFAILOVER command before they can be set to failover mode.

Syntax

SET FAILOVER COPY=*controller*

Parameters

THIS_CONTROLLER
OTHER_CONTROLLER

Specifies which controller 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 one minute to complete.

Example

This example shows how to set the controllers in a dual-redundant configuration and copy the configuration information from “this controller” to “other controller:”

```
SET FAILOVER COPY=THIS_CONTROLLER
```

See also

SET MULTIBUS_FAILOVER

SET NOFAILOVER

SET NOMULTIBUS_FAILOVER

SET *mirrorset-name*

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

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 during copying.

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 specify a replacement policy with the POLICY= switch, 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.

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.

If the mirrorset won’t have a normal or normalizing member remaining after you remove the disk drive, the controller reports an error and no action is taken. A normal or normalizing member is a mirrorset member whose contents are the same as all other normal members.

For each reduced mirrorset, there must be at least one remaining normal member after the reduction.

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 the 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 data existing before the mirrorset was created on the normal member to the normalizing members. The controller recognizes the normalizing members as normal when the mirrorset member's blocks are all the same.

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.

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 meet the criteria, the controller selects a 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 the 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 (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 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.

Examples

This example shows how to change the replacement policy of mirrorset MIRR1 to BEST_FIT:

```
SET MIRR1 POLICY=BEST_FIT
```

This example shows how to remove member DISK30000 from mirrorset MIRR1 created above. If the mirrorset has a replacement policy and an acceptable disk drive is in the spareset, the controller automatically adds the spare disk drive to the mirrorset.

```
SET MIRR1 REMOVE=DISK30000
```

This example shows how to add disk DISK30200 to the mirrorset MIRR1:

```
SET MIRR1 REPLACE=DISK30200
```

A copy operation begins immediately on DISK30200.

See also

ADD MIRRORSET

MIRROR

REDUCE

SHOW MIRRORSET

UNMIRROR

SET MULTIBUS_FAILOVER

Places “this controller” and the “other controller” into a dual-redundant (failover) configuration within a multiple-bus environment. This allows both controllers to access the storage devices and provide greater throughput. If one controller fails, the devices and cache attached to the failed controller become available to and accessible through the remaining controller. Both controllers must be configured for nofailover before you enter the SET MULTIBUS_FAILOVER command.

NOTE: Partitioned storagesets and partitioned single-disk units cannot function in multiple bus failover dual-redundant configurations. Because they are not supported, you must delete your partitions before configuring the controllers for multiple bus failover.

Syntax

```
SET MULTIBUS_FAILOVERCOPY=controller
```

Parameters

controller

Specifies which controller (“this controller” or “other controller”) 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 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 one minute to complete.

Example

This example shows how to configure two controllers to operate in dual-redundant mode within a multiple bus environment:

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

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

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

See also

SET FAILOVER

SET NOFAILOVER

SET NOMULTIBUS_FAILOVER

SET NOFAILOVER

Reconfigures both controllers to operate in a non-dual-redundant (non-failover) configuration. Immediately after entering this command, remove one controller from the shelf because the sharing of devices is not supported by nonredundant controllers.

> **NOTE:** SET NOFAILOVER and SET NOMULTIBUS_FAILOVER have the same effect. Either command exits from transparent or multiple bus failover mode.

It is recommended that both controllers be present when this command is carried out. Otherwise, the controllers become misconfigured with each other, requiring additional steps later to allow the “other controller” to be configured for failover.

This command affects both controllers, regardless of the controller on which the command is carried out. All units accessed through the “other controller” failover to “this controller” and the “other controller” is shut down. No configuration information is lost when the SET NOFAILOVER command is carried out.

Syntax

SET NOFAILOVER

Switches

DESTROY_UNFLUSHABLE_DATA
NODESTROY_UNFLUSHABLE_DATA (Default)

Instructs the controller how to handle unflushable data contained within the write-back cache. Unflushable data may be present if, for example, a member of a stripeset has failed. If the other controller is not running, use this switch to specify how to handle this unflushable data when taking the controller pair out of failover mode. If the other controller is running, unflushable data must be cleared from the other controller with the CLEAR_ERRORS command, which is described on pages page 7–39 to page 7–42.

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

This example shows how to terminate failover mode between two controllers in a dual-redundant configuration:

```
SET NOFAILOVER
```

See also

SET FAILOVER

SET NOMULTIBUS_FAILOVER

Reconfigures both controllers to operate in a non-dual-redundant (non-failover) configuration. Immediately after entering this command, remove one controller from the shelf because the sharing of devices is not supported by nonredundant controllers.

> **NOTE:** SET NOFAILOVER and SET NOMULTIBUS_FAILOVER have the same effect. Either command exits from transparent or multiple bus failover mode.

It is recommended that both controllers be present when this command is carried out. Otherwise, the controllers become misconfigured with each other, requiring additional steps later to allow the “other controller” to be configured for failover.

This command affects both controllers, regardless of the controller on which the command is carried out. All units accessed through the “other controller” failover to “this controller” and the “other controller” is shut down. No configuration information is lost when the SET NOMULTIBUS_FAILOVER command is carried out.

Syntax

SET NOMULTIBUS_FAILOVER

Switches

DESTROY_UNFLUSHABLE_DATA
NODESTROY_UNFLUSHABLE_DATA (Default)

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, SET NOMULTIBUS_FAILOVER, 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

This example shows how to terminate failover mode between two controllers in a dual-redundant configuration and destroy any cache data that remains in either controller’s cache:

```
SET NOMULTIBUS_FAILOVER DESTROY_UNFLUSHABLE_DATA
```

See also

SET FAILOVER

SET MULTIBUS_FAILOVER

SET NOFAILOVER

SET *RAIDset-name*

Changes 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

POLICY=BEST_FIT
POLICY=BEST_PERFORMANCE (Default)
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 (Default)

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

Specify 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 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

This example shows how to change the replacement policy for RAIDset RAID9 to BEST_FIT:

```
SET RAID9 POLICY=BEST_FIT
```

This example shows how to remove member DISK10000 from the RAID9 RAIDset:

```
SET RAID9 REMOVE=DISK10000
```

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

This example shows how to add disk DISK20100 to the reduced RAIDset, RAID9:

```
SET RAID9 REPLACE=DISK20100
```

Reconstruction immediately begins on DISK20100.

See also

ADD RAIDSET

SHOW RAIDSETS

SET *remote-copy-set-name*

IMPORTANT: This command works in a remote copy environment only using HSG80 ACS V8.4P code, where local (initiator) and remote (target) sites coexist.

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=target path-name

Adds a controller pair into the remote copy environment. This command is most useful when a failed controller pair has been repaired and must be added back into the remote copy environment.

The target path-name consists of two parts: 1) the controller pair name specified with the SET *controller* REMOTE_COPY= command and 2) the remote unit name, which you create using the ADD REMOTE_COPY_SET command.

ERROR_MODE=FAILSAFE
ERROR_MODE=NORMAL (Default)

Select ERROR_MODE=FAILSAFE to lock in the remote copy set. This way, when one controller pair fails, the unit in the other remote copy set cannot be overwritten.

Select NORMAL to unlock the remote copy center and to change the error mode so that the unit in the remote copy set can be written to.

INITIATOR=initiator unit-name

Sets the name of the controller pair as the initiator.

REMOVE=target path-name

Removes a controller pair from an existing remote copy set. The controller pair specified by *target path-name* is removed from the remote copy set specified by *remote-copy-set-name*.

See also

ADD REMOTE_COPY_SETS

SET *controller* REMOTE_COPY

SET *unit-number*

Changes the characteristics of a unit.

Syntax

SET *unit-number*

Parameter

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 7–6 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 7–6 SET *unit-number* Switches for Existing Containers

Container Type / Switch	ENABLE_ACCESS_PATH DISABLE_ACCESS_PATH	MAXIMUM_CACHED_ TRANSFER_SIZE	IDENTIFIER NOIDENTIFIER	PREFERRED_PATH NOPREFERRED_PATH	READ_CACHE NOREAD_CACHE	READAHEAD_CACHE NOREADAHEAD_CACHE	WRITE_PROTECT NOWRITE_PROTECT	WRITEBACK_CACHE NOWRITEBACK_CACHE	RUN NORUN
RAIDset	✔	✔	✔	✔	✔	✔	✔	✔	✔
Stripeset	✔	✔	✔	✔	✔	✔	✔	✔	✔
Mirrorset	✔	✔	✔	✔	✔	✔	✔	✔	✔
NoTransportable Disk	✔	✔	✔	✔	✔	✔	✔	✔	✔
Transportable Disk	✔	✔	✔	✔	✔	✔	✔		✔

NOTE: Regardless of storageset type, the RUN and NORUN switches cannot be specified for partitioned units.

ENABLE_ACCESS_PATH=
DISABLE_ACCESS_PATH=

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.

If you enable another host ID, previously enabled hosts are not disabled. The new ID is added. If you wish to enable only certain IDs, disable all access paths (DISABLE_ACCESS_PATH=ALL), then enable the desired IDs. The system will display the following message:

```
Warning 1000: Access IDs in addition to the one(s) specified are still
enabled. If you wish to enable ONLY the id(s) listed, disable all
access paths (DISABLE_ACCESS_PATH=ALL), then enable the ones
previously listed.
```

MAXIMUM_CACHED_TRANSFER_SIZE=n
MAXIMUM_CACHED_TRANSFER_SIZE=32 (Default)

Sets the largest number of write blocks to be cached by the controller. The controller does not cache any transfers over the set size. Accepted values are 1 through 1024.

The MAXIMUM_CACHED_TRANSFER_SIZE 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 (Default)

Used only in Windows NT operating systems.

May be set only when dual-redundant controllers are operating in a multiple bus failover configuration. In a multiple bus failover configuration, the host determines which controller the units are accessed through. The host's unit-to-controller settings always take precedence over the preferred path assigned to units with this switch. The target ID numbers assigned with the SET *controller* PORT_1_ALPA= (or PORT_2) command determines which target ID number the controller uses to respond to the host.

> **NOTE:** When the controllers are configured to operate in transparent-failover mode, if you set the PREFERRED_PATH switch with the ADD UNIT or SET *unit-number* command, an error message displays because you can only assign a preferred controller path at the unit level only when in multiple bus failover mode.

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 unit online.

Select PREFERRED_PATH=OTHER_CONTROLLER to instruct the “other controller” to bring the unit online.

See Chapter 2 for information regarding multiple bus failover.

READ_CACHE (Default)
NOREAD_CACHE

Switches enable or disable the read-cache function for the unit.

Read caching improves performance in almost all situations, so it is generally recommended to leave it enabled. However, under certain types of 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 read cache and remove the processing overhead associated with caching.

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 instead of disk. Read ahead caching is the default for units.

If you do not expect this unit to get sequential I/O requests, select NOREADAHEAD_CACHE for the unit.

RUN (Default)
NORUN

Controls the disk drive's operation and 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 the devices making up a 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 partitioned units.

WRITE_PROTECT
NOWRITE_PROTECT (Default)

Assigns to the unit's a write-protect policy.

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. However, 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

Enable or disable the write-back data caching function of the controller. The controller's write-back caching feature improves write performance. NOWRITEBACK_CACHE is the default on transportable disks.

Specify WRITEBACK_CACHE for all new RAIDsets, mirrorsets, and units you want to take advantage of write-back caching.

Specify NOWRITEBACK_CACHE for units you want to receive data directly from the host without being cached.

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.

NOTE: The controller may take up to five minutes to flush data contained within the write-back cache when you specify the NOWRITEBACK_CACHE switch.

Example

This example shows how to enable write protect and turn off the read cache on unit D102:

```
SET D102 WRITE_PROTECT NOREAD_CACHE
```

See also

SHOW UNITS

SHOW *unit-number*

SHOW

Displays information about controllers, 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 *connection*
SHOW *controller*
SHOW *device-name*
SHOW *device-type*
SHOW EMU
SHOW *storageset-name*
SHOW *storageset-type*
SHOW *unit-number*
SHOW UNITS

Parameters

connection

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

controller

Specifies the controller to be displayed.

THIS_CONTROLLER
OTHER_CONTROLLER

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 (EMU).

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:

- FAILEDSET—Shows the failedset configured to the controller.
- MIRRORSETS—Shows all mirrorsets configured to the controller.
- RAIDSETS—Shows all RAIDsets configured to the controller.
- SPARESET—Show the spareset configured to the controller.
- STORAGESETS—Shows all storagesets configured with the controller.
- STRIPESETS—Shows all stripesets configured to the controller.

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.

In addition to the unit name you defined for the unit, the information includes the unique 128-bit subsystem unit ID. This ID consists of the controller node ID plus a 64-bit unit ID generated by the subsystem. You name the units, however, the subsystem identifies them internally using this identifier.

A unit on controller 1234 5678 9ABC EF00 would have an ID like the following:

1234 5678 9ABC EF00 0001 0001 3056 00D2

Each single disk unit or storage device in your subsystem is assigned a unique unit ID number. The controller constructs a unit ID number for each device you add to the subsystem. The ID number consists of the controller's worldwide node ID and a unique, internally generated serial stamp. You cannot set or change unit ID numbers.

Unit ID numbers stay with the unit when you move the unit from one slot to another in the enclosure.

Switches

FULL

Displays additional information about each device, storageset, or controller.

Examples

This example shows how to display a listing of disks:

```
           SHOW DISKS
Name          Type       Port Targ Lun       Used by
-----------------------------------------------------------
DISK20300     disk           1   0    0       D100
DISK10100     disk           1   1    0          D101
```

This example shows a full listing of devices attached to the controller:

```
           SHOW DEVICES FULL
Name          Type                     Port Targ  Lun        Used by
-------------------------------------------------------------------
DISK10300     disk                      1    3     0          R0

   Switches:
    NOTRANSPORTABLE
    TRANSFER_RATE_REQUESTED = ASYNCHRONOUS (ASYNCHRONOUS negotiated)
   Size: 8378028 blocks
DISK20100     disk                        2     1     0          S0

   Switches:
    NOTRANSPORTABLE
    TRANSFER_RATE_REQUESTED = ASYNCHRONOUS (ASYNCHRONOUS negotiated)
   Size: 8377528 blocks
   Configuration being backed up on this container
```

This example shows how to display a complete listing of the mirrorset named MIRR1:

```
                    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
```

This example shows the full information for a controller:

```
              SHOW THIS_CONTROLLER FULL
Controller:
    HSG80    (C) DEC CX00001905 Software R024F-0, Hardware 0000
    NODE_ID = XXXX-XXXX-XXXX-XXXX
    ALLOCATION_CLASS = 767
    SCSI_VERSION = SCSI-2
    Configured for dual-redundancy with ZG83301871
            In dual-redundant configuration
    Device Port SCSI address 7
    Time: NOT SET
    Command Console LUN is lun 0 (IDENTIFIER = 8888)
Host PORT_1:
    Reported PORT_ID = XXXX-XXXX-XXXX-XXXX
    PORT_1_PROFILE = PLDA
    PORT_1_TOPOLOGY = FABRIC (FABRIC UP)
    Address = 200113
    NOREMOTE_COPY
Host PORT_2:
    Reported PORT_ID = XXXX-XXXX-XXXX-XXXX
    PORT_2_PROFILE = PLDA
    PORT_2_TOPOLOGY = FABRIC (standby)
    Address = 200413
Cache
    256 megabyte write cache, version 0012
    Cache is GOOD
    Unflushed data in cache
```

```
        CACHE_FLUSH_TIMER = DEFAULT (10 seconds)
    Mirrored Cache:
        Not enabled
    Battery:
        FULLY CHARGED
        Expires:       WARNING: UNKNOWN EXPIRATION DATE!
        WARNING: AN UNKNOWN NUMBER OF DEEP DISCHARGES HAVE OCCURRED!
        NOCCACHE_UPS
    Extended information:
        Terminal speed 19200 baud, eight bit, no parity, 1 stop bit
        Operation control: 00000000 Security state code: 6985
        Configuration backup enabled on 1 devices
```

This example shows how to display the current settings for the EMU:

```
                    SHOW EMU
EMU CABINET SETTINGS
    SENSOR_1_SETPOINT   35  DEGREES C
    SENSOR_2_SETPOINT   35  DEGREES C
    SENSOR_3_SETPOINT   35  DEGREES C
    FANSPEED AUTOMATIC
```

SHUTDOWN *controller*

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 failover to the surviving controller.

Syntax

SHUTDOWN *controller*

Parameter

controller

Indicates which controller is to shut down. Specify OTHER_CONTROLLER or THIS_CONTROLLER.

Switches

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 shutdown 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 to cause the controller to shutdown immediately without checking for online devices or before flushing data from the write-back cache to devices.

Specify NOIMMEDIATE_SHUTDOWN to cause 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

This example shows how to shut down “this controller:”

```
SHUTDOWN THIS_CONTROLLER
```

This example shows how to shut down the other controller, even if it cannot write all of the write-back cached data to the units:

```
SHUTDOWN OTHER_CONTROLLER IGNORE_ERRORS
```

See also

RESTART *controller*

SELFTEST *controller*

UNMIRROR

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 be used on mirrorsets already members of higher-level containers (stripesets or units).

The UNMIRROR command is not valid for disk drives having 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*

Parameter

disk-name

Specifies the name of the normal mirrorset member to be removed from a mirror storageset.

Example

This example shows how to convert DISK10300 back to a single device:

```
UNMIRROR DISK10300
```

See also

ADD MIRRORSET

MIRROR

REDUCE

RUN CLONE

SET *mirrorset-name*

Appendix A
System Profiles

This appendix contains device and storageset profiles you can use to create your system profiles. It also contains an enclosure template you can use to help keep track of the location of devices and storagesets in your shelves.

Device Profile

Type of Device

____ Platter disk drive

____ Optical disk drive

____ Tape Drive

____ CD-ROM

Device Name ______________________________

Unit Number ______________________________

Device Switches

Transportability

____ No (default)

____ Yes

Initialize Switches

Chunk size

____ Automatic (default)

____ 64 blocks

____ 128 blocks

____ 256 blocks

____ Other:

Save Configuration

____ No (default)

____ Yes

Metadata

____ Destroy (default)

____ Retain

Unit Switches

Read Cache

____ Yes (default)

____ No

Write Cache

____ Yes (default)

____ No

Maximum Cache Transfer

____ 32 blocks (default)

____ Other:

Availability

____ Run (default)

____ NoRun

Write Protection

____ No (default)

____ Yes

Read Ahead Cache

____ Yes (default)

____ No

Storageset Profile

Type of storageset

_____ Mirrorset _____ RAIDset _____ Stripeset _____ Striped Mirrorset

Storageset Name ____________________

Disk Drives ____________________

Unit Number ____________________

Partitions

Unit #	Unit #	Unit #	Unit #	Unit #	Unit #	Unit #	Unit #
%	%	%	%	%	%	%	%

RAIDset Switches

Reconstruction Policy	Reduced Membership	Replacement Policy
___ Normal (default)	___ No (default)	___ Best performance (default)
___ Fast	___ Yes, missing:	___ Best fit
		___ None

Mirrorset Switches

Replacement Policy	Copy Policy	Read Source
___ Best performance (default)	___ Normal (default)	___ Least busy (default)
___ Best fit	___ Fast	___ Round robin
___ None		___ Disk drive:

Initialize Switches

Chunk size	Save Configuration	Metadata
___ Automatic (default)	___ No (default)	___ Destroy (default)
___ 64 blocks	___ Yes	___ Retain
___ 128 blocks		
___ 256 blocks		
___ Other:		

Unit Switches

Read Cache	Read-Ahead Cache	Maximum Cache Transfer
___ Yes (default)	___ Yes (default)	___ 32 blocks(default)
___ No	___ No	___ Other
Write-Back Cache	**Write Protection**	**Availability**
___ Yes (default)	___ No (default)	___ Run (default)
___ No	___ Yes	___ NoRun

Enclosure Template

Power Supply							**Power Supply**
Power Supply							**Power Supply**
Power Supply							**Power Supply**
Power Supply							**Power Supply**

Appendix B

Controller Specifications

This chapter contains physical, electrical, and environmental specifications for the HSG80 array controller.

Physical and Electrical Specifications for the Controller

Table B–1 lists the physical and electrical specifications for the controller and cache modules.

Table B–1 Controller Specifications

Hardware	Length	Width	Power	Current at +5 V	Current at +12 V
HSG80 Array Controller module	12 inches	9.5 inches	23.27 W	6.04 A	
Write-back Cache, 512 MB (Battery charging)	5.5 inches	9.5 inches	2.48 W 8.72 W		Cache idle, no battery 230 mA Cache running diagnostics, no battery 400 mA Cache idle, fully discharged battery 710 mA

Voltage measurements in Table B–1 are nominal measurements (at +5 V and +12 V). This table does not included tolerances.

Environmental Specifications

The HSG80 array controller is intended for installation in a Class A computer room environment.

The optimum operating environmental specifications are listed in Table B–2; the maximum operating environmental specifications are listed in Table B–3; and the maximum nonoperating environmental specifications are listed in Table B–4. These are the same as for other Compaq storage devices.

Table B–2 Optimum Operating Environmental Specifications

Condition	Specification
Temperature	+18° to +24°C (+65° to +75°F)
Temperature rate of change	11°C (20°F per hour)
Relative humidity	40% to 60% (noncondensing) with a step change of 10% or less (noncondensing)
Altitude	From sea level to 2400 m (8000 ft)
Air quality	Maximum particle count 0.5 micron or larger, not to exceed 500,000 particles per cubic foot of air
Inlet air volume	0.026 cubic m per second (50 cubic ft per minute)

Table B–3 Maximum Operating Environmental Specifications

Condition	Specification
Temperature	+10° to +40°C (+50° to +104°F) Derate 1.8°C for each 1000 m (1.0°F for each 1000 ft) of altitude Maximum temperature gradient 11°C/hour (20°F/hour) ±2°C/hour (4°F/hour)
Relative humidity	10% to 90% (noncondensing) Maximum wet bulb temperature: 28°C (82°F) Minimum dew point: 2°C (36°F)

Table B–4 Maximum Nonoperating Environmental Specifications

Condition	Specification
Temperature	-40° to +66°C (-40° to +151°F) (During transportation and associated short-term storage)
Relative Humidity	8% to 95% in original shipping container (noncondensing); otherwise, 50% (noncondensing)
Altitude	From -300 m (-1000 ft) to +3600 m (+12,000 ft) Mean Sea Level (MSL)

Glossary

This glossary defines terms pertaining to the HSG80 Fibre Channel array controller. It is not a comprehensive glossary of computer terms.

8B/10B A type of byte encoding and decoding to reduce errors in data transmission patented by the IBM Corporation. This process of encoding and decoding data for transmission has been adopted by ANSI.

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.

alias address An AL_PA value recognized by an Arbitrated Loop port in addition to its assigned AL_PA.

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.

arbitrate A process of selecting one L_Port from a collection of several ports that request use of the arbitrated loop concurrently.

arbitrated loop A loop type of topology where two or more ports can be interconnected, but only two ports at a time can communicate.

arbitrated loop physical address Abbreviated AL_PA. A one-byte value used to identify a port in an Arbitrated Loop topology. The AL_PA value corresponds to bits 7:0 of the 24-bit Native Address Indentifier.

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.

CCITT Acronym for Consultive Committee International Telephone and Telegraph. An international association that sets worldwide communication standards, recently renamed International Telecommunications Union (ITU).

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 *See* coaxial cable.

coaxial cable A two-conductor wire in which one conductor completely wraps the other with the two separated by insulation.

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 terminal An EIA-423-compatible terminal used with the controller. This terminal is used to 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 mode A mode within an L_Port that inhibits the port from participating in loop activities. 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 membership The desired number of mirrorset members when the mirrorset is fully populated with 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.

quiesce The act of rendering bus activity inactive or dormant. For example, “quiesce the SCSI bus operations during a device warm-swap.”

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 converter Sometimes referred to as an adapter. (1) A device used to interface between the 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 number A bit-significant representation of the SCSI address referring to one of the signal 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