



HSZ80 Array Controller ACS Version 8.3

Configuration and CLI Reference Guide

**First Edition (December 1998)
Part Number EK-HSZ80-RG. A01/388222-001
Compaq Computer Corporation**

While Compaq Computer Corporation believes the information included in this manual is correct as of the date of publication, it is subject to change without notice. Compaq 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. No responsibility is assumed for the use or reliability of firmware on equipment not supplied by Compaq or its affiliated companies. Possession, use, or copying of the software or firmware described in this documentation is authorized only pursuant to a valid written license from Compaq, an authorized sublicensee, or the identified licensor.

Commercial Computer Software, Computer Software Documentation and Technical Data for Commercial Items are licensed to the U.S. Government with Compaq's standard commercial license and, when applicable, the rights in DFAR 252.227 7015, "Technical Data-Commercial Items."

© 1998 Compaq Computer Corporation.

All rights reserved. Printed in U.S.A.

Compaq, the Compaq logo, DIGITAL, DIGITAL UNIX, DECconnect, HSZ, HSG, StorageWorks, VMS, OpenVMS Registered in the United States Patent and Trademark Office.

UNIX is a registered trademark in the United States and other countries exclusively through X/Open Company Ltd. Windows NT is a registered trademark of the Microsoft Corporation. Sun is a registered trademark of Sun Microsystems, Inc. Hewlett-Packard, TACHYON, and HP-UX are registered trademarks of the Hewlett-Packard Company. IBM and AIX are registered trademarks of International Business Machines Corporation. All other trademarks and registered trademarks are the property of their respective owners.

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

この装置は、情報処理装置等電波障害自主規制協議会（VCCI）の基準に基づくクラスA情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよ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.

About This Guide

Getting Help	xv
Precautions	xvi
Conventions	xviii
Required Tools	xx
Related Publications	xxi
Revision History	xxii

Chapter 1

General Description

Typical Installation	1–2
Summary of HSZ80 Features	1–5
The HSZ80 Array Controller	1–7
Cache Module	1–18
External Cache Battery	1–29

Chapter 2

Configuring an HSZ80 Array Controller

Introduction	2–2
Configuring an HSZ80 Array Controller	2–3
Setting the PVA Module ID Switch	2–25
Establishing a Local Connection to the Controller	2–26
Selecting a Failover Mode	2–29
Enabling Mirrored Write-Back Cache	2–31
Setting SCSI Target ID Numbers	2–33
Using Preferred ID Numbers	2–35

Chapter 3

Creating Storagesets

Introduction	3-2
Planning and Configuring Storagesets.	3-4
Creating a Storageset and Device Profile	3-5
Determining Storage Requirements.	3-7
Choosing a Storageset Type.	3-8
Cloning Data for Backup.	3-20
Backing Up Your Subsystem Configuration	3-25
Node IDs	3-30
Assigning Unit Numbers for Host Access to Storagesets	3-32
Creating a Storageset Map.	3-34
Planning Partitions.	3-40
Choosing Switches for Storagesets and Devices.	3-42
RAIDset Switches	3-43
Mirrorset Switches.	3-45
Device Switches.	3-48
Initialize Switches	3-51
Unit Switches.	3-58

Chapter 4

Configuring Storagesets

Introduction	4-2
Configuring with the Command Console LUN.	4-19
Configuring Units with Multiple Hosts.	4-23
Moving Storagesets	4-35

Chapter 5

CLI Commands

CLI Overview	5-2
ADD DISK	5-7
ADD MIRRORSET	5-11
ADD PASSTHROUGH	5-16
ADD RAIDSET	5-19
ADD SPARESET	5-23
ADD STRIPESET	5-24
ADD UNIT	5-27
CLEAR_ERRORS CLI	5-35
CLEAR_ERRORS <i>controller</i> INVALID_CACHE	5-36
CLEAR_ERRORS <i>device-name</i> UNKNOWN	5-38
CLEAR_ERRORS <i>unit-number</i> LOST_DATA	5-40
CLEAR_ERRORS <i>unit-number</i> UNWRITEABLE_DATA	5-42
CONFIGURATION RESET	5-44
CONFIGURATION RESTORE	5-45
CONFIGURATION SAVE	5-47
CREATE_PARTITION	5-48
DELETE <i>container-name</i>	5-51
DELETE FAILEDSET	5-53
DELETE SPARESET	5-54
DELETE <i>unit-number</i>	5-55
DESTROY_PARTITION	5-57
DIRECTORY	5-59
HELP	5-60
INITIALIZE	5-62
LOCATE	5-67

MIRROR	5-70
POWEROFF	5-74
REDUCE	5-77
RENAME	5-81
RESTART <i>controller</i>	5-83
RETRY_ERRORS unit-number UNWRITEABLE_DATA	5-85
RUN	5-86
SELFTEST <i>controller</i>	5-89
SET <i>controller</i>	5-91
SET <i>device-name</i>	5-102
SET EMU	5-105
SET FAILEDSET	5-108
SET FAILOVER	5-110
SET <i>mirrorset-name</i>	5-112
SET MULTIBUS_FAILOVER	5-118
SET NOFAILOVER	5-120
SET NOMULTIBUS_FAILOVER	5-122
SET <i>RAIDset-name</i>	5-124
SET unit-number	5-128
SHOW	5-136
SHUTDOWN <i>controller</i>	5-143
UNMIRROR	5-145



Appendix A

System Profiles

Device Profile A-2
StorageSet Profile..... A-3
Enclosure Template A-4

Appendix B

Controller Specifications

Physical and Electrical Specifications for the Controller B-2
Environmental Specifications B-3

Glossary

Index

Figures

Basic Building Block	1-3
Bridging the Gap Between the Host and Its Storage Subsystem	1-7
HSZ80 Array Controller	1-9
Parts Used in Configuring the HSZ80 Array Controller	1-10
Optional Maintenance Port Cable for a Terminal Connection	1-12
Location of Controllers and Cache Modules	1-13
HSZ80 Controller Operator Control Panel (OCP)	1-14
Cache Module	1-19
ECB for Dual-Redundant Controller Configurations	1-30
Cabling for a Single-Controller Configuration	2-5
Cabling for a Dual-Redundant Controller Configuration in Transparent Failover	2-10
Cabling for a Dual-Redundant Controller Configuration in Multiple-Bus Failover	2-16
SCSI Target ID Numbers on the Controller Device Bus and PVA Settings in an Extended Subsystem	2-25
PC/Terminal to Maintenance Port Connection	2-27
“This Controller” and “Other Controller”	2-28
Host SCSI Bus and Controller Device Bus	2-33
Units Created from Stagesets, Partitions, and Drives	3-3
Stageset Profile	3-6
Striping Lets Several Disk Drives Participate in Each I/O Request	3-9
Distribute Members across Ports	3-11
Mirrorsets Maintain Two Copies of the Same Data	3-13
First Mirrorset Members on Different Buses	3-14
Parity Ensures Availability; Striping Provides Good Read Performance	3-16
Striping and Mirroring in the Same Stageset	3-18
CLONE Steps for Duplicating Unit Members	3-21

Storageset Map	3-35
PTL Naming Convention	3-37
PTL Addressing in an Extended Configuration	3-38
Locating Devices using PTLs	3-39
Partitioning a Single-Disk Unit	3-40
Chunk Size Larger than the Request Size	3-52
Chunk Size Smaller than the Request Size	3-53
Setting Host Modes	4-25
Accessing Units on a Given Port	4-27
Accessing Units with the Host Port's SCSI ID Number	4-29
Accessing Units through Port Access and the Host Port's SCSI ID Number ...	4-31
Assigning Units through Multiple Host Adapters	4-33
Moving a Storageset from one Subsystem to Another	4-35

Tables

Basic Building Blocks List	1-4
Summary of Controller Features	1-5
HSZ80 Array Controller	1-9
Description of Parts	1-10
Parts of the Optional Maintenance Port Cable	1-12
Cache Module Memory Configurations	1-18
Location of Cache Module Parts	1-19
Cache Policies and Cache Module Status	1-23
Resulting Cache Policies and ECB Status	1-25
ECB Capacity Based on Memory Size	1-29
Location of Parts	1-30
Location of Parts for a Single Controller Configuration	2-5
Location of Parts for Transparent Failover	2-11
Location of Parts for Multiple-Bus Failover	2-17
Controller Limitations for RAIDsets	3-3
A Comparison of Different Kinds of Storagesets	3-8
Unit Numbering Examples	3-33
Maximum Chunk Sizes for a RAIDset	3-54
UNIT Switches for Storagesets	3-58
Recall and Edit Command Keys	5-4
Unit Numbering Examples	5-5
ADD UNIT Switches for Storagesets	5-28
Poweroff Switch Settings	5-75
SET <i>controller</i> Switches	5-91
Maximum SCSI-Bus Lengths for Given Data Transfer Rate	5-100
EMU Set Point Temperatures	5-106
SET UNIT Switches for Existing Containers	5-129
Controller Specifications	B-2

Optimum Operating Environmental SpecificationsB-3
Maximum Operating Environmental SpecificationsB-4
Maximum Nonoperating Environmental SpecificationsB-4



About This Guide

This book describes the features of the HSZ80 Array Controller, configuration procedures for the controller and storagesets running Array Controller Software (ACS) 8.3Z, and the CLI commands used in configuring.

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

Getting Help

If you have a problem and have exhausted the information in this reference 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 technical support:

In the United States and Canada, call 1-800-652-6672.

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 you're 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 and 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.
- Do not 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.

VHDCI Cable Precautions

All of the cables to the controller, cache module, and external cache battery use very-high-density cable interconnect connectors (VHDCI). These connectors have extraordinarily small mating surfaces that can be adversely affected by dust and movement.

Use the following precautions when you're connecting cables that use VHDCI connectors:

- Clean the mating surfaces with a blast of clean air.
- Mate the connectors by hand, then tighten the retaining screws to 1.5 inch-pounds—approximately 1/4 additional turn after the connectors have fully mated.
- Test the assembly by gently pulling on the cable, which should not produce visible separation.

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.

Conventions

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

Typographical Conventions

Convention	Meaning
ALLCAPS	Command syntax that must be entered exactly as shown and for commands discussed within text, for example: <code>SET FAILOVER COPY=OTHER_CONTROLLER</code> “Use the SHOW SPARESET command to show the contents of the spareset.”
Monospaced	Screen display.
<i>Sans serif italic</i>	Command variable or numeric value that you supply, for example: <code>SHOW RAIDset-name or</code> set this_controller id=(<i>n,n,n,n</i>)
<i>italic</i>	Reference to other books, for example: “See 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: An *important* note is a type of note that provides information essential to the completion of a task. Users can disregard information in a note and still complete a task, but they should not disregard an important note.

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

Required Tools

You'll need the following tools to service the controller, cache module, external cache battery (ECB), the Power Verification and Addressing (PVA) module and the I/O module:

- A flathead screwdriver for loosening and tightening the I/O module retaining screws.
 - A small phillips screwdriver for loosening and tightening the GLM access door screws.
 - An antistatic wrist strap.
 - An antistatic mat on which to place modules during servicing.
 - A Storage Building Block (SBB) Extractor for removing StorageWorks building blocks. This tool is not required, but it will enable you to perform more efficiently.
-

Related Publications

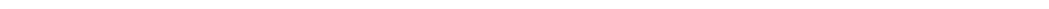
The following table lists some of the Compaq StorageWorks documents related to the use of the controller, cache module, external cache battery, graphical user interface, and the subsystem.

Document Title	Part Number	
BA370 Enclosure Rack Template (Compaq 42U Rack)	355224-001	EK-RKTMP-TP
Command Console Version 2.1 HSZ80 User's Guide	388725-001	AA-RF9TA-TE
Disaster Tolerant Solutions Getting Started Guide for DIGITAL UNIX	N/A	AA-RC3CA-TE
Disaster Tolerant Solutions User's Guide for DIGITAL UNIX	N/A	EK-SWXDT-OP
External Cache Battery Shelf Installation Card (Compaq 42U Rack)	355222-001	EK-HSECB-IC
Hardware Configuration Poster for HSZ80	388724-001	EK-HSZ80-CP
HSZ80 Array Controller ACS V8.3 for DIGITAL UNIX CD-ROM	N/A	AG-RFA0A-BE
HSZ80 Array Controller ACS V8.3 for DIGITAL UNIX Release Notes	388713-001	AA-RF9YA-TE
HSZ80 Array Controller ACS V8.3 for IBM-AIX Release Notes	388711-001	AA-RFALA-TE
HSZ80 Array Controller ACS V8.3 for OpenVMS Release Notes	388712-001	AA-RFAEA-TE
HSZ80 Array Controller ACS V8.3 Maintenance and Service Guide	388221-001	EK-HSZ80-SV
HSZ80 Array Controller Illustrated Parts Map	388220-001	EK-HSZ80-MP
Installation Card (Compaq 42U Rack)	355210-001	EK-H80RM-IC
Installing a Ferrite Bead on a Host Bus	N/A	EK-SWXES-IG
The RAIDBOOK—A Source for RAID Technology	N/A	RAID Advisory Board
RA8000/ESA12000 HSZ80 ACS V8.3 for DIGITAL UNIX Installation Reference Manual	388701-001	AA-RF9ZA-TE

Document Title	Part Number	
RA8000/ESA12000 HSZ80 ACS V8.3 for DIGITAL UNIX Quick Setup Guide	388700-001	AA-RF9XA-TE
RA8000/ESA12000 HSZ80 ACS V8.3 for IBM-AIX Installation Reference Manual	388710-001	AA-RFAMA-TE
RA8000/ESA12000 HSZ80 ACS V8.3 for IBM-AIX Quick Setup Guide	388709-001	AA-RFAKA-TE
RA8000/ESA12000 HSZ80 ACS V8.3 for OpenVMS Installation Reference Manual	388733-001	AA-RFAFA-TE
RA8000/ESA12000 HSZ80 ACS V8.3 for OpenVMS Quick Setup Guide	388732-001	AA-RFADA-TE
RA8000/ESA12000 Storage Subsystem User's Guide	387404-001	EK-SMCPR-UG
Rail Mounting Installation Card (Compaq 42U Rack)	355223-001	EK-H8RMB-IC
Ultra SCSI RAID Enclosure (DS-BA370 Series) User's Guide	387403-001	EK-BA370-UG
Warranty Terms and Conditions	N/A	EK-HSXSW-WC

Revision History

This is a new document.



Chapter 1

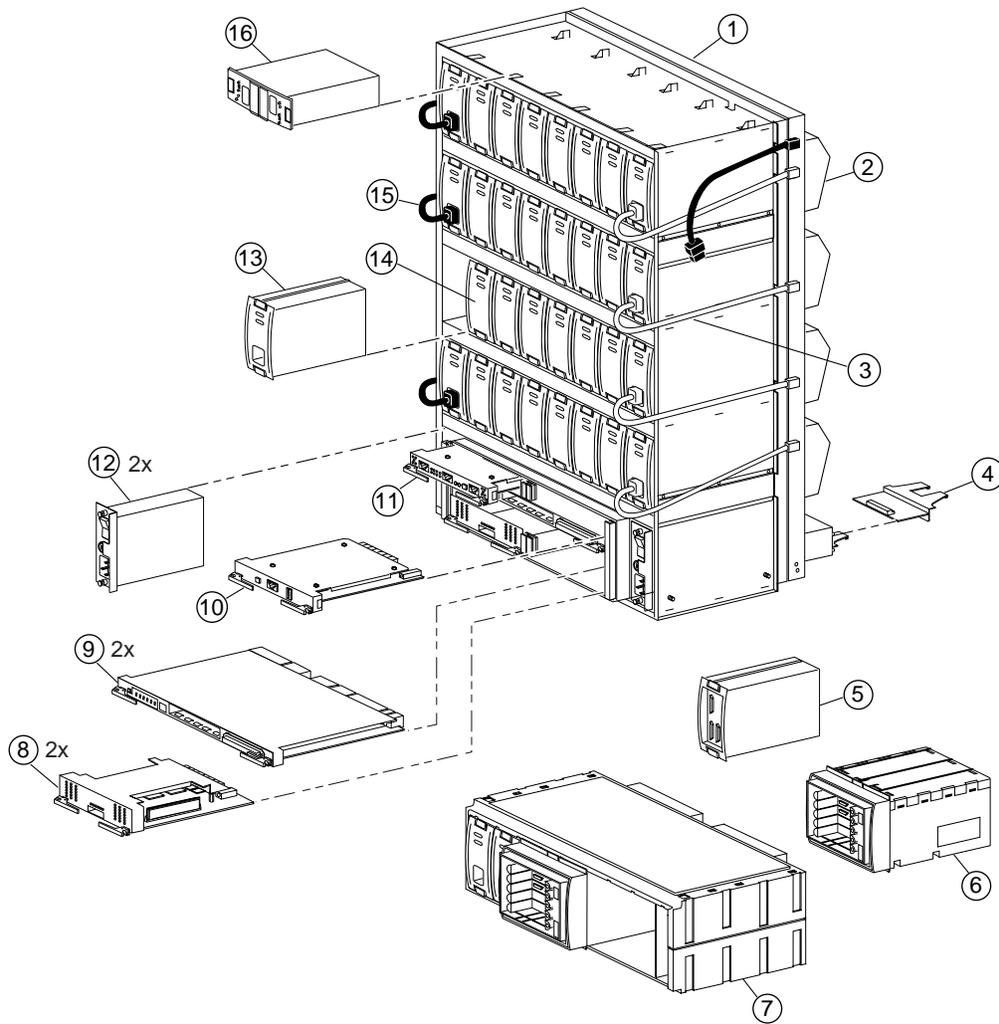
General Description

This chapter illustrates and describes in general terms your subsystem and its major components: the HSZ80 Array Controller, its cache module, and its external cache battery (ECB).

Typical Installation

Figure shows the major components, or basic building blocks, of the storage subsystem that is comprised of the following:

- One BA370 rack-mountable enclosure.
 - Two controllers, each supported by a cache module.
 - 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 SBB shelf with either an 8-bit single-ended, 16-bit single-ended, or 16-bit differential SCSI bus
 - Two cache modules, which support nonvolatile memory and dynamic cache policies to protect the availability of its unwritten (write-back) data
-



CXO6742A

Figure 1-1. Basic Building Block

1-4 *General Description*

Table 1-1 Basic Building Blocks List

Item	Description
1	BA370 rack-mountable enclosure
2	Cooling fan
3	Power cable kit
4	I/O module
5	SCSI hub, 3 port
6	SCSI hub, 5 port
7	SCSI hub, 9 port, upgrade
8	Cache module
9	HSZ80 controller
10	PVA module
11	EMU
12	AC input module
13	180-watt power supply
14	Disk drive, 4 GB, 7200 Disk drive, 9 GB, 7200 Disk drive, 18 GB, 7200 Disk drive, 9 GB, 10K Disk drive, 18 GB, 10K
15	Power cable
16	ECB, dual (shown) ECB, single (not shown)

Summary of HSZ80 Features

Table 1-2 summarizes the features of the controller:

Table 1-2 Summary of Controller Features	
Feature	Supported
Topology	Dual host port Ultra SCSI
Host protocol	SCSI-2, Limited SCSI-3
Host bus interconnect	Wide Ultra SCSI, Differential
Number of host ports	2 on Wide Ultra SCSI, Differential
Device protocol	SCSI-2
Device bus interconnect	Wide Ultra SCSI, Single-ended
Number of SCSI device ports (or I/O modules)	6 Wide Ultra SCSI, Single-ended
Number of SCSI device targets per port	12
Maximum number of SCSI target devices	72 Ultra SCSI disks
RAID levels supported	0, 1, 0+1, 3/5
Cache module memory sizes supported	64 MB, 128 MB, 256 MB, 512 MB
Maximum cache module size	512 MB
PCMCIA program card software upgrades	Yes
Device warm swap	Yes
Exercisers available for testing disks	Yes
Maximum number of RAID-5 storagesets	20
Maximum number of RAID-5 and RAID-1 storagesets	30
Maximum number of RAID-5, RAID-1, and RAID-0 storagesets	45
Maximum number of units presented to the host	128
Maximum number of partitions per storageset	8
Maximum members per mirrorset	6
Maximum number of members per RAID-5 storageset	14

1-6 *General Description*

Table 1-2 Summary of Controller Features (Continued)	
Feature	Supported
Maximum number of members in a stripeset	24
Maximum number of physical devices in a striped mirrorset	48
Maximum host port transfer speed	20 MHz
Largest device, storageset, or unit size	512 GB

The HSZ80 Array Controller

Your controller is the intelligent bridge between your host and the devices in your subsystem.

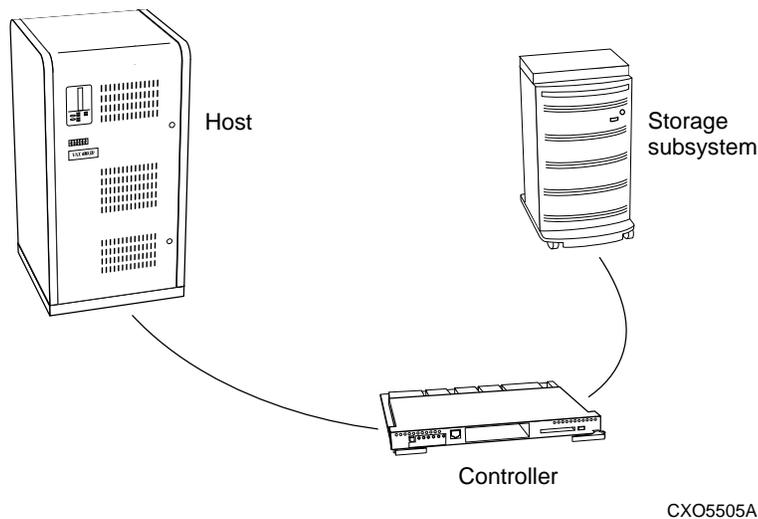


Figure 1-2. Bridging the Gap Between the Host and Its Storage Subsystem

The controller shown in Figure 1-2 is an integral part of any storage subsystem because it provides a host with high-performance and high-availability access to storage devices. See the *HSZ80 Array Controller ACS Version 8.3 Release Notes* for the most recent list of supported devices and operating systems.

The controller provides the ability to combine several ordinary disk drives into a single, high-performance storage unit called a *storageset*. Storagesets are implementations of RAID technology, which ensures that every unpartitioned storageset, whether it uses two disk drives or ten, looks like a single storage unit to the host. See Chapter 3 and Chapter 4 for more information about storagesets and how to configure them.

1-8 General Description

From the host's perspective, the controller is simply another device connected to one of its I/O buses. Consequently, the host sends its I/O requests to the controller just as it would to any SCSI device. From the subsystem's perspective, the controller receives the I/O requests and directs them to the devices in the subsystem. Because the controller processes the I/O requests, the host isn't burdened by the processing that's typically associated with reading and writing data to multiple storage devices.

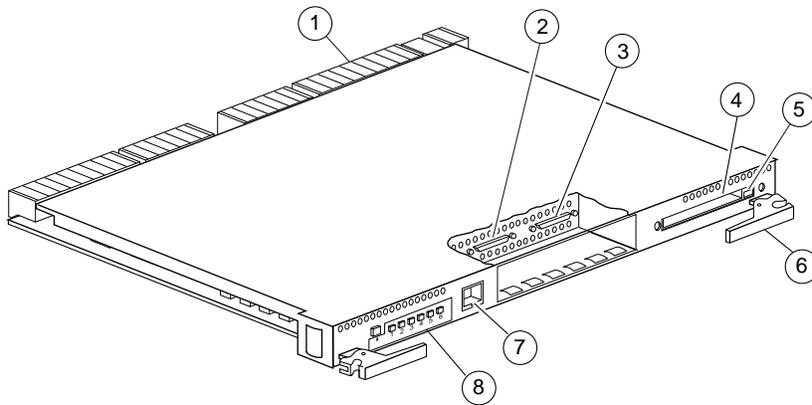
Figure 1-3 shows an HSZ80 Array Controller. Figure 1-4 shows the parts used in configuring it and Table 1-4 lists the descriptions and part numbers.

NOTE: The maintenance port cable shipped with the controller has a 9-pin connector for a PC connection only. If you are using a terminal instead of a PC or if you have a system with a DB25 connection, you can obtain a cable as shown in Figure 1-5 on page 1-12 from the local field service office.

To determine which parts you need for your configuration, you must have an idea of the type of configuration you will be running in your subsystem. See the following sections for your configuration type:

- “Configuring a Single Controller,” page 2-5
- “Configuring Dual-Redundant Controllers in Transparent Failover Mode,” page 2-10
- “Configuring Dual-Redundant Controllers in Multiple-Bus Failover Mode,” page 2-16

The components that you'll use most often are conveniently located on the controller's front panel, such as the maintenance port and the operator control panel (OCP). 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.



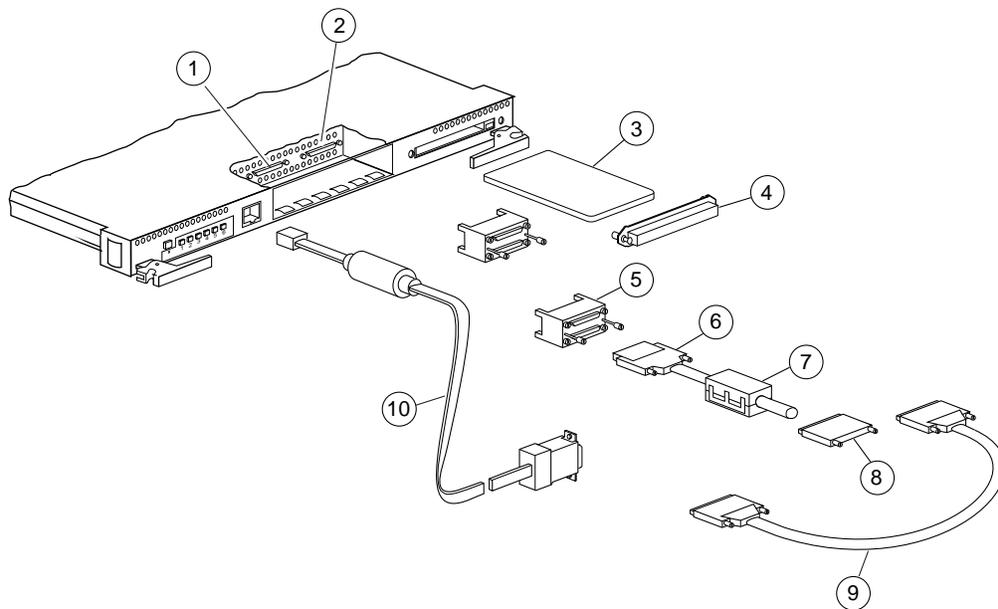
CXO6284B

Figure 1-3. HSZ80 Array Controller

Table 1-3 HSZ80 Array Controller

Item	Description
①	Backplane connector
②	Host port 1
③	Host port 2
④	Program card slot
⑤	Program card ejection button
⑥	Release lever
⑦	Maintenance port
⑧	Operator Control panel

1-10 General Description



CXO6492B

Figure 1-4. Parts Used in Configuring the HSZ80 Array Controller

Table 1-4 Description of Parts	
Item	Description
①	Host port 1
②	Host port 2
③	ESD/PCMCIA program card
④	ESD/PCMCIA program card cover
⑤	Trilink connectors
⑥	Host bus cable

Table 1-4 Description of Parts (Continued)

Item	Description
⑦	Ferrite bead
⑧	Terminator
⑨	Jumper cable
⑩	Maintenance port cable

1-12 General Description

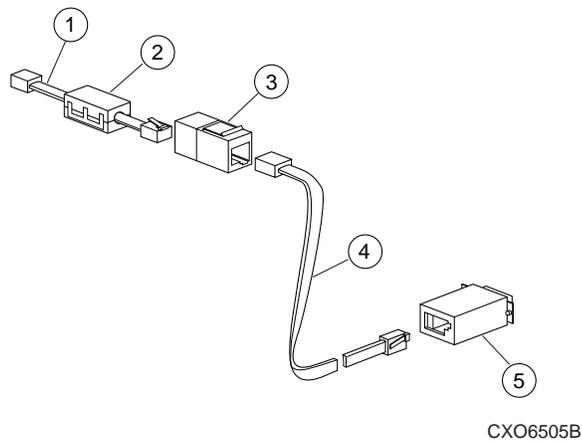


Figure 1-5. Optional Maintenance Port Cable for a Terminal Connection

Table 1-5 Parts of the Optional Maintenance Port Cable	
Item	Description
①	Cable assembly
②	Ferrite bead
③	RJ-11 Adapter
④	RJ-11 Extension Cable
⑤	PC Serial Port Adapter

Each controller is supported by its own cache module. Figure 1-6 shows which cache module supports which controller in a dual-redundant controller configuration in a BA370 rack-mountable enclosure.

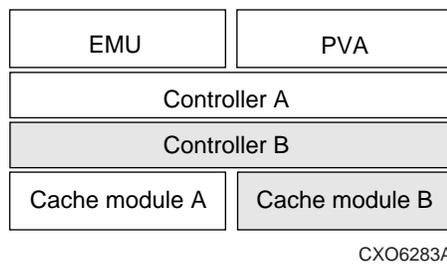


Figure 1-6. Location of Controllers and Cache Modules

IMPORTANT: Compaq recommends that you use the slots for controller A and cache module A for single controller configurations. Slot A responds to SCSI target ID number 7 on the device buses and 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 button LEDs as shown in Figure 1-7. 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 LEDs will illuminate in a solid or flashing pattern to help you diagnose the problem.

1-14 General Description

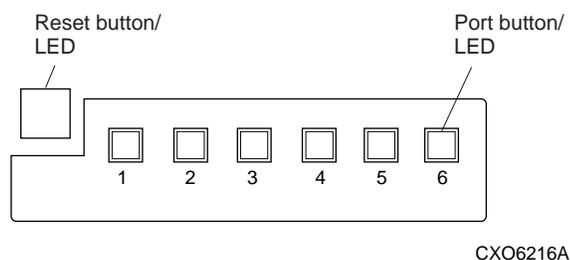


Figure 1-7. HSZ80 Controller Operator Control Panel (OCP)

See Figure 1-3 on page 1-9 for the location of the OCP on the HSZ80 Array Controller.

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 LEDs on the control panel will flash in a pattern that will help you to diagnose the problem. See the *HSZ80 Array Controller ACS Version 8.3 Maintenance and Service Guide* for details about troubleshooting your controller.

Maintenance Port

You can access the controller in two ways: through a PC or 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 terminal to carry out the troubleshooting and servicing procedures in this manual. See “Establishing a Local Connection to the Controller,” page 2-26, for more information about connecting the controller with a maintenance port cable.

Utilities and Exercisers

The controller's software includes the following utilities and exercisers to assist in troubleshooting and maintaining the controller and the other modules that support its operation. For more details on each of the utilities and exercisers, see the *HSZ80 Array Controller ACS Version 8.3 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 the last-failure and memory-system-failure entries that the fault management software stores in the controller's nonvolatile memory.
- Translate many of the event messages that are contained in the entries related to the significant events and failures. For example, entries may contain codes that indicate the cause of the event, the software component that reported the event, the repair action, and so on.
- Set the display characteristics of spontaneous events and failures that the fault management system sends to the PC or terminal, or to the host.

Virtual Terminal Display

Use the virtual terminal display (VTDPY) utility to aid in troubleshooting the following issues:

- Communication between the controller and its hosts.
- Communication between the controller and the devices in the subsystem.
- 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.

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.

Code Load and Code Patch Utility

Use the 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. Whenever you install a new controller, you must have the correct software version and patch number.

IMPORTANT: 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 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, external cache batteries, and the PVA module.

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.

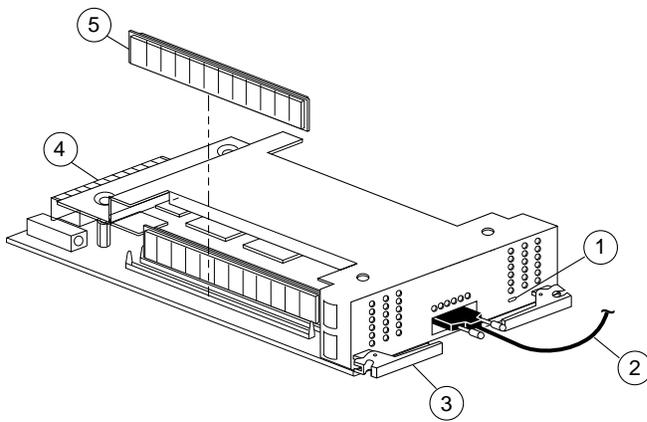
Cache Module

Each controller requires a companion cache module as shown in Figure 1-8. Table 1-7 lists the descriptions and part numbers of the cache module. Figure 1-6 on page 1-13 shows the location of a controller's companion cache module.

The cache module, which contains 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 1-6. For placement of the DIMMs, see the *HSZ80 Array Controller ACS Version 8.3 Maintenance and Service Guide*.

Table 1-6 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



CXO6306B

Figure 1-8. Cache Module

Table 1-7 Location of Cache Module Parts

Item	Description
①	Cache memory power LED
②	ECB Y cable
③	Release lever
④	Backplane connector
⑤	DIMM

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 also stores the data in its cache module. 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.

To set the maximum cache that can be transferred, see "SET unit-number" in Chapter 5, "CLI Commands."

Read-Ahead Caching

Read-ahead caching begins once the controller has already processed a read request and it receives a sequential 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.

The controller then 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 through 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 stores 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.

If you enable read caching for a storage unit, write-through caching is automatically enabled. Likewise, if you disable read caching, write-through caching is automatically disabled.

Write-Back Caching

This caching technique decreases 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 disk units. In either case, the controller will not provide write-back caching to a unit unless the cache memory is nonvolatile, as described in the next section.

If the mirrorset is a disaster-tolerant mirrorset, then write-back caching cannot be enabled.

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

Except for disaster-tolerant supported mirrorsets, the controller can provide write-back caching for storage units 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 1-29, for more information on the ECB.

However, if your subsystem is backed up by an uninterruptible power supply (UPS), you can tell the controller to use the UPS as the backup power source with the command, SET Controller, and using the CACHE_UPS switch.

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.

Table 1-8 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.

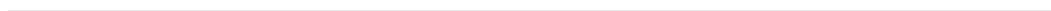


Table 1-9 shows the cache policies resulting from a full or partial failure of cache module A's ECB in a dual-redundant controller configuration. When cache module A is at least 50% charged, the ECB is still good and is charging. When it is less than 50% charged, the ECB is low, but still charging. The consequences shown are the opposite for a failure of cache module B's ECB.

Table 1-8 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: Forced error and loss of write-back data for which the 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 1-8 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	<p>Data integrity: Write-back data that was not written to media when failure occurred was not recovered.</p> <p>Cache policy: Controller A supports write-through caching only; controller B supports write-back caching.</p> <p>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.</p> <p>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.</p>	<p>Data integrity: Controller A recovers all of its write-back data from the mirrored copy on cache B.</p> <p>Cache policy: Controller A supports write-through caching only; controller B supports write-back caching.</p> <p>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.</p>

Table 1-8 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.	<p>Data integrity: Controller A recovers all of its write-back data from the mirrored copy on cache B.</p> <p>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.</p> <p>Failover: No.</p>

Table 1-9 Resulting Cache Policies and ECB Status

Cache Module Status		Cache Policy	
Cache A	Cache B	Unmirrored Cache	Mirrored Cache
At least 50% charged	At least 50% charged	<p>Data loss: No.</p> <p>Cache policy: Both controllers continue to support write-back caching.</p> <p>Failover: No.</p>	<p>Data loss: No.</p> <p>Cache policy: Both controllers continue to support write-back caching.</p> <p>Failover: No.</p>

Table 1-9 Resulting Cache Policies and ECB Status (Continued)

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

Table 1-9 Resulting Cache Policies and ECB Status (Continued)

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

Table 1-9 Resulting Cache Policies and ECB Status (Continued)

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

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 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 1-9. 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, which are listed in Table 1-10.

Table 1-10 ECB Capacity Based on Memory Size

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

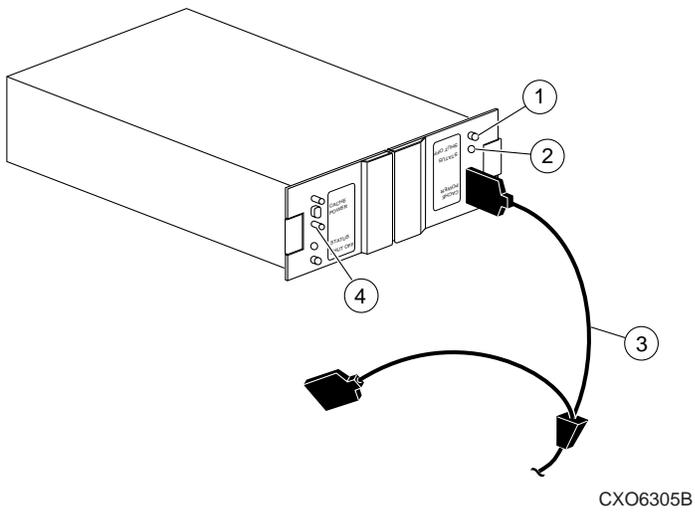


Figure 1-9. ECB for Dual-Redundant Controller Configurations

Table 1-11 Location of Parts	
Item	Description
①	Shut off button
②	Status LED
③	ECB Y cable
④	Male port

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 fully charged and rechecks them every 24 hours. If the batteries are charging, the controller rechecks them every 4 minutes. Batteries are reported as being either above or below 50 percent in capacity. Batteries below 50 percent in capacity are also referred to as being low.

This 4-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.

IMPORTANT: If a UPS is used for backup power, the controller does not check the battery. To set the subsystem to use a UPS or to instruct the controller on performing regular battery checks, see "SET controller" in Chapter 5, "CLI Commands."

Chapter 2

Configuring an HSZ80 Array Controller

This chapter explains how to configure an HSZ80 Array Controller and the modules that support its operation in a StorageWorks subsystem.

Introduction

Use the *Quick Setup Guide* that came with your subsystem to unpack and set up your subsystem prior to configuring your controller.

Unless you specifically requested a preconfigured subsystem, you will have to configure your controller and its subsystem before you can use them. Use the procedure in this chapter to configure your controller. The procedure contains references to more detailed information, should you need it.

For the complete syntax and descriptions of the CLI commands used in the configuration procedure, see Chapter 5, “CLI Commands.”

Configuration Rules

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

- Maximum 15 targets with up to 32 LUNs on each target, depending on the operating system
 - Maximum 128 assignable, host-visible LUNs across all 15 targets
 - Maximum 512-GB LUN capacity
 - Maximum 72 physical devices
 - Maximum 20 RAID-5 storage sets
 - Maximum 30 RAID-5 and RAID-1 storage sets
 - Maximum 45 RAID-5, RAID-1, and RAID-0 storage sets
 - Maximum 8 partitions per storage set or individual disk
 - Maximum 6 members per mirror set
 - Maximum 14 members per RAID-5 storage set
 - Maximum 24 members per stripe set
 - Maximum 48 physical devices per striped mirror set
-

Configuring an HSZ80 Array Controller

You can use this procedure to configure your controller in one of three configurations: 1) single controller, 2) dual controllers in transparent failover mode, or 3) dual controllers in multiple-bus failover (host-assisted) mode. When you are done configuring the controller, you can then add devices, plan storagesets, and configure storagesets. These tasks can be performed either with SWCC or using CLI commands from a PC or terminal.

See the following sections for your configuration type:

- “Configuring a Single Controller,” page 2-5
- “Configuring Dual-Redundant Controllers in Transparent Failover Mode,” page 2-10
- “Configuring Dual-Redundant Controllers in Multiple-Bus Failover Mode,” page 2-16

A Few 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 SCSI host bus cables connected to the host, or you can configure with the SCSI host bus cables connected:
 - SWCC—You can configure the controller using the SWCC graphical user interface. See the *Command Console for HSZ80 Solutions Getting Started Guide* for more details.
 - CLI Commands—You can configure the controller with CLI commands using a terminal program.

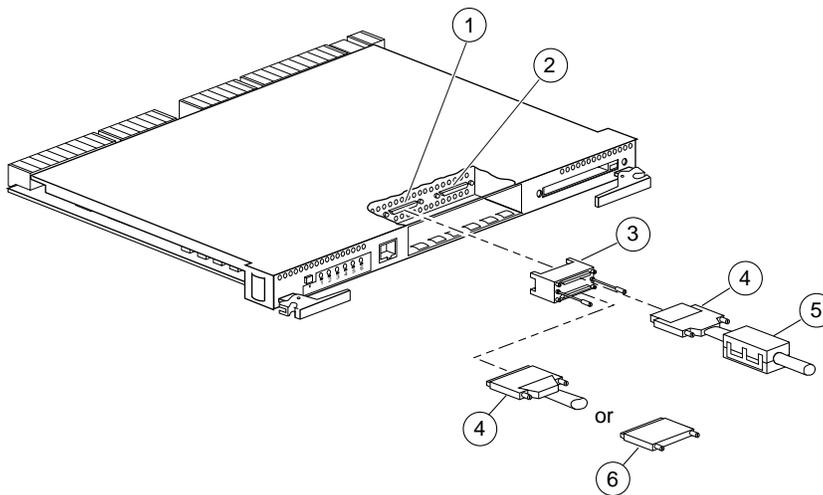
NOTE: Eventually, you must connect the SCSI host bus cables between the controllers and the host. You can connect the SCSI host bus cables after configuring the controllers using a maintenance port cable.

- 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.

2-4 Configuring an HSZ80 Array Controller

- CLI Commands—A few commands can be used 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 deleting it.
 - **SHOW THIS_CONTROLLER FULL** and **SHOW OTHER_CONTROLLER FULL**—Lists the full information about the controller.
-

Configuring a Single Controller



CXO6285B

Figure 2-1. Cabling for a Single-Controller Configuration

Table 2-1 Location of Parts for a Single Controller Configuration

Item	Description
①	Host port 1
②	Host port 2
③	Trilink connector
④	Host bus cable
⑤	Ferrite bead
⑥	Terminator

2-6 Configuring an HSZ80 Array Controller

Follow these steps to configure a 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 *HSZ80 Array Controller ACS Version 8.3 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 2-25, for details about PVA switch settings.

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

See “Establishing a Local Connection to the Controller,” page 2-26, for details about creating a local connection.

5. Set the SCSI target ID numbers for the controller. Use the following commands:
SET THIS_CONTROLLER PORT_1_ID = (n,n)

or

SET THIS_CONTROLLER PORT_2_ID = (n,n)

See “Setting SCSI Target ID Numbers,” page 2-33, for details about valid SCSI target ID numbers.

6. Set the host function mode for the controller using the target ID numbers set in step 5. Use the following command:

NOTE: The default host function mode defaults to Host Mode A. The `HOST_FUNCTION` switch changes the host function mode to a different mode.

```
SET THIS_CONTROLLER HOST_FUNCTION= (ID, MODE)
```

To change the host function mode, see “SET controller,” page 5-91 for information on the `HOST_FUNCTION` switch.

7. Restart the controller using the following command:

```
RESTART THIS_CONTROLLER
```

See “RESTART controller” in Chapter 5, “CLI Commands.”

8. 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.

9. 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.

2-8 Configuring an HSZ80 Array Controller

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

See “Optional Steps,” page 2-22 and Chapter 5, “CLI Commands.”

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

```
SHOW THIS_CONTROLLER FULL
```

See “SHOW” in Chapter 5, “CLI Commands.”

IMPORTANT: If you are using a PC or terminal with the maintenance port cable, you must connect the SCSI bus cables in the desired configuration. See “Connecting a Single Controller to the Host,” page 2-9, for more information about configurations.

Connecting a Single Controller to the Host

Follow these steps and Figure 2-1 to connect a single, nonredundant controller to the host:

1. Configure the controller. See “Configuring a Single Controller,” page 2-5.
2. Ensure the host and adapter are also configured. See the host user’s guide for details.
3. Connect a trilink to the controller going into host Port 1:
 - a. Connect the host bus cable onto one end of the trilink connector.
 - b. For a BA370 enclosure residing in an SW600 cabinet, snap the ferrite bead on the bus cable within one inch of the controller.
4. Connect a terminator on the trilink connector for host Port 1:
 - a. If the controller is at the end of the host bus, connect a terminator to the other connector on the front of the trilink connector.
 - b. Otherwise, connect a cable that continues to the next device on the bus. Be sure to install a terminator at the end of the bus.
5. Repeat steps 3 and 4 for host Port 2.

IMPORTANT: The HSZ80 Array Controller has dual-host ports, Port 1 and Port 2, which may be configured at the same time. The configuration in Figure 2-1 only shows a connection to host Port 1. However, you may also configure host Port 2 the same way, depending on your system requirements.

Configuring Dual-Redundant Controllers in Transparent Failover Mode

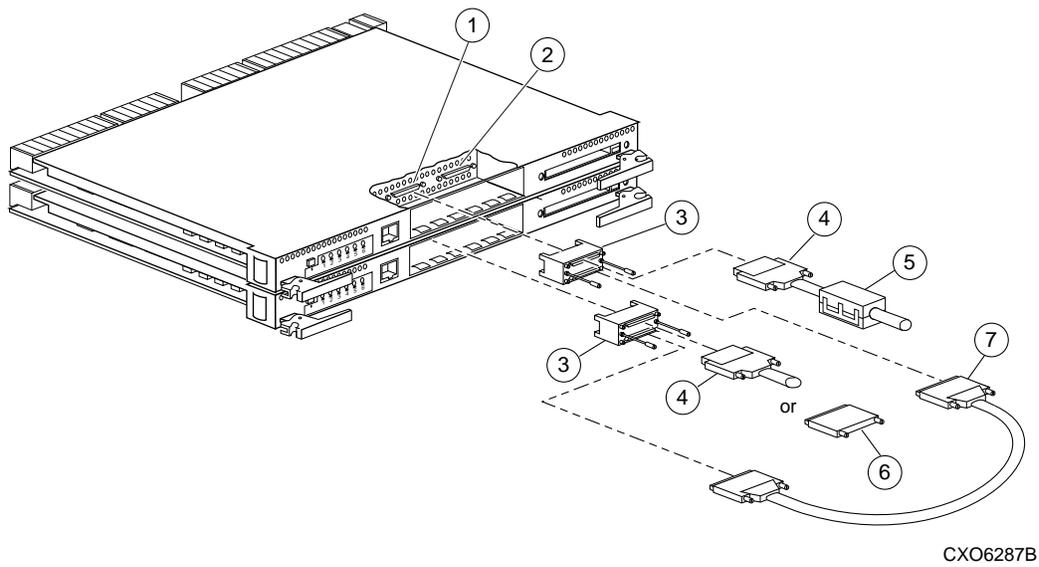


Figure 2-2. Cabling for a Dual-Redundant Controller Configuration in Transparent Failover

Table 2-2 Location of Parts for Transparent Failover

Item	Description
①	Host port 1
②	Host port 2
③	Trilink connector
④	Host bus cable
⑤	Ferrite bead
⑥	Terminator
⑦	Jumper cable

Follow these steps to configure a 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 *HSZ80 Array Controller ACS Version 8.3 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 2-25, for details about PVA switch settings.

2. Remove the program card ESD cover, and insert the program card into the controller. Replace the ESD cover. Repeat for the other controller.
3. Turn on the power to the enclosure or cabinet by turning on the ac input boxes.
4. Establish a local connection to the controllers.

See “Establishing a Local Connection to the Controller,” page 2-26, for details about creating a local connection.

2-12 Configuring an HSZ80 Array Controller

5. Put this controller into transparent failover mode. Use the following command:

CAUTION: Issuing the SET FAILOVER COPY command will overwrite data on the companion controller. If you are reconfiguring the controllers, see “CONFIGURATION RESET” in Chapter 5, “CLI Commands,” before proceeding and for more information.

```
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 2-29.

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

```
SET THIS_CONTROLLER MIRRORRED_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,” page 1-20.

For details about mirrored write-back cache, see “Enabling Mirrored Write-Back Cache,” page 2-31.

7. Set the SCSI target ID numbers for the controller. Use the following command:

```
SET THIS_CONTROLLER PORT_1_ID= (n,n)
```

or

```
SET THIS_CONTROLLER PORT_2_ID= (n,n)
```

See “Setting SCSI Target ID Numbers,” page 2-33, for details about valid SCSI target ID numbers.

8. If you are configuring controllers in transparent failover mode, prefer some or all SCSI target ID numbers to the controllers. Use the following command:

```
SET THIS_CONTROLLER PORT_1_PREFERRED_ID = (n, n)
```

or

```
SET THIS_CONTROLLER PORT_2_PREFERRED_ID = (n,n)
```

where *n,n* is a subset of the target ID numbers you declared in step 5.

See “Using Preferred ID Numbers,” page 2–35, for details about preferred SCSI target ID numbers.

9. Set the host function mode for the controller using the target ID numbers set in step 7 and in step 8. Use the following command:

NOTE: The default host function mode defaults to Host Mode A. The `HOST_FUNCTION` switch changes the host function mode to a different mode.

```
SET THIS_CONTROLLER HOST_FUNCTION= (ID, MODE)
```

or

```
SET OTHER_CONTROLLER HOST_FUNCTION= (ID, MODE)
```

To change the host function mode, see “SET controller,” page 5–91 for information on the `HOST_FUNCTION` switch.

10. Restart the controllers using the following command:

NOTE: After restarting the other controller, the EMU alarm will sound. To stop it, restart this controller.

```
RESTART OTHER_CONTROLLER
```

```
RESTART THIS_CONTROLLER
```

See “RESTART controller” in Chapter 5, “CLI Commands.”

2-14 Configuring an HSZ80 Array Controller

11. 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.

12. 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.

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

See the following section, “Optional Steps,” and Chapter 5, “CLI Commands.”

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

```
SHOW THIS_CONTROLLER FULL
```

See “SHOW” in Chapter 5, “CLI Commands.”

IMPORTANT: If you are using a PC or terminal with the maintenance port cable, you must connect the SCSI bus cables in the desired configuration. See “Connecting in Transparent Failover Mode to the Host,” page 2-15, for more information about configurations.

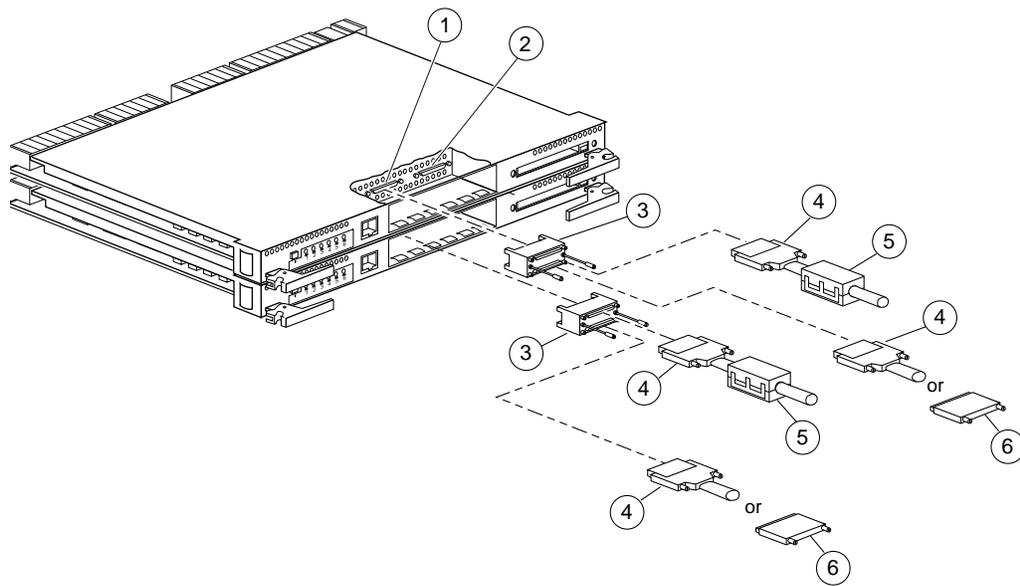
Connecting in Transparent Failover Mode to the Host

Use the following steps and Figure 2-2 to connect a pair of dual-redundant controllers to the host:

1. Configure both controllers. See “Configuring Dual-Redundant Controllers in Transparent Failover Mode,” page 2-10.
2. Ensure the host and adapter are also configured. See the host user’s guide for details.
3. Connect two trilink connectors between both controllers going into host Port 1 with a jumper cable:
 - a. Attach one end of the jumper cable to one trilink.
 - b. Attach the other end of the jumper cable to the other trilink.
4. Connect the host bus cable onto one end of the trilink connector:
 - a. For a BA370 enclosure residing in an SW600 cabinet, snap the ferrite bead on the bus cable within one inch of the controller.
5. Connect a terminator on the end of the trilink connector for host Port 1:
 - a. If the controller is at the end of the host bus, connect a terminator to the other connector on the front of the trilink connector.
 - b. Otherwise, connect another host bus cable that continues to the next device on the bus. Be sure to install a terminator at the end of the bus.
6. Repeat steps 3, 4, and 5 for host Port 2 on both controllers.

IMPORTANT: The HSZ80 Array Controller has dual-host ports, Port 1 and Port 2, which may be configured at the same time. The configuration in Figure 2-2 only shows a connection to host Port 1. However, you may also configure host Port 2 the same way, depending on your system requirements.

Configuring Dual-Redundant Controllers in Multiple-Bus Failover Mode



CXO6286f

Figure 2-3. Cabling for a Dual-Redundant Controller Configuration in Multiple-Bus Failover

Table 2-3 Location of Parts for Multiple-Bus Failover

Item	Description
①	Host port 1
②	Host port 2
③	Trilink connector
④	Host bus cable
⑤	Ferrite bead
⑥	Terminator

Follow these steps to configure a 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 *HSZ80 Array Controller ACS Version 8.3 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 2-25, for details about PVA switch settings.

2. Remove the program card ESD cover, and insert the program card into the controller. Replace the ESD cover. Repeat for the other controller.
3. Turn on the power to the enclosure or cabinet by turning on the ac input boxes.
4. Establish a local connection to the controllers.

See “Establishing a Local Connection to the Controller,” page 2-26, for details about creating a local connection.

2-18 Configuring an HSZ80 Array Controller

5. Put this controller in multiple-bus failover mode, using the following command:

CAUTION: Issuing the SET MULTIBUS_FAILOVER COPY command will overwrite data on the companion controller. If you are reconfiguring the controllers, see “CONFIGURATION RESET” in Chapter 5, “CLI Commands,” before proceeding and for more information.

```
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 “Selecting a Failover Mode,” page 2-29, for details about failover modes.

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

```
SET THIS_CONTROLLER MIRRORRED_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,” page 1-20.

For details about mirrored write-back cache, see “Enabling Mirrored Write-Back Cache,” page 2-31.

7. Set the SCSI target ID numbers for the controller. Use the following command:

```
SET THIS_CONTROLLER PORT_1_ID= (n,n)
```

or

```
SET THIS_CONTROLLER PORT_2_ID= (n,n)
```

See “Setting SCSI Target ID Numbers,” page 2-33, for details about valid SCSI target ID numbers.

8. Set the host function mode for the controller using the target ID numbers set in step 7. Use the following command:

NOTE: The default host function mode defaults to Host Mode A. The `HOST_FUNCTION` switch changes the host function mode to a different mode.

```
SET THIS_CONTROLLER HOST_FUNCTION= (ID, MODE)
```

or

```
SET OTHER_CONTROLLER HOST_FUNCTION= (ID, MODE)
```

To change the host function mode, see “SET controller,” page 5-91 for information on the `HOST_FUNCTION` switch.

9. Restart the controllers using the following command:

NOTE: After restarting the other controller, the EMU alarm will sound. To stop it, restart this controller.

```
RESTART OTHER_CONTROLLER
```

```
RESTART THIS_CONTROLLER
```

See “RESTART controller” in Chapter 5, “CLI Commands.”

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.

2-20 Configuring an HSZ80 Array Controller

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 the following section, “Optional Steps,” and Chapter 5, “CLI Commands.”

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

```
SHOW THIS_CONTROLLER FULL
```

See “SHOW” in Chapter 5, “CLI Commands.”

IMPORTANT: If you are using a PC or terminal with the maintenance port cable, you must connect the SCSI bus cables in the desired configuration. See “Connecting in Multiple-Bus Failover Mode to the Host,” page 2-21, for more information about configurations.

Connecting in Multiple-Bus Failover Mode to the Host

Use the following steps and Figure 2-3 to connect a pair of multiple-bus failover, dual-redundant controllers to the host:

1. Configure both controllers. See “Configuring Dual-Redundant Controllers in Multiple-Bus Failover Mode,” page 2-16.
2. Ensure the host and adapter are also configured. See the host user’s guide for details.
3. Connect a trilink connector on host Port 1 or host Port 2 for one controller:
 - a. Connect the host bus cable onto one end of each trilink connector.
 - b. For a BA370 enclosure residing in an SW600 cabinet, snap the ferrite bead on the bus cable within one inch of the controller.
4. Connect another trilink connector on host Port 1 or host Port 2 for a second controller:
 - a. Connect the host bus cable onto one end of each trilink connector.
 - b. For a BA370 enclosure residing in an SW600 cabinet, snap the ferrite bead on the bus cable within one inch of the controller.
5. Connect a terminator on the end of each trilink connector:
 - a. If the controller is at the end of the host bus, connect a terminator to the other connector on the front of the trilink connector.
 - b. Otherwise, connect another host bus cable that continues to the next device on the bus. Be sure to install a terminator at the end of the bus.

2-22 Configuring an HSZ80 Array Controller

- Repeat steps 3, 4, and 5 for connecting a third and a fourth host bus.

IMPORTANT: The HSZ80 Array Controller has dual-host ports, Port 1 and Port 2, which may be configured at the same time. The configuration in Figure 2-3 shows a two-bus connection going into host Port 1. However, you may also configure host Port 2 the same way, depending on your system requirements. For a two-bus connection, the bus connections can occur on either host port on two controllers. A third and fourth bus connection can also be made if three or four buses are desired.

Optional Steps

You can perform these steps at any time during configuring.

Change 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 "HSZ>."

Type the following command:

```
SET THIS_CONTROLLER PROMPT = "new prompt"
```

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

```
SET OTHER_CONTROLLER PROMPT = "new prompt"
```

See "SET controller" in Chapter 5, "CLI Commands."

Set the maximum data-transfer rate

Use the following command:

```
SET THIS_CONTROLLER PORT_1_TRANSFER_RATE_REQUESTED = speed
```

or

```
SET THIS_CONTROLLER PORT_2_TRANSFER_RATE_REQUESTED = speed
```

If you are configuring dual-redundant controllers, also set the transfer rate for the other controller. However, if you are running more than one HSZ80 SCSI host port on the same bus, you must set their transfer rates to be identical. Use the following command:

```
SET OTHER_CONTROLLER PORT_1_TRANSFER_RATE_REQUESTED=speed
```

or

```
SET OTHER_CONTROLLER PORT_2_TRANSFER_RATE_REQUESTED=speed
```

See Chapter 5, “CLI Commands,” for more information about setting the data transfer rate.

UPS Support

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.

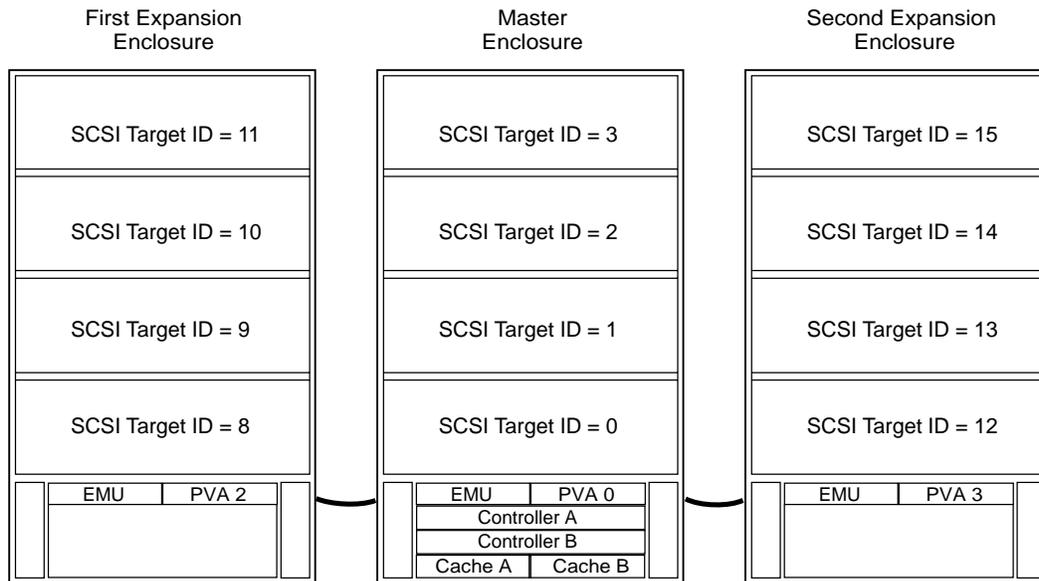
See "SET controller" in Chapter 5, "CLI Commands," for details about setting the UPS.

See Table 1-8 on page 1-23 and Table 1-9 on page 1-25 for information about cache policies.

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 controllers. Use PVA IDs 2 and 3 for the additional enclosures. Figure 2-4 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 2-4. SCSI Target ID Numbers on the Controller Device Bus and PVA Settings in an Extended Subsystem

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, as shown in Figure 2-5, 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.

The maintenance port cable shown in Figure 2-5 has a 9-pin connector molded onto the end for a PC connection. If you need a terminal connection or a 25-pin connection, optional cabling is available and can be ordered. See Figure 1-5 on page 1-12 for an illustration.

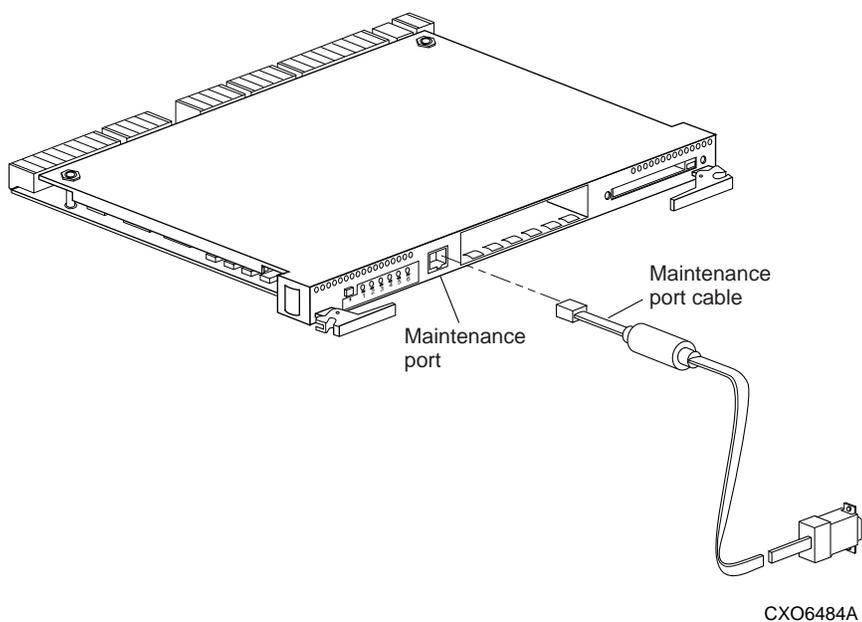


Figure 2-5. PC/Terminal to Maintenance Port Connection



CAUTION: The maintenance 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. Disconnect all maintenance port cables 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 2-6.
 - a. Plug the maintenance port cable into the maintenance port for a PC connection.

2-28 Configuring an HSZ80 Array Controller

- b. If you are using a terminal instead of a PC, you can order a cable from the local field service office as shown in figure 1-5.
2. Turn on the PC or terminal.
3. Configure the PC or terminal for 9600 baud, 8 data bits, 1 stop bit, and no parity. The default baud rate for the HSZ80 Array Controller is 9600, which makes communication between the two compatible.
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: 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 other controller is the "other controller" as shown in Figure 2-6.

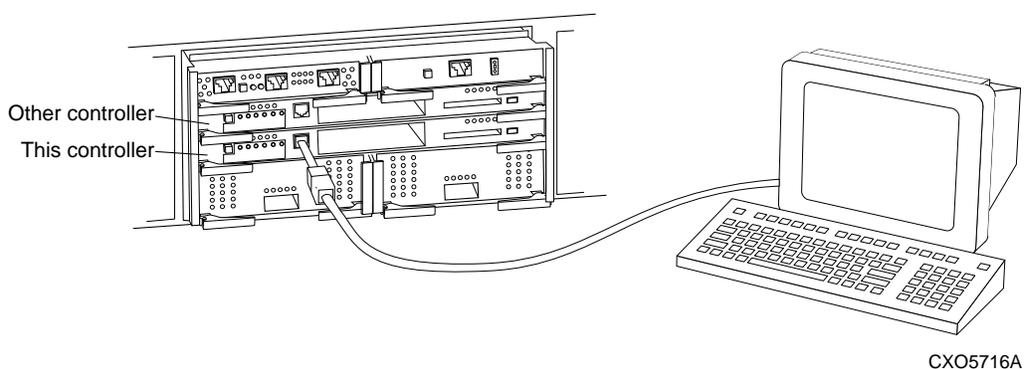


Figure 2-6. "This Controller" and "Other Controller"

Selecting a Failover Mode

In selecting a failover mode, you have two choices: transparent failover or multiple-bus failover. If you want the failover to occur without any intervention from the host, then transparent failover is best suited. The devices attached to the failed controller fail over to the surviving controller. With multiple-bus failover, the host intervenes during failover and sends commands to the companion controller.

Using Transparent Failover Mode

Transparent failover is a dual-redundant controller configuration in which two controllers are connected to the same host and device buses. Use this configuration if you want to use two controllers to service the entire group of storagesets, single-disk units, and other storage devices. Because both controllers service the same storage units, either controller can continue to service all of the units if its companion controller fails.

Distribute the I/O load between the two controllers in a transparent failover configuration using the following commands to specify which controller handles which targets and units:

```
SET THIS_CONTROLLER PORT_1_PREFERRED_ID = (n,n)
```

or

```
SET THIS_CONTROLLER PORT_2_PREFERRED_ID = (n,n)
```

Keep the following tips in mind if you configure controllers for transparent failover:

- Set your controllers for transparent failover before configuring devices. By doing this, devices, storagesets, and units added to one controller's configuration are automatically added to the other controller's configuration.
- If you decide to configure your devices before setting the controllers for transparent failover, make sure you know which controller has the good configuration information before specifying SET FAILOVER COPY command. See Chapter 5, "CLI Commands," for details about setting the failover.

2-30 *Configuring an HSZ80 Array Controller*

- For better subsystem performance, balance your assignment of target ID numbers across your dual-redundant pair of controllers. For example, if you are presenting four targets to the host, prefer two to one controller and two to the other controller.
- Balance your assignment of devices. For example, in an 18-device subsystem, place 3 devices on each of the 6 ports, rather than placing 6 devices on each of 3 ports.
- The controller to which you copy configuration information restarts after you enter the SET FAILOVER command.

Using Multiple-Bus Failover Mode

Multiple-bus (or host-assisted) failover is a dual-redundant controller configuration in which each controller has its own connection to the host. Thus, if one of the host connections to a controller fails, the host can cause units that became inaccessible to failover to the remaining viable connection. Because both controllers service the same storage units, either controller can continue to service all of the units if the other controller fails.

Keep the following points in mind when configuring controllers for multiple-bus failover:

- The host distributes the I/O load between the controllers.
 - The host must have two SCSI adapters as well as operating-system software to support the multiple-bus failover, dual-redundant controller configuration.
 - SCSI target ID number and LUN configurations on each controller must be identical.
 - Mount both controllers in the same BA370 rack-mountable enclosure and follow the steps in “Configuring an HSZ80 Array Controller,” page 2-3.
 - Partitioning is not supported.
-

Enabling Mirrored Write-Back Cache

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

- Both controllers support the same size cache: 64 MB, 128 MB, 256 MB, or 512 MB.
- Diagnostics indicates that both caches are good.
- If you have not enabled the CACHE_UPS switch, then both caches have a battery present. 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.

Selecting a Cache Mode

Before selecting a cache mode, you should understand the caching techniques supported by the cache module.

The cache module supports read, read-ahead, write-through, and write-back caching techniques that you can enable separately for each storage unit in a subsystem. For example, you can enable only read and write-through caching for some units while enabling only write-back caching for other units.

Fault-Tolerance

The cache module supports the following features to protect the availability of its unwritten (write-back) data:

- Nonvolatile memory (required for write-back caching)
- Dynamic caching techniques (automatic)

For details about these features, see “Fault-Tolerance for Write-Back Caching,” page 1–22.



Setting SCSI Target ID Numbers

In a subsystem, every SCSI device is assigned its own SCSI ID number. Setting an ID number, though, really depends on which SCSI bus you are talking about—the host SCSI bus or the controller device bus, which are both shown in Figure 2-7.

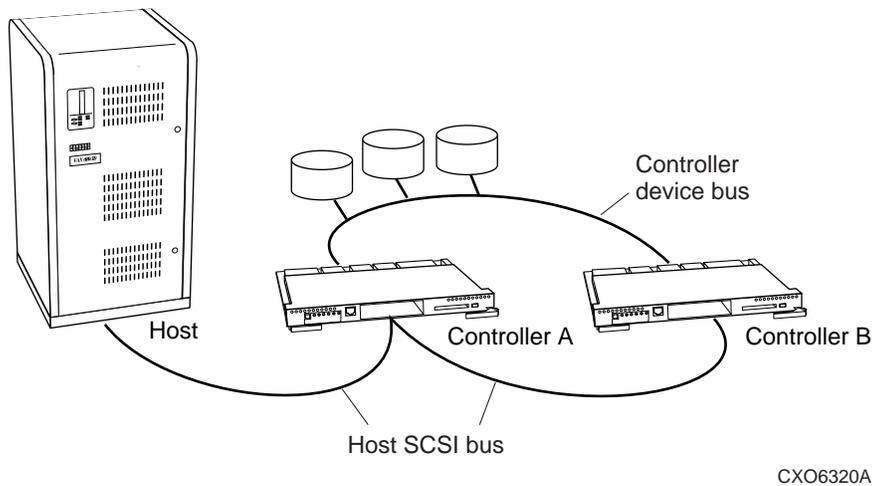


Figure 2-7. Host SCSI Bus and Controller Device Bus

Host SCSI Bus

Each BA370 rack-mountable enclosure in a configuration supports a range of SCSI target ID numbers. Each controller can access the target ID numbers that you specify with either of the following commands:

```
SET THIS_CONTROLLER PORT_1_ID = (n,n)
```

or

```
SET THIS_CONTROLLER PORT_2_ID = (n,n)
```

2-34 Configuring an HSZ80 Array Controller

You can view IDs on this controller with the following command:

```
SHOW THIS_CONTROLLER FULL
```

You can view IDs on the other controller with the following command:

```
SHOW OTHER_CONTROLLER FULL
```

The valid target ID numbers are 0–15, with one ID reserved for each host controller on the host bus. When both ports are in use, the 16 target ID numbers are shared across ports. Once a target ID has been set on one port, that same target may not be used on the other port.

Using more than one target ID number allows the controller to present more units to the host. Enclose multiple ID numbers in parentheses and separate each by a comma: *(n1, n2, n3, . . .)*

Device SCSI Bus

Device addressing allows the controller to identify the location of devices based on a port-target-LUN (PTL) number described in “Device PTL Addressing Convention within the Controller,” page 3–36.

Using Preferred ID Numbers

Use preferred SCSI target ID numbers to balance the I/O load of storage units among controllers and thereby improve the throughput for the dual-redundant pair of controllers.

You can specify a total of 15 SCSI target ID numbers across both host ports or you can specify 15 SCSI target ID numbers to either Port 1 or Port 2. Use the ID numbers you defined with the SET THIS_CONTROLLER PORT_1_ID (or PORT_2_ID) command.

Enclose multiple IDs in parentheses and separate them by a comma.

In a transparent failover configuration, the other controller automatically inherits the newly defined ID numbers. During normal operation, each controller services only those storage units that are associated with its preferred ID numbers.

You can also use the PREFERRED_ID switch to effectively make the other controller a hot standby by declaring that it has no preferred SCSI target ID numbers. Use the following commands:

```
SET OTHER_CONTROLLER NOPORT_1_PREFERRED_ID = (n,n)
```

or

```
SET OTHER_CONTROLLER NOPORT_2_PREFERRED_ID = (n,n)
```

By declaring that it has no preferred IDs, this controller will not respond to any SCSI target ID numbers on that port's host SCSI bus. Instead, the other controller will process all I/O during normal operation.

Chapter 3

Creating StorageSets

This chapter provides information to help you create storageSets for your subsystem. The procedure in this chapter takes you through the planning steps and procedures for creating storageSets.

Introduction

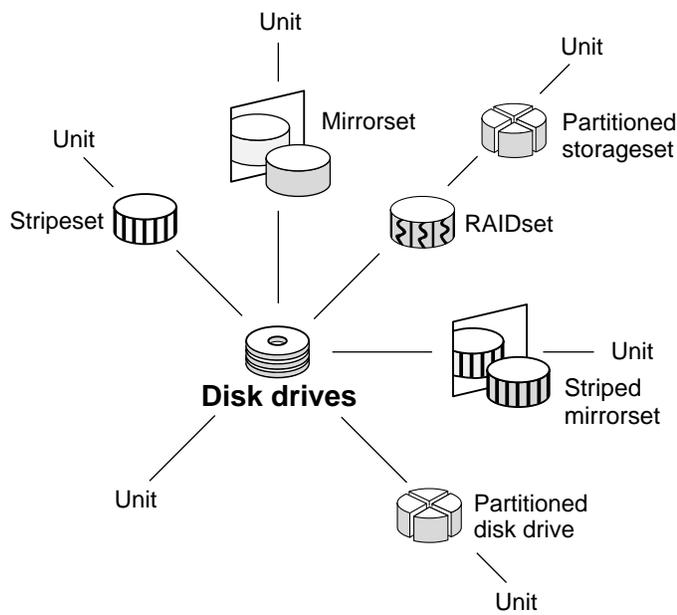
Storage sets are implementations of RAID technology, also known as a *Redundant Array of Independent Disks*. Every storage set shares one important feature: each one looks like a single storage unit to the host, regardless of the number of drives it uses.

You can create storage units by combining disk drives into storage sets, such as stripe sets, RAID sets, and mirror sets, or by presenting them to the host as single-disk units, as shown in Figure 3-1.

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

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

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



CXO5368B

Figure 3-1. Units Created from Storagesets, Partitions, and Drives

Table 3-1 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 Storaagesets

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

1. Create a storaageset and device profile. See “Creating a Storaageset and Device Profile,” page 3-5 for suggestions about creating a profile.
 2. Determine your storage requirements. Use the questions in “Determining Storage Requirements,” page 3-7 to help you.
 3. Choose the type of storaagesets you need to use in your subsystem. See “Choosing a Storaageset Type,” page 3-8 for a comparison and description of each type of storaageset.
 4. Select names for your storaagesets and units. See “Creating a Storaageset Map,” page 3-34 for details about selecting names.
 5. Assign unit numbers to storaagesets so the host can access the units. See “Node IDs,” page 3-30 for information about how to assign unit numbers to storaagesets.
 6. Create a storaageset map to help you configure your subsystem. See “Creating a Storaageset Map,” page 3-34 for suggestions about creating a storaageset map.
 7. If you are going to partition your storaagesets, plan the partitions. See “Planning Partitions,” page 3-40 for information about partitions and how to plan for them.
 8. Choose the switches that you will want to set for your storaagesets and devices. See “Choosing Switches for Storaagesets and Devices,” page 3-42 for a description of the switches you can select for storaagesets.
 9. Configure the storaagesets you have planned using one of these methods:
 - ❑ Use StorageWorks Command Console (SWCC) to set up and manage RAID storage subsystems. See the *SWCC Getting Started Guide* for details about using SWCC to configure your storaagesets.
 - ❑ Use CLI commands. This method allows you flexibility in defining and naming storaagesets. See “Configuring Storaagesets,” page 4-1 for information about configuring each type of storaageset 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 helps you to 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 3-3.

Appendix contains blank profiles that you can copy and use to record the details for your storageSets. Use the information in this chapter to help you make decisions when creating storageSet profiles.

3-6 Creating StorageSets

Type of StorageSet

<input type="checkbox"/> Mirrorset	<input checked="" type="checkbox"/> RAIDset	<input type="checkbox"/> Stripeset	<input type="checkbox"/> Striped Mirrorset
------------------------------------	---	------------------------------------	--

StorageSet Name.....accept default values

Disk Drives.....DISK10300, DISK20300, DISK30300

Unit Number.....accept default

Partitions

| Unit # |
|--------|--------|--------|--------|--------|--------|--------|--------|
| % | % | % | % | % | % | % | % |

RAIDset Switches

<input checked="" type="checkbox"/> Normal (default)	<input checked="" type="checkbox"/> No (default)	<input checked="" type="checkbox"/> Best performance (default)
<input type="checkbox"/> Fast	<input type="checkbox"/> Yes, missing:	<input type="checkbox"/> Best fit
		<input type="checkbox"/> None

Mirrorset Switches

<input type="checkbox"/> Best performance (default)	<input type="checkbox"/> Normal (default)	<input type="checkbox"/> Least busy (default)
<input type="checkbox"/> Best fit	<input type="checkbox"/> Fast	<input type="checkbox"/> Round robin
<input type="checkbox"/> None		<input type="checkbox"/> Disk drive:

Initialize Switches

<input checked="" type="checkbox"/> Automatic (default)	<input type="checkbox"/> No (default)	<input checked="" type="checkbox"/> Destroy (default)
<input type="checkbox"/> 64 blocks	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Retain
<input type="checkbox"/> 128 blocks		
<input type="checkbox"/> 256 blocks		
<input type="checkbox"/> Other:		

Unit Switches

<input checked="" type="checkbox"/> Yes (default)	<input checked="" type="checkbox"/> Yes (default)	<input checked="" type="checkbox"/> 32 blocks (default)	<input type="checkbox"/> Normal
<input type="checkbox"/> No	<input type="checkbox"/> No	<input type="checkbox"/> Other:	<input type="checkbox"/> Failsafe
<input type="checkbox"/> Write Cache	<input checked="" type="checkbox"/> No (default)	<input checked="" type="checkbox"/> Run (default)	*Valid only for mirrorsets with DT support enabled
<input checked="" type="checkbox"/> No	<input type="checkbox"/> Yes	<input type="checkbox"/> NoRun	

Figure 3-2. 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 Table 3-2 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 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 3-2 compares different kinds of storageSets to help you determine which ones satisfy your requirements.

Table 3-2 A Comparison of Different Kinds of StorageSets

StorageSet Type	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 noncritical data
Mirrorset (RAID 1)	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. Large I/O requests are 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 large I/O request or 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.

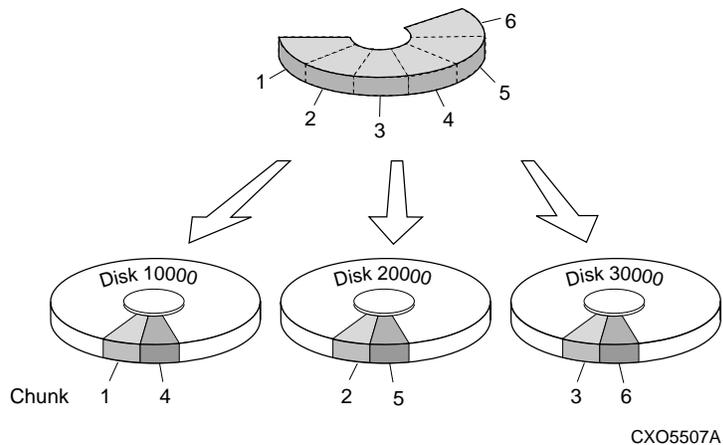


Figure 3-3. Striping Lets Several Disk Drives Participate in Each I/O Request

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 3-51, for information about setting the chunk size.

3-10 Creating Stagesets

A major benefit of striping is that it balances the I/O load across all of the disk drives in the stageset. 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 stagesets, consisting of stripesets, mirrorsets and RAIDsets (see Table 3-1).
- Reporting methods and size limitations prevent certain operating systems from working with large stripesets. See the *HSZ80 Array Controller ACS Version 8.3 Release Notes* or the *Quick Setup Guide* that came with your software platform kit for details about these restrictions.
- A stageset 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 stageset when the stageset is initialized (the base member size). Thus, if you combine 9 GB disk drives with 4 GB disk drives in the same stageset, the 4 GB disk drive will be the base member size and you will waste 5 GB of capacity on each 9 GB member.

If you need high performance and high availability, consider using a RAIDset, striped mirrorset, or a host-based shadow of a stripeset.

NOTE: A striped mirrorset cannot be created if the mirrorsets are enabled for DT_SUPPORT.

- 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 1/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 the Figure 3-4.

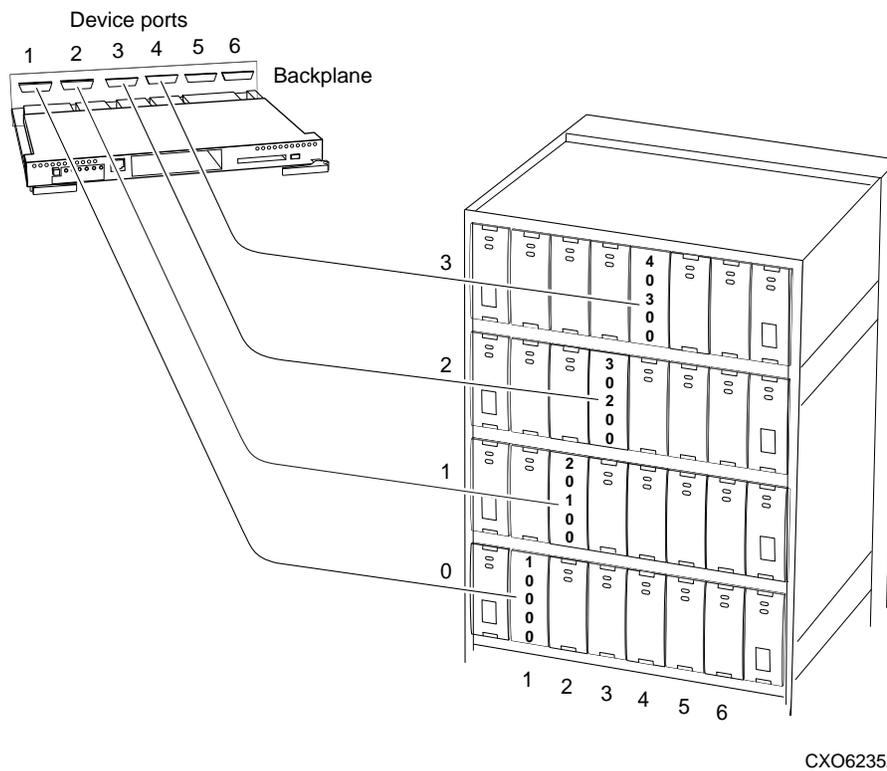


Figure 3-4. Distribute Members across Ports

- Stripesets contain between 2 and 24 members.
- Stripesets are well-suited for the following applications:

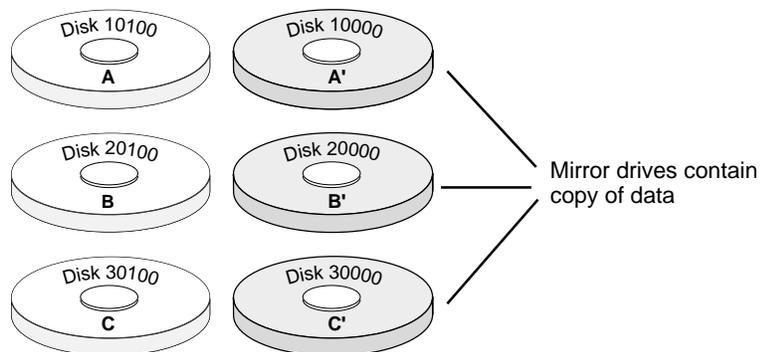
3-12 *Creating Stagesets*

- 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
- Stagesets 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 stageset.

Using Mirrorsets to Ensure Availability

Mirrorsets use redundancy to ensure availability, as illustrated in Figure 3-5. 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.



CXO5511A

Figure 3-5. Mirrorsets Maintain Two Copies of the Same Data

Considerations for Planning a Mirrorset

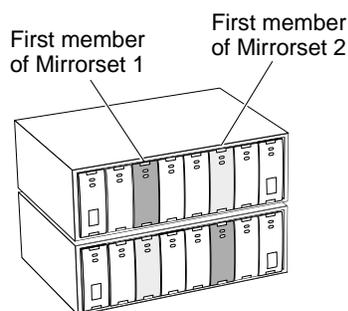
Keep these points in mind as you plan your mirrorsets:

- A controller can support up to 30 storage sets, consisting of mirrorsets and RAID sets. Mirrorsets that are members of a stripeset count against this limitation (see Table 3-1).
- 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 member 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.

3-14 Creating Storagesets

- If you are using more than one mirrorset in your subsystem, you should put the first member of each mirrorset on different buses as shown in Figure 3-6. 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 (see “Read Source,” page 3-46). 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 read traffic to your mirrorsets. When a controller receives a request to write data to a mirrorset, it accesses and writes to all members.



CXO5506A

Figure 3-6. First Mirrorset Members on Different Buses

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 3-46, 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 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.
- Evenly distribute the members across the device ports to balance load and provide multiple paths as shown in Figure 3-4 on page 3-11.
- Mirrorsets are well-suited for the following:
 - Any data for which availability 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 (worse than JBODs, better than RAID5 RAID-sets)
 - Applications for which cost is a primary issue

Using RAIDsets to Increase Performance and Availability

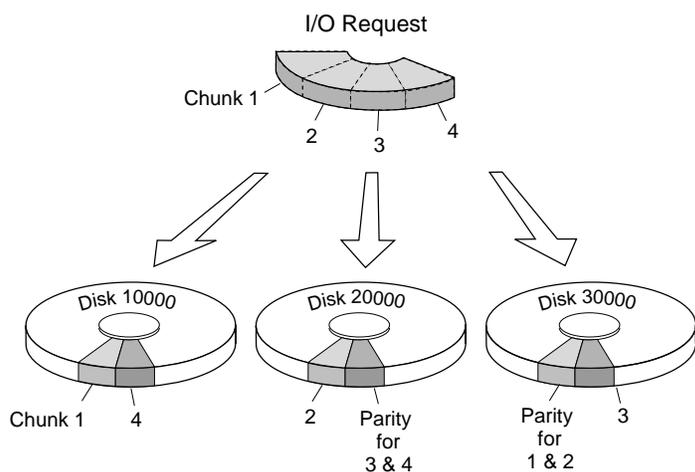
RAIDsets are enhanced stripesets—they use striping to increase I/O performance and distributed-parity data to ensure data availability. Figure 3-7 illustrates the concept of RAIDsets and parity data.

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 and reconstructs the first failed disk drive.)

3-16 Creating Stagesets

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 rate. You can set the chunk size or let the controller set it automatically. See “Chunk Size,” page 3-51, for information about setting the chunk size.



CXO5509A

Figure 3-7. Parity Ensures Availability; Striping Provides Good Read Performance

Considerations for Planning a RAIDset

Keep these points in mind as you plan your RAIDsets:

- A controller can support up to 20 stagesets, consisting of RAIDsets (see Table 3-1).

- Reporting methods and size limitations prevent certain operating systems from working with large RAIDsets. See the *HSZ80 Array Controller ACS Version 8.3 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 3-4 on page 3-11.
- 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.
- RAIDset units are set to WRITEBACK_CACHE by default which increases a unit's performance.
- RAIDset and mirrorset members should be placed 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

Striped mirrorsets are simply stripesets whose members are mirrorsets. Consequently, this kind of stageset combines the performance of striping with the reliability of mirroring. The result is a stageset with very high I/O performance and high data availability (see Figure 3-8).

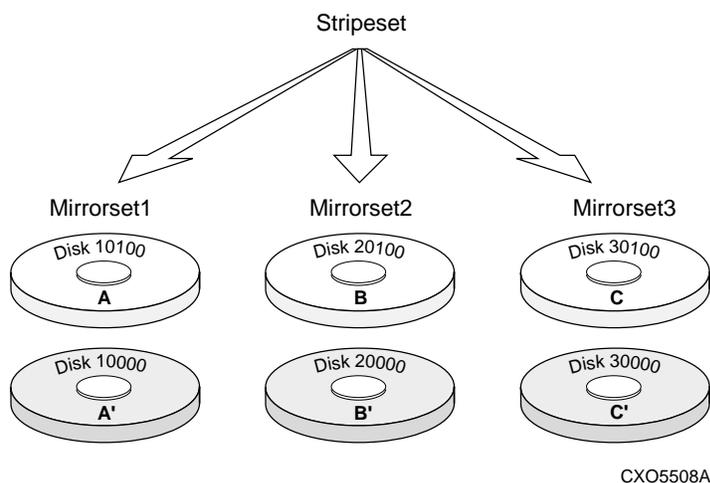


Figure 3-8. Striping and Mirroring in the Same Stageset

The failure of a single disk drive has no effect on this stageset'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 stageset 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 3-10, and “Considerations for Planning a Mirrorset,” page 3-13.

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

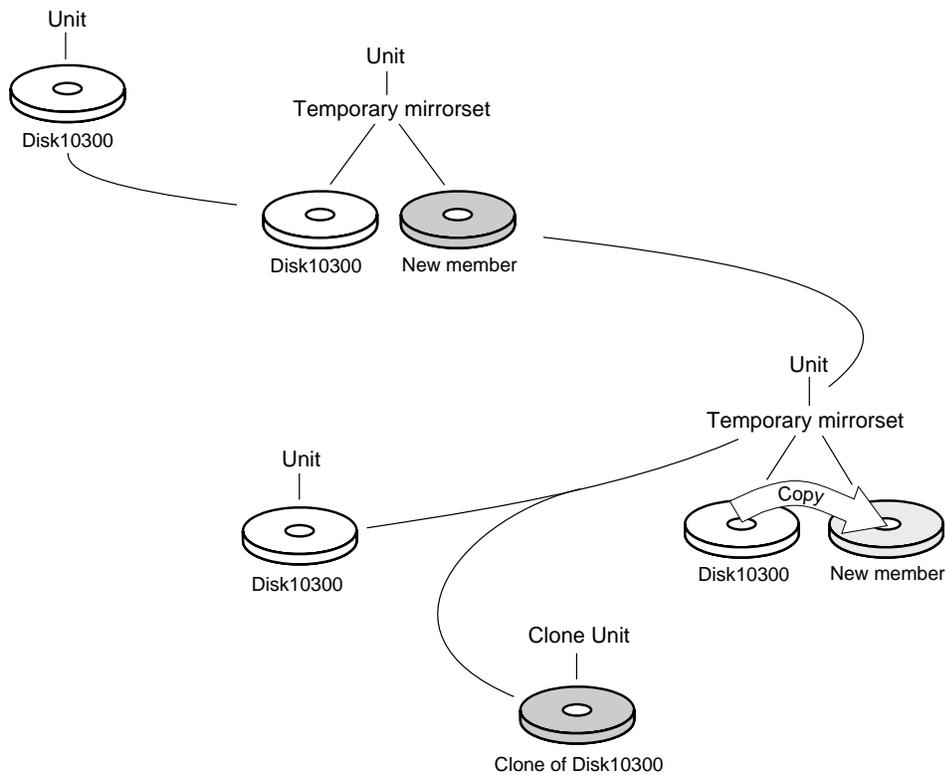
Cloning Data for Backup

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 stageset 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 3-9 to duplicate each member of a unit.





CX05510A

Figure 3-9. CLONE Steps for Duplicating Unit Members

Use the following steps to clone a single-disk unit, stripeset, or mirrorset:

1. Establish a connection to the controller that accesses the unit you want to clone.

3-22 *Creating Stagesets*

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.

Example

This 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

3-24 *Creating Stagesets*

```
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 DISK10000 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 STRIPESSET 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.

See “Considerations for Saving the Configuration,” page 3-55, and “Saving Configuration Information in Dual-Redundant Controller Configurations,” page 3-56, for more information.

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 switch 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 DISK $_{mmn}$  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 DISKnnnn 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 that indicates status and how many devices have copies of the configuration, as shown in the following example.

```
SHOW THIS_CONTROLLER FULL
```

Controller:

```
HSZ80 (C) DEC ZG07010098 SOFTWARE XXXX-0, HARDWARE 0000
NODE ID = XXXX-XXXX-XXXX-XXXX
ALLOCATION_CLASS = 0
SCSI_VERSION = SCSI-2
CONFIGURED FOR DUAL-REDUNDANCY WITH ZG06309900
IN DUAL-REDUNDANT CONFIGURATION
DEVICE PORT SCSI ADDRESS 7
TIME: NOT SET
```

Host port_1:

```
SCSI TARGET(S) (0,1)
PREFERRED TARGET(S) (0)
TRANSFER_RATE_REQUESTED = 20MHZ
HOST FUNCTIONALITY MODE FOR ID 0 = A
HOST FUNCTIONALITY MODE FOR ID 1 = D
```

Host port_2:

```
SCSI TARGET(S) (2,3)
PREFERRED TARGET(S) (3)
TRANSFER_RATE_REQUESTED = 20MHZ
HOST FUNCTIONALITY MODE = A
COMMAND CONSOLE LUN IS TARGET 2, LUN 5
```

3-28 Creating Storagesets

Cache:

128 MEGABYTE WRITE CACHE, VERSION UNKNOWN
CACHE IS GOOD
UNFLUSHED DATA IN CACHE
CACHE_FLUSH_TIMER = DEFAULT (10 SECONDS)

Mirrored Cache:

NOT ENABLED

Battery:

FULLY CHARGED
EXPIRES: 08-MAY-2000
NOCACHE_UPS

Extended information:

TERMINAL SPEED 19200 BAUD, EIGHT BIT, NO PARITY, 1 STOP BIT
OPERATION CONTROL: 00000000 SECURITY STATE CODE: 36415
CONFIGURATION BACKUP ENABLED ON 2 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:					
NOTTRANSPORTABLE					
TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 10.00 MHZ negotiated)					
LOCAL					
Size: 4108970 blocks					
Configuration being backed up on this container					
DISK30300	disk	3	3	0	S2
DEC	RZ28M	(C)	DEC	1003	
Switches:					

NOTTRANSPORTABLE
TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 10.00 MHZ negotiated)
LOCAL
Size: 4108970 blocks
Configuration being backed up on this container

Node IDs

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. The node ID assigned to a subsystem never changes.

Each subsystem's node ID ends in zero, for example, 5000-1FE1-FF0C-EE00. The controller port IDs are derived from the node ID. In a subsystem with two controllers (a dual-redundant configuration), the port ID of Port 1 for both controllers is the node ID plus 1. In this example, both controllers' Port 1 port ID would be 5000-1FE1-FF0C-EE01. Similarly, both controllers would have the same port ID for Port 2, 5000-1FE1-FF0C-EE02. The controllers automatically assign their port IDs.

Use the CLI command `SHOW THIS_CONTROLLER` to display the subsystem's node ID. See Chapter 5, "CLI Commands," for more information about the `SHOW` command and node IDs. When you enter the `SHOW` command, the subsystem node ID displays as the `REPORTED NODEID` and will look like the following:

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

Restoring Node IDs

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

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

If a situation occurs that requires you to restore the node ID, you can restore it using the node ID and check sum printed on the sticker on the frame into which your controller is inserted. See the `SET controller` command in Chapter 5, "CLI Commands," for details about setting the node ID.



CAUTION: Never set two subsystems to the same node ID, or data corruption will occur. Each subsystem has its own unique node ID. This name is printed on the sticker affixed to the horizontal top edge of the Controller Card Cage above the EMU/PVA into which your controller is inserted. If you attempt to set the subsystem node ID to a name other than the one that came with the subsystem, the data on the subsystem will not be accessible.

Assigning Unit Numbers for Host Access to Stagesets

The HSZ80 controller supports up to 15 SCSI target IDs, each of which can present up to 32 logical unit numbers (LUNs) to a host, depending on the requirements of your operating system. However, a maximum number of 128 storage units can be presented to a host by a controller or a dual-redundant pair of controllers. One of these is reserved for the StorageWorks Command Console (SWCC) interface, which leaves 127 available for use.

You will need to assign a unique unit number to each stageset, single disk unit, or storage device that you want your host to know about in your subsystem. A unit number is an alphanumeric name that identifies each storage unit in your subsystem, such as D102 for a disk-based storage unit. The host uses these names 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: use D for disk drives or P for passthrough devices such as tape drives, loaders, and libraries.
 - A first number that indicates which controller accesses the storage unit during normal operation. Use one of the controller's SCSI target IDs: 0 to 15.
By carefully choosing the first number, you can establish preferred paths for all of your storage units in a dual-redundant configuration.
 - A second number that is always zero.
 - A third number that identifies the LUN for the device or storage unit (0 to 7).
-

Omit the leading zeroes for storage units associated with the controller's SCSI target ID zero. For example, use D2 instead of D002 for a storage set that's accessed through LUN 2 of the controller's SCSI target ID 0. Table 3-3 shows additional unit numbering examples.

Table 3-3 Unit Numbering Examples

Unit Number	Device Type	Target ID Number	LUN
D401	Disk	4	1
D1207	Disk	12	7
D5	Disk	0	5

NOTE: The host communicates with a logical unit based on its LUN address. The controller communicates with a device based on its Port-Target-LUN (PTL) address



CAUTION: Selecting SCSI-3 mode enables access to the Command Console LUN (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, systems administrators of each host must not attempt to access the CCL simultaneously.

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 3-10. 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	R1	R1	R1	spare			Power Supply
Power Supply	R2	R2	R2	spare			Power Supply
Power Supply	R2	R2	R2	spare			Power Supply
Power Supply	M1	M1	M2	M1			Power Supply

Figure 3-10. Storageset Map

To create a storageset map:

1. Copy the template from “Enclosure Template” in Appendix A.
2. Establish a local or remote connection to one of the controllers in your subsystem.

3-36 Creating Stagesets

3. Show the devices that are assigned to the controller. Use the following syntax:
SHOW DEVICES
4. Locate each device assigned to the controller and record its location on your copy of the cabinet template. Use the following syntax:
LOCATE *device_name*

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

5. Turn off the LED using the following syntax:
LOCATE CANCEL

The controller names each device based on its 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 stageset map by circling each group of disk drives that you want to combine into a stageset or put into the spareset. Label each group with a stageset name, for example: RAID1 for a RAIDset; Mirr1 for a mirrorset; and Stripe1 for a stripeset.

Device PTL Addressing Convention within the Controller

Your controller has six SCSI-2 device ports. Each device port supports one to four devices or "targets" per port. Every device uses LUN 0, except remote devices in a DT configuration, which can use nonzero LUN values.

The controller identifies the location of devices based on a PTL numbering scheme. The controller uses the PTL address to locate devices.

- P—Designates the controller's SCSI device port number (1 through 6).
 - T—Designates the target 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 numbers 6 and 7 are used for the controllers; ID numbers 4 and 5 are never used.
-

NOTE: The controller operates with BA370 rack-mountable enclosures assigned enclosure ID numbers 0, 2, and 3. You set these ID numbers using 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.

- L—Designates the LUN of the device.

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 3-11.

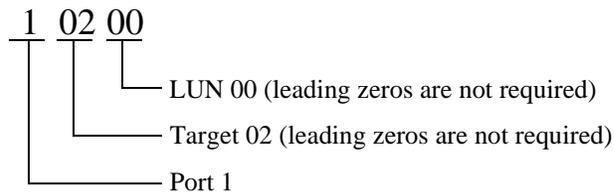


Figure 3-11. PTL Naming Convention

Figure 3-12 shows the addresses for each device in an extended configuration. Use this figure along with “Configuration Rules,” page 2-2, to help you work with the devices in your configuration.



CAUTION: Selecting SCSI-3 mode enables LUN 0 of each target for the Command Console LUN (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, systems administrators of each host must not attempt to access the CCL simultaneously.

3-38 Creating Stagesets

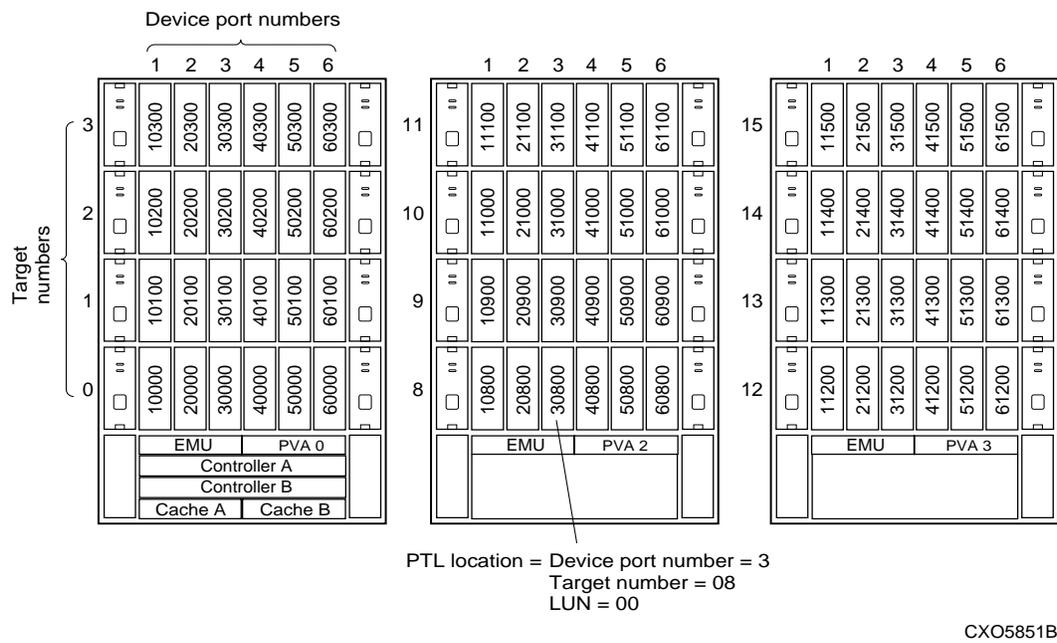
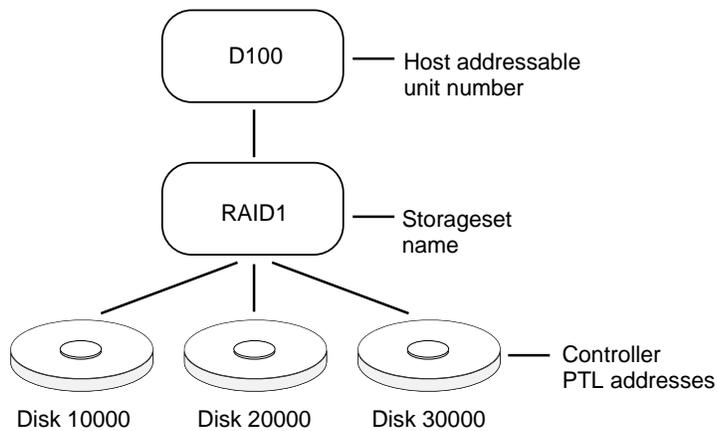


Figure 3-12. PTL Addressing in an Extended Configuration

In Figure 3-12, 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, it names it DISK10200.

When your controller receives an I/O request, it identifies the storageset unit number for the request, then correlates the unit number to the storageset name. From the storageset name, the controller locates the appropriate device for the I/O request. (For example, the RAIDset “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 3-13 illustrates the concept of PTL addressing.



CXO6186A

Figure 3-13. Locating Devices using PTLs

Planning Partitions

Use partitions to divide a storage set or disk drive into smaller pieces, which can be presented to the host as its own storage unit. Figure 3-14 shows the conceptual effects of partitioning a single-disk unit.

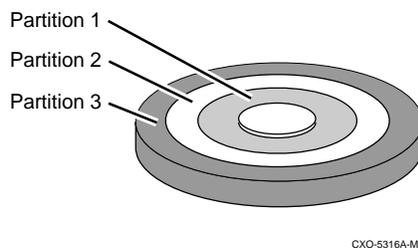


Figure 3-14. Partitioning a Single-Disk Unit

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 storage set or device. Because partitions are separately-addressable storage units, you can partition a single storage set to service more than one user group or application.

Defining a Partition

Partitions are expressed as percentages of the storage set 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 “Partitioning a StorageSet or Disk Drive,” page 4–10, 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 target ID (host-addressable SCSI ID). Thus, if you set a preferred controller for that ID, all the partitions in that storageSet will inherit that preferred controller. 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.

When you use CLI commands to manually configure the storageSet or device, the procedures in Chapter 4, “Configuring StorageSets”, 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 for a StorageSet or Device,” page 4-17, 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 a 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.

3-44 *Creating Storagesets*

- **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
- Disaster Tolerance Support

Replacement Policy

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

- NOTE:** If DT_SUPPORT is enabled, no policy can be selected.
- 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 a drive that also provides the best fit. This switch is not valid if DT_SUPPORT is enabled.
 - 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 mirrorset was initialized). If more than one disk drive meets this criterion, this switch selects one that also provides the best performance. This switch is not valid if DT_SUPPORT is enabled.
 - 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 they are 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 they are available.
 - **READ_SOURCE=DISKnnnn** 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.
-

Disaster Tolerance Support

Specify if disaster tolerant functionality is supported by this mirrorset:

- DT_SUPPORT is enabled if some members are or may be remote and disaster tolerant functionality is supported. The mirrorset has the following restrictions: NOPOLICY is enforced and NOWRITEBACK_CACHE is enforced
- NODT_SUPPORT is enabled if all members are—and intend to remain—local, and if disaster tolerant functionality is not supported

Device Switches

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

- Transportability (only disk drives)
- Device transfer rate
- Local/Remote (only disk drives)

Transportability

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

- NOTTRANSPORTABLE 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 NOTTRANSPORTABLE switch:
 - When you bring nontransportable 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 stagesets within the units contain metadata that matches the configuration. If there is no match or metadata is not present, the controller displays a message; initialize the disk or stageset 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 stagesets 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 storageset or spareset.
- ❑ Cannot be partitioned.
- ❑ 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 NOTTRANSPORTABLE for all disks used in RAIDsets, stripesets, and mirrorsets.

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

Local/Remote

Specify the disk characteristics in a mirrorset, either local (default) or remote.

- Disks set to REMOTE can only be added to mirrorsets and JBOD (that is, not RAID5 raidsets).
 - Disks set to REMOTE can only be added to mirrorsets that are set to DT_SUPPORT.
 - Disks set to LOCAL can be added to mirrorsets that are set to either NODT_SUPPORT or DT_SUPPORT.
 - Disks set to REMOTE that are part of a mirrorset will not normally be the read source.
 - Setting the switch to REMOTE forces a disk to be NOTTRANSPORTABLE.
-

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

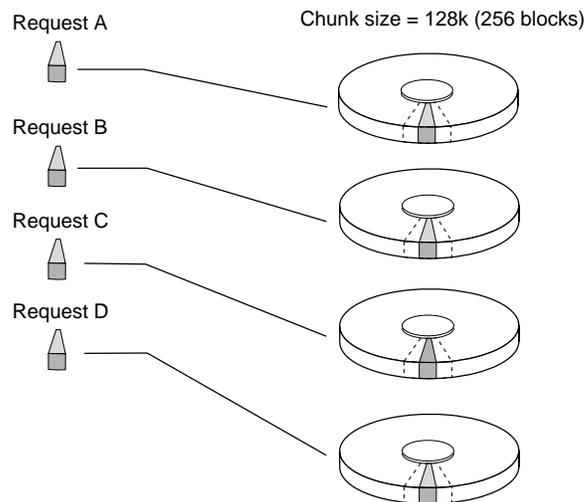
Specify a chunk size to control the stripesize used for RAIDsets and stripesets:

- `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.

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 stageset 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 3-15.



CXO-5135A-MC

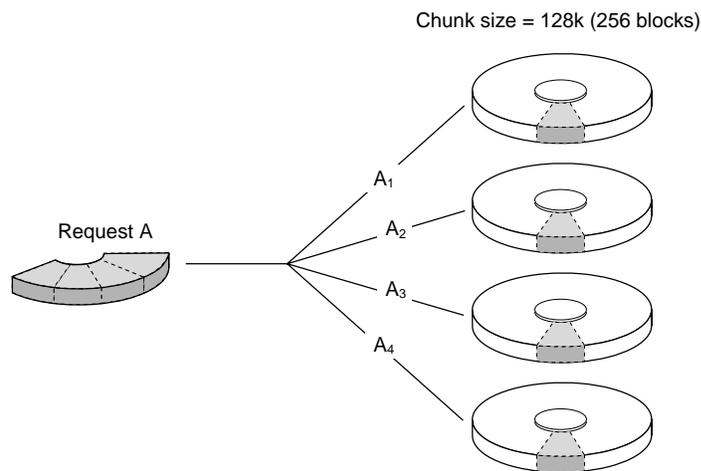
Figure 3-15. Chunk Size Larger than the Request Size

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 3-16.



CXO-5172A-MC

Figure 3-16. 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 3-4 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 3-4 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 for single-controller configuration configurations, the configuration could be jeopardized if the controller fails. For this reason, you may 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 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.

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 Controller Configurations," page 3-56 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 stageset 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.
- To enable SAVE_CONFIGURATION on a stageset containing user data, you must back up the data, re-initialize the stageset, and restore the user data.
- If you previously configured stagesets with the SAVE_CONFIGURATION option, you do not need to initialize them again after you reconfigure your devices with a new controller. The stageset retains the option until it is re-initialized, even if the controller is replaced or the stageset is moved to a new subsystem.
- When you replace a controller, make sure the replacement controller does not contain any configuration data. If the controller is not new, initialize it with the SET THIS_CONTROLLER INITIAL_CONFIGURATION command. If you do not take this precaution, you can lose configuration data if nonvolatile memory changes.

Saving Configuration Information in Dual-Redundant Controller Configurations

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

- The controller-unique data for both controllers is saved.
 - The restore feature (holding in port button 6 and pressing the reset button on the controller's operator control panel) will not function in a dual-redundant controller configuration. If the saved configuration was for a dual-redundant controller configuration, the data will be restored to a single controller and that controller will be set to NOFAILOVER. You must use the SET FAILOVER COPY command to restore the configuration information to the other controller.
-

- When replacing both controllers, you can replace the first and restart it alone by holding in port button 6 and 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 process prevents conflicting data transfer.

Destroy/Nodestroy

Specify whether to destroy or retain (nodestroy) 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. See the REDUCE command in Chapter 5, "CLI Commands," for information on removing disk drives from a mirrorset.

The NODESTROY switch is not valid for RAIDsets and single-disk configurations.

Unit Switches

You can enable (either add or set) the Unit switches listed in Table 3-5 for the listed storagesets and devices. See Chapter 5, “CLI Commands,” for an explanation of the following Unit switches.

Table 3-5 UNIT Switches for Storagesets

Container Type	Switch																
	ACCESS_PATH	PARTITION=partition-number	MAXIMUM_CACHED_TRANSFER	PREFERRED_PATH	NOPREFERRED_PATH	ERROR_MODE=NORMAL	ERROR_MODE=FAILSAFE	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	✓		✓	✓			✓	✓	✓	✓	✓				✓	✓	
Passthrough	✓			✓											✓	✓	

NOTE: ERROR_MODE is only valid on mirrorsets that have been enabled for DT_SUPPORT.

Regardless of storageset type, the RUN and NORUN switches cannot be specified for partitioned units.

Chapter 4

Configuring Storagesets

This chapter provides information to help you configure storagesets for your subsystem.

Introduction

One method of configuring storagesets is manual configuration. This method allows you the most flexibility in defining and naming storagesets. See Chapter 5, “CLI Commands,” for complete information about the CLI commands shown in this chapter.

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:

```
ADD DISK  
RUN CONFIG
```

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 DISKnnnn 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
```

Configuring a Stripese

See “Using Striped Mirrorsets for Highest Performance and Availability,” page 3–18 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 syntax:

```
ADD STRIPESET stripeset-name DISKnnnn DISKnnnn
```

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

```
INITIALIZE stripeset-name switch
```

See “INITIALIZE,” page 5–62, for valid switches and values.

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, default values are applied.

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

See “ADD UNIT,” page 5–27, for valid switches and values.

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

```
SHOW stripeset-name
```

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

```
SHOW unit-number
```

4-4 Configuring Storagesets

Example

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

```
ADD STRIPESSET STRIPE1 DISK10000 DISK20000 DISK30000
INITIALIZE STRIPE1 CHUNKSIZE=128
ADD UNIT D100 STRIPE1 MAXIMUM_CACHED_TRANSFER=16
SHOW STRIPE1
SHOW D100
```

Configuring a Mirrorset

See “Creating a Storageset and Device Profile,” page 3–5, 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, default values are applied.

Use the following syntax to create a mirrorset:

```
ADD MIRRORSET mirrorset-name DISKnnnn DISKnnnn switch
```

See “ADD MIRRORSET,” page 5–11, for valid switches and values.

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

```
INITIALIZE mirrorset-name switch
```

See “INITIALIZE,” page 5–62, for valid switches and values.

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, default values are applied. Use the following syntax:

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

See “ADD UNIT,” page 5-27, for valid switches and values.

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

```
SHOW mirrorset-name
```

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

```
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
```

Configuring a RAIDset

See “Creating a Storageset and Device Profile,” page 3–5 for information about creating a profile and understanding the switches you can set for this kind of storage unit.

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, default values are applied.

Use the following syntax to create a RAIDset:

```
ADD RAIDSET RAIDset-name DISKnnnn DISKnnnn DISKnnnn switch
```

See “ADD RAIDSET,” page 5–19 for switches and values.

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

```
INITIALIZE RAIDset-name switch
```

See “INITIALIZE,” page 5–62, for valid switches and values.

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 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, default values are applied.

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

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

See “ADD UNIT,” page 5–27, for valid switches and values.

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

```
SHOW RAIDset-name
```

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

```
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 D300 RAID1
```

```
SHOW RAID1
```

```
SHOW D300
```

Configuring a Striped Mirrorset

See “Creating a Stageset and Device Profile,” page 3–5 for information about creating a profile and understanding the switches you can set for this kind of storage unit.

NOTE: A striped mirrorset cannot be created if the mirrorsets are enabled for DT_SUPPORT.

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 syntax:

```
ADD STRIPESET stripeset-name mirrorset_1 mirrorset_2
```

4-8 Configuring Storagesets

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

INITIALIZE stripeset-name switch

See “INITIALIZE,” page 5-62, for valid 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, default values are applied.

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

ADD UNIT unit-number stripeset-name switch

See “ADD UNIT,” page 5-27, for valid switches and values.

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

SHOW stripeset-name

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

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
```

Configuring a Single-Disk Unit

Use the following steps to use a single disk drive as a single-disk unit in your subsystem:

1. Add the disk drive using the following syntax:

```
ADD DISK DISKnnnn ptl-location switch_value
```

Optionally, you can append Device switch values. If you do not specify switch values, default values are applied. See “ADD DISK,” page 5–7, for valid switches and values.

2. Initialize the disk drive using the following syntax:

```
INITIALIZE DISKnnnn 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, default values are applied. Use the following syntax:

```
ADD UNIT unit-number DISKnnnn switch_value
```

4-10 Configuring Storagesets

See “ADD UNIT,” page 5-27, for valid switches and values.

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

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
```

Partitioning a Storageset or Disk Drive

See “Planning Partitions,” page 3-40, for details about partitioning a storage unit. Partitioning cannot be used in multiple bus failover mode.

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 syntax:

```
ADD storageset-name DISKnnnn DISKnnnn
```

or

```
ADD DISK DISKnnnn ptl-location
```

Do not split partitioned units across ports. They must be on a single port.

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

```
INITIALIZE storageset-name switch
```

- Create each partition in the storageset or disk drive by indicating the partition's size. Use the following syntax:

```
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.

- Verify the partitions using the following syntax:

```
SHOW storageset-name
```

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

- 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, default values are applied.

Use the following syntax to present partitions to the host:

```
ADD_UNIT unit-number storageset-name PARTITION=partition-number  
switch
```

See "ADD UNIT," page 5-27, for valid switches and values.

- Verify the unit numbers for the partitions using the following syntax:

```
SHOW storageset-name
```

4-12 Configuring Storagesets

Example

The following example shows the commands you would use to create RAID1, a three-member RAIDset partitioned 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	
2	1915 (0.98 MB)	1920	
3	1915 (0.98 MB)	3840	
4	2371 (1.21 MB)	5760	
.			
.			
.			

```

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
.	.	.	.
.	.	.	.
.	.	.	.

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. 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 syntax:

```
ADD SPARESET DISK nnnn
```

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

4-14 *Configuring Storagesets*

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

```
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 syntax:

```
SHOW SPARESET
```

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

```
DELETE SPARESET DISKnnnn
```

Verify the contents of the spareset using the following syntax:

```
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 syntax:

```
SET FAILEDSET AUTOSPARE
```

To disable autospare, use the following syntax:

```
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 syntax:

```
SHOW STORAGESETS
```

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

```
DELETE unit-number
```

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

```
DELETE storageset-name
```

4. Verify the configuration using the following syntax:

```
SHOW STORAGESETS
```

Example

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

```
SHOW STORAGESETS
```

Name	Storageset	Uses	Used by
STRIPE1	stripeset	DISK10000 DISK20000 DISK30000	D100

```
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 3-42, for an explanation of the switches. Remember to update the storageSet’s profile (hardcopy) when you change its switches.

Displaying the Current Switches

To display the current switches for a storageSet or a single disk, 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) acts as a type of access device in the form of a LUN for communicating between the host and the controller. You can access the CCL using either StorageWorks Command Console (SWCC) or CLI commands. When the CCL is enabled, you can communicate between the controller and the host without a 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 by running VTDPY, and running FMU for troubleshooting. You should not use the CCL for maintenance using the following utilities and exercisers: HSUTIL, FRUTIL, and DILX. For these tasks, you would need to communicate with the controller through the maintenance port cable.

Enabling and Disabling the CCL

IMPORTANT: You cannot disable the CCL in SCSI-3 mode. It is always enabled.

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

```
SET THIS_CONTROLLER COMMAND_CONSOLE_LUN
```

To turn it off, use the following command:



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.

```
SET THIS_CONTROLLER NOCOMMAND_CONSOLE_LUN
```

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-Port and Multiple-Host Use

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

SCSI-2 Mode

If you are in SCSI-2 mode and have enabled the `SET FAILOVER` command, only one CCL will be enabled. However, in SCSI-2 mode with `SET MULTIBUS_FAILOVER` enabled, all ports will see the CCL and will be able to access it. In addition, the CCL appears to the host as a direct access device.

SCSI-3 Mode

In SCSI-3 mode, a CCL will appear at LUN 0 of each target ID. If you are in transparent failover mode, each CCL will be accessible from the port that has the SCSI target ID enabled. Ports with multiple target IDs enabled will have multiple CCLs, and they will appear to the host as an array controller.

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



CAUTION: Selecting SCSI-3 mode enables multiple CCLs—one for each target at LUN 0. If the hosts access the CCL simultaneously, unpredictable consequences can occur. In cases where the CCL can be accessed through multiple paths and LUNs, systems administrators of each host must not attempt to access the CCL simultaneously.

Troubleshooting with the CCL

Troubleshooting and maintaining the controller should not be done using the CCL. 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 2–26.

Adding Storage Units with the CCL

If you have not configured any units and have not yet enabled the CCL, you must establish a local connection as described previously in “Establishing a Local Connection to the Controller,” page 2–26. This local connection through the maintenance port cable provides a means of enabling the CCL, so that you can communicate either through SWCC or through a PC or terminal.

To start configuring storage units, you must first set target ID numbers. See “Setting SCSI Target ID Numbers,” page 2–33.

SCSI-2 Mode

As you add storage units and set SCSI IDs 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 Target ID/LUN setting.

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

SCSI-3 Mode

The CCL always remains in a fixed location. It is always located at LUN 0 of every target ID. If you have the CCL enabled, the controller will let you configure a device above LUN 0 of each target ID.

Configuring Units with Multiple Hosts

The HSZ80 Array Controller allows equal but separate access to the targets and units through multiple hosts running on different operating systems, which is called *heterogeneous host support*. The array controller allows hosts equal access to their targets and units by designating the host mode using the `HOST_FUNCTION` switch. The controller prevents one host from accessing another host's units through the use of separate SCSI buses or through the `ENABLE_ACCESS_PATH` switch.

Host Modes

Each target may have only one host mode assigned to it, making it possible for multiple units to be accessible by one host. This process allows designated targets to talk to their host and vice versa. Host modes allow hosts that implement different SCSI-2 behavior to have equal access to their targets on a shared controller or controller pair.

All hosts are allowed equal access to the controller, but not all hosts implement the same SCSI-2 behavior. This process is achieved by setting a host mode for each target, except when you use the default setting. The default host mode is Host Mode A, which includes DIGITAL UNIX and Open VMS. You only set the host mode on targets if you are using a host operating system that is different than the default.

For targets being accessed by the default host, Host Mode A, you will not need to use the `SET controller HOST_FUNCTION` switch.

4-24 *Configuring Storaagesets*

The following modes can be assigned:

- A—DIGITAL UNIX[®], OpenVMS, and Hewlett-Packard[®] HP-UX (Default)
- B—IBM AIX[®]
- C—HSZ15
- D—Windows NT[™] Server
- E—Silicon Graphics Inc. IRIX
- F—Sun Solaris

For more information on the modes, see “SET controller,” page 5-91. To view the host mode setting on the controller, use the following syntax:

```
SHOW THIS_CONTROLLER FULL
```

or

```
SHOW OTHER_CONTROLLER FULL
```

To set the host mode, use one of the following syntax:

```
SET THIS_CONTROLLER HOST_FUNCTION = (ID, MODE)
```

or

```
SET OTHER_CONTROLLER HOST_FUNCTION = (ID, MODE)
```

Example

In Figure 4-1, two different hosts are connected on the same SCSI host bus going into Port 1 of a shared controller. Targets 0 and 1 (units D0-D2, D100-D102) are configured in DIGITAL Open VMS mode and targets 2 and 3 (D200-D202 and D300-D302) are configured in Windows NT Server mode. Because the default host mode includes DIGITAL Open VMS in Host Mode A, all of the targets are automatically assigned to Host Mode A.

NOTE: Figure 4-1 shows an example of a heterogeneous host configuration. For supported host operating systems, refer to the *HSZ80 Array Controller ACS Version 8.3 Release Notes*.

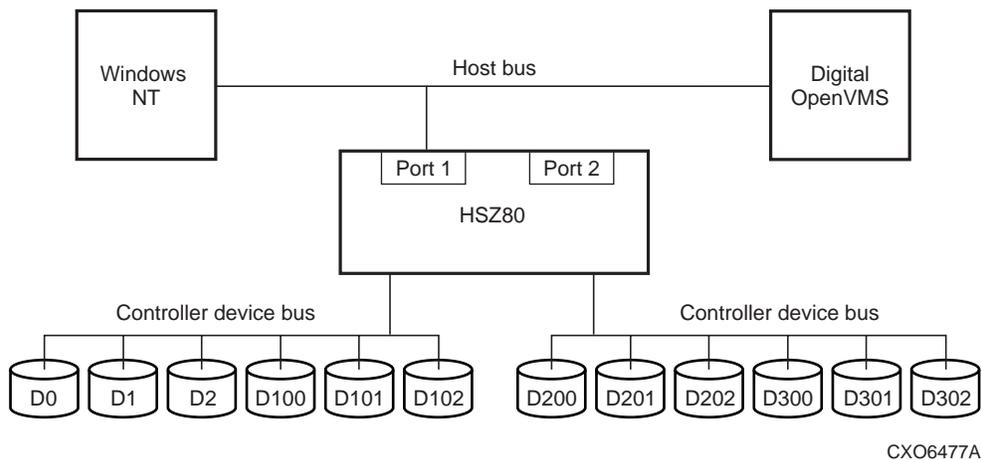


Figure 4-1. Setting Host Modes

To reassign targets 2 and 3 to the Windows NT Server mode, use the following syntax:

```
SET THIS_CONTROLLER HOST_FUNCTION = (2,D)
```

```
SET THIS_CONTROLLER HOST_FUNCTION = (3,D)
```

The controller will then use default SCSI-2 behavior when units D0–D2 and D100–D102 are accessed by a host. It will also use Windows-NT-compatible SCSI-2 behavior when units D200–D202 and D300–D302 are accessed by a Windows NT host.

Transparent Failover Mode

In transparent failover mode, units can be assigned to a host in a variety of setups, including these three common ways:

- by designating the port of the controller, either Port 1 or Port 2 or both.
- by designating the host SCSI ID number of the hosts configured on a shared bus.
- by designating the port of the controller and by designating the host SCSI ID number of the hosts configured on a shared bus.

Designating the Port

The HSZ80 Array Controller has two host ports where separate hosts can then be attached to the controller through separate buses. Units can be assigned to either port, depending on their target ID and by setting PORT_1_ID and PORT_2_ID.

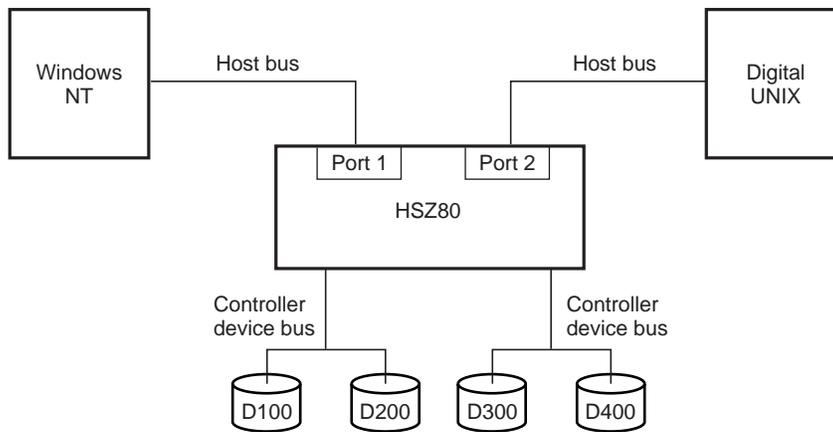
The configurations best suited to using the PORT_*n*_ID switch and separate SCSI buses include the following:

- Separate operating systems accessing units on the same pair of HSZ80 Array Controllers.
- The same operating systems (nonclustered, only) accessing units on the same pair of HSZ80 Array Controllers.
- Any environment in which multiple hosts access the pair of HSZ80 Array Controllers, but where a specific host must have sole access to units.

Example

In Figure 4-2, a Windows NT host is connected to Port 1 while a DIGITAL UNIX host is connected to Port 2. Targets 1 and 2 (D100 and D200) have been assigned to Port 1 and targets 3 and 4 (D300 and D400) have been assigned to Port 2.

NOTE: Figure 4-2 shows an example of a heterogeneous host configuration. For supported host operating systems, refer to the *HSZ80 Array Controller ACS Version 8.3 Release Notes*.



CXO6478A

Figure 4-2. Accessing Units on a Given Port

To assign the units to Port 1 and Port 2, use the following syntax:

```
SET THIS PORT_1_ID = (1,2)
```

```
SET THIS PORT_2_ID = (3,4)
```

To reassign target 1 and 2 to the Windows NT server host mode, use the following syntax:

```
SET THIS_CONTROLLER HOST_FUNCTION (1,D)
```

```
SET THIS_CONTROLLER HOST_FUNCTION (2,D)
```

Designating the Host SCSI ID Number

The HSZ80 Array Controller allows multiple hosts on the same SCSI host bus to access units individually and based on a host port's SCSI ID number. The `ENABLE_ACCESS_PATH` syntax implements this access, but restricts each host on the same bus from accessing other hosts' units. For each unit, you must use two commands:

```
SET UNIT unit number DISABLE_ACCESS_PATH = ALL
```

```
SET UNIT unit number ENABLE_ACCESS_PATH = SCSI ID
```

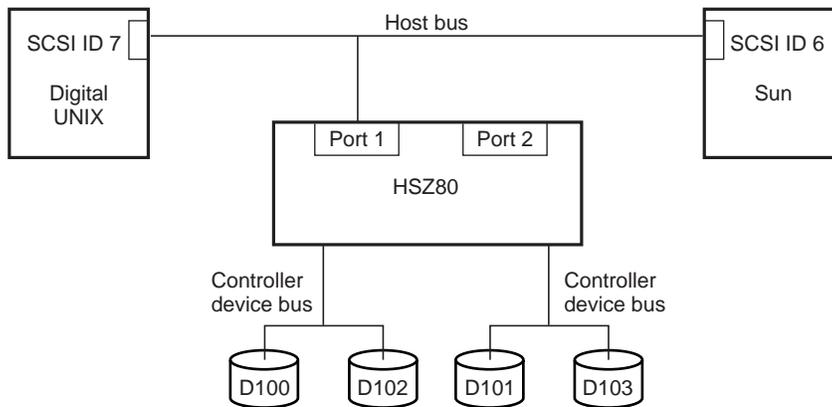
The `DISABLE_ACCESS_PATH` switch disables all hosts from accessing a unit. But, the `ENABLE_ACCESS_PATH` switch restricts access to a specific host.

NOTE: By default, units are enabled for all SCSI ID numbers. You must first disable the unit for all SCSI ID numbers, then followed by enabling the unit for a specific host SCSI ID number.

Example

In Figure 4-3, a DIGITAL UNIX host and a Sun host are connected to Port 1 on the same host bus. Units D100 and D101 have been assigned to SCSI ID 7, the DIGITAL UNIX host. Units D102 and 103 have been assigned to SCSI ID 6, the Windows NT host.

NOTE: Figure 4-3 shows an example of a heterogeneous host configuration. For supported host operating systems, refer to the *HSZ80 Array Controller ACS Version 8.3 Release Notes*.



CXO6479A

Figure 4-3. Accessing Units with the Host Port's SCSI ID Number

To assign the units to one host using SCSI ID numbers, yet restrict the other host's access, use the following syntax:

```
SET D100 DISABLE_ACCESS_PATH = ALL
```

```
SET D100 ENABLE_ACCESS_PATH = 7
```

```
SET D101 DISABLE_ACCESS_PATH = ALL
```

```
SET D101 ENABLE_ACCESS_PATH = 7
```

```
SET D102 DISABLE_ACCESS_PATH = ALL
```

```
SET D102 ENABLE_ACCESS_PATH = 6
```

```
SET D103 DISABLE_ACCESS_PATH = ALL
```

```
SET D103 ENABLE_ACCESS_PATH = 6
```

Designating the Port and Designating the Host SCSI ID Number

Using a two-step process, port access can be combined with shared host access on the same bus further restricting one host from accessing another host's units. You start by assigning units on a port using the SET THIS PORT_ *n* _ID command, followed by restricting access between hosts on the same host SCSI bus using the DISABLE_ACCESS_PATH and ENABLE_ACCESS_PATH switches.

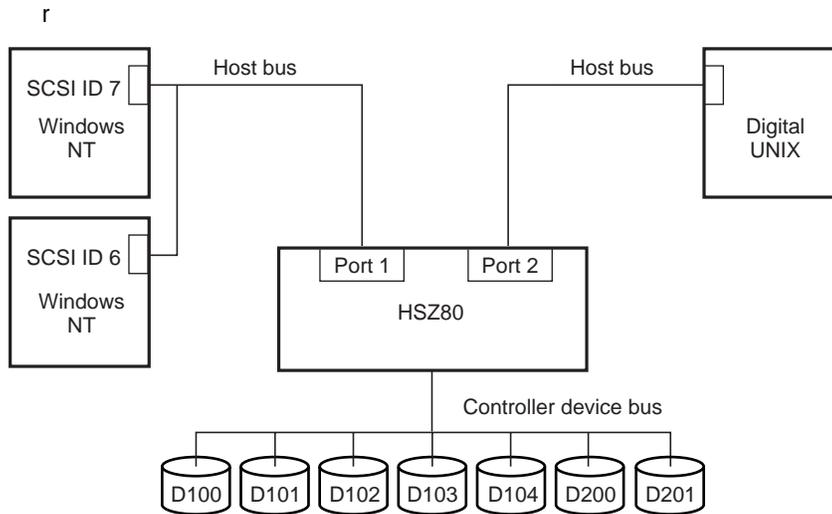
Example

In Figure 4-4, two Windows NT hosts are connected to Port 1 on the same host bus and a DIGITAL UNIX host is connected to Port 2 on a separate SCSI host bus. Because all targets default to Host Mode A—which includes DIGITAL UNIX—you only need to set targets to other host operating systems, such as Windows NT.

Units D100 and D101 have been assigned to Port 1 and SCSI ID 7, a Windows NT host. Units D102, D103, and D104 have been assigned to Port 1 and SCSI ID 6, another Windows NT host. Units D200 and D201 have been assigned to Port 2, the DIGITAL UNIX host.

NOTE: By default, units are enabled for all SCSI ID numbers. You must first disable the unit for all SCSI ID numbers, followed by enabling the unit for a specific host SCSI ID number.

NOTE: Figure 4-4 shows an example of a heterogeneous host configuration. For supported host operating systems, refer to the *HSZ80 Array Controller ACS Version 8.3 Release Notes*.



CXO6480A

Figure 4-4. Accessing Units through Port Access and the Host Port's SCSI ID Number

To assign the units to Port 1 and Port 2, use the following syntax:

```
SET THIS PORT_1_ID = 1
```

```
SET THIS PORT_2_ID = 2
```

To reassign target 1 to the Windows NT server host mode, use the following syntax:

```
SET THIS_CONTROLLER HOST_FUNCTION (1,D)
```

4-32 *Configuring Storagesets*

To assign the units to one of the Windows NT hosts using SCSI ID numbers, yet restrict the other Window NT host's access, use the following syntax:

```
SET D100 DISABLE_ACCESS_PATH = ALL
SET D100 ENABLE_ACCESS_PATH = 7
SET D101 DISABLE_ACCESS_PATH = ALL
SET D101 ENABLE_ACCESS_PATH = 7
SET D102 DISABLE_ACCESS_PATH = ALL
SET D102 ENABLE_ACCESS_PATH = 6
SET D103 DISABLE_ACCESS_PATH = ALL
SET D103 ENABLE_ACCESS_PATH = 6
SET D104 DISABLE_ACCESS_PATH = ALL
SET D104 ENABLE_ACCESS_PATH = 6
```

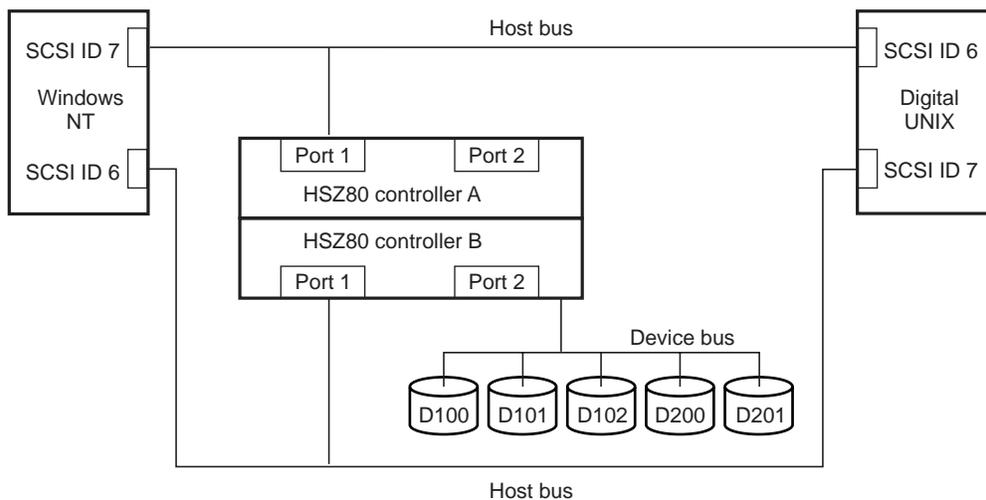
Multiple-Bus Failover Mode

Multiple-bus failover is a dual-redundant controller configuration where each controller in a controller pair has its own connection to the host. In such a dual-redundant configuration, units can be made available on both ports of the controller pair. Regardless of which controller a host is using, that host will always have access to its units while also being restricted from accessing other hosts's units. The command that limits host access is `THIS_PORT_n_ENABLE_ACCESS_PATH`.

Example

In Figure 4-5, a Windows NT host and a DIGITAL UNIX host are connected to a dual-redundant pair of HSZ80 Array Controllers in multiple-bus failover mode. Each host has two host port controllers. One host port is connected to controller A and the other host port is connected to controller B. All commands are entered from controller A.

NOTE: Figure 4-5 shows an example of a heterogeneous host configuration. For supported host operating systems, refer to the *HSZ80 Array Controller ACS Version 8.3 Release Notes*.



CXO6488A

Figure 4-5. Assigning Units through Multiple Host Adapters

To reassign target 1 to the Windows NT server host mode, use the following syntax:

```
SET THIS HOST_FUNCTION (1,D)
```

To assign the units to each host using SCSI ID numbers, yet restrict host access by the other hosts, use the following syntax:

```
SET D100 THIS_PORT_1_DISABLE_ACCESS_PATH = ALL
```

```
SET D100 OTHER_PORT_1_DISABLE_ACCESS_PATH = ALL
```

```
SET D100 THIS_PORT_1_ENABLE_ACCESS_PATH = 7
```

```
SET D100 OTHER_PORT_1_ENABLE_ACCESS_PATH = 6
```

4-34 *Configuring Storagesets*

```
SET D101 THIS_PORT_1_DISABLE_ACCESS_PATH = ALL
SET D101 OTHER_PORT_1_DISABLE_ACCESS_PATH = ALL
SET D101 THIS_PORT_1_ENABLE_ACCESS_PATH = 7
SET D101 OTHER_PORT_1_ENABLE_ACCESS_PATH = 6
```

```
SET D102 THIS_PORT_1_DISABLE_ACCESS_PATH = ALL
SET D102 OTHER_PORT_1_DISABLE_ACCESS_PATH = ALL
SET D102 THIS_PORT_1_ENABLE_ACCESS_PATH = 7
SET D102 OTHER_PORT_1_ENABLE_ACCESS_PATH = 6
```

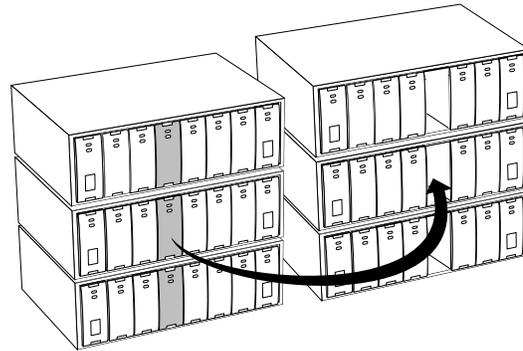
```
SET D200 THIS_PORT_1_DISABLE_ACCESS_PATH = ALL
SET D200 OTHER_PORT_1_DISABLE_ACCESS_PATH = ALL
SET D200 THIS_PORT_1_ENABLE_ACCESS_PATH = 6
SET D200 OTHER_PORT_1_ENABLE_ACCESS_PATH = 7
```

```
SET D201 THIS_PORT_1_DISABLE_ACCESS_PATH = ALL
SET D201 OTHER_PORT_1_DISABLE_ACCESS_PATH = ALL
SET D201 THIS_PORT_1_ENABLE_ACCESS_PATH = 6
SET D201 OTHER_PORT_1_ENABLE_ACCESS_PATH = 7
```

Moving Stagesets

You can move a stageset from one subsystem to another without destroying its data as shown in Figure 4-6. You also can follow the steps in this section to move a stageset to a new location within the same subsystem.

NOTE: You can use the procedure in this section to migrate wide devices from an HSZ70 controller in a BA370 rack-mountable enclosure to an HSZ80 environment. However, if you have an HSZ40 or HSZ50 subsystem, you cannot migrate to an HSZ80 in a BA370 rack-mountable enclosure. Refer to the *HSZ80 Array Controller ACS Version 8.3 Release Notes* for drives that can be supported.



CXO5595A

Figure 4-6. Moving a Stageset from one Subsystem to Another



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

4-36 Configuring Storagesets

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 syntax:
`SHOW storageset-name`
2. Physically 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 syntax:

```
LOCATE disk-name
```

To cancel the locate command, use the following syntax:

```
LOCATE CANCEL
```

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

```
DELETE unit-number
```

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

```
DELETE storageset-name
```

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

```
DELETE disk-name
```

```
DELETE disk-name
```

```
DELETE disk-name
```

6. Remove the disk drives and move them to their new PTL locations.
7. Add each disk drive to the controller’s list of valid devices. Use the following syntax:

```
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 syntax:

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, if it is not already in use, or you can create a new one. Use the following syntax:

ADD UNIT unit-number storageset-name

Example

The following example show the commands you would use to move unit D100 to another cabinet. D100 is the RAIDset RAID99 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

4-38 *Configuring Storagesets*

(...move 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 shows the commands you would use to move 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 5

CLI Commands

This appendix contains the Command Line Interpreter (CLI) commands you can use to interact with your controller. Each command description contains the full syntax and examples of the use of the command. The Overview provides a general description of the CLI and how to use it.

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 xviii, for an explanation of these terms.

Using the CLI

You can access the CLI by connecting a maintenance terminal to the maintenance port on the front of the controller. See “Establishing a Local Connection to the Controller,” page 2–26 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.
- Stageset Commands—Create and configure complex containers made from groups of device containers. There are four basic types of stagesets: stripesets, RAIDsets, striped-mirrorsets, and mirrorsets. Stageset 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 containers.
- Failover Commands—Configure the controllers to operate in transparent or multiple bus failover while also providing support for dual-redundant configurations.
- Diagnostic and Utility Commands—Perform general controller support functions

Getting Help

Help for 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 the HELP command on 5-60 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.

5-4 CLI Commands

- 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.

NOTE: Due to space limitations, some examples in this manual contain commands that exceed one line in length. In these instances, the paragraph preceding the example identifies which command is continued on the next line.

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 5-1 lists the keys used to recall and edit commands.

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	Toggles between insert and overstrike.
Ctrl/E	Moves the cursor to the end of the line.
Ctrl/H or Backspace	Moves the cursor to the beginning of the line.
Ctrl/J or Linefeed	Deletes the word to the left of the cursor.
Ctrl/U	Deletes all characters on the same line as the cursor.

Unit Naming

The host operating system identifies the location of units based on a logical unit numbering (LUN) scheme.

Each unit number contains the following:

- The first character is a letter indicating the kind of device in the storage unit: use D for disk devices or P to indicate passthrough devices such as tape devices, loaders, and libraries.
- If entered, the next one or two digits indicate which target ID number is to access the unit during normal operation. Use one of the controller's SCSI target ID numbers assigned to the controller with the SET *controller* ID command. Omit leading zeroes for SCSI target ID numbers less than ten. For example, use 2 instead of 02 for a storageset being accessed through the controller SCSI target ID number 0, LUN 2. The SCSI target ID number is initially set with the SET *controller* PORT_1_ID (or *PORT_2*) command. See 5-91 for an explanation of the SET *controller* commands.

By carefully choosing the SCSI target ID number when adding units, the preferred paths for all of your storage units in a dual-redundant configuration can be established. Data I/O load to the units can likewise be controlled by equally distributing unit access between the controllers.

NOTE: When operating in multiple bus failover mode, settings specified with the PORT_1_PREFERRED_ID (or PORT_2) switch takes priority over settings assigned with the SET *controller* PORT_1_ID (or PORT_2) switch.

- The digit following the SCSI target ID number is always zero.
- The last digit identifies the logical unit number (LUN) for the device or storage unit. Acceptable LUN numbers are 0 through 31, depending on your operating system.

Table 5-2 Unit Numbering Examples

Unit Number	Device Type	Target ID Number	LUN
D1207	disk	12	7
D401	disk	4	1
D39	(not an acceptable unit number)		
D5	disk	0	5

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 "HSZ>."

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 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 nine characters including letters A through Z, numbers 0 through 9, periods (.), dashes (-), or underscores (_).

IMPORTANT: It is common to name a disk drive DISK*ptll*, where *ptll* 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 the "Device PTL Addressing Convention within the Controller," page 3-36, for an explanation of the PTL addressing naming format.

NOTE: See the *HSZ80 Array Controller ACS Version 8.3 Release Notes* to determine whether the disk drive you are planning to use is compatible with the controller.

Switches

NOTTRANSPORTABLE (Default)

TRANSPORTABLE

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

If the NOTTRANSPORTABLE 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) 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 NOTTRANSPORTABLE 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 NOTTRANSPORTABLE, you should initialize the disk.

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

LOCAL (Default)

REMOTE

Specifies the disk characteristics in a mirrorset, either local or remote.

- Disks set to REMOTE can only be added to mirrorsets and JBOD (that is, not RAID5 raidsets).
 - Disks set to REMOTE can only be added to mirrorsets that are set to DT_SUPPORT.
 - Disks set to LOCAL can be added to mirrorsets that are set to either NODT_SUPPORT or DT_SUPPORT.
 - Disks set to REMOTE that are part of a mirrorset will not be the read source.
-

- Setting the switch to REMOTE forces a disk to be NOTTRANSPORTABLE.

```
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

This example shows how to add DISK10000 at port 1, target 0, LUN 0:

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

This example shows how to add DISK40200 as a transportable disk drive to port 4, target 2, LUN 0.

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

This example shows how to:

- Add a disk drive named DISK30200 as a non-transportable disk to port 3, target 2, LUN 0
- Set the data transfer rate to 10 MHz.

Enter the following command on one line.

```
ADD DISK DISK30200 3 2 0 NOTTRANSPORTABLE  
TRANSFER_RATE_REQUESTED=10MHZ
```

5-10 CLI Commands

This example creates a host-addressable unit after the disk is added:

```
INITIALIZE DISK20000 2 0 0
```

```
ADD UNIT D199 DISK20000
```

This example shows how to add DISK200 as a remote disk drive to port 2, target 0, LUN 0:

```
ADD DISK DISK200 2 0 0 REMOTE
```

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 nine 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 in an intuitive manner.

disk-name1 [*disk-nameN*]

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

Switches

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.

5-12 CLI Commands

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

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

DT_SUPPORT
NODT_SUPPORT

Specifies if disaster tolerant functionality is enabled on this mirrorset. If some members are or may be remote and disaster tolerant functionality is supported, set `DT_SUPPORT`. If all members are—and intend to remain—local and disaster tolerant functionality is not supported, set `NODT_SUPPORT`.

If `DT_SUPPORT` is enabled, the mirrorset has the following restrictions: `NOPOLICY` is enforced and `NOWRITEBACK_CACHE` is enforced.

If `NODT_SUPPORT` is enabled and any of the members have `REMOTE` set, then the command will fail with an error.

NOTE: This command will fail if you change the setting while the mirrorset is in use by a higher level.

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. If `DT_SUPPORT` is enabled, no policy can be selected.

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. This switch is not valid if `DT_SUPPORT` is enabled.

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. This switch is not valid if `DT_SUPPORT` is enabled.

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 5-112).

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. A member with `REMOTE` set may be selected.

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 local mirrorset disk with the least amount of work in its 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 the remote disks.

5-14 CLI Commands

Specify `READ_SOURCE=ROUND_ROBIN` to sequentially direct read requests to each local mirrorset disk. The controller equally queries all normal local disks for each read request. If no normal local disk exists, then the controller will query the remote disks.

Examples

This example shows how to add `DISK10000`, `DISK20100`, and `DISK30200` as a mirrorset with the name `MIRR1`:

```
ADD DISK DISK10000 1 0 0  
ADD DISK DISK20100 2 1 0  
ADD DISK DISK30200 3 2 0  
ADD MIRRORSET MIRR1 DISK10000 DISK20100 DISK30200
```

This example creates 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 PASSTHROUGH

Creates a passthrough container. A passthrough container bypasses controller processing and allows the host to have direct access to the device. A passthrough device appears as if it were connected directly to the host bus rather than to the controller. Use passthrough containers to communicate with tape devices and tape loaders because they do not require the controller for optimum operation performance.

Passthrough units can be created for passthrough Command Console LUNs to allow SWCC to configure and monitor the controllers and devices on the backend in a remote mirroring implementation. The passthrough devices are automatically configured by the controller if they exist. The unit above the passthrough is not automatically configured by the controller.

Syntax

ADD PASSTHROUGH *passthrough-name scsi-port-target-lun*

Parameter

Names the device being added to or changed in a subsystem configuration. This is the same name used with the ADD UNIT command to create a host-addressable unit. You may want to use a name indicating the type of device is being added as a passthrough container, such as TAPE or LOADER.

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

IMPORTANT: It is common to name a tape passthrough device TAPE*ptll*, where *ptll* is the tape's Port-Target-LUN address. Although other naming conventions are acceptable, this naming convention offers the advantage of presenting to the user both the type of device and its SCSI location in an intuitive manner.

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 dress. See the “Device PTL Addressing Convention within the Controller,” page 3–36, for an explanation of the PTL addressing naming format.

NOTE: See the *HSZ80 Array Controller ACS Version 8.3 Release Notes* to determine whether the disk drive you are planning to use is compatible with the controller.

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. For example, the transfer rate may need to be limited to accommodate long cables between the controller and the device, such as with a tape library. The transfer rate is negotiated immediately and does not require a restart in order to take effect.

Example

This example shows how to add tape device TAPE20300 at port 2, target 3, LUN 0:

```
ADD PASSTHROUGH TAPE20300 2 3 0  
ADD UNIT P100 TAPE20300
```

5-18 *CLI Commands*

See also

ADD UNIT

DELETE *container-name*

SHOW DEVICES

SHOW PASSTHROUGH

SHOW *passthrough-name*



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-name3
[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 nine 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-name3 [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 a disk drive that equals or exceeds the base member size of the RAIDset.

Specify *NOPOLICY* to prevent the controller from automatically replacing a failed disk device. This RAIDset operates in a reduced state until you select either *POLICY=BEST_PERFORMANCE* or *POLICY=BEST_FIT*, or manually place a member in the RAIDset. See “SET RAIDset-name,” page 5-124, 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: 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

This example shows how to create a RAIDset named RAID9 that contains disks DISK10000, DISK20100, and DISK30200.

```
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 POLICY=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

INITIALIZE

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. Disks set to REMOTE cannot be added to the spareset.

Example

This example shows how to add a disk drive named DISK20200 and DISK30300 to a spareset:

```
ADD DISK DISK20200 2 2 0
ADD DISK DISK30300 3 3 0
ADD SPARESET DISK20200
ADD SPARESET DISK30300
```

See also

DELETE SPARESET

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. Stripesets can contain between 2 and 24 members.

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

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

NOTE: A 240-character limit exists 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

This example shows how to create a stripeset named STRIPE1 with three disks: DISK10000, DISK20100, and DISK30200:

```
ADD DISK DISK10000 1 0 0  
ADD DISK DISK20100 2 1 0  
ADD DISK DISK30200 3 2 0  
ADD STRIPESET STRIPE1 DISK10000 DISK20100 DISK30200
```

This example show how to create a stripeset named STRIPE1 and then create a logical unit from it:

```
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 STRIPESSET

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 a logical unit.

Syntax

ADD UNIT *unit-number container-name*

Parameters

unit-number

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.

Containers must have units that are on a single port. Do not split partitioned units across ports.

container-name

Specifies the name of the container (disk drive, device, storageset, or partition) being used to create the unit.

A maximum of 48 devices can make up one unit.

Switches

Table 5-3 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 5-3 ADD UNIT Switches for Storagesets

Container Type	Switch													
	ACCESS_PATH	PARTITION=partition-number	MAXIMUM_CACHED_TRANSFER	PREFERRED_PATH	NONPREFERRED_PATH	READ_CACHE	NOREAD_CACHE	READAHEAD_CACHE	NOREADAHEAD_CACHE	WRITE_PROTECT	NOWRITE_PROTECT	WRITEBACK_CACHE	NOWRITEBACK_CACHE	RUN
RAIDset	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Stripeset	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Mirrorset	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NoTransportable Disk	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Transportable Disk	✓		✓	✓	✓	✓	✓	✓	✓				✓	
Passthrough	✓			✓									✓	

NOTE: Regardless of the storageset type, you cannot specify RUN and NORUN for partitioned units.

ENABLE_ACCESS_PATH=
DISABLE_ACCESS_PATH=
THIS_PORT_1_ENABLE_ACCESS_PATH=
THIS_PORT_1_DISABLE_ACCESS_PATH=
THIS_PORT_2_ENABLE_ACCESS_PATH=
THIS_PORT_2_DISABLE_ACCESS_PATH=
OTHER_PORT_1_ENABLE_ACCESS_PATH=
OTHER_PORT_1_DISABLE_ACCESS_PATH=
OTHER_PORT_2_ENABLE_ACCESS_PATH=
OTHER_PORT_2_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.

ENABLE_ACCESS_PATH and DISABLE_ACCESS_PATH switches are available only when in transparent mode; the other switches are available only when in multiple bus failover mode.

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 or THIS(OTHER)_PORT_1(2)_DISABLE_ACCESS_PATH=ALL), then enable the desired IDs. The system will display the following message (transparent failover mode, typical):

```
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.
```

5-30 CLI 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. Partitioned units must be on a single port.

MAXIMUM_CACHED_TRANSFER=32 (Default)
MAXIMUM_CACHED_TRANSFER=n

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

PREFERRED_PATH=OTHER_CONTROLLER
PREFERRED_PATH=THIS_CONTROLLER
NOPREFERRED_PATH (Default)

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_ID* (or *PORT_2*) command determines which target ID number the controller uses to respond to the host.

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.

See Chapter 2 for information regarding multiple bus failover.

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 read-cache 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 (Default)

NOWRITE_PROTECT

Tells the controller whether data contained on the unit can be overwritten.

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

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

WRITEBACK_CACHE (Default)

NOWRITEBACK_CACHE

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 the controller's write-back caching feature.

`WRITEBACK_CACHE` cannot be enabled on units with an underlying `DT_SUPPORT` mirrorset container.

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.

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
```

5-34 *CLI Commands*

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

This example shows how to clear the message “All NVPM components initialized to their default settings” from the CLI prompt:

All NVPM components initialized to their default settings
CLEAR_ERRORS CLI

See also

CLEAR_ERRORS *controller* INVALID_CACHE

CLEAR_ERRORS *unit-number* LOST_DATA

CLEAR_ERRORS *device-name* UNKNOWN

CLEAR_ERRORS *unit-number* UNWRITEABLE_DATA

CLEAR_ERRORS *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 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 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 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

This example shows how to force the controller to recognize a previously unknown device named DISK30000:

```
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 and 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

This example shows how to 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

This example shows how 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 5-62.

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 "CREATE_PARTITION," page 5-48, 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 "Configuring an HSZ80 Array Controller," page 2-3).

NOTE: The INITIALIZE *container-name* SAVE_CONFIGURATION must be used to save the controller's configuration to a disk (see "SAVE_CONFIGURATION," page 5-64), in order to reset the configuration (see "CREATE_PARTITION," page 5-48) or to restore the configuration (see "CREATE_PARTITION," page 5-48).

Syntax

```
CONFIGURATION RESTORE
```

5-46 *CLI Commands*

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 STRIPESSET, 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 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 storage set.
- 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 to 16,777,215; `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

This 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_PARTITIOIN 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 *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 member of a RAIDset, or a RAIDset unit; you must first delete the higher-level container or containers.

NOTE: This command does not delete spareset 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

This example shows how to delete a disk drive named DISK10000:

DELETE DISK10000

This example shows how to delete a stripeset named STRIPE1:

DELETE STRIPE1

5-52 *CLI Commands*

This example shows how to delete a RAIDset named RAID9:

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 members from the storage shelf for testing, repair, or replacement.

You should consider defective all disk drives in the failedset. 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

This example shows how to delete DISK20200 from the failedset:

```
DELETE FAILEDSET DISK20200
```

See also

```
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 example shows how to remove DISK20300 from the spareset:

```
DELETE SPARESET DISK20300
```

See also

```
ADD SPARESET
```

```
SHOW SPARESET
```

DELETE *unit-number*

Deletes a logical unit from the controller configuration. The host cannot address deleted units. If the controller'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

This example shows how to delete disk unit number D103:

```
DELETE D103
```

See also

ADD UNIT

CLEAR_ERRORS LOST_DATA

CLEAR_ERRORS UNWRITEABLE_DATA

DELETE *container-name*

DELETE FAILEDSETS

DELETE SPARESETS

DESTROY_PARTITION



DESTROY_PARTITION

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 STRIPESSET, 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

This 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

CREATE_PARTITION

DELETE *unit-number*

SHOW

DIRECTORY

Lists the diagnostics and utilities available on “this controller.”

Syntax

DIRECTORY

Example

This example shows how to display a directory listing:

DIRECTORY

```
HSUTIL V83Z D
FRUTIL V83Z D
CHVSN V83Z D
CLCP V83Z D
CLONE V83Z D
CONFIG V83Z D
DILX V83Z D
DIRECT V83Z D
DSTAT V83Z D
FMU V83Z D
VTDPY V83Z D
```

NOTE: CHVSN and DSTAT are not user utilities. They may 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

This example shows how to display information regarding the HELP command:

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

This example shows how to get help on the SET command using the question mark (?):

SET ?

Your options are:

```
EMU
FAILEDSET
FAILOVER
NOFAILOVER
MULTIBUS_FAILOVER
NOMULTIBUS_FAILOVER
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 nottransportable disks.

Use the INITIALIZE command when:

- Creating a unit from a newly installed disk.
- Creating a unit from a newly created RAIDset, stripeset, or mirrorset.
- Initializing the data structure of a previously partitioned container.

Do not use the INITIALIZE command when:

- Creating a unit from the same disks previously initialized, such as when a RAIDset is moved.
 - Creating a storage set 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 to 16,777,215; *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.

5-64 CLI Commands

The default chunk size for storagesets with less than nine members is 256 blocks, or 128 kilobytes (K). The default chunk size for storagesets with more than nine members is 128 blocks, or 64K. 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 the “Chunk Size,” page 3-51, 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 stripesets and 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 disk, stripeset, or 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 3-25, for more information regarding SAVE_CONFIGURATION.

Examples

This example shows how to initialize container DISK10000 and how to save a copy of the controller configuration on it.

```
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
DISK10000disk		1	0	0	S2
	DEC RZ28M (C)	DEC 1003			
	Switches:				
	NOTTRANSPORTABLE				

5-66 CLI Commands

```
TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 10.00 MHZ negotiated)
LOCAL
Size: 4108970 blocks
Configuration being backed up on this container
DISK30300    disk          3    3    0    S2
DEC    RZ28M    (C) DEC 1003
Switches:
NOTTRANSPORTABLE
TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 10.00 MHZ negotiated)
LOCAL
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 STRIPESSET 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 amber 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.

NOTE: LOCATE issued on a frontend controller will not locate devices hanging off a backend controller.

Syntax

LOCATE *parameter*

Parameters

Only one of the following parameters may be entered with each LOCATE command.

ALL

Causes the green 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 green device fault LEDs turned on with the LOCATE command.

DISKS

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

PTL (SCSI-location)

Causes the green device fault LED on the device at the given SCSI location to flash. See “Device PTL Addressing Convention within the Controller,” page 3–36, 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 green 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 amber 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 green 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 amber 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. 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.

NOTE: This command is also valid for a REMOTE disk, resulting in a single member mirrorset with DT_SUPPORT enabled.

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 MIRRORn, 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.

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

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

Sets the selection criteria the controller uses to choose a replacement disk from the spareset when a mirrorset member fails. If *DT_SUPPORT* is enabled, no policy can be selected.

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. This switch is not valid if *DT_SUPPORT* is enabled.

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. This switch is not valid if `DT_SUPPORT` is enabled.

Specify `NOPOLICY` to prevent the controller from automatically replacing a failed disk device. This causes the mirrorset to operate in a reduced state until either `POLICY=BEST_PERFORMANCE` or `POLICY=BEST_FIT` is selected, or a member is manually placed in the mirrorset (see “SET mirrorset-name,” page 5-112).

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 add a second disk to each mirrorset. It is not necessary to initialize the mirrorsets or add them as units; the higher-level structure of the stripeset is carried down to the mirrorsets.

```
ADD 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 in all cabinets associated with the controller.

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 powering the cache module after the cabinet is powered off.

Specify *BATTERY_OFF* to disable the cache batteries after the cabinet is powered off.

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, cabinet power is turned off.

Table 5-4 shows what action will be taken depending on the switch settings and the results of the attempted flush:

CAUTION: If the BATTERY_OFF switch is used with the OVERRIDE_BAD_FLUSH switch, unwritten data could be lost. Do not power off the subsystem until the unwritten data has been recovered.

Table 5-4 Poweroff Switch Settings

Battery Switch	Override Switch	Flush Results	Action
BATTERY_ON	OVERRIDE_BAD_FLUSH	Success	Controller and units in the cabinet are shutdown and the ECB is powering the cache module.
BATTERY_ON	OVERRIDE_BAD_FLUSH	Failure	Controller and units in the cabinet are shutdown and the ECB is powering the cache module.
BATTERY_ON	NO_OVERRIDE_BAD_FLUSH	Success	Controller and units in the cabinet are shutdown and the ECB is powering the cache module.
BATTERY_ON	NO_OVERRIDE_BAD_FLUSH	Failure	Nothing is shutdown, the ECB is powering the cache module, and the user is notified of a bad flush.

Table 5-4 Poweroff Switch Settings (Continued)

Battery Switch	Override Switch	Flush Results	Action
BATTERY_OFF	OVERRIDE_BAD_FLUSH	Success	Controller and units in the cabinet are shutdown and the ECB is not powering the cache module.
BATTERY_OFF	OVERRIDE_BAD_FLUSH	Failure	Controller and units in the cabinet are shutdown and the ECB is not powering the cache module.
BATTERY_OFF	NO_OVERRIDE_BAD_FLUSH	Success	Controller and units in the cabinet are shutdown and the ECB is not powering the cache module.
BATTERY_OFF	NO_OVERRIDE_BAD_FLUSH	Failure	Nothing is shutdown, the ECB is powering the cache module, and user is notified of a bad flush.

In dual-redundant controller configurations, if both controllers can't be shutdown, then both controllers and their ECB chargers remain on.

Example

This example shows how to power off the disk units and the cabinet in 10 seconds:

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.

NOTE: An error is displayed if you attempt to reduce the only remote or local normal member from the mirrorset if `ERROR_MODE= FAILSAFE` is enabled on the unit.

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 any 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 MIRR2 MIRR3	D104

SHOW MIRRORSETS

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

REDUCE DISK20100 DISK20200 DISK40200**SHOW MIRRORSETS**

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

See also

ADD MIRRORSET

MIRROR

RUN CLONE

SHOW MIRRORSET

SET *mirrorset-name*



RENAME

Renames a container.

Syntax

RENAME *old-container-name new-container-name*

Parameters

old-container-name

Specifies the existing name of the container.

new-container-name

Assigns the new name for the container.

See “Changing the CLI Prompt,” page 5–6, for information regarding container naming rules.

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

5-82 *CLI Commands*

RENAME DISK10000 MYDISK
SHOW DISKS

Name	Type	Port	Targ	Lun	Used by
MYDISK	disk	1	0	0	D100
DISK10100	disk	1	1	0	D101

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 when to shutdown.



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 1-22 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 *unit-number* 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 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 Upgrading the Subsystem in the 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 3-20, 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 4-2, 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 option.
 - 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 *HSZ80 Array Controller ACS Version 8.3 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 1-22, 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 1-22, 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 controller

Changes parameters on the specified controller.

Syntax

SET *controller*

Parameter

controller

Indicates which controller is to be set. Specify OTHER_CONTROLLER or THIS_CONTROLLER.

Switches

Table 5-5 lists the switches available with this command. Descriptions of the switches follow the table.

Table 5-5 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
HOST_FUNCTION	A, B, C, D, E, F
NODE_ID	assigned during manufacturing

Table 5-5 SET controller Switches (Continued)

Switch	Values
PORT_1_ID	
NOPORT_1_ID	
PORT_2_ID	
NOPORT_2_ID	
MIRRORED_CACHE	None
NOMIRRORED_CACHE	
PORT_1_PREFERRED_ID	
NOPORT_1_PREFERRED_ID	
PORT_2_PREFERRED_ID	
NOPORT_2_PREFERRED_ID	
PORT_1_TRANSFER_RATE_ REQUESTED	5MHz
NOPORT_1_TRANSFER_RATE_ REQUESTED	10MHz
PORT_2_TRANSFER_RATE_ REQUESTED	20MHz (default)
NOPORT_2_TRANSFER_RATE_ REQUESTED	
PROMPT	1-16 characters
SCSI_VERSION	SCSI-2 (default) SCSI-3
TERMINAL_PARITY	odd, even
NOTERMINAL_PARITY	
TERMINAL_SPEED	4800, 9600, 19200
TIME	dd-mm-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 1–4294967295. 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 on a unit may elapse before the write-back cache flushes data for that unit. 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). 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)

Enable 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.

HOST_FUNCTION=mode

HOST_FUNCTION=(ID,mode)

Configures the host compatibility mode for a controller target ID number. When changed, you must restart both controllers in order for the new setting to take effect.

Select the *HOST_FUNCTION=mode* switch to set all of the targets on the host interface to the same mode. The default mode is A.

Select the *HOST_FUNCTION=(ID,mode)* switch to configure the host port on a target-by-target basis for compatibility with various operating systems.

The `HOST_FUNCTION` switch assigns only one host function to the specified target ID number with each `SET` command. You must enter separate `SET` commands to set multiple modes to different target ID number—for example, to support hosts with different operating systems on a single SCSI bus.

Each controller or pair of controllers can service a maximum of two hosts simultaneously as long as they are assigned to different targets. If the `HOST_FUNCTION` switch is changed on a single controller in a dual-redundant configuration, corresponding mode changes are automatically passed on to the other controller. The controller does not have to restart for these changes to take effect.

The following modes can be assigned:

- A—DIGITAL UNIX[®], OpenVMS, and Hewlett-Packard[®] HP-UX (Default)
- B—IBM AIX[®]
- C—HSZ15
- D—Windows NT[™] Server
- E—Silicon Graphics Inc. IRIX
- F—Sun Solaris

Enter the `SHOW controller` command to display the current Host Function Mode settings.

NODE_ID=nnnn-nnnn-nnnn-nnnn checksum

Sets the subsystem worldwide ID (node ID). If a situation occurs that requires you to reset the subsystem worldwide ID (node ID), use the ID and check sum that appears on the sticker on your enclosure.



CAUTION: Each subsystem has its own unique node ID. If you attempt to set the subsystem node ID to an ID 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 node ID; data corruption will occur.

PORT_1_ID=(n1,n2,n3...)

NOPORT_1_ID

PORT_2_ID=(n1,n2,n3...)

NOPORT_2_ID

Assigns one to fifteen SCSI target ID numbers (0 through 15) to either port 1 or to port 2. A total of 15 target ID numbers can be assigned to both ports. For example, port 1 has eight target ID numbers and port 2 has seven target ID numbers for a total of 15. Another example, port 1 has 15 target ID numbers and port 2 has none. Hosts on port 1 can only see targets assigned to port 1 and hosts on port 2 can only see targets assigned to port 2. If two or more target ID numbers are entered, the numbers must be enclosed in parentheses and separated with commas. When changed, you must restart both controllers in order for the new setting to take effect.

A controller can present up to 32 units to the host on each target ID number. Unit numbers identify both the target ID number and the LUN. However, access to LUNs 8 through 31 requires support by both the host adapter and operating system. For example, D203 indicates a disk is addressed through target 2, LUN 3. D5 is target 0, LUN 5.

If two controllers are in a dual-redundant configuration, both controllers support the same target ID numbers. When the ID numbers are changed on one controller, the other controller is automatically updated.

NOTE: Use the *PORT_1_PREFERRED_ID* (or *PORT_2*) switch along with the *PORT_1_ID* (or *PORT_2*) switch to specify which targets are assigned to each of your controllers on port 1 (or 2). If you do not specify the preference of your targets as you add them, the controller automatically prefers all of the targets to the “other controller.”

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.

PORT_1_PREFERRED_ID=(n1,n2,n3...)

NOPORT_1_PREFERRED_ID

PORT_2_PREFERRED_ID=(n1,n2,n3...)

NOPORT_2_PREFERRED_ID

May be set only when dual-redundant configurations are operating in a transparent failover configuration. It specifies which targets and units are handled by the specified controller port. First, specify, the total set of ID numbers with the preferred ID number switch. If you enter two or more *PORT_1_PREFERRED_ID* (or *PORT_2*) numbers, the numbers must be enclosed within parentheses and separated with commas. When changed, the new setting for this switch takes effect immediately.

NOTE: Only target ID numbers set with the *PORT_1_ID* (or *PORT_2*) switch can be preferred with the *PORT_1_PREFERRED_ID* (or *PORT_2*) switch.

The other controller is automatically updated to support the remaining ID numbers, if any. For example, if a dual-redundant pair of controllers is set to ID numbers 0, 1, and 2, one controller can be set to respond to ID numbers 0 and 1; the other controller automatically responds to ID number 2. The first controller presents unit numbers 0 through 7 and 100 through 107. The other controller presents unit numbers 200 through 207 to the controllers.

Specify *NOPORT_1_PREFERRED_ID* (or *NOPORT_2*) for one controller so it is immediately available if the other controller fails. The other controller processes all of the data during normal operation. However, if the other controller fails, the controller assigned the *NOPORT_1_PREFERRED_ID* (or *NOPORT_2*) switch assumes control of the targets assigned to the failed controller.

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.

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. 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 LLC 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-mm-yyy:hh:mm:ss

Sets the date and time using a 24-hour clock. The time is set on both controllers in a dual-redundant configuration. When changed, the new value entered for this switch takes effect immediately.

```

PORT_1_TRANSFER_RATE_REQUESTED=20MHZ
PORT_1_TRANSFER_RATE_REQUESTED=10MHZ
PORT_1_TRANSFER_RATE_REQUESTED=5MHZ
PORT_1_TRANSFER_RATE_REQUESTED=DEFAULT
NOPORT_1_TRANSFER_RATE_REQUESTED
PORT_2_TRANSFER_RATE_REQUESTED=20MHZ
PORT_2_TRANSFER_RATE_REQUESTED=10MHZ
PORT_2_TRANSFER_RATE_REQUESTED=5MHZ
PORT_2_TRANSFER_RATE_REQUESTED=DEFAULT
NOPORT_2_TRANSFER_RATE_REQUESTED

```

Specifies the maximum data transfer rate the controllers use when communicating with the host. The transfer rate may need to be limited to accommodate long cables between the subsystem and the host system. Set the same transfer rate for both controllers in a dual-redundant configuration. When changed in a dual-redundant configuration, you must restart both controllers in order for the new setting to take effect. When changed in a single configuration, restart only the controller on which the new setting will take effect.

The controller must be restarted for any changes to the data transfer rate to take effect.

Table 5-6 lists the maximum transfer rates for different lengths of fast and slow SCSI buses. These lengths represent cable lengths plus shelf-bus lengths.

Transfer Rate	Meter	Feet
5 MHz	6	19.7
10 MHz	3	9.8
20 MHz	25	82

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 "this controller" on host port 1, so it responds to requests for target ID numbers 2 and 5:

```
SET THIS_CONTROLLER PORT_1_ID=(2,5)
```

See also

```
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

LOCAL (Default)

REMOTE

Specifies the disk characteristics in a mirrorset, either local or remote.

- Disks set to REMOTE can only be added to mirrorsets and JBOD (that is, not RAID5 raidsets).
 - Disks set to REMOTE can only be added to mirrorsets that are set to DT_SUPPORT.
 - Disks set to LOCAL can be added to mirrorsets that are set to either NODT_SUPPORT or DT_SUPPORT.
 - Disks set to REMOTE that are part of a mirrorset will not be the read source.
 - Setting the switch to REMOTE forces a disk to be NOTTRANSPORTABLE.
-

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
NOTTRANSPORTABLE (Default)

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

NOTE: Setting the switch to *TRANSPORTABLE* forces a disk to be *LOCAL*.

Set the *TRANSPORTABLE* switch for disks only.

StorageSets cannot be made transportable. Specify *NOTTRANSPORTABLE* 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 *NOTTRANSPORTABLE* 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 *NOTTRANSPORTABLE*, you should initialize the disk.

If you specify *NOTTRANSPORTABLE* and there is no metadata on the unit, the unit must be initialized.

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
```

This example shows how to set DISK10300 to remote:

```
SET DISK10300 REMOTE
```

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 5-7 lists the valid EMU set-point temperatures in both Fahrenheit and Celsius.

Table 5-7 EMU Set Point Temperatures

°C	°F	°C	°F	°C	°F	°C	°F	°C	°F
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. Disks set to *REMOTE* cannot be autospared.

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

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

Disks that you plan to use for AUTOSPARE must not have valid metadata on them. If you suspect a disk does have metadata on it (it was used in a stripeset or was initialized as NOTTRANSPORTABLE) 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. Remove DISK10000 from the cabinet until it is time to replace a failed disk.

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

DELETE FAILEDSET

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.

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.

DT_SUPPORT
NODT_SUPPORT

Specifies if disaster tolerant functionality is enabled on this mirrorset. If some members are or may be remote and disaster tolerant functionality is supported, set *DT_SUPPORT*. If all members are—and intend to remain—local and disaster tolerant functionality is not supported, set *NODT_SUPPORT*.

If *DT_SUPPORT* is enabled, the mirrorset has the following restrictions: *NOPOLICY* is enforced and *NOWRITEBACK_CACHE* is enforced.

If *NODT_SUPPORT* is enabled and any of the members have *REMOTE* set, then the command will fail with an error.

NOTE: This command will fail if changing the setting while the mirrorset is in use by a higher level.

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—it will be specified if *DT_SUPPORT* is enabled—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.

NOTE: An error is displayed if you attempt to remove the only remote or local normal member from the mirrorset if ERROR_MODE= FAILSAFE is enabled on the unit.

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

NOTE: An error is displayed if you attempt to replace a REMOTE disk into a NODT_SUPPORT mirrorset.

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. If DT_SUPPORT is enabled, NOPOLICY must be selected.

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. This switch is not valid if DT_SUPPORT is enabled.

Specify POLICY=BEST_PERFORMANCE to choose a replacement disk drive from the spareset with the best performance. The controller attempts to select a disk on a different port than existing mirrorset members. If there is more than one disk drive in the spareset matching the best performance criteria, the controller selects a disk drive that equals or exceeds the base member size of the mirrorset. This switch is not valid if DT_SUPPORT is enabled.

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. A member with `REMOTE` set may be selected.

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. If no normal local disk exists, then the controller will query the remote disks.

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. If no normal local disk exists, then the controller will query the remote disks.

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_FAILOVER COPY=controller
```

Parameters

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 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 PORT_1_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_PATH=THIS_CONTROLLER  
SET D101 PREFERRED_PATH=OTHER_CONTROLLER
```

See also

SET FAILOVER

SET NOFAILOVER

SET NOMULTIBUS_FAILOVER

SET NOFAILOVER

Reconfigures both controllers to operate in a nondual-redundant (nonfailover) 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 CLI command, `CLEAR_ERRORS controller INVALID_CACHE`, 5-36.

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 MULTIBUS_FAILOVER
```

```
SET NOMULTIBUS_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 on 5-124 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 *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 5-8 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 5-8 SET UNIT Switches for Existing Containers

Container Type	Switch																
	ACCESS_PATH	PARTITION=partition-number	MAXIMUM_CACHED_ TBANSFER	PREFERRED_PATH	NO PREFERRED_PATH	ERROR_MODE=NORMAL	ERROR_MODE=FAILSAFE	READ_CACHE	NO READ_CACHE	READAHEAD_CACHE	NO READAHEAD_CACHE	WRITE_PROTECT	NO WRITE_PROTECT	WRITEBACK_CACHE	NO WRITEBACK_CACHE	RUN	NORUN
RAIDset	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Stripeset	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Mirrorset	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
NoTransportable Disk	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Transportable Disk	✓		✓	✓			✓	✓	✓	✓	✓				✓		
Passthrough	✓			✓													

NOTE: ERROR_MODE is only valid on mirrorsets that have been specified for DT_SUPPORT.

Regardless of storageset type, the RUN and NORUN switches cannot be specified for partitioned units.

ENABLE_ACCESS_PATH=
DISABLE_ACCESS_PATH=
THIS_PORT_1_ENABLE_ACCESS_PATH=
THIS_PORT_1_DISABLE_ACCESS_PATH=
THIS_PORT_2_ENABLE_ACCESS_PATH=
THIS_PORT_2_DISABLE_ACCESS_PATH=
OTHER_PORT_1_ENABLE_ACCESS_PATH=
OTHER_PORT_1_DISABLE_ACCESS_PATH=
OTHER_PORT_2_ENABLE_ACCESS_PATH=
OTHER_PORT_2_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.

ENABLE_ACCESS_PATH and DISABLE_ACCESS_PATH switches are available only when in transparent mode; the other switches are available only when in multiple bus failover mode.

If you enable another host ID, previously enabled hosts are not disabled. The new ID are added. If you wish to enable only certain IDs, disable all access paths (DISABLE_ACCESS_PATH=ALL or THIS(OTHER)_PORT_1(2)_DISABLE_ACCESS_PATH=ALL), then enable the desired IDs. The system will display the following message (transparent failover mode, typical):

```
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.
```



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. Partitioned units must be on a single port.

MAXIMUM_CACHED_TRANSFER=n
MAXIMUM_CACHED_TRANSFER=32 (Default)

The controller does not cache any transfers over the set size. Accepted values are 1 through 2048.

PREFERRED_PATH=OTHER_CONTROLLER
PREFERRED_PATH=THIS_CONTROLLER
NOPREFERRED_PATH (Default)

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_ID* (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.

ERROR_MODE=NORMAL
ERROR_MODE=FAILSAFE

Valid only on mirrorsets that have been enabled for `DT_SUPPORT`. Controls how the mirroring software treats member failure.

Specify `NORMAL` and the mirroring software will fail any I/O—forcing the mirrorset inoperative—when the normal/normalizing local membership falls to zero.

Specify `FAILSAFE` and the mirroring software will fail any I/O—forcing the mirrorset inoperative—whenever either normal/normalizing remote membership or normal/normalizing local membership falls to zero.

If `FAILSAFE` is specified, the `DT_SUPPORT` must be enabled on the mirrorset and there must be at least one normal/normalizing local member and one normal/normalizing remote member, otherwise an error is displayed.

READ_CACHE (Default)
NOREAD_CACHE

Sets the controller’s read-cache function.

Read caching improves performance in almost all situations. Therefore, it is recommended to 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 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 unit'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 partitions.

WRITE_PROTECT (Default)
NOWRITE_PROTECT

Tells the controller whether data contained on the unit can be overwritten.

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

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

WRITEBACK_CACHE (Default)
NOWRITEBACK_CACHE

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 the controller's write-back caching. `WRITEBACK_CACHE` cannot be enabled on units with an underlying `DT_SUPPORT` mirrorset container.

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 5 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 *controller*

SHOW *device-name*

SHOW *device-type*

SHOW EMU

SHOW *storage-set-name*

SHOW *storage-set-type*

SHOW *unit-number*

SHOW UNITS

Parameters

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.
- PASSTHROUGH—Shows all passthrough containers attached to the controller.

EMU

Displays information regarding the status of the environmental monitoring unit (EMU).

storage-set-name

Specifies the name of a particular storage set to be displayed. For example, SHOW STRIPE1 displays information about the stripe set named STRIPE1.

storage-set-type

Specifies the type of storage set to be displayed. Valid types are:

- FAILEDSET—Shows the failed set configured to the controller.
- MIRRORSETS—Shows all mirror sets configured to the controller.
- RAIDSETS—Shows all RAID sets configured to the controller.
- SPARESET—Shows the spare set configured to the controller.
- STORAGESETS—Shows all storage sets configured with the controller.
- STRIPESETS—Shows all stripe sets 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.

If the underlying container is a mirrorset with DT_SUPPORT, it displays the ERROR_MODE setting. It reports if the mirrorset is failsafe locked or normal. A unit that is inoperative due to the failsafe locked condition will only be shown at the unit level (not mirrorset level). Since WRITEBACK_CACHE is not an allowable switch for DT_SUPPORT mirrorset, the switch is not displayed.

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 IDs.

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
DISK100	disk	1	0	0	RO
DEC	HSZ80	(C)	DEC	V10Z	
Switches:					
NOTTRANSPORTABLE					
TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 20.00 MHZ negotiated)					
LOCAL					
Size: 8378028 blocks					
DISK200	disk	2	0	0	RO
DEC	HSZ80	(C)	DEC	V10Z	
Switches:					
NOTTRANSPORTABLE					
TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 20.00 MHZ negotiated)					
LOCAL					
Size: 8378028 blocks					
DISK300	disk	3	0	0	RO
DEC	HSZ80	(C)	DEC	V10Z	
Switches:					
NOTTRANSPORTABLE					
TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 20.00 MHZ negotiated)					
REMOTE					
Size: 8378028 blocks					
DISK400	disk	4	0	0	RO
DEC	HSZ80	(C)	DEC	V10Z	

Switches:
 NOTTRANSPORTABLE
 TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 20.00 MHZ negotiated)
 REMOTE
 Size: 8378028 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 DISK60300	S0

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 mirrorset:

SHOW MIRRORSET FULL

Name	Storageset	Uses	Used by
M0	mirrorset	DISK100 DISK200 DISK300 DISK400	D0

Switches:
 COPY (priority) = NORMAL
 READ_SOURCE = LEAST_BUSY
 MEMBERSHIP = 4, 4 MEMBER PRESENT

```

DT_SUPPORT
State:
RECONSTRUCTING
LOCAL DISK100 (member 0) is NORMAL
REMOTE DISK200 (member 1) is NORMALIZING 2% complete
LOCAL DISK300 (member 2) is NORMALIZING 0% complete
REMOTE DISK400 (member 3) is NORMALIZING 0% complete
Size: 8378028 blocks

```

This example shows the full information for a controller:

SHOW THIS_CONTROLLER FULL

Controller:

```

Not configured for dual-redundancy
Device port SCSI address 7
Time: NOT SET

```

Host port:

```

SCSI target(s) (0, 2, 4, 5)
Preferred target(s) (0, 2, 4)
TRANSFER_RATE_REQUESTED = 20MHZ
Host Functionality Mode = A
Command Console LUN is target 0, lun 4

```

Cache:

```

64 megabyte write cache, version 4
Cache is GOOD
Battery is GOOD
No unflushed data in cache
CACHE_FLUSH_TIMER = DEFAULT (10 seconds)
NOCACHE_UPS

```

Mirrored Cache:

```

64 megabyte write cache, version 4
Cache is GOOD
Battery is GOOD
No unflushed data in cache

```

Extended information:

```

Terminal speed 19200 baud, eight bit, no parity, 1 stop bit
Operation control: 00000001 Security state code: 75580
Configuration backup disabled

```

5-142 *CLI Commands*

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*

SELFTTEST *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 only 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 storage set 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 storage sets in your shelves.

Device Profile

Type of Storageset

<input type="checkbox"/> Platter disk drive	<input type="checkbox"/> Optical Disk Drive
---	---

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

Storage Profile

Type of Storage Profile

<input type="checkbox"/> Mirrorset	<input type="checkbox"/> RAIDset	<input type="checkbox"/> Stripeset	<input type="checkbox"/> Striped Mirrorset
------------------------------------	----------------------------------	------------------------------------	--

Storage Profile Name.....

Disk Drives.....

Unit Number.....

Partitions

| Unit # |
|--------|--------|--------|--------|--------|--------|--------|--------|
| % | % | % | % | % | % | % | % |

RAIDset Switches

Reconstruction Policy		Reduced Membership		Replacement Policy	
<input type="checkbox"/> Normal (default)	<input type="checkbox"/> Fast	<input type="checkbox"/> No (default)	<input type="checkbox"/> Yes, missing:	<input type="checkbox"/> Best performance (default)	<input type="checkbox"/> Best fit
				<input type="checkbox"/> None	

Mirrorset Switches

Replacement Policy		Copy Policy		Read Source	
<input type="checkbox"/> Best performance (default)	<input type="checkbox"/> Best fit	<input type="checkbox"/> Normal (default)	<input type="checkbox"/> Fast	<input type="checkbox"/> Least busy (default)	<input type="checkbox"/> Round robin
<input type="checkbox"/> None				<input type="checkbox"/> Disk drive:	

Initialize Switches

Chunk size		Save Configuration		Metadata	
<input type="checkbox"/> Automatic (default)	<input type="checkbox"/> 64 blocks	<input type="checkbox"/> No (default)	<input type="checkbox"/> Yes	<input type="checkbox"/> Destroy (default)	<input type="checkbox"/> Retain
<input type="checkbox"/> 128 blocks	<input type="checkbox"/> 256 blocks				
<input type="checkbox"/> Other:					

Unit Switches

Read Cache		Read-Ahead Cache		Maximum Cache Transfer		Error Mode*	
<input type="checkbox"/> Yes (default)	<input type="checkbox"/> No	<input type="checkbox"/> Yes (default)	<input type="checkbox"/> No	<input type="checkbox"/> 32 blocks (default)	<input type="checkbox"/> Other:	<input type="checkbox"/> Normal	<input type="checkbox"/> Failsafe
Write Cache		Write Protection		Availability		*Valid only for mirrorsets with DT support enabled	
<input type="checkbox"/> Yes (default)	<input type="checkbox"/> No	<input type="checkbox"/> No (default)	<input type="checkbox"/> Yes	<input type="checkbox"/> Run (default)	<input type="checkbox"/> NoRun		

Appendix B **Controller Specifications**

This appendix contains physical, electrical, and environmental specifications for the HSZ80 Array Controller.

B-2 *Controller Specifications*

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
HSZ80 Array Controller module	12.5 inches	8.75 inches	23.27 W	6.04 A	
Write-back Cache, 512 MB (Battery charging)	12.5 inches	7.75 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 and +12 V). This table does not included tolerances.

Environmental Specifications

The HSZ80 array controller is intended for installation in a Class A computer room environment.

The optimum 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)

B-4 *Controller Specifications*

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 HSZ80. It is not a comprehensive glossary of computer terms.

adapter	A device that converts the protocol and hardware interface of one bus type into another without changing the function of the bus.
ACS	Array Controller Software. <i>See</i> array controller software.
array controller	<i>See</i> controller.
array controller software	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. <i>See also</i> synchronous.
bad block	A data block that contains a physical defect.
bad block replacement	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.
battery hysteresis	The ability of the software to allow write-back caching during the time when a battery is charging, but only when a previous down time has not drained more than 50 percent of a battery's rated capacity.
BBR	<i>See</i> bad block replacement.
BIST	<i>See</i> built-in self-test.

GL-2 *Glossary*

bit	A single binary digit having a value of either 0 or 1. A bit is the smallest unit of data a computer can process.
block	Also called a sector. The smallest collection of consecutive bytes addressable on a disk drive. In integrated storage elements, a block contains 512 bytes of data, error codes, flags, and the block's address header.
bootstrapping	A method used to bring a system or device into a defined state by means of its own action. For example, a machine routine whose first few instructions are enough to bring the rest of the routine into the computer from an input device.
built-in self-test	A diagnostic test performed by the array controller software on the controller's policy processor.
byte	A binary character string made up of 8 bits operated on as a unit.
cache memory	A portion of memory used to accelerate read and write operations.
CDU	Cable Distribution Unit. The power entry device for Compaq 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. <i>See also</i> 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.
CLI	<i>See</i> command line interpreter.
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 SBB device can be replaced. <i>See also</i> 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. Stripsets and mirrorsets are examples of storageset containers the controller uses to create units.
controller	<p>A hardware device that, with proprietary software, facilitates communications between a host and one or more devices organized in an array.</p> <p>A state in which data to be copied to the mirrorset is inconsistent with other members of the mirrorset. <i>See also</i> normalizing.</p>
copying member	Any member that joins the mirrorset after the mirrorset is created. 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. <i>See also</i> normalizing member.
CSR	Control and Status Register.
DAEMON	A program usually associated with UNIX systems and 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 Compaq subsystem cabinets, such as the SW600-series and 800-series cabinets in which StorageWorks components can be mounted.

GL-4 *Glossary*

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	<i>See</i> 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. <i>See also</i> 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	Write-back cached data that has not been written to storage media, even though the host operation processing the data has completed.
disaster tolerance	The ability to resume data center operations shortly after a significant event occurs at the primary data center. The ability to restart processing can require an alternate data center, with current copies of critical data in a usable state.
DMA	Direct Memory Access.
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.
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 one or two cabinets are connected to the primary cabinet.
external cache battery	<i>See</i> ECB.
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.

GL-6 *Glossary*

FD SCSI	The fast, narrow, differential SCSI bus with an 8-bit data transfer rate of 10 MB/s. <i>See also</i> FWD SCSI and SCSI.
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.
FRU	Field Replaceable Unit. A hardware component that can be replaced at the customer's location by Compaq service personnel or qualified customer service personnel.
full duplex	A communications system in which there is a capability for 2-way transmission and acceptance between two sites at the same time.
FWD SCSI	A fast, wide, differential SCSI bus with a maximum 16-bit data transfer rate of 20 MB/s. <i>See also</i> SCSI and FD SCSI.
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.
half-duplex	Pertaining to a communications system in which data can be either transmitted or received but only in one direction at one time.
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.

host compatibility mode	A setting used by the controller to provide optimal controller performance with specific operating systems. This improves the controller's performance and compatibility with the specified operating 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. <i>See also</i> 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. <i>See also</i> 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. <i>See also</i> 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.
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.

GL-8 *Glossary*

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 vice versa), the actual transfer of the data, and the processing and overlaying activity to make both of those happen.
JBOD	Just a Bunch of Disks. Describes a group of single-device logical units.
LBN	Logical Block Number.
LED	Light Emitting Diode.
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. <i>See also</i> 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.
LRU	Least Recently Used. A cache term used to describe the block replacement policy for read cache.
Mb/s	Megabits per second. Approximately one million (10^6) bits per second.

MB/s	Megabytes per second. Approximately one million (10^6) bytes 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. <i>See also</i> 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."
mirrored write-back caching	A method of caching data that maintains two copies of the cached data. The copy is available if either cache module fails.
mirroring	The act of creating an exact copy or image of data.
mirrorset	<i>See</i> RAID level 1.
MIST	Module Integrity Self-Test.
network	A data communication, a configuration in which two or more terminals or devices are connected to enable information transfer.
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.
nonredundant controller configuration	(1) A single controller configuration. (2) A controller configuration that does not include a second controller.

GL-10 Glossary

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.
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 becomes a normal member when either all of the normal members fail or all of the normal members are removed from the mirrorset. <i>See also</i> copying member.
NVM	Nonvolatile Memory. A type of memory where the contents survive power loss. Also 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. <i>See also</i> this controller.
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.
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 bit	A binary digit added to a group of bits that checks to see if there are errors in the transmission.
parity RAID	<i>See</i> 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.
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.
primary cabinet	The primary cabinet is the subsystem enclosure that contains the controllers, cache modules, external cache batteries, and the PVA module.
program card	The PCMCIA card containing the controller's operating software.

GL-12 Glossary

protocol	The conventions or rules for the format and timing of messages sent and received.
PTL	Port-Target-LUN. The controller's method of locating a device on the controller's device bus.
PVA module	Power Verification and Addressing module.
RAID	Redundant Array of Independent Disks. Represents multiple levels of storage access developed to improve performance or availability or both.
RAID level 0	A RAID storage set 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 are excellent, this RAID level is the only one that does not provide redundancy. RAID level 0 storage sets are sometimes referred to as stripe sets.
RAID level 0+1	A RAID storage set 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. RAID level 0+1 storage sets are sometimes referred to as striped mirror sets.
RAID level 1	A RAID storage set of two or more physical disks that maintains a complete and independent copy of the entire virtual disk's data. This type of storage set has the advantage of being highly reliable and extremely tolerant of device failure. RAID level 1 storage sets are sometimes referred to as mirror sets.
RAID level 3	A RAID storage set 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 storage set. <i>See also</i> RAID level 5.
RAID Level 5	A RAID storage set that, unlike RAID level 3, stores the parity information across all of the disk drives within the storage set. <i>See also</i> RAID level 3.

RAID level 3/5	A Compaq-developed RAID storage set that stripes data and parity across three or more members in a disk array. A RAID set combines the best characteristics of RAID level 3 and RAID level 5. A RAID set is the best choice for most applications with small to medium I/O requests, unless the application is write intensive. A RAID set is sometimes called parity RAID. RAID level 3/5 storage sets are sometimes referred to as RAID sets.
RAID set	<i>See</i> RAID level 3/5.
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 spare set disk and then incorporates the spare set disk into the mirror set, striped mirror set, or RAID set from which the failed member came. <i>See also</i> regeneration.
reduced	Indicates that a mirror set or RAID set 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 RAID set is considered to be redundant when all of the parity blocks contain the correct parity for the user data.
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 storage set. The regeneration of an entire RAID set member is called reconstruction. <i>See also</i> reconstruction.
request rate	The rate at which requests are arriving at a servicing entity.

GL-14 Glossary

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	Storage 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. <i>See also</i> 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. <i>See also</i> 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 data transmission	A method transmission in which each bit of information is sent sequentially on a single channel rather than simultaneously as in parallel transmission.
signal converter	<i>See</i> SCSI bus signal converter.
single ended I/O module	A 16-bit I/O module. <i>See also</i> 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 storage set.
storage array	An integrated set of storage devices.
storage array subsystem	<i>See</i> storage subsystem.
storage set	(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.

GL-16 Glossary

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	<p>A family of Compaq 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.</p> <p>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.</p>
stripe	The data divided into blocks and written across two or more member disks in an array.
striped mirrorset	<i>See</i> RAID level 0+1.
stripeset	<i>See</i> 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	<p>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.</p> <p>Each stripeset member receives an equal share of the I/O request load, improving performance.</p>

surviving controller	The controller in a dual-redundant configuration pair that serves its companion's devices when the companion controller fails.
synchronous	Pertaining to a method of data transmission which allows each event to operate in relation to a timing signal. <i>See also</i> 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.
Target ID number	The address a bus initiator uses to connect with a bus target. Each bus target is assigned a unique target address.
this controller	The controller that is serving your current CLI session through a local or remote terminal. <i>See also</i> other controller.
transfer data rate	The speed at which data may be exchanged with the central processor, expressed in thousands of bytes per second.
Ultra SCSI bus	A wide, Fast-20 SCSI bus.
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 32 units on each target. <i>See also</i> target and target ID number.
unwritten cached data	Sometimes called unflushed data. <i>See</i> 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.

GL-18 *Glossary*

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 on the controller.
VTDPY	Virtual Terminal Display utility.
warm swap	A device replacement method that allows the complete system to remain 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.
Worldwide name	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. 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

A

- AC input module
 - parts, 1–4
- ADD DISK, 5–7
 - LOCAL, 5–8
 - NOTTRANSPORTABLE, 5–8
 - REMOTE, 5–8
 - TRANSFER_RATE_REQUESTED, 5–9
 - TRANSPORTABLE, 5–8
- ADD DISK *container-name scsi-port-target-lun*, 5–7
- ADD MIRRORSET, 5–11
 - COPY, 5–11
 - DT_SUPPORT, 5–12
 - NODT_SUPPORT, 5–12
 - POLICY, 5–12
 - READ_SOURCE, 5–13
- ADD PASSTHROUGH, 5–16
 - TRANSFER_RATE_REQUESTED, 5–17
- ADD PASSTHROUGH *passthrough-name scsi-port-target-lun*, 5–16
- ADD RAIDSET, 5–19
 - NOPOLICY, 5–20
 - NOREduced, 5–21
 - POLICY, 5–20
 - RECONSTRUCT, 5–20
 - REDUCED, 5–21
- ADD RAIDSET *RAIDset-name container-nameN*, 5–19
- ADD SPARESET, 5–23
- ADD SPARESET *disk-name*, 5–23
- ADD STRIPESET, 5–24
- ADD STRIPESET *stripeset-name container-nameN*, 5–24
- ADD UNIT, 5–27
 - DISABLE_ACCESS_PATH, 5–29
 - ENABLE_ACCESS_PATH, 5–29
 - MAXIMUM_CACHED_TRANSFER, 5–30
 - NOPREFERRED_PATH, 5–30
 - NOREAD_CACHE, 5–31
 - NOREADAHEAD_CACHE, 5–31
 - NORUN, 5–32
 - NOWRITE_PROTECT, 5–32
 - NOWRITEBACK_CACHE, 5–32
 - PARTITION, 5–30
 - PREFERRED_PATH, 5–30
 - READ_CACHE, 5–31
 - READAHEAD_CACHE, 5–31
 - RUN, 5–32
 - WRITE_PROTECT, 5–32
 - WRITEBACK_CACHE, 5–32
- ADD UNIT *unit-number container-name*, 5–27
- Adding
 - disks, 5–7
 - mirrorsets, 5–11
 - passthrough, 5–16
 - RAIDsets, 5–19
 - sparesets, 5–23
 - stripesets, 5–24
 - units, 5–27
- Adding disk drives
 - as eligible devices, 4–2
 - to spareset using CLI, 4–13

Addresses
 providing with the PVA module, 2–25
Addressing PTL convention, 3–36
ALLOCATION_CLASS
 SET *controller*, 5–93
Array of disk drives, 3–8
AUTOSPARE, 4–15
 SET FAILEDSET, 5–108
Autospare
 failedset, 4–15
Availability, 3–15

B

BA370 rack-mountable enclosure
 parts, 1–4
Backing up data, 3–20
Backing up data with the Clone utility, 1–16
Backplane
 location, 1–8
Backup power source
 enabling write-back caching, 1–22
Battery hysteresis, 1–31
BATTERY_OFF
 POWEROFF, 5–74
BATTERY_ON
 POWEROFF, 5–74
Building blocks
 AC input module, 1–4
 BA370 rack-mountable enclosure, 1–4
 cache module, 1–4
 controller, 1–4
 cooling fan, 1–4
 disk drives, 1–4
 dual-battery ECB, 1–4
 EMU, 1–4
 I/O module, 1–4

 power cable, 1–4
 power cable kit, 1–4
 power supply, 1–4
 PVA module
 SCSI hub, 3-port, 1–4
 SCSI hub, 5-port, 1–4
 SCSI hub, 9-port, 1–4
 single-battery ECB, 1–4
Bus
 device bus interconnect, 1–5
 distribute members across, 3–15, 3–17
 distributing first mirrorset members, 3–14
 distributing members across, 3–11

C

Cables
 maintenance port cable part number for a
 terminal, 1–12
 tightening, xvii
Cache module
 caching techniques, 1–20
 companion cache module, 1–18
 controller and cache module location, 1–13
 DIMMs supported, 1–5, 1–18
 general description, 1–18
 illustration of parts, 1–19
 location, 1–2, 1–13
 maximum memory size supported, 1–5
 memory configurations, 1–18
 memory sizes supported, 1–5
 parts, 1–4
 read caching, 1–20
 read-ahead caching, 1–20
 relationship to controller, 1–13
 replacing cache modules with FRUTIL, 1–17
 write-back caching, 1–21

- write-through caching, 1–21
- Cache policies
 - fault-tolerance for write-back caching, 1–22
- Cache, setting flush timer, 5–93
- CACHE_FLUSH_TIMER
 - SET *controller*, 5–93
- CACHE_UPS
 - SET *controller*, 5–93
- Caching techniques
 - general description, 1–20
 - read caching, 1–20
 - read-ahead caching, 1–20
 - write-back caching, 1–21
 - write-through caching, 1–21
- CAPACITY
 - CREATE_PARTITION, 5–49
 - INITIALIZE, 5–63
- Change volume serial number utility. *See*
 - CHVSN utility
- Changing switches
 - devices, 3–42
 - initialize, 4–18
 - storage sets, 3–42
 - unit, 4–18
- Charging diagnostics
 - battery hysteresis, 1–31
 - general description, 1–31
- Chunk size, 3–51
 - choosing for RAIDsets and stripesets, 3–51
 - controlling stripesize, 3–51
 - maximum for RAIDsets, 3–54
 - using to increase data transfer rate, 3–53
 - using to increase request rate, 3–52
 - using to increase write performance, 3–54
- CHUNKSIZE, 3–51
- INITIALIZE, 5–63
- Chunksize, setting storage set size, 5–63
- CHVSN utility
 - general description, 1–17
- CHVSN, running, 5–86
- CLCP utility
 - general description, 1–16
- CLCP, running, 5–86
- CLEAR_ERRORS CLI, 5–35
- CLEAR_ERRORS *controller*
 - INVALID_CACHE, 5–36
 - data-retention-policy, 5–36
 - DESTROY_UNFLUSHED_DATA, 5–36
 - NODESTROY_UNFLUSHED_DATA, 5–36
- CLEAR_ERRORS *device-name* UNKNOWN, 5–38
- CLEAR_ERRORS *unit-number*
 - LOST_DATA, 5–40
- CLI
 - definition, 5–2
 - overview, 5–2
- CLI commands
 - abbreviating commands, 5–3
 - ADD DISK, 5–7
 - ADD MIRRORSET, 5–11
 - ADD PASSTHROUGH, 5–16
 - ADD RAIDSET, 5–19
 - ADD SPARESET, 5–23
 - ADD STRIPESSET, 5–24
 - ADD UNIT, 5–27
 - CLEAR_ERRORS CLI, 5–35
 - CLEAR_ERRORS *controller*
 - INVALID_CACHE, 5–36
 - CLEAR_ERRORS *device-name*
 - UNKNOWN, 5–38
 - CLEAR_ERRORS *unit-number*
 - LOST_DATA, 5–40

- CLEAR_ERRORS *unit-number*
 - UNWRITEABLE_DATA, 5-42
 - CONFIGURATION RESET, 5-44
 - CONFIGURATION RESTORE, 5-45
 - CONFIGURATION SAVE, 5-47
 - CREATE_PARTITION, 5-48
 - customizing the prompt, 5-98
 - DELETE *container-name*, 5-51
 - DELETE FAILEDSET, 5-53
 - DELETE SPARESET, 5-54
 - DELETE *unit-number*, 5-55
 - DESTROY_PARTITION, 5-57
 - DIRECTORY, 5-59
 - getting help, 5-3
 - HELP, 5-60
 - INITIALIZE, 5-62
 - LOCATE, 5-67
 - MIRROR, 5-70
 - overview, 5-2
 - POWEROFF, 5-74
 - REDUCE, 5-77
 - RENAME, 5-81
 - RESTART *controller*, 5-83
 - RETRY_ERRORS *unit-number*
 - UNWRITEABLE_DATA, 5-85
 - rules for entering, 5-3
 - RUN, 5-86
 - SELFTTEST *controller*, 5-89
 - SET *controller*, 5-91
 - SET *device-name*, 5-102
 - SET EMU, 5-105
 - SET FAILEDSET, 5-108
 - SET FAILOVER, 5-110
 - SET *mirrorset-name*, 5-112
 - SET MULTIBUS_FAILOVER, 5-118
 - SET NOFAILOVER, 5-120
 - SET NOMULTIBUS_FAILOVER, 5-122
 - SET RAID*set-name*, 5-124
 - SET *unit-number*, 5-128
 - SHOW, 5-136
 - SHUTDOWN *controller*, 5-143
 - syntax, 5-6
 - UNMIRROR, 5-145
 - CLI prompt
 - changing the character string, 2-22
 - CLONE
 - procedure, 3-21
 - utility, 3-20
 - Clone utility
 - general description, 1-16
 - CLONE, running, 5-86
 - Cloning data, 3-20
 - Code load and code patch utility. *See* CLCP utility
 - Command line interpreter. *See* CLI
 - COMMAND_CONSOLE_LUN
 - SET *controller*, 5-94
 - Communicating with a controller
 - from a local terminal, 2-27
 - Comparison of storagesets, 3-8
 - Components. *See* Controller
 - CONFIG utility
 - general description, 1-16
 - CONFIG, running, 5-86
 - Configuration
 - modifying controller configurations, 5-2
 - resetting, 5-44
 - restoring, 5-45, 5-64
 - saving, 5-47
-

- CONFIGURATION RESET, 5–44
- CONFIGURATION RESTORE, 5–45
- Configuration rules
 - devices, 2–2
 - LUN capacity, 2–2
 - mirrorsets, 2–2
 - partitions per storage set, 2–2
 - RAID-5 and RAID-1 storage sets, 2–2
 - RAID-5 storage sets, 2–2
 - RAID-5, RAID-1, and RAID-0 storage sets, 2–2
 - requirements, 2–2
 - striped mirrorsets, 2–2
 - stripesets, 2–2
 - See also* Summary of controller features
- CONFIGURATION SAVE, 5–47
- Configuration utility. *See* CONFIG utility
- Configuring
 - controller, 2–3, 2–29
 - dual-redundant controller configurations, 2–29
 - dual-redundant controller configurations with mirrored cache, 2–31
 - mirrorsets, 4–4
 - multiple-bus failover, 2–30
 - RAIDsets, 4–6
 - single-disk unit, 4–9
 - striped mirrorsets, 4–7
 - stripesets, 4–3
- Configuring using CLI
 - mirrorsets, 4–4
 - RAIDsets, 4–6
 - single-disk units, 4–9
 - storage sets, 4–2
 - striped mirrorsets, 4–7
 - stripesets, 4–3
- Connecting, 2–15
 - dual-redundant controllers to the host, 2–15
 - local connection to the controller, 2–26
 - multiple-bus failover, dual-redundant controllers to the host, 2–21
 - PC connection to the controller, 2–26
 - single controller to the host, 2–15
 - terminal connection to the controller, 2–26
- Container
 - initializing, 5–62
- Controller
 - “this” and “other” defined, xviii
 - addressing, 3–36
 - backplane, 1–8
 - changing the CLI prompt, 2–22
 - communicating from a local terminal, 2–27
 - configuring, 2–3, 2–29
 - controller and cache module location, 1–13
 - displaying information, 5–136
 - dual-redundant controller configuration, 2–15, 2–29
 - dual-redundant controller configurations with mirrored cache, 2–31
 - ECB diagnostics, 1–31
 - fault LEDs, 1–13
 - general description, 1–7
 - host ports, 1–8
 - illustration of parts, 1–9
 - local connection, 2–26
 - location, 1–2, 1–13
 - maintenance port, 1–8
 - multiple-bus failover configuration, 2–30
 - multiple-bus failover mode, 2–30
 - multiple-bus failover, dual-redundant controller configuration, 2–21
 - node IDs, 3–30

- OCP, 1–8, 1–13
 - other controller defined, 2–28
 - part numbers of parts used in configuring, 1–10
 - parts, 1–4
 - parts used in configuring, 1–10
 - patching controller software with the CLCP utility, 1–16
 - program card, 1–8
 - relationship to cache module, 1–13
 - release lever, 1–8
 - replacing a failed controller with FRUTIL, 1–17
 - reset button on the OCP, 1–14
 - showing, 5–136
 - shutting down, 5–143
 - single-controller configuration, 2–15
 - summary of features, 1–5
 - testing with DILX, 1–16
 - this controller defined, 2–28
 - transparent failover mode, 2–29
 - troubleshooting with FMU, 1–15
 - upgrading controller software with the CLCP utility, 1–16
 - fault-management. *See* FMU
 - Conventions
 - typographical, xviii
 - warnings, cautions, tips, notes, xviii
 - Cooling fan
 - parts, 1–4
 - COPY
 - ADD MIRRORSET, 5–11
 - mirrorset switches, 3–46
 - SET *mirrorset-name*, 5–112
 - CREATE_PARTITION, 5–48
 - CAPACITY, 5–49
 - CYLINDERS, 5–49
 - HEADS, 5–49
 - SECTORS_PER_TRACK, 5–49
 - SIZE, 5–48
 - CREATE_PARTITION *container-name*
 SIZE=percent, 5–48
 - Creating
 - disks, 5–7
 - mirrorsets, 5–11
 - partitions, 4–10
 - passthrough, 5–16
 - RAIDsets, 5–19
 - single-disk units, 5–33
 - sparesets, 5–23
 - storageset and device profiles, 3–5
 - stripesets, 5–24
 - units, 5–27
 - CYLINDERS
 - CREATE_PARTITION, 5–49
 - INITIALIZE, 5–63
 - D**
 - Data
 - backing up with the Clone utility, 1–16
 - duplicating with the Clone utility, 1–16
 - Data transfer rate, 3–53
 - Data-retention-policy
 - CLEAR_ERRORS *controller*
 INVALID_CACHE, 5–36
-

- DELETE *container-name*, 5–51
- DELETE FAILEDSET, 5–53
- DELETE FAILEDSET *disk-name*, 5–53
- DELETE SPARESET, 5–54
- DELETE SPARESET *disk-name*, 5–54
- DELETE *unit-number*, 5–55
- Deleting
 - devices, 5–51
 - mirrorsets, 5–51
 - RAIDsets, 5–51
 - storagesets, 5–51
 - stripesets, 5–51
 - units, 5–55
- Deleting storagesets, 4–16
- DESTROY, 3–57
 - INITIALIZE, 5–64
- DESTROY_PARTITION, 5–57
- DESTROY_PARTITION *container-name*
PARTITION=*partition-number*, 5–57
- DESTROY_UNFLUSHABLE_DATA
 - SET NOFAILOVER, 5–120
 - SET NOMULTIBUS_FAILOVER, 5–122
- DESTROY_UNFLUSHED_DATA
 - CLEAR_ERRORS *controller*
INVALID_CACHE, 5–36
- Device bus interconnect, 1–5
- Device ports
 - LEDs, 1–13
 - number supported, 1–5
- Device profile, A–2
- Device protocol, 1–5
- Device statistics utility. *See* DSTAT utility
- Device switches, 3–42, 3–48
 - changing switches, 3–42
 - device transfer rate, 3–49
 - enabling switches, 3–42
- LOCAL, 3–50
 - local/remote, 3–50
- NOTTRANSPORTABLE, 3–48
- REMOTE, 3–50
- TRANSFER_RATE_REQUESTED, 3–49
 - transportability, 3–48
- TRANSPORTABLE, 3–48
- Device targets. *See* Devices
- Device transfer rate, 3–49
- Devices
 - adding with the CONFIG utility, 1–16
 - changing switches, 4–17
 - creating a profile, 3–5
 - generating a new volume serial number with
the CHVSN utility, 1–17
 - largest supported, 1–6, 2–2
 - locating, 5–67
 - maximum number in striped mirrorsets, 1–6,
2–2
 - maximum number supported, 1–5, 2–2
 - number per port, 1–5
 - renaming the volume serial number with the
CHVSN utility, 1–17
 - setting data transfer rate, 5–9, 5–103
- SHOW *device-type*, 5–137
- showing, 5–136
- specifying address, ?? to 3–37
- transfer rate, 3–49
- warm swap, 1–5
- See also* Disk drives, LUN, Partitioning,
Partitions, Storagesets
- Diagnostics
 - ECB charging, 1–31
 - listing of, 5–59
 - running, 5–86

- DILX
 - general description, 1–16
 - DILX, running, 5–86
 - DIMMs
 - cache module memory configurations, 1–18
 - supported, 1–5
 - DIRECT, running, 5–87
 - DIRECTORY, 5–59
 - DISABLE_ACCESS
 - SET *unit-number*, 5–130
 - DISABLE_ACCESS_PATH
 - ADD UNIT, 5–29
 - Disabling
 - autospare, 4–15
 - Disaster tolerance
 - write-back caching not enabled in disaster-tolerant mirrorsets, 1–21
 - Disaster tolerance support, 3–47
 - Disk drives
 - adding, 4–2
 - adding to configuration, 5–7
 - adding to spareset using CLI, 4–13
 - adding with the CONFIG utility, 1–16
 - array, 3–8
 - corresponding storagesets, 3–34
 - deleting, 5–51
 - displaying information, 5–136
 - dividing, 3–40
 - generating a new volume serial number with the CHVSN utility, 1–17
 - generating read and write loads with DILX, 1–16
 - initializing, 5–62
 - investigating data transfer with DILX, 1–16
 - largest device supported, 1–6, 2–2
 - making transportable, 5–103
 - mirroring, 5–70
 - monitoring performance with DILX, 1–16
 - partitioning, 4–10
 - partitions supported, 1–5, 2–2
 - parts, 1–4
 - removing from a mirrorset, 5–77
 - removing from sparesets using CLI, 4–14
 - removing from the failedset, 5–53
 - removing from the spareset, 5–54
 - renaming, 5–81
 - renaming the volume serial number with the CHVSN utility, 1–17
 - setting device data transfer rate, 5–9
 - showing, 5–136 to 5–137
 - transfer rate, 5–9
 - See also* Devices, LUN, Partitioning, Partitions, Storagesets
 - Disk inline exerciser
 - general description, 1–16
 - DISKS
 - SHOW *device-type*, 5–137
 - Display. *See* VTDPY
 - Displaying switches, 4–17
 - Distributing
 - first member of multiple mirrorsets, 3–14
 - members across ports, 3–15, 3–17
 - members of storageset, 3–11
 - Dividing storagesets, 3–40
 - Documentation, related, xxi
 - DSTAT, running, 5–87
 - DT_SUPPORT, 3–47
 - ADD MIRRORSET, 5–12
 - SET *mirrorset-name*, 5–113
 - Dual-battery ECB
 - parts, 1–4
-

Dual-redundant configuration

ECB, 1-29

Dual-redundant controller configuration

disabling, 5-120, 5-122

enabling, 5-110

Dual-redundant controller configurations

connecting to the host, 2-15

multiple-bus failover mode, 2-30

transparent failover mode, 2-29

when to use, 2-29

E

ECB

as a default backup source, 1-22

battery hysteresis, 1-31

diagnostics, 1-31

dual-battery ECB, 1-4

general description, 1-29

maintenance period, 1-29

replacing ECBs with FRUTIL, 1-17

single-battery ECB, 1-4

Electrostatic discharge precautions, xvi

EMU

parts, 1-4

setting, 5-105

ENABLE_ACCESS

SET *unit-number*, 5-130

ENABLE_ACCESS_PATH

ADD UNIT, 5-29

Enabling

AUTOSPARE, 4-15

Enabling switches

devices, 3-42

storage sets, 3-42

Enclosures

addressing with the PVA module, 2-25

PVA ID, 2-25

Erasing metadata, 3-57

Error messages

clearing from CLI, 5-35

clearing unwriteable data errors, 5-42

ERROR_MODE

SET *unit-number*, 5-132

ESD card cover

location, 1-10

Event messages

translating with FMU, 1-15

Events

setting display characteristics with FMU,
1-15

Examples

adding disk drives to a spareset, 4-14

cloning a storage unit, 3-22

configuring a mirrorset, 4-5

configuring a RAIDset, 4-7

configuring a single-disk unit, 4-10

configuring a striped mirrorset, 4-9

configuring a stripeset, 4-4

deleting storage sets, 4-16

partitioning a storage set, 4-12

removing disk drives from a spareset, 4-15

Exercisers

availability, 1-5

DILX, 1-16

See also Utilities and exercisersExternal cache battery. *See* ECB**F**

Failedset

autospare, 4-15

deleting members, 5-53

Failover

- disabling, 5-120, 5-122
- general description, 2-29
- multiple-bus, 2-30
- transparent, 2-29

FAILSAFE, error mode, 5-132

Failures

- displaying with FMU, 1-15
- setting display characteristics with FMU, 1-15

FANSPEED

- SET EMU, 5-106

Fault LEDs, 1-13

Fault management utility. *See* FMU

Fault-management software. *See* FMU

Fault-tolerance for write-back caching

- general description, 1-22
- nonvolatile memory, 1-22

Field Replacement utility. *See* FRUTIL

Finding devices and storagesets, 5-67

Firmware

- formatting disk drives with HSUTIL, 1-16
- upgrading with HSUTIL, 1-16

FMU

- general description, 1-15

FMU, running, 5-87

FRUTIL

- general description, 1-17

FRUTIL, running, 5-87

FULL

- SHOW, 5-138

H

HEADS

- CREATE_PARTITION, 5-49
- INITIALIZE, 5-63

HELP, 5-60

Help

- technical support, xv

Heterogeneous host support

- in multiple-bus failover mode, 4-32
- in transparent failover mode, 4-26

History, revision of this manual, xxii

Host

- configuring units with multiple hosts, 4-23
- heterogeneous host support, 4-23
- how it works with the controller and subsystem, 1-8
- maximum number of units presented, 1-5

Host bus

- setting target ID numbers, 2-33

Host bus interconnect, 1-5

Host ports

- location, 1-8
- maximum transfer speed, 1-6
- number, 1-5

Host protocol

- supported, 1-5

HOST_FUNCTION

- SET controller, 5-94

Host-assisted failover. *See* Failover, Multiple-bus failover

HSUTIL

- general description, 1-16
-

HSUTIL, running, 5–87
 HSZ80 Array Controller Subsystem. *See*
 Storage subsystem
 HSZ80 Array Controller. *See* Controller
 HSZterm. *See* remote connection
 Hysteresis. *See* Battery hysteresis

I

I/O

- distributing the load in transparent failover mode, 2–29
- investigating I/O activity of units with VTDPY, 1–15
- logging I/O activity with DSTAT, 1–17
- using preferred ID numbers to improve throughput, 2–35

I/O module

- parts, 1–4

I/O request routing, 3–39

IGNORE_ERRORS

- RESTART *controller*, 5–83
- SELFTTEST *controller*, 5–89
- SHUTDOWN *controller*, 5–143

IMMEDIATE_SHUTDOWN

- RESTART *controller*, 5–84
- SELFTTEST *controller*, 5–90
- SHUTDOWN *controller*, 5–144

INITIALIZE, 5–62

- CAPACITY, 5–63
- changing, 4–18
- CHUNKSIZE, 5–63
- CYLINDERS, 5–63
- DESTROY, 5–64
- HEADS, 5–63
- NODESTROY, 5–64
- NOSAVE_CONFIGURATION, 5–64

- SAVE_CONFIGURATION, 5–64
- saving user data, 5–64
- SECTORS_PER_TRACK, 5–63
- INITIALIZE *container-name*, 5–62
- Initialize switches, 3–51
- chunk size, 3–51
- CHUNKSIZE, 3–51
- DESTROY, 3–57
- destroy/nodestroy, 3–57
- NODESTROY, 3–57
- save configuration, 3–55
- SAVE_CONFIGURATION, 3–55

J

JBOD, 3–8

L

Largest device supported, 1–6, 2–2

Last failures

- displaying with FMU, 1–15

LEDs, 1–13

Listing diagnostics and utilities, 5–59

LOCAL, 3–50

- ADD DISK, 5–8
- SET *device-name*, 5–102

Local connection

- connecting to the controller, 2–26
- illustration of terminal to maintenance port, 2–27

Local terminal

- connecting through the maintenance port, 1–14

Local terminal port. *See* Maintenance port

Local/Remote, 3–50

LOCATE, 5–67

- ALL, 5–67

- CANCEL*, 5–67
 - container-name*, 5–68
 - DISKS*, 5–68
 - parameter*, 5–67
 - PTL (SCSI-location)*, 5–68
 - unit-number*, 5–68
 - UNITS*, 5–68
 - Lost data error, clearing, 5–40
 - LUN, 2–2
 - largest device, storageset, or unit, 1–6
 - M**
 - Maintenance port
 - establishing a local connection to the controller, 2–26
 - general description, 1–14
 - location, 1–8
 - precautions, xvii
 - terminal or PC connection, 2–26
 - See also* Maintenance port cable, Terminal connection
 - Maintenance port cable
 - establishing a local connection to the array controller, 2–26
 - PC or terminal connection illustration, 1–12
 - terminal connection part number, 1–12
 - See also* Maintenance port, Terminal connection
 - Mapping storagesets, 3–34
 - Maximum LUN capacity, 2–2
 - MAXIMUM_CACHED_TRANSFER
 - ADD UNIT, 5–30
 - SET *unit-number*, 5–131
 - Mean time between failures, 3–10
 - Members
 - distributing first member of mirrorset, 3–14
 - distributing on bus, 3–14 to 3–15, 3–17
 - MEMBERSHIP
 - SET *mirrorset-name*, 5–113
 - Membership
 - RAIDset switches, 3–44
 - Memory-system failures
 - displaying with FMU, 1–15
 - Messages
 - translating with FMU, 1–15
 - Metadata
 - erasing, 4–15
 - retaining, 4–15
 - MIRROR, 5–70
 - COPY, 5–71
 - POLICY, 5–71
 - MIRROR *disk-name mirrorset-name*, 5–70
 - Mirrored write-back cache
 - enabling, 2–31
 - MIRRORED_CACHE
 - SET *controller*, 5–96
 - Mirrorset switches, 3–45
 - COPY, 3–46
 - disaster tolerance support, 3–47
 - DT_SUPPORT, 3–47
 - NODT_SUPPORT, 3–47
 - POLICY, 3–45
 - READ_SOURCE, 3–46
 - Mirrorsets
 - actual number of members, 5–77
 - adding to configuration, 5–11
 - changing switches, 4–17
 - choosing a replacement member, 5–20
 - configuring using CLI, 4–4
 - converting back to a single device, 5–145
 - creating from a single disk, 5–70
 - deleting, 5–51
-

- description, 3–2, 3–12
- displaying information, 5–136
- duplicating data with the Clone utility, 1–16
- initializing, 5–62
- maximum number of members, 2–2
- planning, 3–13
- removing a member, 5–77
- renaming, 5–81
- setting a replacement policy, 5–20
- showing, 5–136
- temporary from CLONE, 3–20
- unmirroring, 5–145
- write-back caching not enabled in disaster-tolerant mirrorsets, 1–21
- Moving storage sets, 4–35
- MTBF, *See* Mean time between failures
- Multiple-bus failover configuration
 - when to use, 2–30
- Multiple-bus failover mode
 - configuring units with multiple hosts, 4–32
 - general description, 2–30
 - See also* Transparent failover mode
- Multiple-bus failover, dual-redundant configuration
 - connecting to the host, 2–21
- N**
- NO_OVERRIDE_BAD_FLUSH
 - POWEROFF, 5–74
- NOAUTOSPARE
 - SET FAILEDSET, 5–108
- NOCACHE_UPS
 - SET *controller*, 5–93
- NOCOMMAND_CONSOLE_LUN
 - SET *controller*, 5–94
- Node IDs, 3–30
- NODE_ID
 - SET *controller*, 5–95
- NODESTROY, 3–57
 - INITIALIZE, 5–64
- NODESTROY_UNFLUSHABLE_DATA
 - SET NOFAILOVER, 5–120
 - SET NOMULTIBUS_FAILOVER, 5–122
- NODESTROY_UNFLUSHED_DATA
 - CLEAR_ERRORS *controller*
 - INVALID_CACHE, 5–36
- NODT_SUPPORT, 3–47
 - ADD MIRRORSET, 5–12
 - SET *mirrorset-name*, 5–113
- NOIGNORE_ERRORS
 - RESTART *controller*, 5–83
 - SELFTEST *controller*, 5–89
 - SHUTDOWN *controller*, 5–143
- NOIMMEDIATE_SHUTDOWN
 - RESTART *controller*, 5–84
 - SELFTEST *controller*, 5–90
 - SHUTDOWN *controller*, 5–144
- NOMIRRORED_CACHE
 - SET *controller*, 5–96
- Nonvolatile memory
 - fault-tolerance for write-back caching, 1–22
- NOPOLICY
 - ADD RAIDSET, 5–20
- NOPORT_1_ID
 - SET *controller*, 5–96
- NOPORT_1_PREFERRED_ID
 - SET *controller*, 5–98
- NOPORT_1_TRANSFER_RATE_REQUEST
 - ED
 - SET *controller*, 5–100

NOPORT_2_ID
SET *controller*, 5-96

NOPORT_2_PREFERRED_ID
SET *controller*, 5-98

NOPORT_2_TRANSFER_RATE_REQUEST
ED
SET *controller*, 5-100

NOPREFERRED_PATH
ADD UNIT, 5-30
SET *unit-number*, 5-131

NOREAD_CACHE
ADD UNIT, 5-31
SET *unit-number*, 5-132

NOREADAHEAD_CACHE
ADD UNIT, 5-31
SET *unit-number*, 5-133

NOREduced
ADD RAIDSET, 5-21

NORUN
ADD UNIT, 5-32
SET *unit-number*, 5-133

NOSAVE_CONFIGURATION, 3-55
INITIALIZE, 5-64

Note, defined, xix

NOTERMINAL_PARITY
SET *controller*, 5-99

NOTRANSPORTABLE, 3-48
ADD DISK, 5-8
SET *device-name*, 5-103

NOWRITE_PROTECT
ADD UNIT, 5-32
SET *unit-number*, 5-133

NOWRITEBACK_CACHE
ADD UNIT, 5-32
SET *unit-number*, 5-134

O

OCP
fault LEDs, 1-13
general description, 1-13
location, 1-8
reset button, 1-13

Operator control panel. *See* OCP

Options
for devices, 3-48
for RAIDsets, 3-43
for storage units, 3-58
initialize, 3-51

Other controller
explained, 5-2

Other controller defined, 2-28

OVERRIDE_BAD_FLUSH
POWEROFF, 5-74

Overwriting data, 3-57

P

Part numbers
maintenance port cable for a terminal
connection, 1-12
parts used in configuring the controller, 1-10
storage subsystem basic building blocks, 1-4

PARTITION
ADD UNIT, 5-30
SET *unit-number*, 5-131

Partitioning
disk drives, 4-10
storage sets using CLI, 4-10

Partitions
creating, 5-48
defining, 3-40
deleting unit, 5-55
displaying size, 5-136

- guidelines, 3-41
- maximum supported, 1-5, 2-2
- planning, 3-40
- setting size, 5-48
- showing, 5-136
- Parts
 - AC input module, 1-4
 - BA370 rack-mountable enclosure, 1-4
 - cache module, 1-4
 - controller, 1-4
 - cooling fan, 1-4
 - disk drives, 1-4
 - dual-battery ECB, 1-4
 - ECB, 1-4
 - EMU, 1-4
 - I/O module, 1-4
 - power cable, 1-4
 - power cable kit, 1-4
 - power supply, 1-4
 - PVA module, 1-4
 - SCSI hub, 3-port, 1-4
 - SCSI hub, 5-port, 1-4
 - SCSI hub, 9-port, 1-4
 - single-battery ECB, 1-4
- Passthrough
 - adding to configuration, 5-16
- Passthrough containers
 - deleting, 5-51
 - renaming, 5-81
- Passthrough devices
 - showing, 5-137
- Passthroughs
 - SHOW *device-type*, 5-137
- Path
 - preferring for storage units in dual-redundant configurations, 3-32
- PC connection
 - optional maintenance port cable, 1-12
 - part number for the optional maintenance port cable, 1-12
 - See also* optional maintenance port cable
- PCMCIA program card. *See* Program card
- Performance, 3-15
- Planning
 - mirrorsets, 3-13
 - overview, 3-5
 - partitions, 3-40
 - RAIDsets, 3-16
 - storage sets, 3-8
 - striped mirrorsets, 3-19
 - stripesets, 3-10
- POLICY, 3-45
 - ADD MIRRORSET, 5-12
 - ADD RAIDSET, 5-20
 - MIRROR, 5-71
 - RAIDset switches, 3-43
 - SET *mirrorset-name*, 5-115
 - SET *RAIDset-name*, 5-124
- PORT_1_ID
 - SET *controller*, 5-96
- PORT_1_PREFERRED_ID
 - SET *controller*, 5-98
- PORT_1_TRANSFER_RATE_REQUESTED
 - SET *controller*, 5-100
- PORT_2_ID
 - SET *controller*, 5-96
- PORT_2_PREFERRED_ID
 - SET *controller*, 5-98
- PORT_2_TRANSFER_RATE_REQUESTED
 - SET *controller*, 5-100
- Ports
 - number on controller, 1-5

- See also* Device ports, Host ports
 - Power cable
 - parts, 1-4
 - Power cable kit
 - parts, 1-4
 - Power source
 - enabling write-back caching, 1-22
 - Power supply
 - parts, 1-4
 - Power, verification, and addressing module. *See*
 - PVA module
 - POWEROFF, 5-74
 - BATTERY_ON, 5-74
 - BATTERY_OFF, 5-74
 - NO_OVERRIDE_BAD_FLUSH, 5-74
 - VERRIDE_BAD_FLUSH, 5-74
 - SECONDS, 5-75
 - Precautions
 - electrostatic discharge, xvi
 - maintenance port, xvii
 - VHDCI cables, xvii
 - Preferred paths for storage units, 3-32
 - PREFERRED_PATH
 - ADD UNIT, 5-30
 - PREFERRRED_PATH
 - SET *unit-number*, 5-131
 - Profiles
 - creating, 3-5
 - description, 3-5
 - device, A-2
 - storage set, A-3
 - Program card
 - ESD cover, 1-10
 - location, 1-8
 - software upgrades, 1-5
 - PROMPT
 - SET *controller*, 5-98
 - Protocol
 - device, 1-5
 - host, 1-5
 - PTL addressing convention, 3-36
 - Publications, related, xxi
 - PVA ID, 2-25
 - PVA module
 - parts, 1-4
 - setting the switch, 2-25
 - R**
 - RAID levels
 - supported, 1-5
 - RAID-5 and RAID-1 storage sets
 - maximum number, 1-5, 2-2
 - RAID-5 storage sets
 - maximum number, 1-5, 2-2
 - maximum number of members, 1-5, 2-2
 - RAID-5, RAID-1, and RAID-0 storage sets
 - maximum number, 1-5, 2-2
 - RAIDset
 - showing, 5-136
 - specifying chunksize, 5-63
 - RAIDset switches, 3-43
 - membership, 3-44
 - NOREDUCED, 3-44
 - POLICY, 3-43
 - RECONSTRUCT, 3-43
 - reconstruction policy, 3-43
 - REDUCED, 3-44
 - replacement policy, 3-43
 - RAIDsets
 - adding to configuration, 5-19
 - adding while missing a member, 5-21
-

- changing characteristics, 5–124
- changing switches, 4–17
- choosing chunk size, 3–51
- configuring using CLI, 4–6
- deleting, 5–51
- description, 3–2, 3–15
- displaying information, 5–136
- initializing, 5–62
- maximum chunk size, 3–54
- maximum membership, 3–17
- planning, 3–16
- removing a member, 5–125
- renaming, 5–81
- replacing a member, 5–126
- specifying replacement policy, 5–124
- switches, 3–43
- Read caching
 - enabled for all storage units, 1–20
 - general description, 1–20
- READ_CACHE
 - ADD UNIT, 5–31
 - SET *unit-number*, 5–132
- READ_SOURCE
 - ADD MIRRORSET, 5–13
 - mirrorset switches, 3–46
 - SET *mirrorset-name*, 5–116
- Read-ahead caching
 - enabled for all disk units, 1–21
 - general description, 1–20
- READAHEAD_CACHE
 - ADD UNIT, 5–31
 - SET *unit-number*, 5–133
- RECONSTRUCT
 - ADD RAIDSET, 5–20
 - RAIDset switches, 3–43
 - SET *RAIDset-name*, 5–125
- REDUCE, 5–77
- REDUCE *disk-nameN*, 5–78
- REDUCED
 - ADD RAIDSET, 5–21
- Related publications, xxi
- Relationship
 - controller to cache module, 1–13
- Release lever
 - location, 1–8
- REMOTE, 3–50
 - ADD DISK, 5–8
 - SET *device-name*, 5–102
- REMOVE
 - SET *mirrorset-name*, 5–114
 - SET *RAIDset-name*, 5–125
- Removing
 - disk drives from sparesets, 4–14
- Removing a mirrorset member, 5–77
- RENAME, 5–81
- RENAME *old-container-name new-container-name*, 5–81
- REPLACE
 - SET *mirrorset-name*, 5–114
- Replacement policy
 - mirrorsets, 3–45
- REPLACESET *RAIDset-name*, 5–126
- Request rate, 3–52
- Required tools, xx
- Resetting configuration, 5–44
- RESTART *controller*, 5–83
- RESTART *controller*
 - IGNORE_ERRORS, 5–83
 - IMMEDIATE_SHUTDOWN, 5–84
 - NOIGNORE_ERRORS, 5–83
 - NOIMMEDIATE_SHUTDOWN, 5–84

Restoring configuration, 5-45
RETRY_ERRORS *unit-number*
 UNWRITEABLE_DATA, 5-85
Revision history, xxii
RUN, 5-86
 ADD UNIT, 5-32
 CHVSN, 5-86
 CLCP, 5-86
 CLONE, 5-86
 CONFIG, 5-86
 DILX, 5-86
 DIRECT, 5-87
 DSTAT, 5-87
 FMU, 5-87
 FRUTIL, 5-87
 HSUTIL, 5-87
 SET *unit-number*, 5-133
 VTDPY, 5-87
RUN *program name*, 5-86

S

Save configuration, 3-55
SAVE_CONFIGURATION, 3-55
 INITIALIZE, 5-64
Saving configurations, 5-47
 dual-redundant configurations, 3-56
SCSI device ports. *See* Device ports
SCSI device targets. *See* Devices
SCSI hub, 3-port
 parts, 1-4
SCSI hub, 5-port
 parts, 1-4
SCSI hub, 9-port
 parts, 1-4

SCSI target ID numbers. *See* Target ID numbers
SCSI_VERSION
 SET *controller*, 5-99
SECONDS
 POWEROFF, 5-75
SECTORS_PER_TRACK
 CREATE_PARTITION, 5-49
 INITIALIZE, 5-63
SELFTEST *controller*, 5-89
 IGNORE_ERRORS, 5-89
 IMMEDIATE_SHUTDOWN, 5-90
 NOIGNORE_ERRORS, 5-89
 NOIMMEDIATE_SHUTDOWN, 5-90
SENSOR_N_SETPOINT
 SET EMU, 5-105
SET *controller*, 5-91
 ALLOCATION_CLASS, 5-93
 CACHE_FLUSH_TIMER, 5-93
 CACHE_UPS, 5-93
 COMMAND_CONSOLE_LUN, 5-94
 HOST_FUNCTION, 5-94
 MIRRORED_CACHE, 5-96
 NOCACHE_UPS, 5-93
 NOCOMMAND_CONSOLE_LUN, 5-94
 NODE_ID, 5-95
 NOMIRRORED_CACHE, 5-96
 NOPORT_1_ID, 5-96
 NOPORT_1_PREFERRED_ID, 5-98
 NOPORT_1_TRANSFER_RATE_REQUESTED, 5-100
 NOPORT_2_ID, 5-96
 NOPORT_2_PREFERRED_ID, 5-98
 NOPORT_2_TRANSFER_RATE_REQUESTED, 5-100
 NOTERMINAL_PARITY, 5-99
 PORT_1_ID, 5-96

PORT_1_PREFERRED_ID, 5-98
 PORT_1_TRANSFER_RATE_REQUESTED, 5-100
 PORT_2_ID, 5-96
 PORT_2_PREFERRED_ID, 5-98
 PORT_2_TRANSFER_RATE_REQUESTED, 5-100
 PROMPT, 5-98
 SCSI_VERSION, 5-99
 TERMINAL_PARITY, 5-99
 TERMINAL_SPEED, 5-99
 TIME, 5-99
 TRANSFER_RATE_REQUESTED, 5-100
 SET *device-name*, 5-102
 LOCAL, 5-102
 NOTTRANSPORTABLE, 5-103
 REMOTE, 5-102
 TRANSFER_RATE_REQUESTED, 5-103
 TRANSPORTABLE, 5-103
 SET EMU, 5-105
 FANSPEED, 5-106
 SENSOR_N_SETPOINT, 5-105
 SET FAILEDSET, 5-108
 AUTOSPARE, 5-108
 NOAUTOSPARE, 5-108
 SET FAILOVER, 5-110
 SET FAILOVER COPY=*controller*, 5-110
 SET *mirrorset-name*, 5-112
 COPY, 5-112
 DT_SUPPORT, 5-113
 MEMBERSHIP, 5-113
 NODT_SUPPORT, 5-113
 POLICY, 5-115
 READ_SOURCE, 5-116
 REMOVE, 5-114
 REPLACE, 5-114
 SET MULTIBUS_FAILOVER, 5-118
 SET NOFAILOVER, 5-120
 DESTROY_UNFLUSHABLE_DATA, 5-120
 NODESTROY_UNFLUSHABLE_DATA, 5-120
 SET NOMULTIBUS_FAILOVER, 5-122
 DESTROY_UNFLUSHABLE_DATA, 5-122
 NODESTROY_UNFLUSHABLE_DATA, 5-122
 SET *RAIDset-name*, 5-124
 POLICY, 5-124
 RECONSTRUCT, 5-125
 REMOVE, 5-125
 REPLACE, 5-126
 SET *unit-number*, 5-128
 DISABLE_ACCESS, 5-130
 ENABLE_ACCESS, 5-130
 ERROR_MODE, 5-132
 MAXIMUM_CACHED_TRANSFER, 5-131
 NOPREFERRED_PATH, 5-131
 NOREAD_CACHE, 5-132
 NOREADAHEAD_CACHE, 5-133
 NORUN, 5-133
 NOWRITE_PROTECT, 5-133
 NOWRITEBACK_CACHE, 5-134
 PARTITION, 5-131
 PREFERRED_PATH, 5-131
 READ_CACHE, 5-132
 READAHEAD_CACHE, 5-133
 RUN, 5-133
 WRITE_PROTECT, 5-133
 WRITEBACK_CACHE, 5-134
 Setting
 cache flush timer, 5-93
 CLI prompt, 5-98
 control of metadata, 5-63 to 5-64

- controller behavior at restart, 5-84
 - controller behavior at shutdown, 5-144
 - controller behavior selftest, 5-90
 - controller cache flush timer, 5-93
 - controller cache UPS policy, 5-93
 - controller configuration handling, 5-64
 - controller error handling at selftest, 5-84, 5-89
 - controller error handling at shutdown, 5-143
 - data retention policy, 5-36
 - device data transfer rate, 5-9, 5-103
 - failedset autospare feature, 5-108
 - fan speed, 5-106
 - full display, 5-138
 - mirrorset copy data, 5-71
 - mirrorset copy speed, 5-11, 5-71, 5-112 to 5-113
 - mirrorset member read source, 5-13, 5-116
 - mirrorset read source, 5-13, 5-116
 - mirrorset spareset replacement policy, 5-12, 5-71, 5-115
 - nofailover cached data policy, 5-120
 - number of blocks cached by controller, 5-30, 5-131
 - number of mirrorset members, 5-113
 - number of unit partitions, 5-30, 5-131
 - partition size, 5-36, 5-48
 - RAIDset member reconstruct policy, 5-20, 5-125
 - RAIDset member replacement policy, 5-13, 5-20, 5-116
 - read cache for units, 5-31
 - storageset chunksize, 5-63
 - subsystem temperature sensor setpoint, 5-105
 - terminal parity, 5-99
 - terminal speed, 5-99
 - time, 5-99
 - transportability of devices, 5-103
 - transportability of disks, 5-8, 5-103
 - unit availability to the host, 5-32, 5-133
 - write protect for units, 5-32
 - write-back cache for units, 5-32, 5-134
 - SHOW, 5-136
 - FULL, 5-138
 - SHOW *controller*, 5-136
 - SHOW *device-name*, 5-136
 - SHOW *device-type*, 5-137
 - DEVICES, 5-137
 - DISKS, 5-137
 - PASSTHROUGH, 5-137
 - SHOW EMU, 5-137
 - SHOW *storageset-name*, 5-137
 - SHOW *storageset-type*, 5-137
 - FAILEDSET, 5-137
 - MIRRORSETS, 5-137
 - RAIDSETS, 5-137
 - SPARESETS, 5-137
 - STORAGESETS, 5-137
 - STRIPESETS, 5-137
 - SHOW *unit-number*, 5-137
 - SHOW UNITS, 5-138
 - SHUTDOWN *controller*, 5-143
 - IGNORE_ERRORS, 5-143
 - IMMEDIATE_SHUTDOWN, 5-144
 - NOIGNORE_ERRORS, 5-143
 - NOIMMEDIATE_SHUTDOWN, 5-144
 - Single configuration
 - ECB, 1-29
 - Single-battery ECB
 - parts, 1-4
 - Single-controller configuration
 - connecting to the host, 2-15
-

- Single-disk units
 - backing up, 3–20
 - configuring using CLI, 4–9
 - displaying switches, 4–17
- SIZE
 - CREATE_PARTITION, 5–48
- Spareset
 - removing a disk drive, 5–54
- Sparesets
 - adding disk drives using CLI, 4–13
 - adding to configuration, 5–23
 - AUTOSPARE, 4–15
 - removing disk drives using CLI, 4–14
- Specify device addresses, ?? to 3–37
- Speed. *See* transfer rate
- Spontaneous events
 - setting display characteristics with FMU, 1–15
- Storage requirements, determining, 3–7
- Storage subsystem
 - typical installation, 1–2
- Storageset map, 3–34
- Storageset profile, 3–5, A–3
- Storageset switches, 3–42
 - changing switches, 3–42
 - enabling switches, 3–42
- Storagesets
 - adding devices with the CONFIG utility, 1–16
 - attributes, 3–8
 - backing up, 3–20
 - backing up data with the Clone utility, 1–16
 - changing switches, 4–17
 - comparison, 3–8
 - configuring using CLI, 4–2
 - creating a profile, 3–5
 - creating map, 3–34
 - deleting, 4–16
 - displaying information, 5–136
 - displaying switches, 4–17
 - dividing, 3–40
 - duplicating data with the Clone utility, 1–16
 - generating a new volume serial number with the CHVSN utility, 1–17
 - how they work with the host, 1–7
 - initializing, 5–62
 - largest device supported, 1–6, 2–2
 - locating, 5–67
 - maximum number of partitions supported, 1–5, 2–2
 - mirrorsets, 3–2, 3–12
 - moving, 4–35
 - partitioning using CLI, 4–10
 - planning, 3–8
 - RAIDsets, 3–2
 - renaming, 5–81
 - renaming the volume serial number with the CHVSN utility, 1–17
 - showing, 5–136
 - striped mirrorsets, 3–2
 - stripesets, 3–2, 3–9
 - See also* Configuration rules
- StorageWorks array controller, 5–2
- Striped mirrorsets
 - configuring using CLI, 4–7
 - description, 3–2, 3–18
 - maximum number of physical devices, 1–6, 2–2
 - planning, 3–19
- Stripesets
 - adding to configuration, 5–24
 - configuring using CLI, 4–3
 - deleting, 5–51

- description, 3–2, 3–9
- displaying information, 5–136
- initializing, 5–62
- maximum number of members, 1–6, 2–2
- mirroring, 5–70
- planning, 3–10
- renaming, 5–81
- showing, 5–136
- specifying chunksize, 5–63

Subsystem

- addressing with the PVA module, 2–25
- connecting a single controller to the host, 2–9
- connecting dual-redundant controllers to the host, 2–15
- connecting multiple-bus failover, dual-redundant controllers to the host, 2–21
- illustration of SCSI target ID numbers and PVA settings, 2–25
- saving configuration, 3–55

Switches

- changing for devices, 4–17
- changing for storage sets, 4–17
- changing initialize, 4–18
- changing mirrorset, 4–17
- changing RAIDset, 4–17
- changing unit, 4–18
- displaying current, 4–17
- NOTTRANSPORTABLE, 3–48
- overview, 3–42
- RAIDset, 3–43
- TRANSFER_RATE_REQUESTED, 3–49
- TRANSPORTABLE, 3–48

T

Target ID numbers

- illustration of SCSI target ID numbers and PVA settings, 2–25
- on the device bus, 2–34
- on the host bus, 2–33
- setting, 2–33
- using preferred ID numbers, 2–35
- See also* PTL addressing convention

Targets. *See* Devices

Technical support, xv

Terminal

- setting parity, 5–99
- setting speed, 5–99

Terminal connection

- optional maintenance port cable, 1–12, 2–26
- part number for the optional maintenance port, 1–12
- See also* Maintenance port, Maintenance port cable

Terminal display. *See* VTDPY

Terminal. *See* Maintenance port

TERMINAL_PARITY

- SET controller, 5–99

TERMINAL_SPEED

- SET controller, 5–99

Testing controllers, 5–89

This controller

- explained, 5–2
- removing from dual-redundant controller configuration, 5–120, 5–122
- starting diagnostic or utility programs, 5–86

- This controller defined, 2–28
- This controller, defined, xviii
- Tightening VHDCI cables, xvii
- TIME
 - SET *controller*, 5–99
- Tip, defined, xix
- Tools, xx
- Topology
 - supported, 1–5
- Transfer rate
 - how chunk size affects, 3–51
 - setting device, 5–9, 5–103
 - switch, 3–49
- TRANSFER_RATE_REQUESTED, 3–49
 - ADD DISK, 5–9
 - ADD PASSTHROUGH, 5–17
 - SET *controller*, 5–100
 - SET *device-name*, 5–103
- Translating messages
 - FMU, 1–15
- Transparent failover mode
 - configuring units with multiple hosts, 4–26
 - designating the host SCSI ID number, 4–28
 - designating the port, 4–26
 - designating the port and host SCSI ID number, 4–30
 - general description, 2–29
 - in configuring units with multiple hosts, 4–26
 - See also* Multiple-bus failover mode
- Transportability, 3–48
- TRANSPORTABLE, 3–48
 - ADD DISK, 5–8
 - SET *device-name*, 5–103
- Troubleshooting
 - backing up data with the Clone utility, 1–16
 - CLCP utility, 1–16
 - communication between controller and devices with VTDPY, 1–15
 - communication between the controller and hosts with VTDPY, 1–15
 - DILX, 1–16
 - displaying last failures with FMU, 1–15
 - displaying memory-system failures with FMU, 1–15
 - FMU, 1–15
 - generating a new volume serial number with the CHVSN utility, 1–17
 - generating read and write loads with DILX, 1–16
 - investigating data transfer with DILX, 1–16
 - investigating I/O activity of units with VTDPY, 1–15
 - logging I/O activity with DSTAT, 1–17
 - monitoring performance with DILX, 1–16
 - patching controller software with the CLCP utility, 1–16
 - renaming the volume serial number with the CHVSN utility, 1–17
 - replacing a failed controller with FRUTIL, 1–17
 - replacing cache modules with FRUTIL, 1–17
 - replacing ECBs with FRUTIL, 1–17
 - setting display characteristics of events and failures with FMU, 1–15
 - testing the controller and disk drives with DILX, 1–16
 - translating event messages with FMU, 1–15
 - upgrading controller software with the CLCP utility, 1–16
 - upgrading EMU software with the CLCP utility, 1–16
 - VTDPY, 1–15

See also Config utility

See also HSUTIL

Troubleshooting and maintaining the controller
utilities and exercisers, 1–15

Typographical conventions, xviii

U

Unit switches

changing, 4–18

overview, 3–58

Units

adding to configuration, 5–27

changing characteristics, 5–128

clearing lost data error, 5–40

configuring with multiple hosts, 4–23

deleting from the configuration, 5–55

displaying configured units, 5–138

displaying information, 5–136

heterogeneous host support, 4–23

largest unit supported, 1–6, 2–2

maximum number presented to the host, 1–5

mirroring, 5–70

showing, 5–136

UNMIRROR, 5–145

UNMIRROR *disk-name*, 5–145

Unpartitioned mirrorsets

duplicating data with the Clone utility, 1–16

Unwriteable data error, retrying, 5–85

Upgrading

controller software with the CLCP utility,
1–16

EMU software with the CLCP utility, 1–16

firmware with HSUTIL, 1–16

Utilities

CHVSN, 5–86

CLCP, 5–86

CLONE, 5–86

CONFIG, 5–86

DILX, 5–86

DIRECT, 5–87

DSTAT, 5–87

FMU, 5–87

FRUTIL, 5–87

HSUTIL, 5–87

listing of, 5–59

running, 5–86

VTDPY, 5–87

Utilities and exercisers

CHVSN utility, 1–17

CLCP utility, 1–16

Clone utility, 1–16

CONFIG utility, 1–16

DILX, 1–16

DSTAT, 1–17

FMU, 1–15

FRUTIL, 1–17

HSUTIL, 1–16

VTDPY, 1–15

V

VHDCI cable precautions, xvii

Virtual terminal display, 1–15

Volume serial number

generating a new one with the CHVSN utility,
1–17

renaming with the CHVSN utility, 1–17

VTDPY

general description, 1–15

VTDPY, running, 5-87

W

Write performance, 3-54

WRITE_PROTECT

ADD UNIT, 5-32

SET *unit-number*, 5-133

Write-back caching

enabled for all disk units, 1-21

fault-tolerance, 1-22

general description, 1-21

not enabled in disaster-tolerant mirrorsets,
1-21

setting the flush timer, 5-93

WRITEBACK_CACHE

ADD UNIT, 5-32

SET *unit-number*, 5-134

Write-through caching

enabling and disabling, 1-21

general description, 1-21

