DIGITAL StorageWorks

HSG80 Array Controller ACS Version 8.0 User’s Guide

Part Number: EK-HSG80-UG. A01

January 1998

Revision/Update Information: This is a new document.

User’s Guide

Digital Equipment Corporation
Maynard, Massachusetts

January 1998

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.

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

Contents

Figures

Tables

Preface

Chapter 1 General Description

Chapter 2 Configuring an HSG80 Array Controller

Chapter 3 Creating Storagesets

Chapter 4 Troubleshooting

Chapter 5 Replacement Procedures

Chapter 6 Upgrading the Subsystem

Appendix A System Profiles

Appendix B CLI Commands

Appendix C LED Codes

Appendix D Event Codes

Appendix E Controller Specifications

Glossary

Index

Figures

Tables

Preface

This book describes the features of the HSG80 array controller and configuration procedures for the controller and storagesets running Array Controller Software (ACS) 8.0.

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

Precautions

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

Electrostatic Discharge Precautions

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

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

Local-Connection Port Precautions

The local-connection 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 BOLD	Command syntax that must be entered exactly as shown, for example: **SET FAILOVER COPY=OTHER_CONTROLLER**
ALLCAPS	Command discussed within text, for example: "Use the SHOW SPARESET command to show the contents of the spareset."
`Monospaced`	Screen display.
Sans serif italic	Command variable or numeric value that you supply, for example: SHOW *RAIDset-name* or **SET THIS_CONTROLLER ID=***(n,n,n,n,)*
italic	Reference to other books, 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.

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

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

Required Tools

You will need the following tools to service the controller, cache module, and external cache battery:

- A small screwdriver for loosening and tightening the cable-retaining 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 provide more efficient service.

Related Publications

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

Document Title	Part Number
Fibre Channel Arbitrated Loop Hub (DS-DHGGA-CA) User's Guide	EK-DHGGA-UG
KGPSA PCI-to-Fibre Channel Host Adapter	EK-KGPSA-UG.A01
DIGITAL StorageWorks Ultra SCSI RAID Enclosure (BA370-Series) User's Guide	EK-BA370-UG
The RAIDBOOK—A Source for RAID Technology	RAID Advisory Board
DIGITAL StorageWorks HSG80 Array Controller ACS V8.0 Release Notes for WindowsNT Intel	AA-R8PWA-TE
DIGITAL StorageWorks HSG80 Array Controller ACS V8.0 Release Notes for WindowsNT Alpha	AA-R93ZA-TE
Getting Started RAID Array 7000 Fibre Channel	AA-R8Q5A-TE
For additional publications and sources of information see the DIGITAL StorageWorks HSG80 Array Controller ACS V8.0 Release Notes	—

Revision History

This is a new document.

CHAPTER 1

General Description

This chapter illustrates and describes in general terms your subsystem and its major components: the HSG80 array controller, its cache module, and its external cache battery. See the *Fibre Channel Arbitrated Loop Hub User's Guide* and *KGPSA PCI-to-Fibre Channel Host Adapter User Guide* for information about the fibre channel arbitrated loop hub and adapter that connect the subsystem to your host.

The HSG80 Subsystem

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

- One pedestal enclosure
- Two controllers, each supported by their own cache module
- One external cache battery (ECB), which provides 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

Figure 1–1 The HSG80 Subsystem

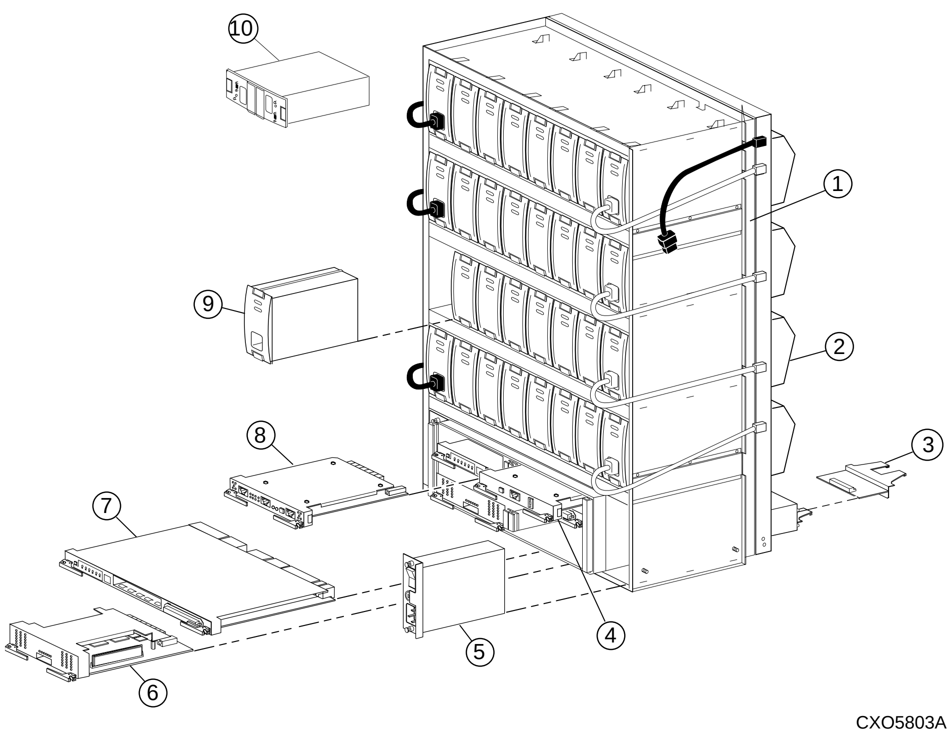


Table 1–1 Key to Figure 1–1 The HSG80 Subsystem

Item	Description	Part No.
1	Pedestal enclosure	—
2	Cooling fan	DS–BA35X–MK
3	I/O module	70–32876–01
4	Power Verification and Addressing(PVA) module	—
5	AC input module	DS–BA35X–HE
6	Cache module	70–33256–01
7	HSG80 controller	70-33259-01
8	Environmental monitoring unit (EMU)	70-32866-01
9	180-watt power supply	DS–BA35X–HH
10	External cache battery (ECB)	DS–HS35X–BD

Summary of HSG80 Features

Table 1–2 summarizes the features of the controller.

Table 1–2 Controller Features

Feature	Supported
Topology	FC-AL 2 nodes per loop single host adapter
Host protocol	FC-AL
Host bus interconnect	Two Fibre Channel ports, copper gigabit link module (GLM)
Device protocol	SCSI–2
Device bus interconnect	Fast Wide Ultra Single-ended
Number of SCSI device ports	6
Number of SCSI device targets per port	4

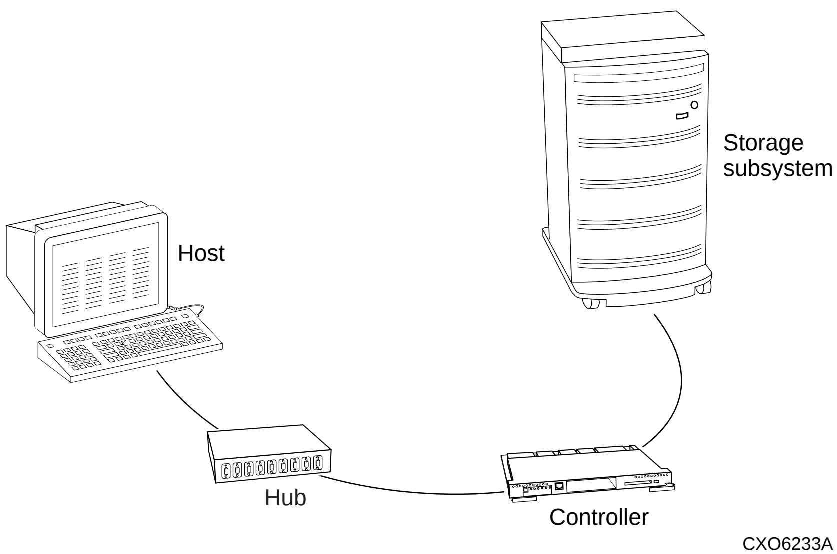
Table 1–2 Controller Features (Continued)

Feature	Supported
Maximum number of SCSI devices	24
RAID levels supported	0, 1, 0+1, 3/5
Cache size	128 MB (four 32 MB DIMMs)
Program card updates	Yes
Device warm swap	No
Exercisers for testing disks	Yes
Number of configuration entities (devices + storagesets + partitions + units)	191
Maximum number of RAID 1 storagesets	12
Maximum number of RAID 5 storagesets	8
Maximum number of RAID 5, RAID 1, and RAID 0 storagesets	45
Maximum number of partitions per storageset or disk drive	8
Maximum number of units presented to host	16 (8 on each of 2 ports) This is a driver limitation.
Maximum number of devices per unit	24
Serial interconnect	1 GB/second
Largest device, storageset, or unit size	512 GB

The HSG80 Controller

Your controller is the intelligent bridge between your host and the devices in your subsystem. It bridges the gap between the host and its storage subsystem as Figure 1–2 illustrates.

Figure 1–2 A Host and Its Storage Subsystem



The controller shown in Figure 1–3 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 product-specific release notes that accompanied the software release 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, “Creating Storagesets,” for more information about storagesets and how to configure them.

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 Fibre Channel 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 HSG80 Array Controller

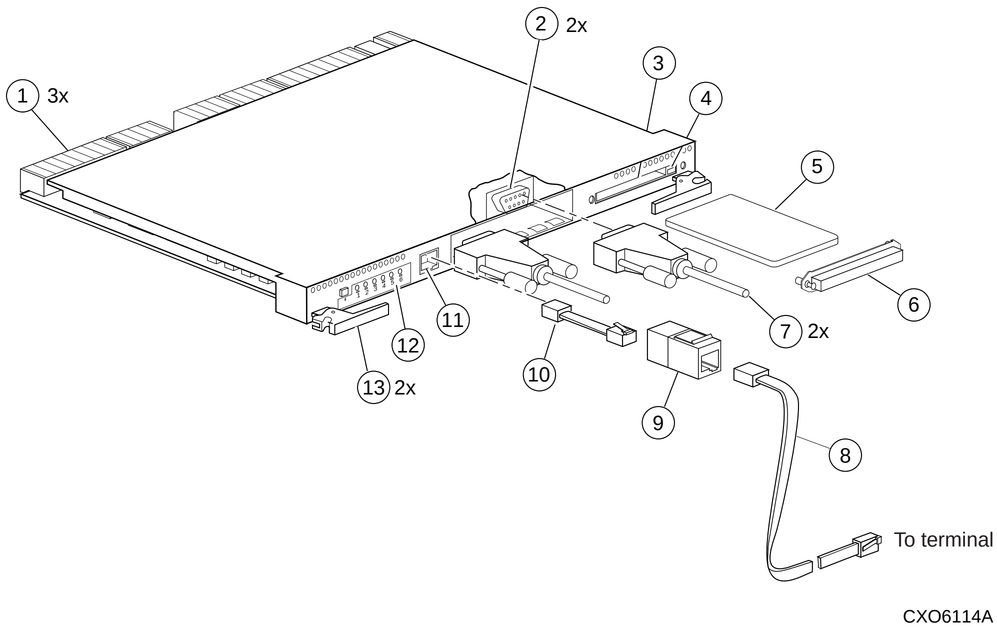


Table 1–3 Key to Figure 1–3 HSG80 Array Controller

Item	Description	Part No.
1	Backplane connectors	—
2	Host ports	—
3	Program-card slot	—
4	Program-card ejection button	—
5	Program card	BG-R8Q3A-BA
6	ESD/PCMCIA card cover	74-52628-01
7	5-meter Fibre Channel cables or 10-meter Fibre Channel cables	17-04718-06 17-04718-07

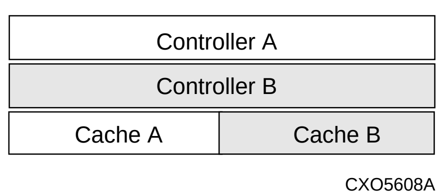
Table 1–3 Key to Figure 1–3 HSG80 Array Controller (Continued)

Item	Description	Part No.
8	Maintenance Port Cable	17-04718-06
9	RJ-11 adapter	12–43346–01
10	DECconnect office cable	BC16E–XX
11	Local connection port	—
12	Operator control panel (OCP)	—
13	Lever for removing, installing, and retaining controller module.	—

The components that you'll use most often are conveniently located on the controller's front panel, such as the local connection 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.

Each controller is supported by its own cache module. Figure 1–4 shows which cache module supports which controller in a dual-redundant configuration in a BA370 rack-mountable enclosure.

Figure 1–4 Location of Controllers and Cache Modules



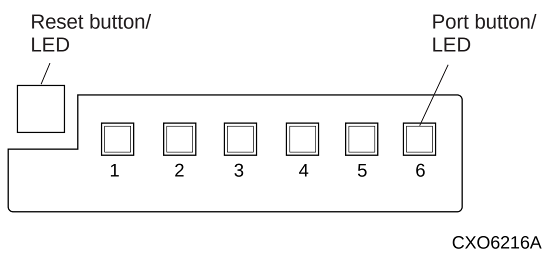
Tip DIGITAL recommends that you use the slots for controller "A" and cache module "A" for single configurations. Slot "A" responds to SCSI target ID number 7; slot "B" responds to SCSI target ID number 6.

Operator Control Panel

The operator control panel (OCP) contains a reset button and six device-port LED quiesce buttons as shown in Figure 1–5. The reset button flashes about once per second to indicate that the controller is operating normally. The 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.

You can also use the six port-quiesce buttons to turn off the controller's device ports. To quiesce a device port, push its port button until its LED is continuously lit. Push the button again to resume I/O activity to the port.

Figure 1–5 HSG80 Controller Operator Control Panel (OCP)



See Figure 1–3 on page 1–6 for the location of the OCP. See "Operator Control Panel LED Codes," page C–2, for an explanation of the LED codes that may appear on the OCP.

Under normal circumstances, you will not need to remove the controller from its cabinet. For this reason, the components that you will use most often are conveniently located on the front panel. For example, the local-connection port provides a convenient way to connect a 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 LED lights on the control panel will flash in a pattern that will help you to diagnose the problem. See Chapter 4, "Troubleshooting," for details about troubleshooting your controller.

Local Connection Port

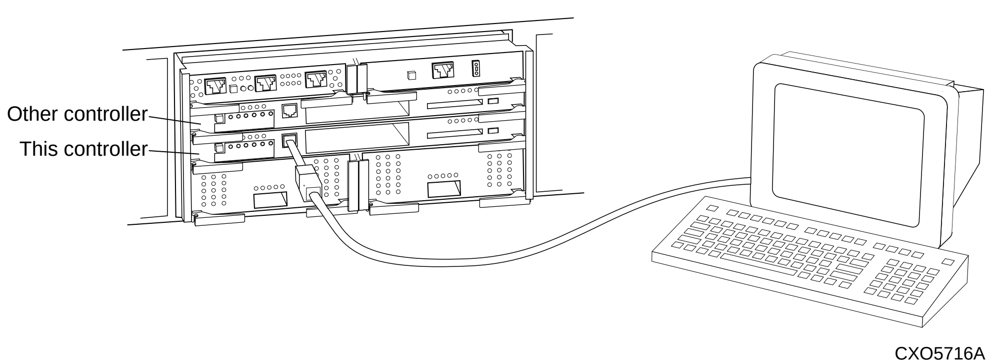
You can access the controller in two ways: through a local terminal via the local connection port, or through a remote terminal—sometimes called a virtual terminal or host console—via the host. DIGITAL recommends that you use a local terminal to carry out the troubleshooting and servicing procedures in this manual.

The local connection port provides a convenient way to connect a 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 port supports serial communications up to 19200 baud using 8 data bits, 1 stop bit, and no parity.

To connect a local terminal to a controller:

1. Turn off the terminal, and connect it to the controller via the DECconnect office cable shown in Figure 1–3 on page 1–6. When you're entering CLI commands in a dual-redundant configuration, remember that the controller you're connected to is “this controller” and the other controller is the “other controller.” This relationship is shown in Figure 1–6.
2. Turn on the terminal and configure it for 19200 baud, 8 data bits, 1 stop bit, and no parity.
3. Press Enter (or Return) until the CLI prompt appears.

Figure 1–6 “This Controller” and “Other Controller”



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:

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 local terminal or host.

See "Checking Failure Entries," page 4–10, for more information about using this utility.

Virtual Terminal Display

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

- Communication between the controller and its host
- 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

See "Using VTDPY to Check for Communication Problems," page 4–16, for more information about using this utility.

Disk Inline Exerciser

Use the disk inline exerciser (DILX) to investigate the data-transfer capabilities of disk drives. The disk inline exerciser (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. See "Checking for Disk-Drive Problems,” page 4–27, for more information about this exerciser.

Configuration Utility

Use the configuration utility (CONFIG) 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. See “Adding Several Disk Drives at a Time,” page 3–44, for more information about using the CONFIG utility.

HSUTIL

Use HSUTIL to upgrade the firmware on disk drives in the subsystem. See “Upgrading Firmware on a Device,” page 6–6, for more information about this utility.

Code Load and Code Patch Utility

Use Code Load/Code Patch utility (CLCP) to upgrade and patch the controller’s software. See “Upgrading Controller Software,” page 6–2, for more information about using this utility.

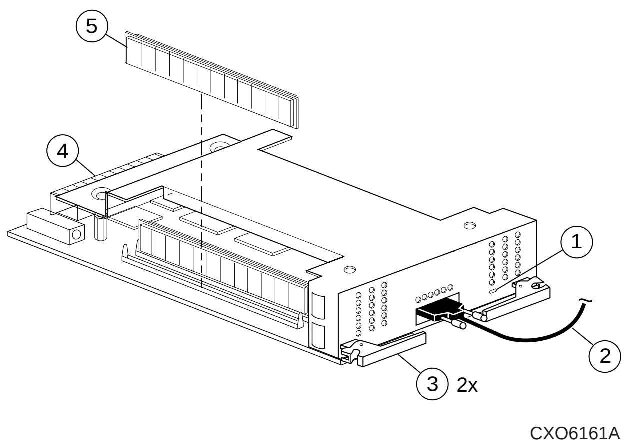
Clone Utility

Use the Clone utility to duplicate the data on any unpartitioned single-disk unit, stripeset, or mirrorset. Backup the cloned data while the actual storageset remains online. See “Cloning Data for Backup,” page 6–12, for more information about using the Clone utility.

Cache Module

The controller requires a companion cache module as shown in Figure 1–7. (Figure 1–4 on page 1–7 shows the location of a controller's companion cache module.) The cache module, which contains 128 MB of memory, increases the subsystem's I/O performance by providing read, write-through, and write-back caching. These caching techniques are described below.

Figure 1–7 Cache Module



Item	Description	Part No.
1	Cache-memory power LED button	—
2	ECB "Y" cable	17-04479-03
3	Retaining lever	—
4	Backplane connector	—
5	DIMM (4) (32 MB DIMM pack)	20-47083-D7

Caching Techniques

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

- read caching
- read-ahead caching
- write-through caching
- write-back caching

Read Caching

When the controller receives a read request from the host, it reads the data from the disk drives, delivers it to the host, and stores the data in its cache module. This process is called read caching.

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

See the SET unit command MAXIMUM_CACHED_TRANSFER switch in Appendix B, “CLI Commands,” for more details.

Read-Ahead Caching

Read-ahead caching begins when 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.

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

Fault-Tolerance for Write-Back Caching

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

Non-Volatile Memory

The controller can provide write-back caching for any storage unit as long as the controller's cache memory is non-volatile. In other words, to enable write-back caching, you must provide a back-up 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 back up 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 1–19, for more information about the ECB. However, if your subsystem is backed up by a UPS, you can tell the controller to use the UPS as the backup power source with the SET THIS CONTROLLER CACHE_UPS command. See Appendix B, "CLI Commands," for instructions on using this command.

Cache Policies Resulting from Cache Module Failures

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

Table 1–5 on page 1–17 shows the cache policies resulting from a full or partial failure of cache module A in a dual-redundant configuration. The consequences shown in this table are the same for cache module B.

Table 1–4 on page 1–16 shows the consequences of a full or partial failure of cache module A's ECB in a dual-redundant configuration. The consequences shown in this table are reciprocal for a failure of cache module B's ECB.

Table 1–4 Cache Policies Resulting from Cache Module Failures

Cache A	Cache B	Cache Policy
Good	Good	**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 multibit error occurred. Controller A detects and reports the lost blocks. **Cache policy:** Both controllers support write-back caching. **Failover:** No.
DIMM or cache memory controller failure	Good	**Data loss:** loss of write-back data that wasn’t written to media when failure occurred. **Cache policy:** Controller A supports write-through caching only; controller B supports write-back caching. **Failover:** In dual-redundant configuration failover, all units failover to controller B. In single 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 non-volatile memory on cache A.
Cache board failure	Good	Same as for DIMM failure.

Table 1–5 Cache Policies Resulting from ECB Failures

ECB Cache A	ECB Cache B	Cache Policy
Good	Good	**Data loss:** No. **Cache policy:** Both controllers continue to support write-back caching. **Failover:** No.
Low	Good	**Data loss:** No. **Cache policy:** Controller A supports write-through caching only; controller B supports write-back caching. **Failover:** In dual-redundant configuration failover, all units failover to controller B and operate normally. In single configurations, the controller only provides write-through caching to its units.
Failed	Good	**Data loss:** No. **Cache policy:** Controller A supports write-through caching only; controller B supports write-back caching. **Failover:** In dual-redundant configuration failover, all units failover to controller B and operate normally. In single configurations, RAIDsets and mirrorsets become inoperative.
Low	Low	**Data loss:** No. **Cache policy:** Both controllers support write-through caching only. **Failover:** No.

Table 1–5 Cache Policies Resulting from ECB Failures (Continued)

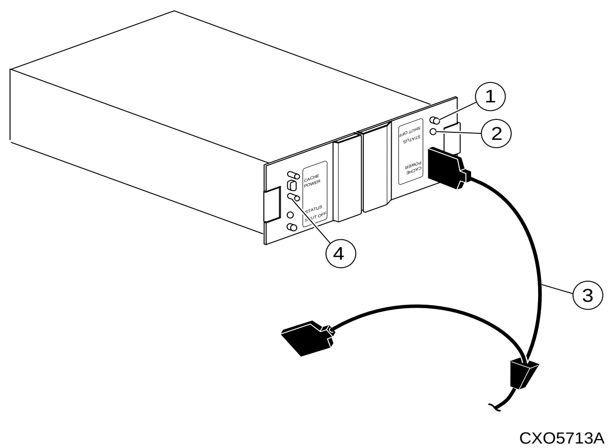
ECB Cache A	ECB Cache B	Cache Policy
Failed	Low	**Data loss:** No. **Cache policy:** Both controllers support write-through caching only. **Failover:** In dual-redundant configuration failover, all units failover to controller B and operate normally. In single configurations, RAIDsets and mirrorsets become inoperative.
Failed	Failed	**Data loss:** No. **Cache policy:** Both controllers support write-through caching only. **Failover:** No. RAIDsets and mirrorsets become inoperative. Other units that use write-back caching operate with write-through caching only.

External Cache Battery

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

DIGITAL supplies two versions of ECBs: a single-battery ECB for single configurations, and a dual-battery ECB for dual-redundant configurations, which is shown in Figure 1–8. When the batteries are fully charged, an ECB can preserve 128 MB of cache memory for 24 hours.

Figure 1–8 ECB for Dual-Redundant Configurations



Item	Description	Part No.
	ECB for dual-redundant configurations	DS–HS35X–BD
	ECB for single configurations	DS–HS35X–BC
1	Shut off button	—
2	Status LED	—
3	ECB cable	17-04479-03
4	VHDCI male port for second battery	—

Charging Diagnostics

Whenever you restart the controller, its diagnostic routines automatically check the charge in the ECB's batteries. If the batteries are fully charged, the controller reports them as good and rechecks them every 24 hours. If the batteries are low, the controller reports them as low and rechecks them every four minutes.

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

Caution DIGITAL recommends that you replace the ECB every two years to prevent battery failure.

Note If an uninterruptible power supply (UPS) is used for backup power, the controller does not check the battery. See Appendix B, “CLI Commands,” for information about the CACHE_UPS and NOCACHE_UPS commands.

CHAPTER 2

Configuring an HSG80 Array Controller

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

Introduction

Use the Getting Started 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. The procedure in this chapter for configuring your controller 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 “CLI Commands,” page –1.

Configuration Rules

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

- Maximum 200 assignable, host-visible LUNs
- Maximum 512 GB LUN capacity
- Maximum 24 physical devices
- Maximum 8 RAID-5 storagesets
- Maximum 12 RAID-1 storagesets
- Maximum 8 partitions of a storageset or individual disk
- Maximum 6 members per mirrorset
- Maximum 14 members per RAIDset
- Maximum 14 members per stripeset

Configuring a Controller

You can use this procedure to configure your controller in a single or dual controller configuration. Use the references in each step to locate details about the commands and concepts.

To configure a controller:

1. Press the power verification and addressing (PVA) module ID switch until it is set to SCSI ID number zero for the pedestal enclosure.
2. Remove the program card ESD cover and insert the controller's program card. Replace the ESD cover.
3. Turn on the power to the pedestal enclosure.
4. Establish a local connection to the controller.

   See "Establishing a Local Connection to the Controller," page 2–5, for details about creating a local connection.
5. Choose a single or failover configuration for the controller:
   a. If you are configuring a single controller, skip to step 10.
   b. If you are configuring dual- redundant controllers in transparent failover mode, proceed to step 9.
6. If the controller reports a node ID of all zeros (0000-0000-0000-0000) set the subsystem worldwide name (node ID) to the worldwide name that came with your subsystem. Use the steps in "Restoring Worldwide Names (Node ID Numbers)," page 3–17.
7. Set the port topology for each port.

   SET THIS_CONTROLLER PORT_1_TOPOLOGY="*topology*"

   SET THIS_CONTROLLER PORT_2_TOPOLOGY="*topology*"

   If this is a single configuration with a single hub, set PORT 2 off-line.

   If this is a dual-redundant configuration, the "other controller" inherits "this controller's" port topology.
   See Appendix B, "CLI Commands," for more information about using the SET THIS_CONTROLLER PORT_n_TOPOLOGY= command.
8. If you selected LOOP_HARD for the port topology, specify the arbitrated loop physical address (ALPA) for the host ports.

   SET THIS_CONTROLLER PORT_1_ALPA="*address*"

   SET THIS_CONTROLLER PORT_2_ALPA="*address*"

If this is a dual-redundant configuration, the “other controller” inherits “this controller’s” port ALPA addresses.

See Appendix B, “CLI Commands,” for more information about using the SET OTHER_CONTROLLER PORT_n_ALPA= command.

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

   SET FAILOVER COPY = THIS_CONTROLLER

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

   See details about failover modes in “Selecting a Failover Mode,” page 2–7.

10. Optional: Change the CLI prompt. Type the following command:

    SET THIS_CONTROLLER PROMPT = *“new prompt”*

    If you’re configuring dual-redundant controllers, also change the CLI prompt on the “other controller.” Use the following syntax:

    SET OTHER_CONTROLLER PROMPT = *“new prompt”*

    See Appendix B, “CLI Commands,” for more information about using the SET OTHER_CONTROLLER PROMPT= command.

11. Optional: Indicate that your subsystem power is supported by a UPS. Use the following syntax:

    SET THIS_CONTROLLER CACHE_UPS

    The “other controller” inherits “this controller’s” cache UPS setting. See “Backing up Power with a UPS,” page 2–8, for more information.

12. Restart the controller, using the following syntax:

    RESTART THIS_CONTROLLER

    If this is a dual-redundant configuration, restart the “other controller” using the following syntax:

    RESTART OTHER_CONTROLLER

    See the RESTART THIS_CONTROLLER command in Appendix B, “CLI Commands,” for more information about using this command.

13. When the CLI prompt reappears, it will display details about the controller you configured. Use the following syntax:

    SHOW THIS_CONTROLLER FULL

    See the SHOW THIS_CONTROLLER FULL command in Appendix B, “CLI Commands,” for more information about using this command.

14. Connect the controller to the host.

    See “Connecting the Subsystem to the Host,” page 2–8 for information about how to complete the connection.

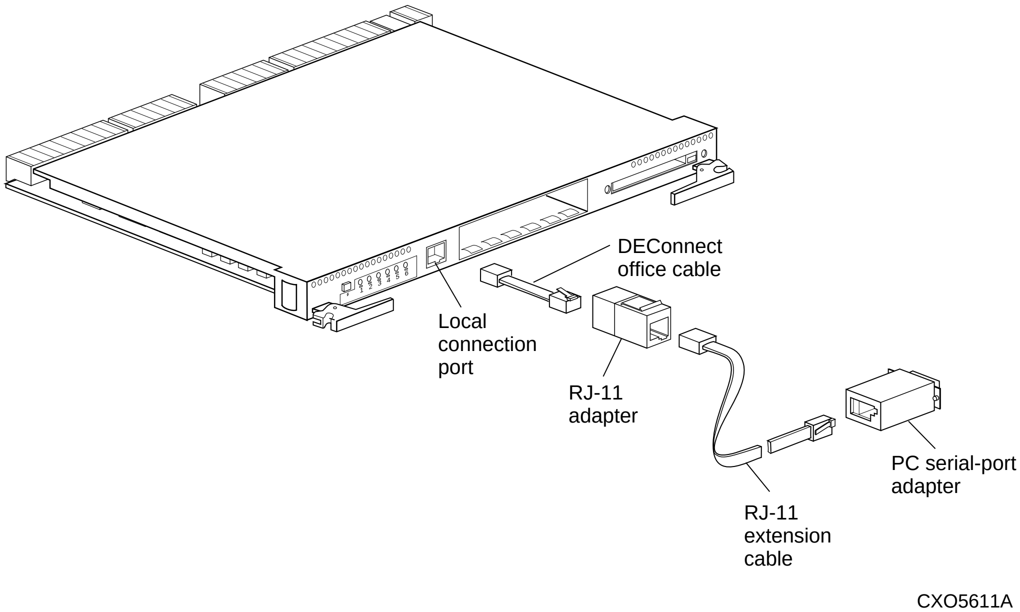
15. Plan and configure storagesets for your subsystem.

    See Chapter 3, “Creating Storagesets,” for detailed information about planning and configuring storagesets.

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 Getting Started Guide that came with your platform kit for details.

Figure 2–1 Terminal to Local-Connection Port Connection



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

To establish a local connection for setting the controller's initial configuration:

1. Turn off the terminal and connect it to the controller as shown in Figure 2–1. Plug one end of a DECconnect Office Cable (BC16E–XX) into the terminal; plug the other end into the RJ–11 adapter (12–43346–01); use the RJ–11 extension (17–03511–04) to connect the adapter to the controller's local-connection port. If you are using a PC instead of a terminal, you need the PC serial-port adapter (H8571–J), as shown in Figure 2–1.

2. Turn on the terminal.

3. Configure the terminal for 19200 baud, 8 data bits, 1 stop bit, and no parity.
4. Press the Enter or Return key. A copyright notice and the CLI prompt appear, indicating that you established a local connection with the controller.

Selecting a Failover Mode

Transparent failover is a dual-redundant 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.

Transparent failover occurs when a controller fails or someone presses the reset button on one of the controllers. To configure controllers for transparent failover, mount both controllers in the same BA370 pedestal and follow the steps in “Configuring a Controller,” page 2–3.

Using Transparent Failover

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

- Set your controllers for transparent failover before configuring devices. Once the devices, storagesets, and units are added to one controller’s configuration, they are automatically added to the other’s.
- 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=. See Appendix B, “CLI Commands,” for details about the SET FAILOVER COPY= command.
- Balance your assignment of devices. For example, in a 12-device subsystem, place 2 devices on each of the 6 ports, rather than placing 4 devices on each of 3 ports.
- The controller to which you copy configuration information restarts after you enter the SET FAILOVER command.

Changing the CLI Prompt

Use the SET THIS_CONTROLLER PROMPT= command to change the CLI prompt. Enter a 1-16 character string as the new prompt. For example, you could use the prompt to indicate the controller's name, such as "HSG>."

See Appendix B, "CLI Commands," for more information about the SET THIS_CONTROLLER command.

Backing up Power with a UPS

By default, the controller expects to use an external cache battery (ECB) as backup power to the cache module. You can also opt to use an uninterruptable power supply (UPS) to provide backup power in the event of a primary power failure. See Appendix B, "CLI Commands," for details about the SET THIS_CONTROLLER CACHE_UPS command. See Table 1–5 on page 1–17 and Table 1–4 on page 1–16 for information about cache policies.

Connecting the Subsystem to the Host

This section describes how to connect your subsystem to a host. It also includes instructions for connecting a single (nonredundant) controller and dual-redundant controllers to the host.

Caution Do not attempt to configure dual-redundant controllers using one hub with a loopback cable. This configuration will cause data corruption and is not supported.

Connecting a Single Controller to the Host Using One Hub

There are two possible configurations for a single controller, one that uses a single hub, and a second that uses two hubs. The second configuration can be used if you have two hosts to which you're connecting to your controller.

Figure 2–2 Cabling for Single Configuration

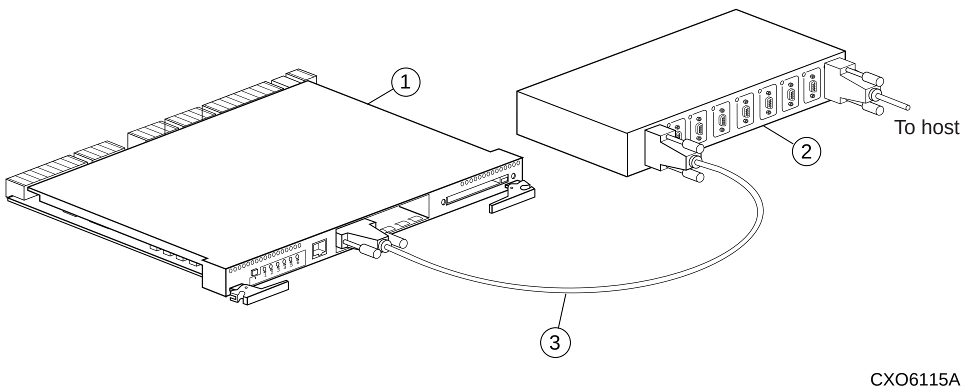


Table 2–1 Key to Figure 2–2 Cabling for Single Configuration

Item	Description	Part No.
1	Single Controller	—
2	9-Port Fibre Channel HUB	DS-DHGGA-CA
3	5-meter copper Fibre Channel cable or 10-meter copper Fibre Channel cable	17-04718-06 17-04718-07

To connect a single, nonredundant controller to the host using one hub:

1. Stop all I/O from the host to its devices on the bus to which you are connecting the controller.
2. Connect the Fibre Channel cable from Port 1 on the controller to Port 1 of the hub.

   For this configuration, set Port 2 off-line using the SET THIS_CONTROLLER PORT_2_TOPOLOGY=OFFLINE command. See Appendix B, “CLI Commands,” for details about the SET command.
3. Follow the procedures in the Getting Started Guide for connecting the Fibre Channel cable from the hub to your host system.
4. Route and tie the cables as desired.
5. Restart the I/O from the host. Some operating systems may require you to restart the host to see the devices attached to the new controller.

Connecting a Dual-Redundant Configuration to the Host

There are two possible ways to connect dual-redundant controllers to your host. The first method requires two hubs; the second method requires one hub.

Using Two Hubs

Use the diagram and procedure in this section to connect your dual-redundant controllers to the host using two hubs.

Figure 2–3 Cabling for Dual-Redundant Configuration with Two Hubs

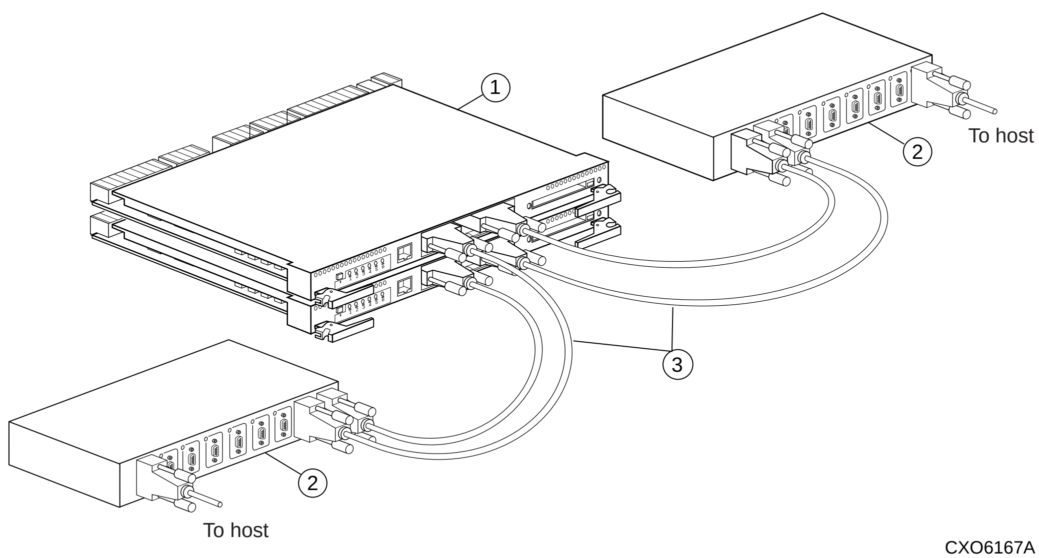


Table 2–2 Key to Figure 2–3 Cabling for Dual-Redundant Configuration with Two Hubs

Item	Description	Part No.
1	Dual Controller	—
2	9-port Fibre Channel HUB	DS-DHGGA-CA
3	5-meter host Fibre Channel cable 10-meter host Fibre Channel cable	17-04718-06 17-04718-07

To connect a pair of dual-redundant controllers to the host:

1. Stop all I/O from the host to its devices on the bus to which you are connecting the controllers.
2. Connect the 5-meter Fibre Channel cable from Port 1 on controller A to Port 9 on hub 1. Repeat this step to connect the second cable from Port 1 on controller B to Port 8 on hub 1.
3. Connect another 5-meter Fibre Channel cable from Port 2 on controller A to Port 1 on hub 2. Repeat this step to connect the final cable from Port 2 on controller B to Port 2 on hub 2.
4. Connect each hub to their respective host according to the instructions in the Getting Started manual.
5. Route and tie the cables as desired.
6. Restart the I/O from the host. Some operating systems may require you to restart the host to see the devices attached to the new controller.

Using One Hub

Use the diagram and procedure in this section to connect your dual-redundant controllers to the host using one hub.

Figure 2–4 Cabling for Dual-Redundant Configuration with One Hub

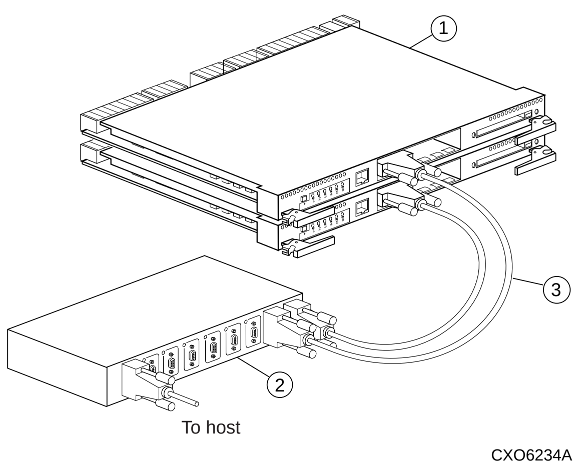


Table 2–3 Key to Figure 2–4 Cabling for Dual-Redundant Configuration with One Hub

Item	Description	Part No.
1	Dual Controller	—
2	9-port Fibre Channel HUB	DS-DHGGA-CA
3	5-meter host Fibre Channel cable 10-meter host Fibre Channel cable	17-04718-06 17-04718-07

To connect a pair of dual-redundant controllers to the host:

1. Stop all I/O from the host to its devices on the bus to which you are connecting the controllers.
2. Connect the 5-meter Fibre Channel cable from Port 1 on controller A to Port 9 on hub 1. Repeat this step to connect the second cable from Port 1 on controller B to Port 8 on hub 1.
3. Connect another 5-meter Fibre Channel cable from Port 2 on controller A to Port 1 on hub 2. Repeat this step to connect the final cable from Port 2 on controller B to Port 2 on hub 2.
4. Connect each hub to their respective host according to the instructions supplied in the Getting Started manual.
5. Route and tie the cables as desired.
6. Restart the I/O from the host. Some operating systems may require you to restart the host to see the devices attached to the new controller.

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.

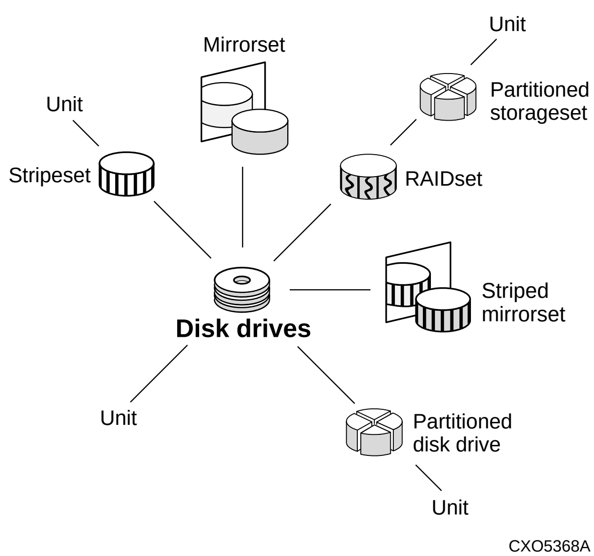
Introduction

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

You can create storage units by combining disk drives into storagesets, such as stripesets, RAIDsets, and mirrorsets, or by presenting them to the host as single-disk units, as shown in Figure 3–1.

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

Figure 3–1 Units Created from Storagesets, Partitions, and Drives



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

Planning and Configuring Storagesets

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

1. Create a storageset and device profile. See “Creating a Storageset and Device Profile,” page 3–4, for suggestions about creating a profile.
2. Determine your storage requirements. Use the questions in “Determining Storage Requirements,” page 3–6, to help you.
3. Choose the type of storagesets you need to use in your subsystem. See “Choosing a Storageset Type,” page 3–7, for a comparison and description of each type of storageset.
4. Select names for your storagesets and units. See “Creating a Storageset Map,” page 3–20, for details about selecting names.
5. Assign unit numbers to storagesets so the host can access the units. See“Creating a Storageset Map,” page 3–20, for information about how to assign units numbers to storagesets.
6. Create a storageset map to help you configure your subsystem. See “Creating a Storageset Map,” page 3–20, for suggestions about creating a storageset map.
7. If you are going to partition your storagesets, plan the partitions. See “Planning Partitions,” page 3–25, for information about partitions and how to plan for them.
8. Choose the switches that you will want to set for your storagesets and devices. See“Choosing Switches for Storagesets and Devices,” page 3–27, for a description of the switches you can select for storagesets.
9. Configure the storagesets you have planned using one of these methods:
   - Use StorageWorks Command Console (SWCC), a Graphical User Interface (GUI), to set up and manage RAID storage subsystems. See the SWCC Getting Started guide for details about using SWCC to configure your storagesets.
   - Use CLI commands. This method allows you flexibility in defining and naming storagesets. See “Configuring Storagesets with CLI Commands,” page 3–44, for information about configuring each type of storageset using CLI commands.

Creating a Storageset and Device Profile

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

Familiarize yourself with the kinds of information contained in a storageset profile, as shown in Figure 3–2.

Appendix A 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.

Figure 3–2 A Typical Storageset Profile

Type of storageset			
_____ Mirrorset	__X_ RAIDset	_____ Stripeset	_____ Striped Mirrorset

Storageset Name ... *accept default values*

Disk Drives........... *DISK10300, DISK20300, DIS30300*

Unit Number......... *accept default*

Partitions

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

RAIDset Switches

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

Mirrorset Switches

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

Initialize Switches

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

Unit Switches

Read Cache	Write Cache	Maximum Cache Transfer
_X_ Yes (default)	___ Yes (default)	_X_ 32 blocks (default)
___ No	_X_ No	___ Other:

Availability	Write Protection
_X_ Run (default)	_X_ No (default)
___ NoRun	___ Yes

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 the table in “Choosing a Storageset Type,” page 3–7, to determine the types of storagesets you should create to satisfy your organization’s requirements.

Choosing a Storageset Type

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

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

Table 3–1 compares different kinds of storagesets to help you determine which ones satisfy your requirements.

Table 3–1 A Comparison of Different Kinds of Storagesets

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

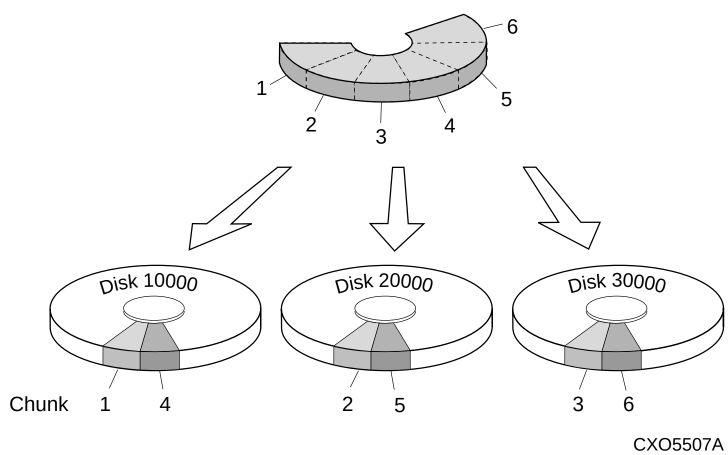
For a comprehensive discussion of RAID, refer to *The RAIDBOOK—A Source Book for Disk Array Technology.*

Using Stripesets to Increase I/O Performance

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

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

Figure 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 3–34, for information about setting the chunk size.

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

Considerations for Planning a Stripeset

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

- Reporting methods and size limitations prevent certain operating systems from working with large stripesets. See the *HSG80 Array Controller ACS Version 8.0 Release Notes* or the Getting Started Guide that came with your platform kit for details about these restrictions.
- A storageset should only contain disk drives of the same capacity. The controller limits the capacity of each member to the capacity of the smallest member in the storageset. Thus, if you combine 9 GB disk drives with 4 GB disk drives in the same storageset, you will waste 5 GB of capacity on each 9 GB member.

  If you need high performance and high availability, consider using a RAIDset, striped mirrorset, or a host-based shadow of a stripeset.
- Striping does not protect against data loss. In fact, because the failure of one member is equivalent to the failure of the entire stripeset, the likelihood of losing data is higher for a stripeset than for a single disk drive.

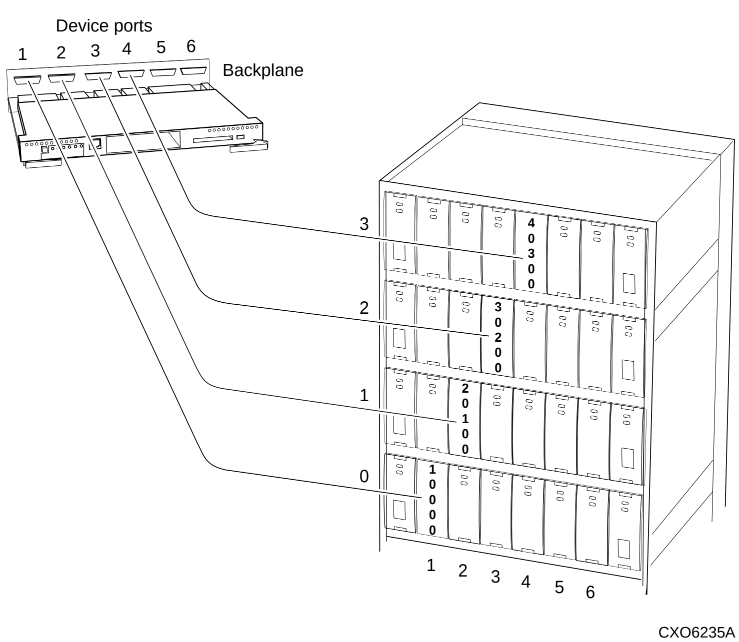
  For example, if the mean time between failures (MTBF) for a single disk is one hour, then the MTBF for a stripeset that comprises N such disks is l/N hours. As another example, if a single disk's MTBF is 150,000 hours (about 17 years), a stripeset comprising four of these disks would only have an MTBF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

  For this reason, you should avoid using a stripeset to store critical data. Stripesets are more suitable for storing data that can be reproduced easily or whose loss does not prevent the system from supporting its critical mission.
- Evenly distribute the members across the device ports to balance load and provide multiple paths as shown in the Figure 3–4.
- Stripesets contain between 2 and 14 members.
- Stripesets are well-suited for the following applications:
  - Storing program image libraries or run-time libraries for rapid loading
  - Storing large tables or other structures of read-only data for rapid application access

- Collecting data from external sources at very high data transfer rates

- Stripesets are not well-suited for the following applications:
  - A storage solution for data that cannot be easily reproduced or for data that must be available for system operation
  - Applications that make requests for small amounts of sequentially-located data
  - Applications that make synchronous random requests for small amounts of data

Figure 3–4 Distribute Members across Ports

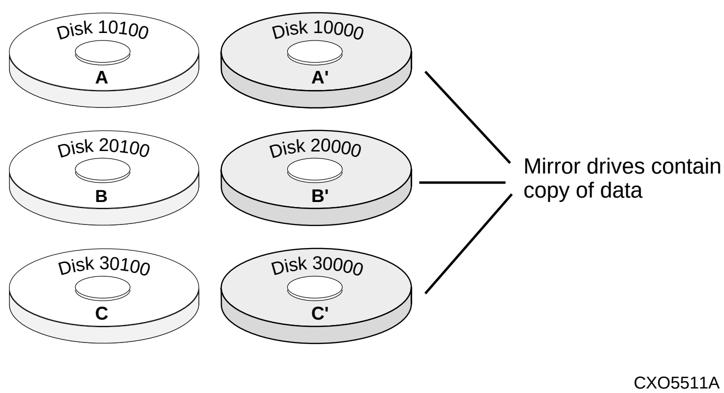


By spreading the traffic evenly across the buses, you will ensure that no bus handles the majority of data to the storageset.

Using Mirrorsets to Ensure Availability

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

Figure 3–5 Mirrorsets Maintain Two Copies of the Same Data



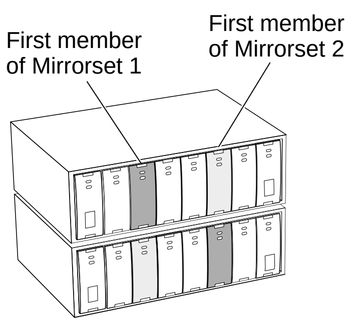
Considerations for Planning a Mirrorset

Keep these points in mind as you plan your mirrorsets:

- 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 20 mirrorsets per controller or pair of dual-redundant controllers. Each mirrorset may contain up to six members.
- A write-back cache module is required for mirrorsets, but write-back cache need not be enabled for the mirrorset to function properly.
- Both write-back cache modules must be the same size.
- If you’re using more than one mirrorset in your subsystem, you should put the first member of each mirrorset on different buses as shown in Figure 3–6. (The first member of a mirrorset is the first disk drive you add.)

When a controller receives a request to read or write data to a mirrorset, it typically accesses the first member of the mirrorset. If you have several mirrorsets in your subsystem and their first members are on the same bus, that bus will be forced to handle the majority of traffic to your mirrorsets.

Figure 3–6 First Mirrorset Members on Different Busses



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 3–31, for more information about this switch.

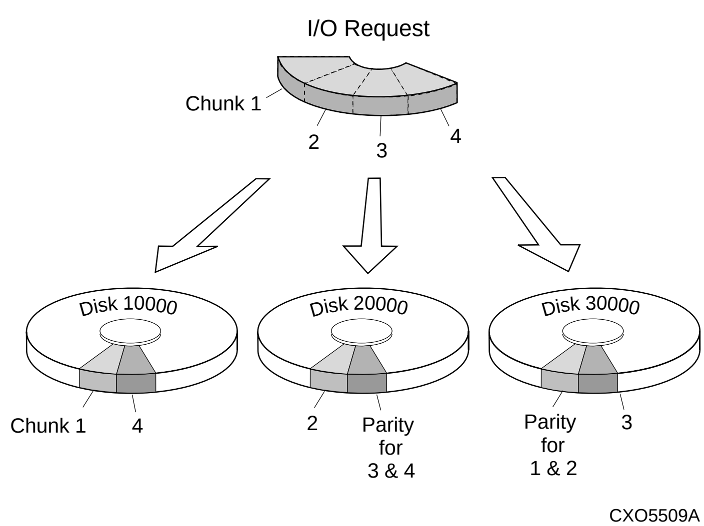
- Place mirrorsets and RAIDsets on different ports to minimize risk in the event of a single port bus failure.
- Mirrorset units are set to WRITEBACK_CACHE by default which increases a unit's performance.
- A storageset should only contain disk drives of the same capacity. The controller limits the capacity of each member to the capacity of the smallest member in the storageset. Thus, if you combine 9 GB disk drives with 4 GB disk drives in the same storageset, you waste 5 GB of capacity on each 9 GB member.
- Evenly distribute the members across the device ports to balance load and provide multiple paths as shown in Figure 3–4 on page 3–10.
- Mirrorsets are well-suited for the following:
  - Any data for which reliability requirements are extremely high
  - Data to which high-performance access is required

- Applications for which cost is a secondary issue
- Mirrorsets are not well-suited for the following applications:
  - Write-intensive applications
  - 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.

Figure 3–7 Parity Ensures Availability; Striping Provides Good Performance



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

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

The relationship between the chunk size and the average request size determines if striping maximizes the request rate or the data-transfer rates. You can set the chunk size or let the controller set it automatically. See “Chunk Size,” page 3–34, for information about setting the chunk size.

Considerations for Planning a RAIDset

Keep these points in mind as you plan your RAIDsets:

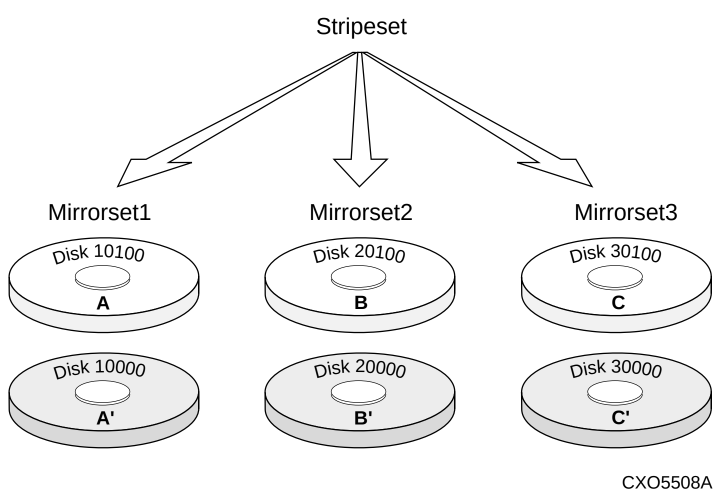
- Reporting methods and size limitations prevent certain operating systems from working with large RAIDsets. See the *HSG80 Array Controller ACS Version 8.0 Release Notes* or the Getting Started Guide that came with your platform kit for details about these restrictions.
- A cache module is required 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 3–10.
- A storageset should only contain disk drives of the same capacity. The controller limits the capacity of each member to the capacity of the smallest member in the storageset. Thus, if you combine 9 GB disk drives with 4 GB disk drives in the same storageset, you’ll waste 5 GB of capacity on each 9 GB member.
- RAIDset units are set to WRITEBACK_CACHE by default which increases a unit’s performance.
- RAIDsets and mirrorsets on different ports to minimize risk in the event of a single port bus failure.
- RAIDsets are particularly well-suited for the following:

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

Using Striped Mirrorsets for Highest Performance and Availability

Striped mirrorsets are simply stripesets whose members are mirrorsets. Consequently, this kind of storageset combines the performance of striping with the reliability of mirroring. The result is a storageset with very high I/O performance and high data availability.

Figure 3–8 Striping and Mirroring in the Same Storageset



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

Considerations for Planning a Striped Mirrorset

Plan the mirrorset members, then plan the stripeset that will contain them. Review the recommendations in “Considerations for Planning a Stripeset,” page 3–9, and “Considerations for Planning a Mirrorset,” page 3–11.

Worldwide Names (Node ID Numbers)

A worldwide name (node ID) is a unique 64-bit number assigned to a subsystem by the Institute of Electrical and Electronics Engineers (IEEE) and set by DIGITAL manufacturing prior to shipping. The worldwide name assigned to a subsystem never changes.

Each subsystem's worldwide name ends in zero, for example 5000-1FE1-FF0C-EE00. The controller port ID numbers are derived from the worldwide name. In a subsystem with two controllers (a dual-redundant configuration) the port ID of Port 1 for both controllers is the worldwide 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 subsytem's worldwide name. See Appendix B, "CLI Commands," for more information about the SHOW command and worldwide names. The CLI uses the term node ID for worldwide names. When you enter the SHOW command, the subsystem worldwide name (node ID) displays as the REPORTED NODEID and will look like the following:

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

Restoring Worldwide Names (Node ID Numbers)

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

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

If a situation occurs that requires you to restore the worldwide name, you can restore it using the worldwide name and check sum printed on the sticker on the frame into which your controller is inserted. See the

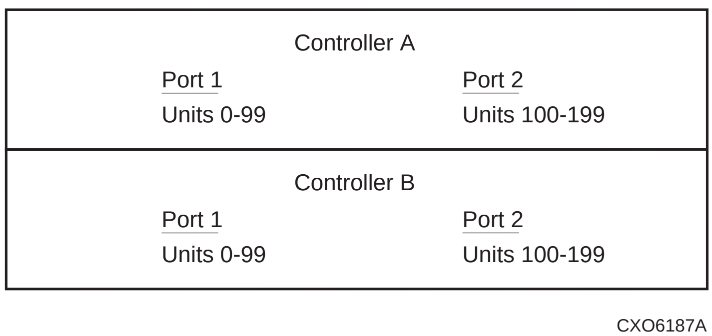
SET *controller* command in Appendix B, "CLI Commands," for details about setting the worldwide name (node ID.)

Caution Each subsystem has its own unique worldwide name (node ID). This name is printed on the sticker affixed to the frame into which your controller is inserted. If you attempt to set the subsystem worldwide name to a name other than the one that came with the subsystem, the data on the subsystem will not be accessible. Never set two subsystems to the same worldwide name; data corruption will occur.

Assigning Unit Numbers for Host Access to Storagesets

Each controller has two ports, Port 1 and Port 2, as shown in Figure 3–9. You can specify a maximum of eight units per port for a total of 16 units per controller or dual-redundant pair of controllers. This maximum is a limitation of the host.

Figure 3–9 Controller Port ID Numbers and Unit Numbers



Note The host sees units 100-199 as units 0-99 as well.

You will need to assign a unit number to each storageset, single disk unit, or storage device that you want your host to know about in your subsystem. The host uses these numbers to indicate the source or target for every I/O request it sends to a controller.

Each unit number contains the following:

- A letter that indicates the kind of devices in the storage unit. For example, D for disk drives.
- A number from 0-199.

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

See Appendix B, “CLI Commands,” for details about the ADD UNIT command.

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 four disk drives in the spareset. Each enclosure also has redundant power supplies.

Figure 3–10 Storageset Map

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**

To create a storageset map:

1. Copy the template from “Enclosure Template,” page A–4.

2. Establish a local or remote connection to one of the controllers in your subsystem.
3. Show the devices that are assigned to the controller. Use the following 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 Port-Target-LUN (PTL) location. See the next section for details about the controller's PTL addressing convention. Repeat step 2 through step 4 for each controller or dual-redundant pair of controllers.

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

Device PTL Addressing Convention within the Controller

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

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

- P—Designates the controller's SCSI device port number (1 through 6).
- T—Designates the target identification (ID) number of the device. Valid target ID numbers are 0 through 3.
- L—Designates the logical unit (LUN) of the device.

Note The controller operates with a BA370 pedestal assigned ID number 0.

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

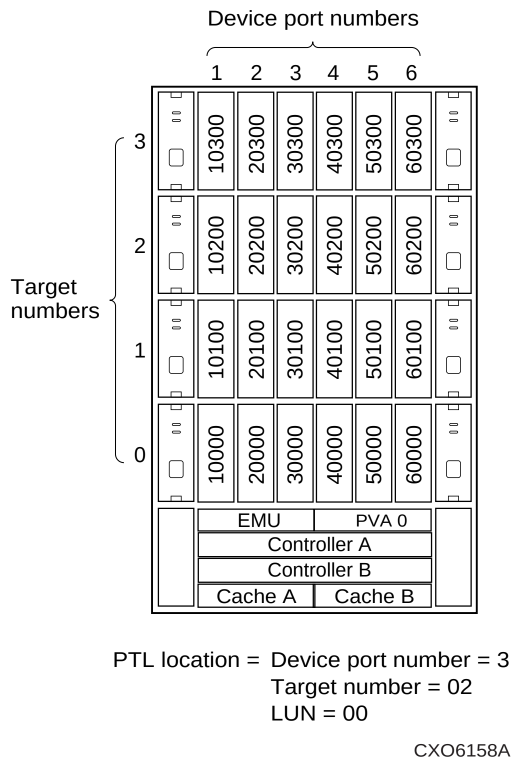
Figure 3–11 PTL Naming Convention

1 02 00

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

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

Figure 3–12 PTL Addressing in a Configuration

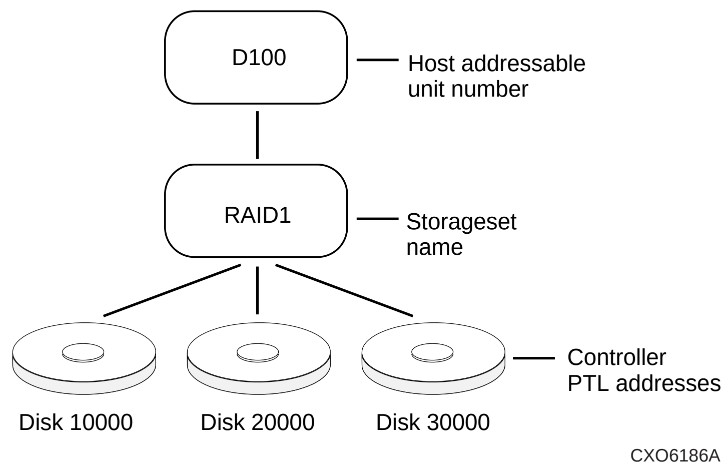


In Figure 3–12, the controller addresses DISK30200 through device port 3, target 02, LUN 00. This PTL location indicates the pathway the controller uses to address a disk drive (device) in the subsystem. It also indicates the device name.

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.

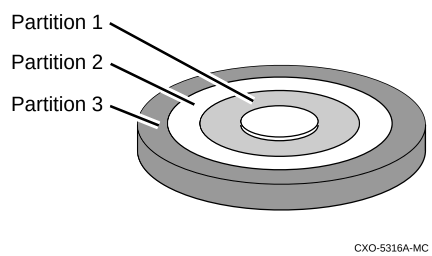
Figure 3–13 Locating Devices using PTLs



Planning Partitions

Use partitions to divide a storageset or disk drive into smaller pieces, which can each be presented to the host as its own storage unit. Figure 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 storageset or device. Because partitions are separately-addressable storage units, you can partition a single storageset to service more than one user group or application.

Defining a Partition

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

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

See "Partitioning a Storageset or Disk Drive," page 3–54, for information on manually partitioning a storageset or single-disk unit.

Guidelines for Partitioning Storagesets and Disk Drives

Keep these points in mind as you plan your partitions:

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

Choosing Switches for Storagesets and Devices

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

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

Enabling Switches

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

If you use CFMENU to configure the device or storageset, it prompts you for the switches during the configuration process and automatically applies them to the storageset or device.

If you use CLI commands to configure the storageset or device manually, the procedures in “Configuring Storagesets with CLI Commands,” page 3–44, 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 3–60, for information about changing switches for a storageset or device.

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

RAIDset Switches

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

- Replacement policy
- Reconstruction policy
- Membership

Replacement Policy

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

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

Reconstruction Policy

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

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

Membership

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

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

Mirrorset Switches

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

- Replacement policy
- Copy speed
- Read source

Replacement Policy

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

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

Copy Speed

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

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

Read Source

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

- READ_SOURCE=LEAST_BUSY (default) forces the controller to read data from the “normal” or operational member that has the least-busy work queue.
- 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.
- READ_SOURCE=DISK*nnnnn* forces the controller to always read data from a particular “normal” or operational member. If the specified member fails, the controller reads from the least busy member.

Device Switches

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

- Transportability
- Transfer rate

Transportability

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

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

  Consider these points when using the NOTRANSPORTABLE switch:

  - When you bring non-transportable devices from another subsystem to your controller subsystem, add the device to your configuration using the ADD command. Do not initialize the device, or you will reset and destroy any forced error information contained on the device.
  - When you add units, the controller software verifies that the disks or storagesets within the units contain metadata. To determine whether a disk or storageset contains metadata, try to create a unit from it. This causes the controller to check for metadata. If no metadata is present, the controller displays a message; initialize the disk or storageset before adding it.
- TRANSPORTABLE disk drives can be used in non-StorageWorks subsystems. Transportable disk drives can be used as single-disk units in StorageWorks subsystems as well as disk drives in other systems. They cannot be combined into storagesets in a StorageWorks subsystem.

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

  Transportable devices have these characteristics:

- Can be interchanged with any SCSI interface that does not use the device metadata, for example, a PC.
- Cannot have write-back caching enabled.
- Cannot be members of a shadowset, storageset, or spareset.
- Do not support forced errors.

Consider these points when using the TRANSPORTABLE switch:

- Before you move devices from the subsystem to a foreign subsystem, delete the units and storagesets associated with the device and set the device as transportable. Initialize the device to remove any metadata.
- When you bring foreign devices into the subsystem with customer data follow this procedure:

1. Add the disk as a transportable device. Do not initialize it.
2. Copy the data the device contains to another nontransportable unit.
3. Initialize the device again after resetting it as nontransportable. Initializing it now places metadata on the device.

- Storagesets cannot be made transportable. Specify NOTRANSPORTABLE for all disks used in RAIDsets, stripesets, and mirrorsets.
- Do not keep a device set as transportable on a subsystem. The unit attached to the device loses forced error support which is mandatory for data integrity on the entire array.

Transfer Rate

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

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

Initialize Switches

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

- Chunk size (for stripesets and RAIDsets only)
- Save configuration
- Overwrite

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:

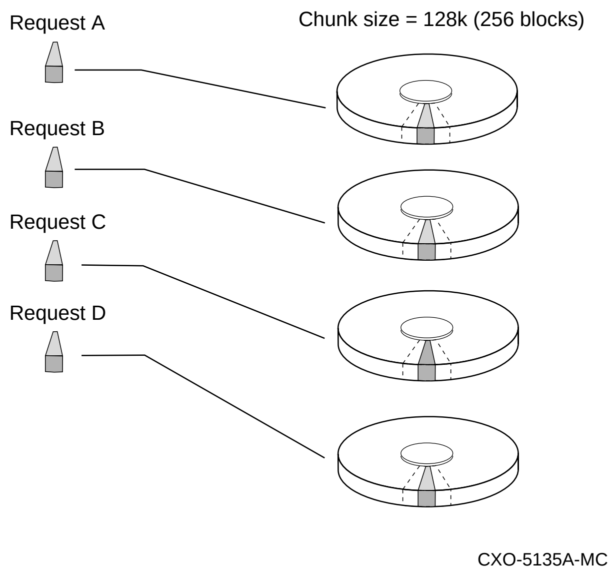
- CHUNKSIZE=DEFAULT lets the controller set the chunk size based on the number of disk drives (d) in a stripeset or RAIDset. If $d < 9$ then chunk size = 256. If $d > 9$ then chunk size = 128.
- CHUNKSIZE=*n* lets you specify a chunk size in blocks. The relationship between chunk size and request size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

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

Increasing the Request Rate

A large chunk size (relative to the average request size) increases the request rate by allowing multiple disk drives to respond to multiple requests. If one disk drive contains all of the data for one request, then the other disk drives in the storageset are available to handle other requests. Thus, in principle, separate I/O requests can be handled in parallel, thereby increasing the request rate. This concept is shown in Figure 3–15.

Figure 3–15 Chunk Size Larger than the Request Size



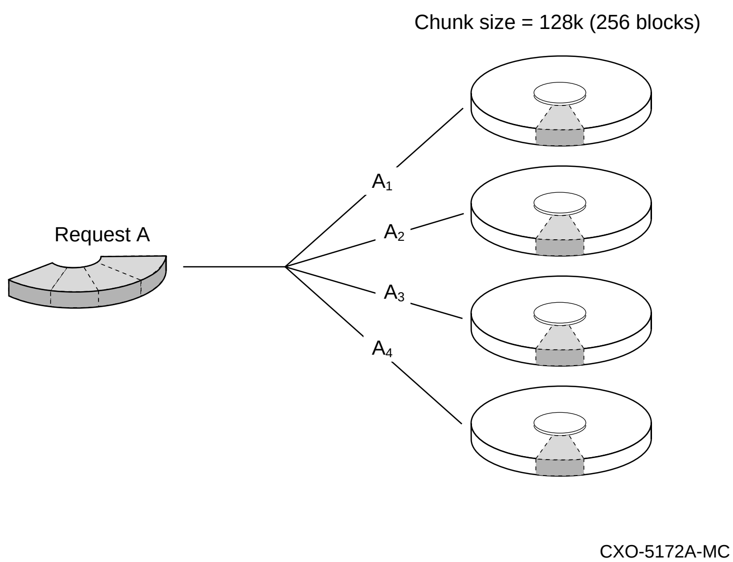
Applications such as interactive transaction processing, office automation, and file services for general timesharing tend to require high I/O request rates.

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.

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 3–2 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 3–2 Maximum Chunk Sizes for a RAIDset

RAIDset Size	Max 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:

Note The SAVE_CONFIGURATION switch is recommended for single configurations only. While the switch will work for dual-redundant configurations, it is not necessary to enable it.

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

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

  The save configuration option saves the following information:

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

  The save configuration option does not save the following information:

  - Software or hardware upgrades
  - Inter-platform conversions

Considerations for Saving the Configuration

- It is not necessary to use the SAVE_CONFIGURATION switch for dual-redundant configurations. Use the SET FAILOVER COPY= command to restore configuration information in a replacement controller. See “Saving Configuration Information in Dual-Redundant Configurations,” page 3–39 for details.
- Do not remove and replace disk devices between the time you save and restore your configuration. This is particularly important for devices that you migrate from another system. The controller could recover and use the wrong configuration information on your subsystem.
- Save your subsystem configuration as soon as possible after removing and replacing any disk devices in your subsystem. This ensures that the devices always contain the latest, valid information for your system.
- When you incorporate a spare into a storageset that you initialized with the INITIALIZE SAVE_CONFIGURATION command, the controller reserves space on the spare for configuration information. The controller updates this information when the configuration changes.

- You cannot use a storageset that contains user data to save your subsystem configuration unless you backup and restore the user data.
- If you previously configured storagesets with the SAVE_CONFIGURATION option, you do not need to initialize them again after you reconfigure your devices with a new controller.
- When you replace a controller, make sure the replacement controller does not contain any configuration data. If the controller is not new, initialize it with the SET THIS_CONTROLLER INITIAL_CONFIGURATION command. If you do not take this precaution, you can lose configuration data if non-volatile memory changes.

Saving Configuration Information in Dual-Redundant Configurations

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

- The controller-unique data for both controllers is saved.
- The autorestore feature does not operate when you restart the second controller in a dual-redundant subsystem. You must use the SET FAILOVER COPY= command to copy the configuration information from the operating controller.
- When replacing both controllers, you can replace the first and restart it alone to enable the autorestore feature. (This controller picks up any previously saved configuration data on disk and uses it to set up the subsystem configuration.) Replace the second controller using the SET FAILOVER COPY= command to copy the configuration information from the operating controller.
- Both controllers update the saved data; each writes to only those devices currently preferred to it. This prevents conflicting data transfer.

Destroy/Retain

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

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

- DESTROY (default) overwrites the user data and forced-error metadata on a disk drive when it is initialized.
- NODESTROY preserves the user data and forced-error metadata when a disk drive is initialized. Use NODESTROY to create a single-disk unit from any disk drive that has been used as a member of a mirrorset. See the REDUCED command in the Appendix B, “CLI Commands,” for information on removing disk drives from a mirrorset.

  NODESTROY is ignored for members of a RAIDset, all of which are destroyed when the RAIDset is initialized.

Unit Switches

You can enable the Unit switches listed in Table 3–3 for the listed storagesets and devices. See Appendix B, “CLI Commands,” for a complete list of Unit switches

Table 3–3 Unit Switches

Container Type / **Switch**	PARTITION=partition-number	MAXIMUM_CACHED_ TRANSFER	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	✓	✓	✓	✓	✓		✓
Partition	✓	✓	✓		✓	✓	

Partition

Specify the partition number that identifies the partition associated with the host-addressable unit number you are adding. Partitioned units must have the same SCSI target ID number and must be part of the same container.

PARTITION=*partition-number* allows you to identify a partition that’s associated with the unit you’re adding.

Maximum Cache Transfer

Specify the amount of data (in blocks) that the controller may cache to satisfy a read request:

- MAXIMUM_CACHED_TRANSFER=n lets you indicate the number of data blocks that the controller will cache to satisfy a read request. Any I/O transfers in excess of the specified size will not be cached. You can specify a value from 1 to 1024.
- MAXIMUM_CACHED_TRANSFER=32 (default) is the default number of data blocks that the controller will cache to satisfy a read request.

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

Read Cache

Enable or disable the caching of read data to the storage unit:

- READ_CACHE (default) enables the caching of read data.
- NOREAD_CACHE disables the caching of read data.

Read-ahead Cache

Enables or disables the prefetching of data from disk which improves the performance of synchronous sequential read streams.

- READAHEAD_CACHE (default) enables the prefetching of data from disk.
- NOREADAHEAD_CACHE disables the prefetching of data from disk.

Availability

Specify whether or not to make the storage unit available to the host. This switch is not valid for partitioned units. Do not specify this switch on the SET or ADD commands for a partitioned unit.

- RUN (default) specifies that as soon as you provide a host-addressable unit number, the storage unit will be made available to the host.
- NORUN specifies that the storage unit will not be made available to the host until you specify the RUN switch.

Write Protection

Enable or disable write protection for the storage unit:

- NOWRITE_PROTECT enables the controller to write new data to the storage unit.
- WRITE_PROTECT (default) prevents the controller from writing any new data to the storage unit. (The controller can write to a protected unit if it needs to reconstruct data.)

Write-back Cache

Enable or disable the controller's write-back caching for a storage unit:

- WRITEBACK_CACHE (default) enables write-back caching.
- NOWRITEBACK_CACHE disables write-back caching.

Note If you disable write-back caching for a storage unit that previously used it, it may take up to five minutes to flush the unwritten data from the cache to the devices in the storage unit.

Configuring Storagesets with CLI Commands

One method of configuring storagesets is manual configuration. This method allows you the most flexibility in defining and naming storagesets. See Appendix B, “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:

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 DISK*nnnnn 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 Stripeset

See “Using Striped Mirrorsets for Highest Performance and Availability,” page 3–15 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* **DISK***nnnnn* **DISK***nnnnn*

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

   INITIALIZE *stripeset-name switch*

Table 3–4 contains the valid Initialize switches and values.

Table 3–4 Initialize Switches for Stripesets

Initialize Switch	Value and Syntax
Chunk size	CHUNKSIZE=DEFAULT* CHUNKSIZE=*N*
Save configuration	NOSAVE_CONFIGURATION* SAVE_CONFIGURATION
Destroy	NODESTROY* DESTROY

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

   ADD UNIT *unit-number stripeset-name switch*

   Table 3–5 contains the valid Unit switches and values.

Table 3–5 Unit Switches for Stripesets

Unit Switch	Value and Syntax
Maximum cached transfer	MAXIMUM_CACHED_TRANSFER=32* MAXIMUM_CACHED_TRANSFER=*n*
Read cache	READ_CACHE* NOREAD_CACHE
Write-back cache	WRITEBACK_CACHE* NOWRITEBACK_CACHE
Availability	RUN* NORUN
Read-ahead cache	READAHEAD_CACHE* NOREADAHEAD_CACHE

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*

Example

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

ADD STRIPESET STRIPE1 DISK10000 DISK20000 DISK30000

INITIALIZE STRIPE1 CHUNKSIZE=128

ADD UNIT D100 STRIPE1 MAXIMUM_CACHED_TRANSFER=16

SHOW STRIPE1

SHOW D100

Configuring a Mirrorset

See Chapter 3, “Creating Storagesets,” for information about creating a profile and understanding the switches you can set for this kind of storage unit.

To configure a mirrorset:

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

   Use the following syntax to create a mirrorset:

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

Table 3–6 contains the valid Mirrorset switches and values.

Table 3–6 Mirrorset Switches

Mirrorset Switch	Value and Syntax
Replacement policy	POLICY=BEST_PERFORMANCE* POLICY=BEST_FIT NOPOLICY
Copy speed	COPY=NORMAL* COPY=FAST
Read source	READ_SOURCE=LEAST_BUSY* READ_SOURCE=ROUND_ROBIN READ_SOURCE=DISK*nnnnn*

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*

   Table 3–7 contains the valid Initialize switches and values.

Table 3–7 Initialize Switches for Mirrorsets

Initialize Switch	Value and Syntax
Save configuration	NOSAVE_CONFIGURATION* SAVE_CONFIGURATION
Destroy	NODESTROY* DESTROY

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

   ADD UNIT *unit-number mirrorset-name switch*

   Table 3–8 contains the valid Unit switches and values.

Table 3–8 Unit Switches for Mirrorsets

Unit Switch	Value and Syntax
Maximum cached transfer	MAXIMUM_CACHED_TRANSFER=32* MAXIMUM_CACHED_TRANSFER=*n*

Table 3–8 Unit Switches for Mirrorsets (Continued)

Unit Switch	Value and Syntax
Read cache	READ_CACHE* NOREAD_CACHE
Write-back cache	WRITEBACK_CACHE* NOWRITEBACK_CACHE
Write protection	NOWRITE_PROTECT* WRITE_PROTECT
Read-ahead cache	READAHEAD_CACHE* NOREADAHEAD_CACHE
Availability	RUN* NORUN

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

To configure a RAIDset:

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

Use the following syntax to create a RAIDset:

ADD RAIDSET *RAIDset-name* **DISK***nnnnn* **DISK***nnnnn* **DISK***nnnnn* *switch*

Table 3–9 contains the valid RAIDset switches and values.

Table 3–9 RAIDset Switches

RAIDset Switch	Value and Syntax
Replacement policy	POLICY=BEST_PERFORMANCE* POLICY=BEST_FIT NOPOLICY
Reconstruction speed	RECONSTRUCT=NORMAL* RECONSTRUCT=FAST

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*

   Table 3–10 contains the valid Initialize switches and values.

Table 3–10 Initialize Switches for RAIDsets

Initialize Switch	Value and Syntax
Chunk size	CHUNKSIZE=DEFAULT* CHUNKSIZE=*n*
Save configuration	NOSAVE_CONFIGURATION* SAVE_CONFIGURATION

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

---

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

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

   ADD UNIT *unit-number RAIDset-name switch*

Table 3–11 contains the valid Unit switches and values.

Table 3–11 Unit Switches for RAIDsets

Unit switch	Value and syntax
Maximum cached transfer	MAXIMUM_CACHED_TRANSFER=32* MAXIMUM_CACHED_TRANSFER=n
Read cache	READ_CACHE* NOREAD_CACHE
Read-ahead cache	READAHEAD_CACHE* NOREADAHEAD_CACHE
Write-back cache	WRITEBACK_CACHE* NOWRITEBACK_CACHE
Availability	RUN* NORUN

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 D99 RAID1

SHOW RAID1

SHOW D99

Configuring a Striped Mirrorset

See Chapter 3, “Creating Storagesets,” for information about creating a profile and understanding the switches you can set for this kind of storage unit.

To configure a striped mirrorset:

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

   ADD STRIPESET *mirrorset_1 mirrorset_2*
3. Initialize the stripeset. If you want to set any Initialize switches, you must do so in this step. Use the following syntax:

   INITIALIZE *stripeset-name switch*

   Table 3–12 contains the valid Initialize switches and values.

Table 3–12 Initialize Switches for Striped Mirrorsets

Initialize Switch	Value and Syntax
Chunk size	CHUNKSIZE=DEFAULT* CHUNKSIZE=*n*
Save configuration	NOSAVE_CONFIGURATION* SAVE_CONFIGURATION
Destroy	NODESTROY* DESTROY

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

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

   ADD UNIT *unit-number stripeset-name switch*

Table 3–13 contains the valid Unit switches and values.

Table 3–13 Unit Switches for Striped Mirrorsets

Unit Switch	Value and Syntax
Maximum cached transfer	MAXIMUM_CACHED_TRANSFER=32* MAXIMUM_CACHED_TRANSFER=*n*
Read cache	READ_CACHE* NOREAD_CACHE
Read-ahead cache	READAHEAD_CACHE* NOREADAHEAD_CACHE
Write-back cache	WRITEBACK_CACHE* NOWRITEBACK_CACHE
Write protection	NOWRITE_PROTECT* WRITE_PROTECT
Availability	RUN* NORUN

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

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

1. Add the disk drive by following the steps in “Adding Disk Drives,” page 3–44. Optionally, you can append Device switch values. If you do not specify switch values, the default values (*) are applied.
   Table 3–14 contains the valid Device switches and values.

Table 3–14 Device Switches for Single-Disk Units

Device Switch	Value and Syntax
Transportability	NOTRANSPORTABLE* TRANSPORTABLE
Transfer rate	TRANSFER_RATE_REQUESTED=20MHZ* TRANSFER_RATE REQUESTED=10MHZ TRANSFER_RATE_REQUESTED=5MHZ TRANSFER_RATE_REQUESTED=ASYNCHRONOUS

2. Initialize the disk drive using the following syntax:

   INITIALIZE DISK*nnn switch*

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

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

Table 3–15 contains the valid Unit switches and values.

Table 3–15 Unit Switches for Single-Disk Units

Unit Switch	Value and Syntax
Maximum cached transfer	MAXIMUM_CACHED_TRANSFER=32* MAXIMUM_CACHED_TRANSFER=*n*
Read cache	READ_CACHE* NOREAD_CACHE
Read-ahead cache	READAHEAD_CACHE* NOREADAHEAD_CACHE
Write-back cache	WRITEBACK_CACHE* NOWRITEBACK_CACHE
Write protection	NOWRITE_PROTECT* WRITE_PROTECT
Availability	RUN* NORUN

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 3–25, for details about partitioning a storage unit.

To partition a storageset or disk drive:

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

   ADD *storageset-name* **DISK***nnnnn* **DISK***nnnnn*

   or

   ADD DISK DISK*nnnnn ptl-location*

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

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

   INITIALIZE *storageset-name switch*

3. 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.

4. 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.

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

   Use the following syntax to present partitions to the host:

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

Table 3–16 contains the valid Unit switches and values for partitioned storagesets and disk drives.

Table 3–16 Unit Switches for Partitioned Storagesets and Disk Drives

Unit Switch	Value and Syntax
Maximum cached transfer	MAXIMUM_CACHED_TRANSFER=32* MAXIMUM_CACHED_TRANSFER=*n*
Read cache	READ_CACHE* NOREAD_CACHE
Read-ahead cache	READAHEAD_CACHE* NOREADAHEAD_CACHE
Write-back cache	WRITEBACK_CACHE* NOWRITEBACK_CACHE

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

SHOW *storageset-name*

Example

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

ADD RAIDSET RAID1 DISK10000 DISK20000 DISK30000

INITIALIZE RAID1

CREATE_PARTITION RAID1 SIZE=25

CREATE_PARTITION RAID1 SIZE=25

CREATE_PARTITION RAID1 SIZE=25

CREATE_PARTITION RAID1 SIZE=LARGEST

SHOW RAID1

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

```
 Partition number       Size       Starting Block   Used by
        .
        .
        .
```

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.

Follow these 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*nnnnn*

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

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 DISK*nnnnn*

   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. Follow these 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 that is comprised of DISK10000, DISK20000, and DISK30000.

SHOW STORAGESETS

Name	Storageset	Uses	Used by
STRIPE1	stripeset	DISK10000	D100
		DISK20000	
		DISK30000	

DELETE D100

DELETE STRIPE1

SHOW STORAGESETS

Changing Switches for a Storageset or Device

You can optimize a storageset or device at any time by changing the switches that are associated with it. See "Choosing Switches for Storagesets and Devices," page 3–27, for an explanation of the switches. Remember to update the storageset's profile when you change its switches.

Displaying the Current Switches

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

SHOW *storageset-name* **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
```

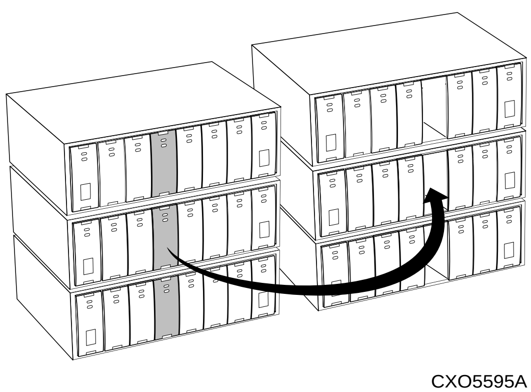
Moving Storagesets

You can move a storageset from one subsystem to another without destroying its data as shown in Figure 3–17. You also can follow the steps in this section to move a storageset to a new location within the same subsystem.

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

You can use the procedure in this section to migrate wide devices from an HSZ70 controller in a BA370 rack-mountable enclosure to an HSG80 environment. However, if you have an HSZ40 or HSZ50 subsystem, you cannot migrate to an HSG80 in a BA370 rack-mountable enclosure. Refer to the *HSG80 Array Controller ACS Version 8.0 Release Notes* for drives that can be supported.

Figure 3–17 Moving a Storageset from one Subsystem to Another



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

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

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

   SHOW *storageset-name*

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

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

   LOCATE *disk-name*

   To cancel the locate command, enter the following:

   LOCATE CANCEL

3. Delete the unit-number shown in the “Used by” column of the SHOW storageset-name command. Use the following 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 again 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 or create a new one. Use the following syntax:

   ADD UNIT *unit-number storageset-name*

Example

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

SHOW RAID99

Name	Storageset	Uses	Used by
RAID99	raidset	disk10000	D100
		disk20000	
		disk30000	

DELETE D100

DELETE RAID99

DELETE DISK10000

DELETE DISK20000

DELETE DISK30000

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

ADD DISK DISK20000 2 0 0

ADD DISK DISK30000 3 0 0

ADD DISK DISK40000 4 0 0

ADD RAIDSET RAID99 DISK20000 DISK30000 DISK40000

ADD UNIT D100 RAID99

Example

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

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

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

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

CHAPTER 4

Troubleshooting

This chapter provides guidelines for troubleshooting the controller, cache module, and external cache battery. It also describes the utilities and exercisers that you can use to aid in troubleshooting these components. See the appendixes for a list of LEDs and event codes. See the documentation that accompanied the enclosure for troubleshooting its hardware, such as the power supplies, cooling fans, and environmental monitoring unit.

Maintenance Features

Use these maintenance features to troubleshoot and service a controller:

- “Operator Control Panel,” page 1–7
- “Establishing a Local Connection to the Controller,” page 2–5
- “Utilities and Exercisers,” page 1–9

Troubleshooting Checklist

The following checklist provides a general procedure for diagnosing the controller and its supporting modules. If you follow this checklist, you’ll be able to identify many of the problems that occur in a typical installation. When you’ve identified the problem, use Table 4–1 to confirm your diagnosis and fix the problem.

If your initial diagnosis points to several possible causes, use the tools described later in this chapter to further refine your diagnosis. If the problem can’t be diagnosed using the checklist and tools, call customer service for additional support.

To troubleshoot the controller and its supporting modules:

1. Check the power to the cabinet and its components. Are the cords connected properly? Is the power within specifications?
2. Check the component cables. Are the bus cables to the controllers connected properly? Are the external cache battery (ECB) cables connected properly?
3. Check the program cards to ensure they’re fully seated.
4. Check the operator control panel and devices for LED codes. See Appendix C to interpret the LED codes.
5. Connect a local terminal to the controller and check its configuration with the following command:

   SHOW THIS_CONTROLLER FULL

   Ensure that the ACS software version is correct and that pertinent patches have been installed. Also, check the status of the cache module and its ECB.
6. Using FMU, check for last failure or memory-system failure entries. Show these codes and translate the last failure codes they contain. See page 4-10, Checking Failure Entries.

   If the controller has failed to the extent it cannot support a local terminal for FMU, check the host’s error log for the instance or last-failure codes. See Appendix D to interpret the event codes.
7. Check the status of the devices with the following command:

   SHOW DEVICES FULL

Look for errors such as “misconfigured device” or “No device at this PTL.” If a device reports misconfigured or missing, check its status with the following command:

SHOW *device-name*

8. Check the status of the storagesets with the following command:

   SHOW STORAGESETS FULL

   Ensure that all storagesets are “normal” (or “normalizing” if it’s a RAIDset or mirrorset). Check again for misconfigured or missing devices.

9. Check the status of the units with the following command:

   SHOW UNITS FULL

   Ensure that all of the units are “available” or “online.” If the controller reports that a unit is unavailable or off-line, re-check the storageset it belongs to with the following command:

   SHOW *storageset-name*

   If the controller reports that a unit has “lost data” or is “unwriteable,” re-check the status of the devices that make up the storageset. If the devices are OK, re-check the status of the cache module. If the unit reports a “media format error,” re-check the status of the storageset and its devices.

Troubleshooting Table

Use the troubleshooting checklist that begins on page 4–2 to find a symptom, then use this table to verify and fix the problem.

Table 4–1 Troubleshooting Table

Symptom	Possible Cause	Investigation	Remedy
Reset button not lit	No power to subsystem.	Check power to subsystem and power supplies on controller’s shelf.	Replace cord or AC input power module.
		Ensure that all cooling fans are installed. If one or more fans are missing or all are inoperative for more than 8 minutes, the EMU shuts down the subsystem.	Turn off power switch on AC input power module. Replace cooling fan. Restore power to subsystem.
		Verify that the standby power switch on the PVA was not depressed for more than 5 seconds.	Depress the alarm control switch on the EMU.
	Failed controller.	If the foregoing check fails to produce a remedy, check for OCP LED codes.	Replace controller.
Reset button lit steadily; other LEDs also lit	Various	See “Operator Control Panel LED Codes,” page C–2.	Follow repair action.

Table 4–1 Troubleshooting Table (Continued)

Symptom	Possible Cause	Investigation	Remedy
Can’t set failover to create dual-redundant configuration	Incorrect command syntax.	See Appendix B, “CLI Commands,”for the SET FAILOVER command.	Use correct command syntax.
	Different software versions on controllers.	Check software versions on both controllers.	Update one or both controllers so that both controllers are using the same software version.
	Incompatible hardware.	Check hardware versions.	Upgrade controllers so that they’re using compatible hardware.
	Controller previously set for failover.	Ensure that neither controller is configured for failover.	SET NOFAILOVER on both controllers, then reset “this” controller for failover.
	Failed controller.	If the foregoing checks fail to produce a remedy, check for OCP LED codes.	Follow repair action.

Table 4–1 Troubleshooting Table (Continued)

Symptom	Possible Cause	Investigation	Remedy
	Node ID is all zeros.	SHOW_THIS to see if node ID is all zeros.	Set node ID using the node ID (bar code) that is located on the frame in which the controller sits. See SET THIS CONTROLLER NODE_ ID in Appendix B, “CLI Commands”. Also, be sure that you are copying in the right direction. If you are cabled to the new controller, use SET FAILOVER COPY=OTHER. If cabled to old controller, use SET FAILOVER COPY=THIS.
Invalid cache	Cache module may erroneously contain unflushed write-back data. This may occur after you’ve installed a new controller: its existing cache module may indicate that it contains unflushed write-back data, but the new controller expects to find no data in the existing cache module. (This error may also occur if you install a new cache module for a controller that expects write-back data in the cache.)	SHOW THIS CONTROLLER indicates “invalid cache.” No spontaneous FMU message.	Connect a local terminal to controller reporting the error, and clear the error with a CLEAR_ERRORS THIS_CONTROLLE R command. See Appendix B, “CLI Commands”for more information.

Table 4–1 Troubleshooting Table (Continued)

Symptom	Possible Cause	Investigation	Remedy
Can’t add device	Illegal device.	See product-specific release notes that accompanied the software release for the most recent list of supported devices.	Replace device.
	Device not properly installed in shelf.	Check that SBB is fully seated.	Firmly press SBB into slot.
	Failed device.	Check for presence of device LEDs.	Follow repair action in the documentation provided with the enclosure or device.
	Failed power supplies.	Check for presence of power supply LEDs.	Follow repair action in the documentation provided with the enclosure or power supply.
	Failed bus to device.	If the foregoing checks fail to produce a remedy, check for OCP LED codes.	Replace enclosure or shelf.
Can’t configure storagesets	Incorrect command syntax.	See *Appendix B, “CLI Commands,”* for the ADD *storageset* command.	Reconfigure storageset with correct command syntax.
	Exceeded maximum number of storagesets.	Use the SHOW command to count the number of storagesets configured on controller.	Delete unused storagesets.
	Failed battery on ECB. (An ECB or UPS is required for RAIDsets and mirrorsets.)	Use the SHOW command to check the ECB’s battery status.	Replace ECB if required.

Table 4–1 Troubleshooting Table (Continued)

Symptom	Possible Cause	Investigation	Remedy
Can’t assign unit number to storageset	Incorrect command syntax.	See the *Appendix B, “CLI Commands,”* for correct syntax.	Reassign unit number with correct syntax.
Unit is available but not online	This is normal. Units are “available” until the host accesses them, at which point their status is changed to “online.”	None.	None.
Host cannot access unit	Host files or device drivers not properly installed or configured.	Check for the required device special files.	Configure device special files as described in the getting started manual that accompanied your software release.
	Invalid Cache.	See the description for the Invalid cache symptom.	See the description for the Invalid cache symptom.
	Unit(s) have lost data.	Issue the SHOW_UNIT command.	CLEAR_ERRORS *unit* lost data.

Table 4–1 Troubleshooting Table (Continued)

<table>
<tr><th>Symptom</th><th>Possible Cause</th><th>Investigation</th><th>Remedy</th></tr>
<tr><td rowspan="2">Host’s log file indicates that a forced error occurred when the controller was reconstructing a RAIDset or mirrorset</td><td>Unrecoverable read errors may have occurred when controller was reconstructing the storageset. Errors occur if another member fails while the controller is reconstructing the storageset.</td><td>Conduct a read scan of the storageset using the appropriate utility from the host’s operating system, such as the “dd” utility for a DIGITAL UNIX host.</td><td>Rebuild the storageset, then restore its data from a backup source. While the controller is reconstructing the storageset, monitor the activity with FMU for any unrecoverable errors. If unrecoverable errors persist, note the device on which they occurred, and replace the device before proceeding.</td></tr>
<tr><td>Host requested data from a normalizing storageset that didn’t contain the data.</td><td>SHOW storageset-name to see if all of its members are “normal”.</td><td>Wait for normalizing members to become normal, then resume I/O to it.</td></tr>
</table>

Checking Failure Entries

The controller's fault-management software reports information about significant events and failures that occur during the controller's operation. This information is derived from the sense data, which the controller sends in response to a Request Sense command as described in the SCSI-2 specifications (Information technology, Small Computer System Interface, X3T9.2, 375R Rev. 10k, 10-Mar-93).

The Fault Management Utility (FMU) provides a limited interface to the controller's fault-management software. Use FMU to:

- Display the last-failure and memory-system-failure entries that the fault-management software stores in the controller's non-volatile 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 local terminal.

Displaying Failure Entries

The controller stores the four most recent last-failure and memory-system failure reports as entries in its non-volatile memory. The occurrence of any failure event will terminate the controller on which it occurred. To display the last-failure and memory-system failure entries:

1. Connect a local terminal to the controller.
2. Start FMU with the following command:

   RUN FMU

3. Show one or more of the entries with the following command:

   SHOW *event_type entry# FULL*

   where:

   - *event-type* is LAST_FAIL or MEMORY_SYSTEM_FAIL
   - *entry#* is ALL, MOST_RECENT, or 1 through 4

- *FULL* displays additional information, such as the I960 stack and hardware component register sets (e.g. the memory controller, FX, host port, and device ports, etc.).

4. Exit FMU with the following command:

   EXIT

Example

The following example shows a last-failure entry. The Informational Report—the lower half of the entry—contains the instance code, reporting component, and so forth, which you can translate with FMU to learn more about the event.

```
Last Failure Entry: 4. Flags: 006FF300
Template: 1.(01) Description: Last Failure Event
Power On Time: 0. Years, 14. Days, 19. Hours, 51. Minutes, 31. Seconds
Controller Model: HSG80
Serial Number: AA12345678 Hardware Version: 0000(00)
Software Version: V080G(50)
Informational Report
Instance Code: 0102030A Description:
An unrecoverable software inconsistency was detected or an intentional
restart or shutdown of controller operation was requested.
Reporting Component: 1.(01) Description:
Executive Services
Reporting component’s event number: 2.(02)
Event Threshold: 10.(0A) Classification:
SOFT. An unexpected condition detected by a controller software com-
ponent (e.g., protocol violations, host buffer access errors, internal
inconsistencies, uninterpreted device errors, etc.) or an intentional
restart or shutdown of controller operation is indicated.
Last Failure Code: 20090010 (No Last Failure Parameters)
Last Failure Code: 20090010 Description:
This controller requested this controller to shutdown.
Reporting Component: 32.(20) Description:
Command Line Interpreter
Reporting component’s event number: 9.(09)
Restart Type: 1.(01) Description: No restart
```

Translating Event Codes

To translate the event codes in the fault-management reports for spontaneous events and failures:

1. Connect a local terminal to the controller.
2. Start FMU with the following command:

   RUN FMU

3. Show one or more of the entries with the following command:

   DESCRIBE *code_type code#*

   where *code_type* is one of those listed in Table 4–2 and *code#* is the alpha-numeric value displayed in the entry. The code types marked with an asterisk (*) require multiple code numbers.

Table 4–2 Event-Code Types

Event-Code Type	
instance	event_threshold
repair_action	memory_system_failure
last_failure	restart_type
ASC_ASCQ*	SCSI_command_operation*
controller_unique_ASC_ASCQ*	sense_data_qualifiers*
component	sense_key
device_type	template

Example

The following example shows the FMU translation of a last-failure code.

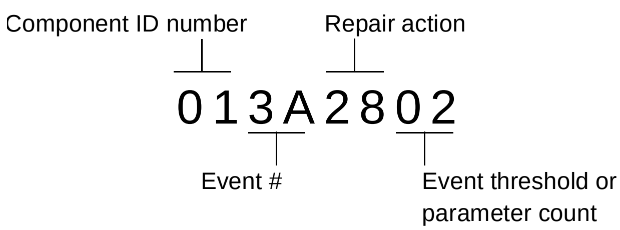
```
Last Failure Code: 206C0020 Descrip-
tion:

 Controller was forced to restart in or-
der for
 new controller code image to take ef-
fect.

Reporting Component: 32.(20) Descrip-
tion:

 Command Line Interpreter

Reporting component's event number:
108.(6C)

Restart Type: 2.(02) Description: Auto-
matic hardware restart
```

Instance Codes and Last-Failure Codes

Instance codes identify and accompany significant events that do not cause the controller to terminate operation; last-failure codes identify and accompany failure events that cause the controller to stop operating. Last-failure codes are sent to the host only after the affected controller is restarted successfully.

You can easily translate an instance or last-failure code yourself, rather than using FMU to translate them, if you understand their structure as shown in Figure 4–1.

Figure 4–1 Structure of an Instance or Last-Failure Code



- The component ID number identifies the software component that detected the event. See Component Identifier Codes on page D-59.
- The event number indicates the event that the component detected. When combined with the component ID, this number uniquely identifies the event (two different components may use the same event number to report different events).
- The repair action indicates the action that should be taken when the event threshold is reached. See Recommended Repair Action Codes on page D-53.
- The event threshold (reported for instance codes only) indicates the severity of the event and when the repair action should be taken. See Event Threshold Codes on page D-60.
- The parameter count (reported for last-failure codes only) indicates the number of fields in the event template that contain other important information about the failure.

Controlling the Display of Significant Events and Failures

You can control how the fault-management software displays significant events and failures with FMU’s SET command. Table 4-3 shows and describes the various SET commands that you can enter while you're running FMU. These commands remain in effect only as long as the current FMU session remains active unless you enter the PERMANENT qualifier—the last entry in Table 4-3.

Table 4–3 FMU SET Commands

Command	Result
SET EVENT_LOGGING SET NOEVENT_LOGGING	enable and disable the spontaneous display of significant events to the local terminal; preceded by "%EVL". When logging is enabled, the controller spontaneously displays information about the events on the local terminal. Spontaneous event logging is suspended during the execution of CLI commands and operation of utilities on a local terminal. Because these event logs are spontaneous, they are not stored anywhere and cannot be recovered.
SET LAST_FAILURE LOGGING SET NOLAST_FAILURE LOGGING	enable and disable the spontaneous display of last failure events; preceded by "%LFL". The controller spontaneously displays information relevant to the sudden termination of controller operation.
SET *log_type* REPAIR_ACTION SET *log_type* NOREPAIR_ACTION	enable and disable the inclusion of repair action information for event logging or last-failure logging. By default, repair actions are not displayed for these log types. If the display of repair actions is enabled, the controller displays any of the recommended repair actions associated with the event.
SET *log_type* VERBOSE SET *log_type* NOVERBOSE	enable and disable the automatic translation of event codes that are contained in event logs or last-failure logs. By default, the additional descriptions are not displayed for these log types. See Translating Event Codes on page 4-12 for instructions to translate these codes manually.
SET PROMPT SET NOPROMPT	enable and disable the display of the CLI prompt string following the log identifier "%EVL" or "%LFL". This command is useful if the CLI prompt string is used to identify the controllers in a dual-redundant configuration (see *Appendix B, "CLI Commands,"* for instructions to set the CLI command string for a controller). If enabled the CLI prompt will be able to identify which controller sent the log to the local terminal.
SET TIMESTAMP SET NOTIMESTAMP	enable and disable the display of the current date and time in the first line of an event or last-failure log.

Table 4–3 FMU SET Commands (Continued)

Command	Result
SET FMU REPAIR_ACTION SET FMU NOREPAIR_ACTION	enable and disable the inclusion of repair actions with SHOW LAST_FAILURE and SHOW MEMORY_SYSTEM_FAILURE commands. By default, the repair actions are not shown. If repair actions are enabled, the command outputs display all of the recommended repair actions associated with the instance or last-failure codes used to describe an event.
SET FMU_VERBOSE SET FMU_NOVERBOSE	enable and disable the inclusion of instance and last failure code descriptive text with SHOW LAST_FAILURE and SHOW MEMORY_SYSTEM_ FAILURE commands. By default, this descriptive text is not displayed. If the descriptive text is enabled, it identifies the fields and their numeric content that comprise an event or last-failure entry.
SET CLI_EVENT_REPORTING SET NOCLI_EVENT_REPORTING	enable and disable the asynchronous errors reported at the CLI prompt (for example, “swap signals disabled” or “shelf has a bad power supply”). Preceded by “%CER”, these errors are, by default, reported and cleared with the CLEAR ERRORS_CLI command.
SET FAULT_LED_LOGGING SET NOFAULT_LED_LOGGING	enable and disable the solid fault LED event log display on the local terminal. Preceded by “%FLL”. When enabled, and a solid fault pattern is displayed in the OCP LEDs, the fault pattern and its meaning are displayed on the maintenance terminal. For many of the patterns, additional information is also displayed to aid in problem diagnosis. In cases of automatic hardware reset (e.g., power failure or pressing the controller’s reset button), the fault LED log display is inhibited because automatic resets do not allow sufficient time to complete the log display.
SHOW PARAMETERS	displays the current settings associated with the SET command.
SET command PERMANENT	preserves the SET command across controller resets.

Using VTDPY to Check for Communication Problems

Use the virtual terminal display (VTDPY) utility to get information about the following communications:

- 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

To run VTDPY:

1. Connect a local terminal to the controller. The terminal must support ANSI control sequences.
2. Set the terminal to NOWRAP mode to prevent the top line of the display from scrolling off of the screen.
3. Start VTDPY with the following command:

   RUN VTDPY

   Use the following key sequences and commands to control VTDPY:

Table 4–4 VTDPY Key Sequences and Commands

Command	Action
Ctrl/C	Enables command mode; after entering Ctrl/C, enter one of the following commands and press Return: CLEAR DISPLAY CACHE DISPLAY DEFAULT DISPLAY DEVICE DISPLAY HOST DISPLAY STATUS HELP INTERVAL *seconds* (to change update interval)
Ctrl/G	Updates screen
Ctrl/O	Pauses (and resumes) screen updates
Ctrl/R	Refreshes current screen display
Ctrl/Y	Exits VTDPY

You may abbreviate the commands to the minimum number of characters necessary to identify the command. Enter a question mark (?) after a partial command to see the values that can follow the supplied command. For example, if you enter DISP ?, the utility will list CACHE, DEFAULT, and so forth. (Separate “DISP” and “?” with a space.) Upon successfully executing a command—other than HELP—VTDPY exits command mode. Pressing Return without a command also causes VTDPY to exit command mode.

Checking Controller-to-Host Communications

Use the display host VTDPY command to see how or if the controller is communicating with the host.

Figure 4–2 Xfer Rate Region of the Default Display

```
VTDPY> DISPLAY DEFAULT

          S/N: ZG74100120 SW: R052G-0 HW: 00-00
                35.8% Idle 98237 KB/S 1559 Rq/S

Pr  Name         Stk/Max Typ  Sta        CPU%            Target  Unit   ASWC  KB/S  Rd%  Wr%  Cm%HT%
0   NULL           0/0         Rn        35.8            111111  D0000  o^a   3288  100   0    00100
4   HP_MAIN  40/2   FNC   Rn  64.1      0123456789012345  D0001   o^a   3288   100   0    0    0100
                                        P1 DDD hH         D0002   o^a   3288   100   0    0    0100
                                        o2 DDD hH         D0003   o^a   3283   100   0    0    0100
                                        r3 DDD hH         D0004   o^a   8800   100   0    0    0100
                                        t4 DDD hH         D0005   o^a   8918   100   0    0    0100
                                         5 DDD hH         D0006   o^a   8795   100   0    0    0100
                                         6 DDD hH         D0007   o^a   8725   100   0    0    0100
                                                          D0100   o^a   6254   100   0    0    0100
                                                          D0101   o^a   6211   100   0    0    0100
                                                          D0102   o^a   6227   100   0    0    0100
                                                          D0103   o^a   6265   100   0    0    0100
                                                          D0104   o^a   6216   100   0    0    0100
                                                          D0105   o^a   6222   100   0    0    0100
                                                          D0106   o^a   6227   100   0    0    0100
                                                          D0107   o^a   6222   100   0    0    0100
```

Checking Controller-to-Device Communications

Use the device display to see how or if the controller is communicating with the devices in the subsystem. This display contains three important regions:

- Device map region (upper left)
- Device status region (upper right)
- Device-port status (lower left)

Figure 4–3 Regions on the Device Display

```
VTDPY>DISPLAY DEVICE


                      S/N: ZG64100176 SW: v7.0 HW: CX-02
        67% I/D Hit  99.9% Idle       0 KB/S       0 Rq/S       Up:
0  5:17.54
      Target                      P TL ASWF Rq/S RdKB/S WrKB/S Que
Tg CR BR TR
         111111              P1120 A^     0     0     0  0  0  0  0  0
 0123456789012345             D1130 A^     0     0     0  0  0  0  0  0
P1     hH   PDD              D1140 A^     0     0     0  0  0  0  0  0
o2     hH   DD D             D2120 A^     0     0     0  0  0  0  0  0
r3  ????hH                   D2130 A^     0     0     0  0  0  0  0  0
t4     hH DDD                D2150 a^     0     0     0  0  0  0  0  0
 5   P  hH                   ?3020  ^ F   0     0     0  0  0  0  0  0
 6 DDD  hH                   ?3030  ^ F   0     0     0  0  0  0  0  0
                             ?3040  ^ F   0     0     0  0  0  0  0  0
                             ?3050  ^ F   0     0     0  0  0  0  0  0
                             D4090 A^     0     0     0  0  0  0  0  0
                             D4100 A^     0     0     0  0  0  0  0  0
Port Rq/S RdKB/S WrKB/S  CR BR TR   D4110 A^      0      0      0   0
0  0  0  0
 1    0     0     0  0  0  0  P5030 A^     0     0     0  0  0  0  0  0
 2    0     0     0  0  0  0  D6010 A^     0     0     0  0  0  0  0  0
 3    0     0     0  0  0  0  D6020 A^     0     0     0  0  0  0  0  0
 4    0     0     0  0  0  0  D6030 A^     0     0     0  0  0  0  0  0
 5      0      0      0   0  0  0
 6      0      0      0   0  0  0
```

Checking Device Type and Location

The device map region of the device display (upper left) shows all of the devices that the controller recognizes through its device ports. Table 4–5 lists the heading and contents for each column of the device map region.

Table 4–5 Device Map Columns

Column	Contents
Port	SCSI ports 1 through 6.

Table 4–5 Device Map Columns (Continued)

Column	Contents
Target	SCSI targets 0 through 15. Single controllers occupy 7; dual-redundant controllers occupy 6 and 7.
	D = disk drive or CD-ROM drive
	F = foreign device
	H = this controller
	h = other controller in dual-redundant configurations
	P = passthrough device
	? = unknown device type
	= no device at this port/target location

Checking Device Status and I/O Activity

The device status region of the device display (upper right) shows the name and I/O characteristics for all of the devices that the controller recognizes. Table 4–6 lists the heading and contents for each column of the device status region.

Table 4–6 Device Status Columns

Column	Contents
PTL	Kind of device and its port-target-lun (PTL) location:
	D = disk drive
	P = passthrough device
	? = unknown device type
	= no device at this port/target location
A	Availability of the device:
	A = available to this controller
	a = available to other controller
	U = unavailable, but configured on this controller
	u = unavailable, but configured on other controller
	= unknown availability state

Table 4–6 Device Status Columns (Continued)

Column	Contents
S	Spindle state of the device: ^ = disk is spinning at correct speed; tape is loaded > = disk is spinning up < = disk is spinning down v = disk is not spinning = unknown spindle state
W	Write-protection state of the device. For disk drives, a W in this column indicates that the device is hardware write-protected. This column is blank for other kinds of devices.
F	Fault state of the device. An F in this column indicates an unrecoverable device fault. If this field is set, the device fault LED should also be lit.
Rq/S	Average request rate for the device during the last update interval. Requests can be up to 32K and generated by host or cache activity.
RdKB/S	Average data transfer rate from the device (reads) during the last update interval.
WrKB/S	Average data transfer rate to the device (writes) during the last update interval.
Que	Maximum number of I/O requests waiting to be transferred to the device during the last update interval.
Tg	Maximum number of requests queued to the device during the last update interval. If the device doesn't support tagged queuing, the maximum value is 1.
CR	Number of SCSI command resets that occurred since VTDPY was started.
BR	Number of SCSI bus resets that occurred since VTDPY was started.
TR	Number of SCSI target resets that occurred since VTDPY was started.

Checking Device-Port Status and I/O Activity

The device-port status region of the device display (lower left) shows the I/O characteristics for the controller's device ports. Table 4–7 lists the heading and contents for each column of the device-port status region.

Table 4–7 Device-Port Status Columns

Column	Contents
Port	SCSI device ports 1 through 6.
Rq/S	Average request rate for the port during the last update interval. Requests can be up to 32K and generated by host or cache activity.
RdKB/S	Average data transfer rate from the devices on the port (reads) during the last update interval.
WrKB/S	Average data transfer rate to the devices on the port (writes) during the last update interval.
CR	Number of SCSI command resets that occurred since VTDPY was started.
BR	Number of SCSI bus resets that occurred since VTDPY was started.
TR	Number of SCSI target resets that occurred since VTDPY was started.

Checking Unit Status and I/O Activity

Use the cache display to see the status and I/O activity for the logical units configured on the controller.

Figure 4–4 Unit Status on the Cache Display

```
VTDPY> DISPLAY CACHE

                      S/N: ZG64100176 SW: v7.0 HW: CX-02
                         66% I/D Hit  99.8% Idle       0 KB/S     0
Rq/S Up:    0  5:16.42
Unit  ASWC  KB/S  Rd% Wr% Cm% HT% PH% MS% Purge BlChd BlHit
P0300 o        0   0   0   0   0   0   0     0     0     0
D0303 o^ b     0   0   0   0   0   0   0     0     0     0
```

```
D0304 o^ b      0   0   0   0   0   0   0    0    0    0
P0400           0   0   0   0   0   0   0    0    0    0
P0401           0   0   0   0   0   0   0    0    0    0
D0402 x^ b      0   0   0   0   0   0   0    0    0    0
```

Table 4–8 lists the heading and contents for each column of the device status region.

Table 4–8 Unit Status Columns

Column	Contents
Unit	Kind of unit (and its unit number):
	D = disk drive or CD-ROM drive
	P = passthrough device
	? = unknown device type
A	Availability of the unit:
	a = available to other controller
	d = disabled for servicing. Offline.
	e = mounted for exclusive access by a user
	f = media format error
	i = inoperative
	m = maintenance mode for diagnostic purposes
	o = online. Host may access this unit through “this controller.”
	r = rundown with the SET NORUN command
	v = no volume mounted due to lack of media
	x = online. Host may access this unit through “other controller.”
	= unknown availability
S	Spindle state of the device:
	^ = disk is spinning at correct speed; tape is loaded
	> = disk is spinning up; tape is loading
	< = disk is spinning down; tape is unloading

Table 4–8 Unit Status Columns (Continued)

Column	Contents
	v = disk is not spinning; tape is unloaded
	= unknown spindle state
W	Write-protection state. For disk drives, a W in this column indicates that the device is hardware write-protected. This column is blank for units that comprise other kinds of devices.
C	Caching state of the device:
	b = read and write-back caching enabled
	r = read caching only
	= caching disabled
KB/S	Average amount of data transferred to and from the unit during the last update interval in 1000-byte increments.
Rd%	Percentage of data transferred between the host and the unit that were read from the unit.
Wr%	Percentage of data transferred between the host and the unit that were written to the unit.
CM%	Percentage of data transferred between the host and the unit were compared. A compare operation can accompany a read or a write operation, so this column is not the sum of columns Rd% and Wr%.
HT%	Cache-hit percentage for data transferred between the host and the unit.
PH%	Partial cache-hit percentage for data transferred between the host and the unit.
MS%	Cache-miss percentage for data transferred between the host and the unit.
Purge	Number of blocks purged from the cache during the last update interval.
BlChd	Number of blocks added to the cache during the last update interval.
BlHit	Number of blocks hit during the last update interval.

Checking Fibre Channel Link Errors

You can also use the display host VTDPY command to check for any channel link errors.

Figure 4–5 Fibre Channel Host Status Display

```
                                                          VTDPY> DISPLAY HOST

   ##          WWN          ID/ALPA   PL    * * * * * * PORT 1 * * * * * *      * * * * * *.PORT 2 * * * * * *
   00    1000-0000-C920-6116    01     2I    Topology              : LOOP     Topology. :          LOOP
   01    1000-0000-C920-2752    02     1I    Current Status        : LOOP     Current Status        : LOOP
   02    1000-0000-C920-611A    04     1I    Current ID/ALPA       :    1     Current ID/ALPA:      2
   03    2000-0000-D10F-F9CA    04     2I         LINK ERROR COUNTERS                LINK ERROR COUNTERS
                                             Link Down             :    0     Link Down :           1
                                             Loss of Signal        :    0     Loss of Signal:       0
                                             Bad Rx Char           :   20     Bad Rx Char:         20
                                             Loss of Sync          :    0     Loss of Sync:         0
                                             Link Fail             :    0     Link Fail :           0
                                             Received EOFa         :    0     Received EOFa:        0
                                             Generated EOFa        :    0     Generated EOFa:       0
                                             Bad CRC               :    0     Cad CRC . :                0
                                             Protocol Error        :    0     Protocol Error:       0

ID world wide name:     alpa

Topology:
LOOP - loop is down
OFFLNE - offline

Current Status:
DOWN - loop is down
LOOP - loop is up and running
STNDBY - standby
```

Note Use the VTDPY>CLEAR command to clear the host display link error counters.

The link error counters displayed in Figure 4–5 are detailed in Figure 4–9 below:

Table 4–9 Frame Manager Link Error Status Counters #1 Register

Field Label	Field Name	Description
Loss of Signal Count	Loss of Signal Count	Indicates the number of times the Frame Manager detected a low to high transition on the lnk_unuse signal.
Bad Rx Char Count	Bad Received Character Count	Indicates the number of times the 8B/10B decode detected an invalid 10-bit code. FC-PH denotes this value as “Invalid Transmission Word during frame reception.” This field may be non-zero after initialization. After initialization, the host should read this value to determine the correct starting value for this error count.
Loss of Sync Count	Loss of Sync Count	Indicates the number of times the loss of sync is greater than RT_TOV.
Link Fail Count	Link Fail Count	Indicates the number of times the Frame Manager detected a NOS or other initialization protocol failure that caused a transition to the Link Failure state.

Table 4–9 Frame Manager Link Error Status Counters #1 Register (Continued)

Field Label	Field Name	Description
Received EOFa	Received EOFa	The number of frames containing an EOFa delimiter that Tachyon has received.
Generated EOFa	Generated EOFa	The number of problem frames that Tachyon has received that caused the Frame Manager to attach an EOFa delimiter. Frames that Tachyon discarded due to internal FIFO overflow are not included in this or any other statistic.
Bad CRC Count	Bac CRC Count	The number of bad CRC frames that Tachyon has received.
Protocol Error Count	Protocol Error Count	The number of protocol errors that the Frame Manager has detected.

Checking for Disk-Drive Problems

Use the disk inline exerciser (DILX) to check the data-transfer capability of disk drives. DILX generates intense read/write loads to the disk drive while monitoring the drive's performance and status. You may run DILX on as many disk drives as you'd like, but because this utility creates substantial I/O loads on the controller, DIGITAL recommends that you stop host-based I/O during the test. You may also use DILX to exercise the read capability of CD-ROM drives.

Finding a Disk Drive in the Subsystem

Follow these steps to find a disk drive or device in the subsystem:

1. Connect a local terminal to the controller.
2. Show the devices that are configured on the controller with the following command:

   SHOW DEVICES
3. Find the device in the enclosure with the following command:

   LOCATE *device-name*

   This command causes the device's LED to blink continuously. Enter the following command to turn off the LED:

   LOCATE CANCEL

Testing the Read Capability of a Disk Drive

To test the read capability of a disk drive:

1. From a host console, dismount logical unit that contains the disk drive you want to test.
2. Connect a local terminal to the controller that accesses the disk drive you want to test.
3. Run DILX with the following command:

   RUN DILX
4. Decline the Auto-configure option so that you can specify the disk drive to test.
5. Accept the default test settings and run the test in read-only mode.
6. Enter the unit number of the disk drive you want to test. For example, to test D107, enter the number 107.

7. If you want to test more than one disk drive, enter their unit numbers when prompted. Otherwise enter “n” to start the test. Use the following control sequences to control DILX during the test:

Table 4–10 DILX Control Sequences

Command	Action
Ctrl/C	Terminates the test
Ctrl/G	Displays the performance summary for the current test and continue testing
Ctrl/Y	Terminates the test and exits DILX

Testing the Read and Write Capabilities of a Disk Drive

Run a DILX Basic Function test to test the read and write capability of a disk drive. During the Basic Function test, DILX runs these four tests. (DILX repeats the last three tests until the time that you set in step 6 on page 4-29 expires.)

- **Write test—**writes specific patterns of data to the disk drive (see Table 4–11 on page 4–30.) DILX does not repeat this test.
- **Random I/O test—**simulates typical I/O activity by issuing read, write, access, and erase commands to randomly-chosen logical block numbers (LBNs). You can set the ratio of these commands as well as the percentage of read and write data that are compared throughout this test. This test takes 6 minutes.
- **Data-transfer test—**tests throughput by starting at an LBN and transferring data to the next LBN that has not been written to. This test takes 2 minutes.
- **Seek test—**stimulates head motion on the disk drive by issuing single-sector erase and access commands. Each I/O uses a different track on each subsequent transfer. You can set the ratio of access and erase commands. This test takes 2 minutes.

To test the read and write capabilities of a specific disk drive:

1. From a host console, dismount the logical unit that contains the disk drive you want to test.
2. Connect a local terminal to the controller that accesses the disk drive you want to test.

3. Run DILX with the following command:

   RUN DILX

4. Decline the auto-configure option so that you can specify the disk drive to test.

Tip Use the auto-configure option if you want to test the read and write capabilities of every disk drive in the subsystem.

---

5. Decline the default settings.
6. Enter the number of minutes you want to run the DILX Basic Function test. (To ensure that DILX accesses the entire disk space, you should enter 120 or more.)
7. Enter the number of minutes between the display of performance summaries.
8. Choose to include performance statistics in the summary.
9. Choose to display both hard and soft errors.
10. Choose to display the hex dump.
11. Accept the hard-error limit default.
12. Accept the soft-error limit default.
13. Accept the queue depth default.
14. Choose option 1 to run a Basic Function test.
15. Enable phase 1, the write test.
16. Accept the default percentage of requests that DILX issues as read requests during phase 2, the random I/O test. DILX issues the balance as write requests.
17. Choose ALL for the data patterns that DILX issues for write requests.
18. Perform the initial write pass.
19. Allow DILX to compare the read and write data.
20. Accept the default percentage of reads and writes that DILX compares.
21. Enter the unit number of the disk drive you want to test. For example, if you want to test D107, you’d enter the number 107.

22. If you want to test more than one disk drive, enter their unit numbers when prompted, otherwise enter "n" to start the test. Use the command sequences shown in Table 4–10 to control the write test.

Table 4–11 Data Patterns for Phase 1: Write Test

Pattern	Pattern in Hexadecimal Numbers
1	0000
2	8B8B
3	3333
4	3091
5	0001, 0003, 0007, 000F, 001F, 003F, 007F, 00FF, 01FF, 03FF, 07FF, 0FFF, 1FFF, 3FFF, 7FFF
6	FIE, FFFC, FFFC, FFFC, FFE0, FFE0, FFE0, FFE0, FE00, FC00, F800, F000, F000, C000, 8000, 0000
7	0000, 0000, 0000, FFFF, FFFF, FFFF, 0000, 0000, FFFF, FFFF, 0000, FFFF, 0000, FFFF, 0000, FFFF
8	B6D9
9	5555, 5555, 5555, AAAA, AAAA, AAAA, 5555, 5555, AAAA, AAAA, 5555, AAAA, 5555, AAAA, 5555, AAAA, 5555
10	DB6C
11	2D2D, 2D2D, 2D2D, D2D2, D2D2, D2D2, 2D2D, 2D2D, D2D2, D2D2, 2D2D, D2D2, 2D2D, D2D2, 2D2D, D2D2
12	6DB6
13	0001, 0002, 0004, 0008, 0010, 0020, 0040, 0080, 0100, 0200, 0400, 0800, 1000, 2000, 4000, 8000
14	FIE, FFFD, FFFB, FFF7, FFEF, FFDF, FFBF, FF7F, FEFF, FDFF, FBFF, F7FF, EFFF, BFFF, DFFF, 7FFF
15	DB6D, B6DB, 6DB6, DB6D, B6DB, 6DB6, DB6D, B6DB, 6DB6, DB6D, B6DB, 6DB6, DB6D
16	3333, 3333, 3333, 1999, 9999, 9999, B6D9, B6D9, B6D9, B6D9, FFFF, FFFF, 0000, 0000, DB6C, DB6C
17	9999, 1999, 699C, E99C, 9921, 9921, 1921, 699C, 699C, 0747, 0747, 0747, 699C, E99C, 9999, 9999
18	FFFF

DILX Error Codes

Table 4–12 explains the error codes that DILX may display during and after testing. Contact your service representative to interpret the end-message fields.

Table 4–12 DILX Error Codes

Error Code	Explanation
1	Illegal Data Pattern Number found in data pattern header. DILX read data from the disk and discovered that the data did not conform to the pattern in which it was previously written.
2	No write buffers correspond to data pattern. DILX read a legal data pattern from the disk, but because no write buffers correspond to the pattern, the data must be considered corrupt.
3	Read data does not match write buffer. DILX compared the read and write data and discovered that they didn’t correspond.

Running the Controller’s Diagnostic Test

During start up, the controller automatically tests its device ports, host port, cache module, and value-added functions. If you’re experiencing intermittent problems with one of these components, you can run the controller’s diagnostic test in a continuous loop rather than restarting the controller over and over again.

To run the controller’s diagnostic test:

1. Connect a local terminal to the controller.
2. Start the self test with one of the following commands:

   SELFTEST THIS CONTROLLER
   SELFTEST OTHER_CONTROLLER

   The self test runs until it detects an error or until you press the controller’s reset button.

   If it detects an error, the self test saves information about the error and produces an OCP LED code for a “daemon hard error.” Restart the controller to write the error information to the host’s error log, then check the log for a “built-in self-test failure” event report. This report will contain an instance code, located at offset 32 through 35, that you can use to determine the cause of the error. See “Translating Event Codes,” page 4–12 for help on translating instance codes.

CHAPTER 5

Replacement Procedures

This chapter describes the procedures for removing, installing, and replacing the controller, cache module, external cache battery, and GLM.

Caution You must shut down the subsystem before removing or replacing any modules. If you remove the controller or any other module without first shutting down the subsystem, data loss will occur.

See the documentation that accompanies the enclosure for information about replacing its hardware, such as the power supplies, cooling fans, cables, and environmental monitoring unit.

Replacing Modules in a Single Configuration

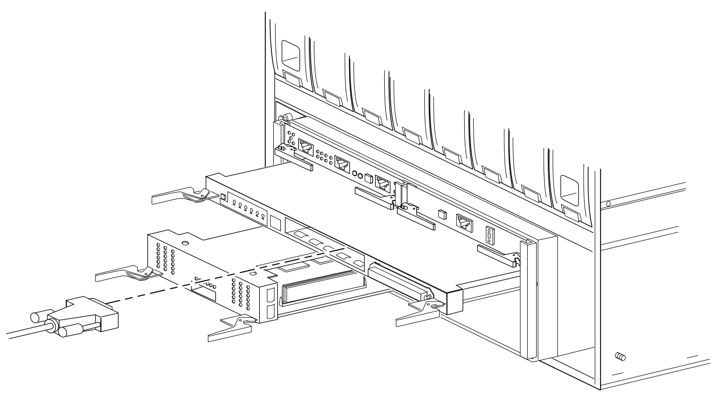
Follow the instructions in this section to replace modules in a single configuration. If you’re replacing modules in a dual-redundant configuration, see “Replacing Modules in a Dual-Redundant Configuration,” page 5–10.

Replacing the Controller and Cache Module

Follow these steps to replace a controller and cache module as shown in Figure 5–1.

You can save the configuration of your controller before you replace it by using the SAVE_ CONFIGURATION switch. This will allow the controller to automatically load the subsystem’s configuration from that device. Otherwise, you’ll have to reconfigure the controller as described in Chapter 2, “Configuring an HSG80 Array Controller.”

Figure 5–1 Replacing the Controller and Cache Module



To remove the controller and cache module...

1. From the host console, dismount the logical units in the subsystem. (If you are using a Windows NT platform, shut down the PC.)
2. If the controller is operating, connect a local terminal to the controller to obtain the last failure codes, if desired.

If the controller is not operating, you must turn off power to the subsystem before removing the controller. Go to Step 4 to turn off power and proceed.

3. Shut down the controller with the following command:

   SHUTDOWN THIS_CONTROLLER

   When the controller shuts down, its reset button and the first three LEDs are lit continuously.
4. Turn off the power to the subsystem.
5. Disconnect the external cache battery (ECB) cable from the cache module.
6. Disable the ECB by pressing its shut off button until the status light stops blinking—about 2 seconds.
7. Remove the ESD cover and program card. Save them for the new controller.
8. Disconnect the hub cables from the controller.
9. Disengage the two retaining levers, and remove the controller.
10. Disengage the two retaining levers, and remove the cache module.

To install the new controller and cache module...

You can save the configuration of your controller before you replace it by using the SAVE_ CONFIGURATION switch. This will allow you to automatically load the subsystem's configuration from that device. Otherwise, you'll have to reconfigure the controller as described in Chapter 2, "Configuring an HSG80 Array Controller."

Caution Make sure you align the controller and cache module in the appropriate pedestal guide rails. If you do not align the modules appropriately, damage to the backplane can occur.

---

1. Insert the new cache module into its slot, and engage its retaining levers.
2. Insert the new controller into its slot, and engage its retaining levers.
3. Connect the ECB cable to the new cache module. Follow the Component Precaution given in the Preface.
4. Connect the hub cables to the new controller.

5. Restore power to the subsystem. The ECB automatically re-enables itself to provide backup power to the cache module.
6. Hold the reset button while inserting the program card into the new controller.
7. Release the reset button, and replace the ESD cover.
8. When the CLI prompt reappears, display details about the controller you configured. Use the following syntax:

   SHOW THIS_CONTROLLER FULL

   See the SHOW THIS_CONTROLLER FULL command in Appendix B, “CLI Commands,” for more information about using this command.
9. If the subsystem worldwide name (node ID) is all zeroes (0000-0000-0000-0000), you must set the ID using the steps in “Restoring Worldwide Names (Node ID Numbers),” page 3–17.
10. Set the port topology for each port.

    SET THIS_CONTROLLER PORT_1_TOPOLOGY=“*topology*”

    SET THIS_CONTROLLER PORT_2_TOPOLOGY=“*topology*”

    If this is a single configuration with a single hub, set PORT 2 off-line.
11. If you selected LOOP_HARD for the port topology, specify the arbitrated loop physical address (ALPA) for the host ports.

    SET THIS_CONTROLLER PORT_1_ALPA=“*address*”

    SET THIS_CONTROLLER PORT_2_ALPA=“*address*”

    See Appendix B, “CLI Commands,” for more information about using the SET OTHER_CONTROLLER PORT_n_ALPA= command.
12. To restore a configuration saved with the SAVE_CONFIGURATION switch, hold button 6 while releasing the reset button.

    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:

    SET THIS_CONTROLLER INITIAL_CONFIGURATION

    The controller shuts down after returning to its initial configuration. Press its reset button to restart the controller.

To restore the configuration, use the CONFIGURATION RESTORE command.

13. Install any patches that you had installed on the previous controller.
14. Mount the logical units on the host. (If you are using a Windows NT platform, restart the PC.)

Replacing the Controller

The instructions for this procedure are similar to those for "Replacing the Controller and Cache Module," page 5–2, except that you should disregard the instructions pertaining to the cache module and the external cache battery.

Replacing the Cache Module

The instructions for this procedure are similar to those for "Replacing the Controller and Cache Module," page 5–2, but the instructions pertaining to the controller do not apply.

Replacing the External Cache Battery

Follow these instructions to remove and replace a single-battery external cache battery (ECB):

To remove the ECB...

1. If the controller and cache module are not operating, go to step 4. Otherwise, go to the next step.
2. Connect a local terminal to the controller.
3. Shut down the controller with the following command:

   ```
   SHUTDOWN THIS_CONTROLLER
   ```

   When the controller shuts down, its reset button and first three LEDs are lit continuously.
4. Turn off the power to the subsystem.

To install the new ECB...

1. Insert the new ECB into its slot.
2. Connect the open end of the ECB "Y" cable to the new ECB.
3. Restore power to the subsystem. The controller automatically restarts.

4. Wait until the new ECB's status light is on continuously, then disconnect the ECB cable from the old ECB.

Caution Do not disconnect the old ECB until the batteries in the new ECB are fully charged. The new ECB's status light will be lit continuously to indicate that its batteries are fully charged. A blinking status light indicates that its batteries are charging.

You may operate the subsystem regardless of the old ECB's battery status, but do not disconnect the old ECB until the batteries in the new ECB are fully charged.

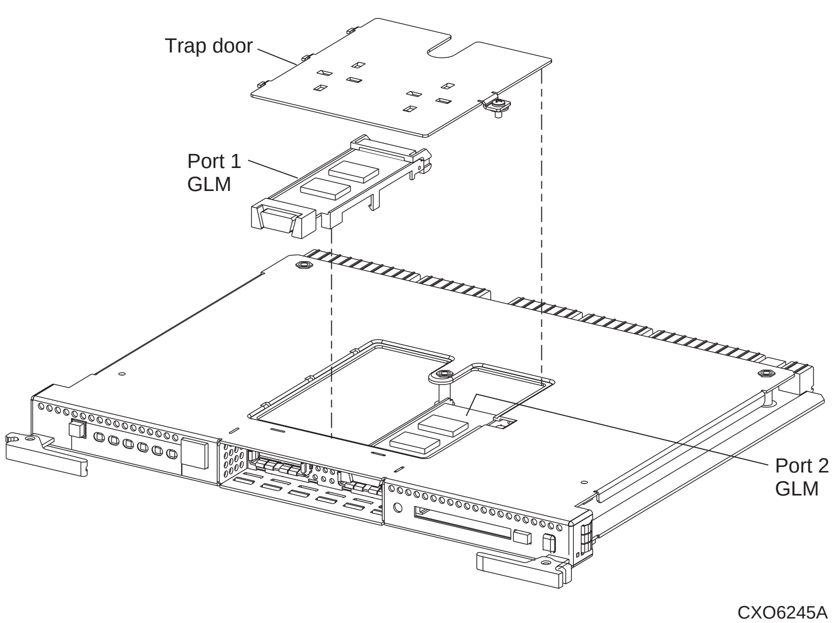
---

5. Remove the old ECB.

Replacing the GLM

Follow these steps to replace a GLM in a single controller subsystem. Figure 5–2 shows the location and orientation of the GLMs.

Figure 5–2 Location of GLMs in Controller



1. Remove the controller using the steps in “Replacing the Controller and Cache Module,” page 5–2. Disregard the steps for removing the cache module.
2. Remove the screw that secures the trap door on the top of the controller.
3. Remove the trap door, and set it aside.
4. Use your index finger and thumb to operate the release lever on the exposed end of the GLM. Press the lower end of the release lever with your index finger while pulling the raised end of the release lever up with your thumb.
5. Remove the GLM.
6. Before inserting the new GLM, notice the holes in the board where the GLM will reside.
7. Insert the new GLM by first placing the cable connection end of the GLM through the opening on the front of the controller.
8. Line up the guide pins on the bottom of the replacement GLM with the holes in the board. Press firmly to seat the GLM.
9. Press the release lever firmly into place to secure the GLM.
10. Insert the controller into its slot, and engage its retaining levers.
11. Connect the hub cables to the controller.
12. Restore power to the subsystem. The ECB automatically re-enables itself to provide backup power to the cache module.
13. Hold the reset button while inserting the program card into the controller.
14. Release the reset button, and replace the ESD cover.
15. Mount the logical units on the host. (If you are using a Windows NT platform, restart the PC.)

Removing the Controller and Cache Module

The instructions for this procedure are the same as the instructions for removing a controller in “Replacing the Controller and Cache Module,” page 5–2.

Removing the Controller

The instructions for this procedure are similar to the instructions for removing a controller in “Replacing the Controller and Cache Module,” page 5–2, except that you should disregard instructions for removing the cache module and the external cache battery.

Caution Your cache module may contain data if the controller crashed and you weren’t able to shut it down with the SHUTDOWN THIS_CONTROLLER command.

If you won’t be installing a new controller within 24 hours, and if the cache module is still holding data, then restore the power to the subsystem after you’ve removed the failed controller. The cache module will hold its data as long as it receives power from the subsystem’s primary or secondary (UPS) power source or an ECB. An ECB only provides power for 24 hours.

Removing the Cache Module

The instructions for this procedure are similar to the instructions for removing a cache module in “Replacing the Controller and Cache Module,” page 5–2, except that you should disregard instructions for removing the controller module.

Installing a Controller and Cache Module

Follow these steps to install a controller and its cache module into an empty subsystem. If you’re upgrading a single-configuration subsystem to a dual-redundant subsystem, follow the instructions for “Upgrading to a Dual-Redundant Controller Configuration,” page 6–10.

1. Turn off the power to the subsystem.
2. Insert the controller into the top slot—controller slot A (see Figure 1–4 on page 1-7). This slot responds to SCSI target ID number 7.
3. Insert the cache module into the bottom left slot—“cache slot A.”
4. Connect the external cache battery (ECB) cable to the cache module.
5. Connect an open end of the ECB “Y” cable to the ECB.
6. Connect the hub cables to the new controller.

7. Restore power to the subsystem. The ECB automatically re-enables itself to provide backup power to the cache module.

   If the subsystem contains a device that was initialized with the SAVE_CONFIGURATION switch, the controller automatically loads the subsystem's configuration from that device. Otherwise, you'll have to reconfigure the controller as described in Chapter 2, "Configuring an HSG80 Array Controller."

   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:

   SET THIS_CONTROLLER INITIAL_CONFIGURATION

   The controller shuts down after returning to its initial configuration. Press its reset button to restart the controller.

   To restore the configuration, use the CONFIGURATION RESTORE command.

8. Hold the reset button while inserting the program card into the new controller.
9. Release the reset button, and replace the ESD cover.
10. Install any patches that you had installed on the previous controller.
11. Mount the logical units on the host.

Installing a Controller

The instructions for this procedure are similar to the instructions for installing a controller in "Installing a Controller and Cache Module," page 5–8, except that you should disregard instructions for installing the cache module and the external cache battery.

Installing a Cache Module

The instructions for this procedure are similar to the instructions for installing a cache module in "Installing a Controller and Cache Module," page 5–8. Disregard instructions for installing the controller module.

Replacing Modules in a Dual-Redundant Configuration

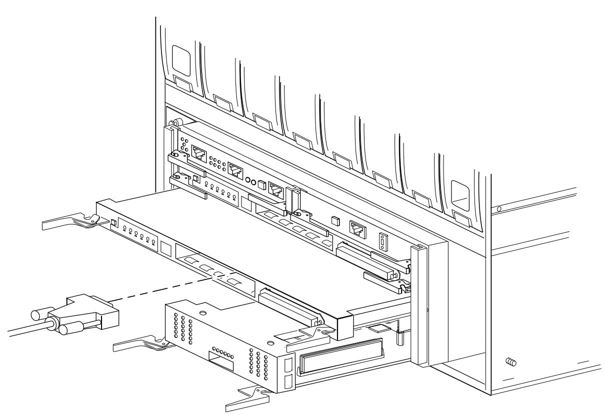
Follow the instructions in this section to replace modules in a dual-redundant configuration. If you’re replacing modules in a single configuration, see “Replacing Modules in a Single Configuration,” page 5–2.

Replacing a Controller and Cache Module

Follow these steps to replace a controller and its cache module as shown in Figure 5–3. Note the following before starting the replacement procedures.

- The new controller’s hardware must be compatible with the functioning controller’s hardware. See the product-specific release notes that accompanied the software release for information regarding hardware compatibility.
- The software versions must be the same on both controllers.
- The new cache module must contain the same amount of memory as the module it’s replacing.

Figure 5–3 Replacing the Controller and its Cache Module



CXO6189A

To remove the controller and its cache module...

1. Connect a local terminal to the operational controller. The controller to which you're connected is "this controller;" the controller that you're removing is the "other controller."
2. Shut down the controllers. First shut down the "other controller," then shut down "this controller" with the following commands:

   SHUTDOWN OTHER_CONTROLLER

   SHUTDOWN THIS_CONTROLLER
3. When the controllers shut down, their reset buttons and their first three LEDs are lit continuously. This may take several minutes, depending on the amount of data that needs to be flushed from the cache modules.
4. Turn off the power to the subsystem.
5. Unplug the subsystem's power cord.
6. Disable the ECBs by pressing their shut-off buttons until their status lights stop blinking—about two seconds.
7. Disconnect the external cache battery (ECB) cable from the "other controller's" cache module.
8. Disengage the two retaining levers, and remove the "other controller."
9. Disengage the two retaining levers, and remove the "other controller's" cache module.

To install the new controller and new cache module...

1. Disable the ECB to which you're connecting the new cache module by pressing its shut-off button until the status light stops blinking—about 2 seconds.
2. Insert the new controller module.
3. Connect the ECB cable to the new cache module. Follow the Component Precaution given in the Preface.
4. Insert the new cache module into its slot and engage its retaining levers.
5. Ensure that the program card is not in the new controller, and insert the new controller into its slot. Engage its retaining levers.
6. Connect the hub cables to the new controller.
7. Plug in the subsystem's power cord.
8. Turn on the subsystem. The ECBs automatically re-enable themselves to provide backup power to the cache modules.

9. Hold both reset buttons while inserting the program cards into both controllers, and replace the ESD covers.
10. If the subsystem worldwide name (node ID) is all zeroes (0000-0000-0000-0000), you must set the ID using the steps in "Restoring Worldwide Names (Node ID Numbers)," page 3–17.
11. Set the port topology for each port.

    SET THIS_CONTROLLER PORT_1_TOPOLOGY="*topology*"

    SET THIS_CONTROLLER PORT_2_TOPOLOGY="*topology*

    If this is a dual-redundant configuration, the "other controller" inherits "this controller's" port topology.
    See Appendix B, "CLI Commands," for more information about using the SET THIS_CONTROLLER PORT_n_TOPOLOGY= command.
12. If you selected LOOP_HARD for the port topology, specify the arbitrated loop physical address (ALPA) for the host ports.

    SET THIS_CONTROLLER PORT_1_ALPA="*address*"

    SET THIS_CONTROLLER PORT_2_ALPA="*address*"

    The "other controller" inherits "this controller's" port ALPA addresses.
13. See Appendix B, "CLI Commands," for more information about using the SET OTHER_CONTROLLER PORT_n_ALPA= command.
14. Enable failover, and re-establish the dual-redundant configuration with the following command:

    SET FAILOVER COPY=THIS_CONTROLLER

    This command copies the subsystem's configuration from "this controller" to the new controller.

Replacing a Controller

The instructions for this procedure are similar to those for "Replacing the Controller and Cache Module," page 5–2, except that you should disregard the instructions pertaining to the cache module and the external cache battery.

Replacing a Cache Module

Follow these steps to replace a cache module. The new cache module must contain the same amount of memory as the module it's replacing.

To remove the cache module...

1. Shut down the controllers. First shut down the “other controller,” then shut down “this controller” with the following commands:

   SHUTDOWN OTHER_CONTROLLER

   SHUTDOWN THIS_CONTROLLER

2. When the controllers shut down, their reset buttons and their first three LEDs are lit continuously. This may take several minutes, depending on the amount of data that needs to be flushed from the cache modules.
3. Turn off the power to the subsystem.
4. Unplug the subsystem’s power cord.
5. Disconnect the ECB cable from the cache module you’re replacing.
6. Disengage the two retaining levers and remove the “other controller’s” cache module.

To install the new cache module...

1. Disable the ECB to which you’re connecting the new cache module by pressing its shut-off button until the status light stops blinking—about 2 seconds.
2. Connect the ECB cable to the new cache module. Follow the Component Precaution given in the Preface.
3. Insert the new cache module into its slot, and engage its retaining levers.
4. Plug in the subsystem’s power cord.
5. Turn on the subsystem. The controllers automatically restart, and the ECBs automatically re-enable themselves to provide backup power to the cache modules.

Replacing Both Controllers (and Their Cache Modules)

If you’re replacing the controllers only, repeat the instructions for “Replacing a Controller,” page 5–12, for each controller module.

If you’re replacing the controllers and their cache modules, repeat the instructions for “Replacing a Controller and Cache Module,” page 5–10, for each controller and cache module.

Replacing an External Cache Battery

Follow these steps to replace an external cache battery (ECB):

To disconnect the failed ECB...

1. Shut down the controllers. First shut down the “other controller,” then shut down “this controller” with the following commands:

   SHUTDOWN OTHER_CONTROLLER

   SHUTDOWN THIS_CONTROLLER

2. When the controllers shut down, their reset buttons and their first three LEDs are lit continuously. This may take several minutes, depending on the amount of data that needs to be flushed from the cache modules.
3. Turn off the power to the subsystem.
4. Unplug the subsystem’s power cord.

To install the new ECB...

1. Insert the new ECB into an empty slot.
2. Connect the open end of the ECB “Y” cable to the new ECB.
3. Plug in the subsystem’s power cord.
4. Turn on the subsystem. The controllers automatically restart, and the ECBs automatically re-enable themselves to provide backup power to the cache modules
5. Wait until the new ECB’s status light is on continuously, then disconnect the ECB cable from the old ECB.

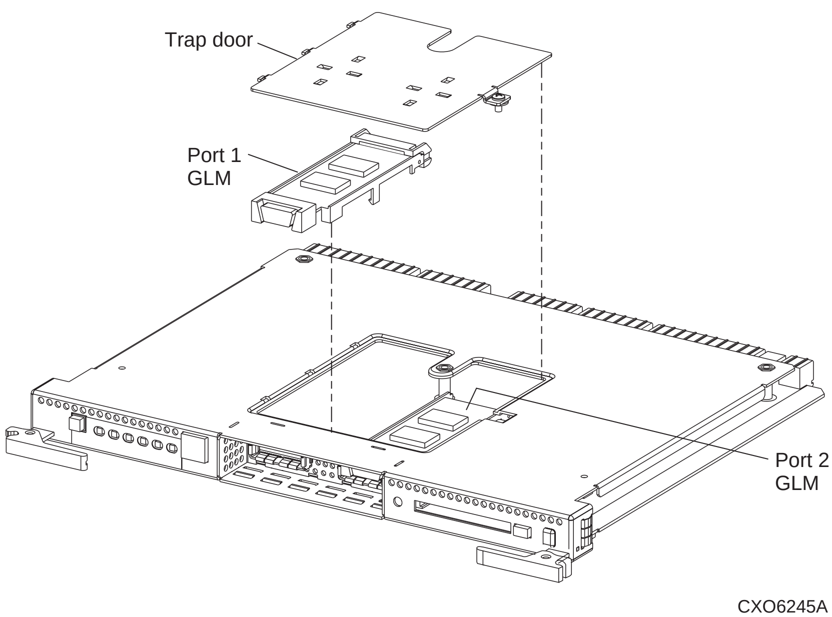
Caution Do not disconnect the old ECB until the batteries in the new ECB are fully charged. The new ECB’s status light will be lit continuously to indicate that its batteries are fully charged. A blinking status light indicates that its batteries are charging. You may operate the subsystem regardless of the old ECB’s status.

6. If you want to backup both cache modules with the new ECB, repeat this procedure for the other controller and its cache module.
7. Remove the old ECB.

Replacing the GLM

Follow these steps to replace a GLM in a controller in a dual-redundant configuration. Figure 5–4 shows the location and orientation of the GLMs.

Figure 5–4 Location of GLMs in Controller



1. Remove the controller using the steps in “Replacing a Controller and Cache Module,” page 5–10, disregarding the steps for removing the cache module.
2. Remove the screw that secures the trap door on the top of the controller.
3. Remove the trap door, and set it aside.
4. Use your index finger and thumb to operate the release lever on the exposed end of the GLM. Press the lower end of the release lever with your index finger while pulling the raised end of the release lever up with your thumb.
5. Remove the GLM.

6. Before inserting the new GLM, notice the holes in the board where the GLM will reside.
7. Insert the new GLM by first placing the cable connection end of the GLM through the opening on the front of the controller.
8. Line up the guide pins on the bottom of the replacement GLM with the holes in the board, and press firmly to seat the GLM.
9. Press the release lever firmly into place to secure the GLM.
10. Ensure that the program card is not in the controller you are re-installing, and insert the controller into its slot. Engage its retaining levers.
11. Connect the hub cables to the controller.
12. Plug in the subsystem’s power cord.
13. Turn on the subsystem. The controllers automatically restart, and the ECBs automatically re-enable themselves to provide backup power to the cache modules.
14. While holding down both controllers’ reset buttons, insert both program cards.
15. Release the reset buttons simultaneously, and replace the ESD cover. (The program cards must be in the controllers when the controllers are restarted.)
16. Restart both controllers. Both controllers must be restarted simultaneously.

Removing a Controller and Cache Module

The instructions for this procedure are the same as the instructions for removing the controller and cache module in “Replacing a Controller and Cache Module,” page 5–10.

Removing a Controller

The instructions for this procedure are the same as the instructions for removing the controller in “Replacing a Controller,” page 5–12.

Removing a Cache Module

The instructions for this procedure are the same as the instructions for removing the cache module in “Replacing a Cache Module,” page 5–12.

Installing a Controller and Cache Module

Follow these steps to install a new controller and cache module into a subsystem that formerly used two controllers in a dual-redundant configuration:

1. Shut down the controllers. First shut down the “other controller,” then shut down “this controller” with the following commands:

   SHUTDOWN OTHER_CONTROLLER

   SHUTDOWN THIS_CONTROLLER

2. When the controllers shut down, their reset buttons and their first three LEDs are lit continuously. This may take several minutes, depending on the amount of data that needs to be flushed from the cache modules.
3. Turn off the power to the subsystem.
4. Unplug the subsystem’s power cord.
5. Connect the ECB cable to the new cache module. Follow the Component Precaution given in the Preface.
6. Ensure that the program card is not in the new controller, and insert the new controller into its slot. Engage its retaining levers.
7. Insert the new cache module into its slot. Engage its retaining levers.
8. Connect the hub cables to the new controller.
9. Plug in the subsystem’s power cord.
10. Turn on the subsystem.

    The controllers automatically restart, and the ECBs automatically re-enable themselves to provide backup power to the cache modules.

11. Insert the program card into the new controller, and replace its ESD cover.
12. Enable failover and re-establish the dual-redundant configuration with the following command:

    SET FAILOVER COPY=THIS_CONTROLLER

    This command copies the subsystem’s configuration from the “this controller” to the new controller.

Installing a Controller

Follow these steps to install a controller into a subsystem that formerly used two controllers in a dual-redundant configuration:

1. Shut down the controllers. First shut down the “other controller,” then shut down “this controller” with the following commands:

   SHUTDOWN OTHER_CONTROLLER

   SHUTDOWN THIS_CONTROLLER

2. When the controllers shut down, their reset buttons and their first three LEDs are lit continuously. This may take several minutes, depending on the amount of data that needs to be flushed from the cache modules.
3. Turn off the power to the subsystem.
4. Unplug the subsystem’s power cord.
5. Ensure that the program card is not in the new controller, and insert the new controller into its slot. Engage the new controller’s retaining levers.
6. Connect the hub cables to the new controller.
7. Plug in the subsystem’s power cord.
8. Turn on the subsystem.
9. While holding both controllers’ reset buttons, insert the program cards into the controllers.
10. Release the reset button simultaneously, and replace the program cards’ ESD covers.
11. Enable failover and re-establish the dual-redundant configuration with the following command:

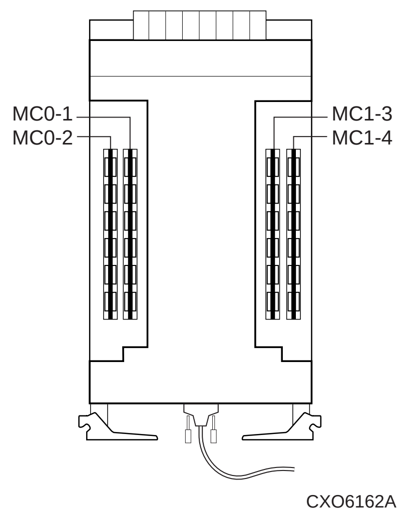
    SET FAILOVER COPY=THIS_CONTROLLER

    This command copies the subsystem’s configuration from the “this controller” to the new controller.

Replacing DIMMs

The cache module contains 128MB (4 x 32MB) of memory as shown in Figure 5–5.

Figure 5–5 Cache-Module Memory Configuration



If a DIMM fails, note which DIMM you need to replace based on the diagram that displays on your console, and follow these steps to replace the DIMM in dual-redundant configurations. For single configurations, disregard references to the “other controller”. Read the plural “controllers,” “cache modules,” etc., as the singular “controller,” “cache module,” and so on.

1. From a host console, dismount the storage units in the subsystem.
2. Connect a local terminal to one of the controllers in the subsystem. The controller to which you’re connected is “this controller.”
3. Shut down both controllers in the following order with these commands:

   SHUTDOWN OTHER_CONTROLLER
   SHUTDOWN THIS_CONTROLLER

   When the controllers shut down, their reset buttons and their first three LEDs are lit continuously.
4. Turn off the power to the subsystem.
5. Disconnect the external cache battery (ECB) cables from both cache modules.

6. Disable both ECBs by pressing their shut-off buttons until their status lights stop blinking—about 2 seconds.
7. Disengage the two retaining levers on each cache module, and remove both cache modules.

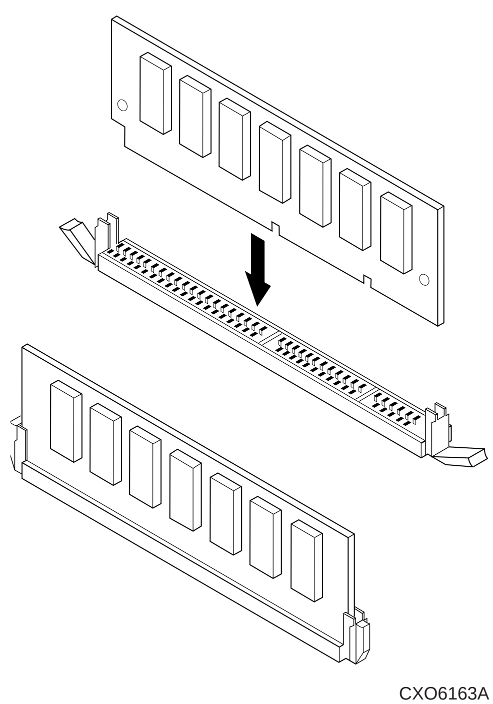
To remove the DIMMs...

1. Press down on the DIMM retaining levers at either end of the DIMM you are removing.
2. Grasp the DIMM, and gently remove it from the DIMM slot.

To install the DIMMs...

1. Insert the DIMM straight into the socket as shown in Figure 5–6.
2. Press the DIMM gently until it’s seated in the socket.
3. Double-check to ensure both ends of the DIMM are firmly seated in the slot and both retaining clips engage the DIMM.

Figure 5–6 Installing a DIMM



To re-install the cache modules...

1. Insert each cache module into its slot, and engage its retaining levers. Connect the ECB cables to the cache modules.
2. Restore power to the subsystem. The controllers automatically restart, and the ECBs automatically re-enable themselves to provide backup power to the cache modules.
3. Mount the storage units on the host.

Replacing a Fibre Cable or Hub

Follow these steps to replace a cable connected to either side of your hub or to replace the hub itself:

To remove an existing fibre cable or a hub...

1. Shut down the host system.
2. Shut down the controllers. If this is a dual-redundant configuration, first shut down the “other controller,” then shut down “this controller” with the following commands. If this a single configuration, shut down “this controller”:

   SHUTDOWN OTHER_CONTROLLER

   SHUTDOWN THIS_CONTROLLER
3. When the controllers shut down, their reset buttons and their first three LEDs are lit continuously. This may take several minutes, depending on the amount of data that needs to be flushed from the cache modules.
4. If you are replacing a cable, unplug the failed cable at each end. If you are replacing a hub, unplug all of the cables connected to the hub.

To install the new fibre cable or hub...

1. If you are replacing a cable, plug the replacement cable into the ports that the removed cable was plugged into.

   If you are replacing a hub, plug all of the cables that were unplugged from the removed hub into the replacement hub.
2. Push both reset buttons to restart the controllers.

   The controllers automatically restart. Your subsystem is now ready for operation.

Replacing Your PCMCIA Card

Follow these steps to replace a PCMCIA card:

1. Shut down the controller that contains the PCMCIA card you want to replace. For example, in a dual-redundant configuration, if the “other controller” contains the PCMCIA card, enter the following command:

   SHUTDOWN OTHER_CONTROLLER

2. When the controller shuts down, its reset button and the first three LEDs are lit continuously. This may take several minutes, depending on the amount of data that needs to be flushed from the cache module belonging to the controller being shut down.
3. When the reset button stays lit continuously, push the button.
4. While pushing in the reset button, eject the PCMCIA card from the controller by pushing in on the program-card eject button.
5. Insert the new PCMCIA card into the PCMCIA card slot.
6. Release the reset button.
7. Restart the controller that contains the replaced card. For example, in a dual-redundant configuration, if the “other controller” contains the new PCMCIA card, enter the following command:

   RESTART OTHER_CONTROLLER

   The controller restarts. Your subsystem is now ready for operation.

Replacing a Failed Storageset Member

If a disk drive fails in a RAIDset or mirrorset, the controller automatically places it into the failedset. If the spareset contains a replacement drive that satisfies the storageset’s replacement policy, the controller automatically replaces the failed member with the replacement drive. If the spareset is empty or doesn’t contain a satisfactory drive, the controller simply “reduces” the storageset so that it can operate without one of its members.

The storageset remains in this reduced state until the spareset contains a satisfactory drive. When the controller senses a satisfactory drive in the spareset, it automatically places the drive into the storageset and restores the storageset to normal. Therefore, replacing a failed storageset member means putting a satisfactory drive into the spareset.

To remove a failed RAIDset or mirrorset member...

1. Connect a local terminal to a controller that accesses the reduced RAIDset or mirrorset.
2. Enable AUTOSPARE with the following command:

   ```
   SET FAILEDSET AUTOSPARE
   ```

   With AUTOSPARE enabled, any new disk drive that you insert into the PTL location of a failed disk drive is automatically initialized and placed into the spareset.
3. Remove the failed disk drive.

To install the new member...

1. Insert a new disk drive that satisfies the replacement policy of the reduced storageset into the PTL location of the failed disk drive.

Note The controller automatically initializes the new disk drive and places it into the spareset. As soon as it becomes a member of the spareset, the controller automatically uses the new disk drive to restore the reduced RAIDset or mirrorset. If initialization fails, the new disk drive is placed into the failedset. Insert another disk drive.

Shutting Down the Subsystem

Caution Controller warm swap is not supported. You must shut down the controller to replace modules.

Follow these steps to shut down a subsystem:

1. From a host console, dismount the logical units in the subsystem.
2. Connect a maintenance terminal to one of the controllers in your subsystem.
3. Shut down the controllers. In single configurations, you only need to shut down “this controller.” In dual-redundant configurations, shut down the “other controller” first, then shut down “this controller” with the following commands:

   ```
   SHUTDOWN OTHER_CONTROLLER
   SHUTDOWN THIS_CONTROLLER
   ```

   When the controllers shut down, their reset buttons and their first three LEDs are lit continuously. This may take several minutes, depending on the amount of data that needs to be flushed from the cache modules.
4. Turn off the power to the subsystem.
5. Unplug the subsystem’s power cord.
6. Disable the ECBs by pressing their shut off buttons until their status lights stop blinking—about two seconds.

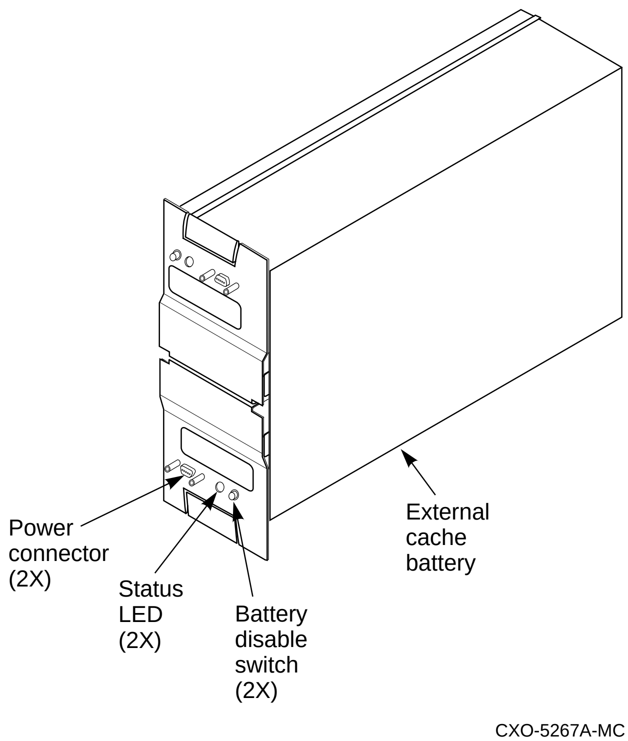
Caution If you are shutting down your controller for longer than one day, complete the additional steps in “Shutting Down the Subsystem for an Extended Time,” page 5–25. This will prevent the write-back cache batteries from discharging.

Shutting Down the Subsystem for an Extended Time

To shut down the subsystem for an extended time, complete the steps for shutting down the subsystem, then continue with these steps:

1. Press the battery disable switch, the small button labeled SHUT OFF next to the flashing status LED on the ECB, and hold it in for approximately 2 seconds. The cache LED will flash once, then shut off.

Figure 5–7 Battery Disable Switch



2. The battery is no longer powering the cache.
3. To return to normal operation, the cache battery is enabled when the battery cable is reconnected and the RAID subsystem is powered on.

Restarting the Subsystem

Follow these steps to restart a subsystem:

1. Plug in the subsystem’s power cord.
2. Turn on the subsystem. The controllers automatically restart, and the ECBs automatically re-enable themselves to provide backup power to the cache modules.

CHAPTER 6

Upgrading the Subsystem

This chapter provides instructions for upgrading subsystems that use the controller, cache module, and external cache battery.

Upgrading Controller Software

You can upgrade the controller’s software two ways:

- Install a new program card that contains the new software.
- Download a new software image, and use the menu-driven Code Load/Code Patch utility (CLCP) to write it onto the existing program card. (You may also use this utility to list, install, and delete patches to the controller software.)

Installing a New Program Card

Follow these steps to install a program card that contains the new software. If you’re only upgrading the software in a single-configuration controller, disregard references to the “other controller” and read the plural “controllers” as the singular “controller.”

To upgrade the software by installing a new program card:

1. From the host console, dismount the storage units in the subsystem.
2. Connect a local terminal to one of the controllers.
3. Shut down the controllers with the following commands:

```
SHUTDOWN OTHER_CONTROLLER
SHUTDOWN THIS_CONTROLLER
```

When the controllers shut down, their reset buttons and first three LEDs are lit continuously.

Caution Do not change the subsystem’s configuration or replace any of its modules until you’ve completed this procedure to upgrade the controller software.

4. Remove the program card electrostatic discharge (ESD) covers.
5. Press and hold the reset button while ejecting the program card on each controller. Release the reset buttons after you’ve removed the program cards.
6. Press and hold the reset button while inserting the new program cards; the controllers automatically restart. Their reset buttons will flash about once per second to indicate that they’re ready to handle I/O.
7. Replace the ESD covers, and push their pins to lock them into place.
8. Mount the storage units on the host.

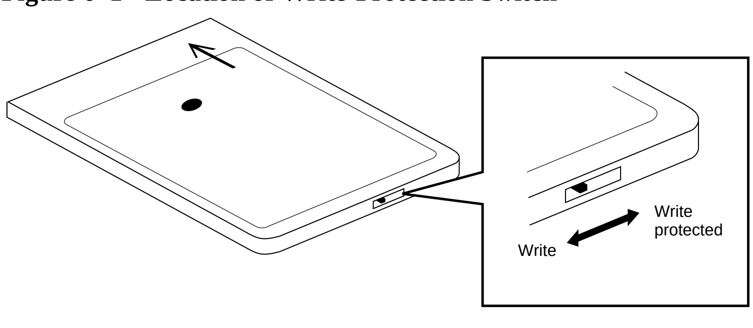
Downloading New Software

Use the CLCP to download new software to the program card while it’s installed in the controller.

To upgrade the software with CLCP:

1. Obtain the new software image file from a customer service representative.
2. Load the image onto a PC using the file- or network-transfer capabilities of your PC.
3. Disable any screen saver or terminate-and-stay-resident programs on your PC that may activate during the CLCP process.
4. From a host console, dismount the storage units in the subsystem.
5. Connect the PC to the controller’s local connection port (you’ll need the PC-serial port adapter shown in Figure 1–3 on page 1–6).
6. Configure the KERMIT transfer protocol on the PC to 19200 baud, eight bits, no parity, and one stop bit.
7. Remove the ESD cover. If your program card is equipped with a write-protection switch, disable write-protection by sliding the switch to the left as shown in Figure 6–1.

Figure 6–1 Location of Write-Protection Switch



8. From a terminal emulator, start CLCP with the following command.

   RUN CLCP

9. Choose option 1 from the CLCP Main Menu to start the Code LOAD local program.

Figure 6–2 CLCP Main Menu

```
Select an option from the following list:
Code Load & Code Patch local program Main Menu
0: Exit
1: Enter Code LOAD local program
2: Enter Code PATCH local program
Enter option number  (0..2) [0] ?
```

10. Use KERMIT to transfer the binary image from the PC to the controller.

    When the download is complete, CLCP automatically writes the new image to the program card and restarts the controller.

11. Verify that the controller is running the new software version with the following command:

    SHOW THIS CONTROLLER

12. Slide the program card’s write-protection switch to the right to re-enable write protection.
13. Replace the ESD cover.
14. Repeat the procedure to upgrade the other controller in dual-redundant subsystems.
15. Mount the storage units in the subsystem.

Using CLCP to List, Install, and Delete Software Patches

Use option 2 from the CLCP Main Menu to manage software patches. These small programming changes are placed into the controller’s NVMEM and become active as soon you restart the controller. The controller reserves enough NVMEM memory for about ten patches, depending upon the size of the patches you’re installing.

Keep the following points in mind while installing or deleting patches:

- Patches are associated with specific software versions. CLCP verifies the patch against the currently installed version.
- Patches are sequential: patch one must be entered before patch two and so on.
- Deleting one patch also deletes all higher-numbered patches. For example, if you delete patch two, you’ll automatically delete patches three, four, and so on.

- Controllers in a dual-redundant configuration must have the same patches applied. You must install patches into each controller separately.

To install, delete, or list software patches:

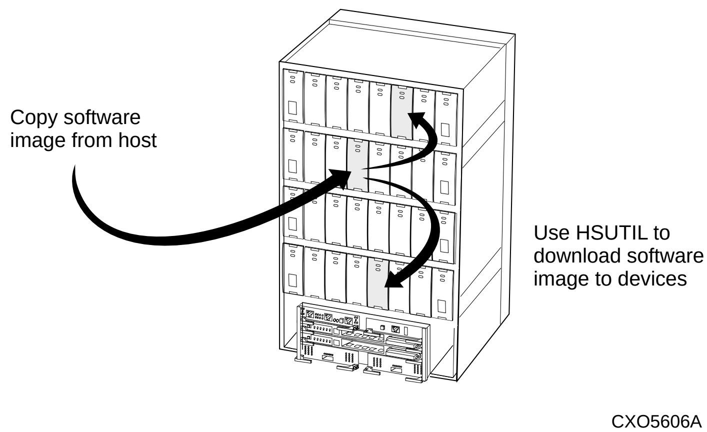
1. Obtain the patch file from a customer service representative.
2. Load the patch file onto a PC using the file- or network-transfer capabilities of your PC.
3. Connect the PC to the controller’s local connection port (you’ll need the PC-serial port adapter shown in Figure 1–3 on page 1–6).
4. Configure the KERMIT transfer protocol on the PC to 19200 baud, eight bits, no parity, and one stop bit.
5. Start CLCP with the following command:

   RUN CLCP
6. Choose option 2 from the CLCP Main Menu to start the Code PATCH local program.
7. Choose option 1, 2, or 3 to install, delete, or list patches, respectively.
8. Press the controller’s reset button to restart the controller if you installed or deleted patches.

Upgrading Firmware on a Device

Use HSUTIL to upgrade a device with firmware located in contiguous blocks at a specific LBN on a source disk drive configured as a unit on the same controller. Upgrading firmware on a disk or tape drive is a two-step process as shown in Figure 6–3: first, copy the new firmware from your host to a disk drive configured as a unit in your subsystem; then use HSUTIL to load the firmware onto the devices in the subsystem.

Figure 6–3 Upgrading Device Firmware



Keep the following points in mind while using HSUTIL to upgrade firmware on a device:

- HSUTIL has been tested with the qualified devices listed in the product-specific release notes that accompanied the software release. You may attempt to install firmware on unsupported devices—HSUTIL won’t prevent this—but if the upgrade fails, the device may be rendered unusable and therefore require the manufacturer’s attention.
- If the power fails or the bus is reset while HSUTIL is installing the new firmware, the device may become unusable. To minimize this possibility, DIGITAL recommends that you secure a reliable power source and suspend all I/O to the bus that services the device you’re upgrading.

- HSUTIL cannot install firmware on devices that have been configured as single disk drive units or as members of a storageset, spareset, or failedset. If you want to install firmware on a device that has previously been configured as a single disk drive, delete the unit number and storageset name associated with it.
- During the installation, the source disk drive is not available for other subsystem operations.
- Some devices may not reflect the new firmware version number when viewed from the "other" controller in a dual-redundant configuration. If you experience this, enter the following CLI command: CLEAR_ERRORS *device-name* UKNOWN.
- Do not issue any CLI commands that access or inspect devices being formatted.

To upgrade firmware with HSUTIL:

1. Connect a local terminal to the controller that accesses the device you want to upgrade.
2. Configure a single-disk unit. (In the next steps, you'll copy the firmware image to this unit, then use HSUTIL to distribute it to the devices you're upgrading. This unit must be a newly initialized disk with no label or file structure to ensure that the firmware image resides in contiguous blocks starting from LBN 0 or another known LBN.)

   See "Configuring a Single-Disk Unit," page 3–53, for instructions on configuring a single-disk unit.
3. Copy the firmware image to the single-disk unit that you configured in step 2. The firmware image must begin at a known LBN—usually 0—and must be contiguous. See the documentation that accompanied your host's operating system for instructions on copying firmware images to a disk drive.
4. Start HSUTIL with the following command:

   RUN HSUTIL
5. Choose option 2 or 3 from the HSUTIL menu.
6. Choose the single-disk unit as the source disk for the download.
7. Enter the starting LBN of the firmware image—usually LBN 0.
8. Enter the product ID of the device you want to upgrade. This ID corresponds to the product information that's reported in the "Type" column when you issue the SHOW DISK FULL command.

HSUTIL lists all devices that correspond to the product ID you entered.

9. Enter the disk or tape name of the device you want to upgrade.
10. Confirm or enter the byte count of the firmware image.
11. Confirm the download.
12. Some disk firmware releases require that you reformat the disk after upgrading its firmware. See the documentation that accompanied the firmware to determine if you need to reformat the device.
13. When HSUTIL finishes downloading the firmware, it displays the new firmware revision for the disk drive.

HSUTIL Messages

While you are formatting disk drives or installing new firmware, HSUTIL may produce one or more of the messages in Table 6-1 (many of the self-explanatory messages have been omitted).

Table 6–1 HSUTIL Messages and Inquiries

Message	Description
Insufficient resources	HSUTIL cannot find or perform the operation because internal controller resources are not available.
Unable to change operation mode to maintenance for unit	HSUTIL was unable to put the source single disk drive unit into maintenance mode to enable formatting or code load.
Unit successfully allocated	HSUTIL has allocated the single disk drive unit for code load operation. At this point, the unit and its associated device are not available for other subsystem operations.
Unable to allocate unit	HSUTIL could not allocate the single disk drive unit. An accompanying message explains the reason.
Unit is owned by another sysop	Device cannot be allocated because it is being used by another subsystem function or local program.
Unit is in maintenance mode	Device cannot be formatted or code loaded because it is being used by another subsystem function or local program.
Exclusive access is declared for unit	Another subsystem function has reserved the unit shown.
The other controller has exclusive access declared for unit	The companion controller has locked out this controller from accessing the unit shown.
The RUNSTOP_SWITCH is set to RUN_DISABLED for unit	The RUN\NORUN unit indicator for the unit shown is set to NORUN; the disk cannot spin up.

Table 6–1 HSUTIL Messages and Inquiries (Continued)

Message	Description
What BUFFER SIZE, (in BYTES), does the drive require (2048, 4096, 8192) [8192]?	HSUTIL detects that an unsupported device has been selected as the target device and the firmware image requires multiple SCSI Write Buffer commands.You must specify the number of bytes to be sent in each Write Buffer command. The default buffer size is 8192 bytes. A firmware image of 256 K, for example, can be code loaded in 32 Write Buffer commands, each transferring 8192 bytes.
What is the TOTAL SIZE of the code image in BYTES [*device default*]?	HSUTIL detects that an unsupported device has been selected as the target device. You must enter the total number of bytes of data to be sent in the code load operation.
Does the target device support only the download microcode and save?	HSUTIL detects that an unsupported device has been selected as the target device. You must specify whether the device supports the SCSI Write Buffer command’s download and save function.
Should the code be downloaded with a single write buffer command?	HSUTIL detects that an unsupported device has been selected as the target device. You must indicate whether to download the firmware image to the device in one or more contiguous blocks, each corresponding to one SCSI Write Buffer command.

Upgrading to a Dual-Redundant Controller Configuration

Follow these steps to upgrade a single-configuration subsystem to a dual-redundant configuration subsystem. Before you complete this procedure, you’ll need the following items:

- Controller with the same software version that’s installed on the subsystem’s current single controller.
- Cache module with the same amount of memory that’s installed in the current cache module.
- External cache battery (ECB) for a dual-redundant configuration.
- ECB cable.

To prepare the single-controller subsystem...

1. From a host console, dismount the storage units in the subsystem.
2. Connect a local terminal to the controller.
3. Shut down the controller with the following command:

   SHUTDOWN THIS_CONTROLLER

   When the controller shuts down, its reset button and first three LEDs stay lighted continuously.
4. Turn off the power to the subsystem.
5. Disable the ECB by pressing its shut-off button until the status light stops blinking—about 2 seconds.
6. Disconnect the ECB cable from the ECB.
7. Remove the single-battery ECB.

To install the new controller, cache module, and ECB...

1. Ensure that the program card is not in the new controller, then insert the new controller into the bottom slot—controller slot B. This slot responds to SCSI target ID number 6. (See Figure 1–4 on page 1–7.) Engage its retaining levers.
2. Insert the new cache module into the bottom right slot—cache slot B—and engage its retaining levers.
3. Insert the new, two-battery ECB into the empty slot at the top of the cabinet.
4. Connect an ECB cable to the new cache module.

5. Connect an open y end of each ECB cable to the new ECB.
6. Connect the controllers to the hub and host as described in “Connecting a Dual-Redundant Configuration to the Host,” page 2–10. Use the correct instructions depending on whether you have one or two hubs.
7. Restore power to the subsystem.
8. Press and hold the reset button while inserting the program card into the new controller. The new controller automatically initializes, and the ECB has automatically enabled itself to provide backup power to the cache modules.

   The reset button flashes about once per second when initialization is complete.
9. Replace the ESD cover on the new controller.
10. See Chapter 2, “Configuring an HSG80 Array Controller,” for instructions on configuring the controller and connecting it to the host.

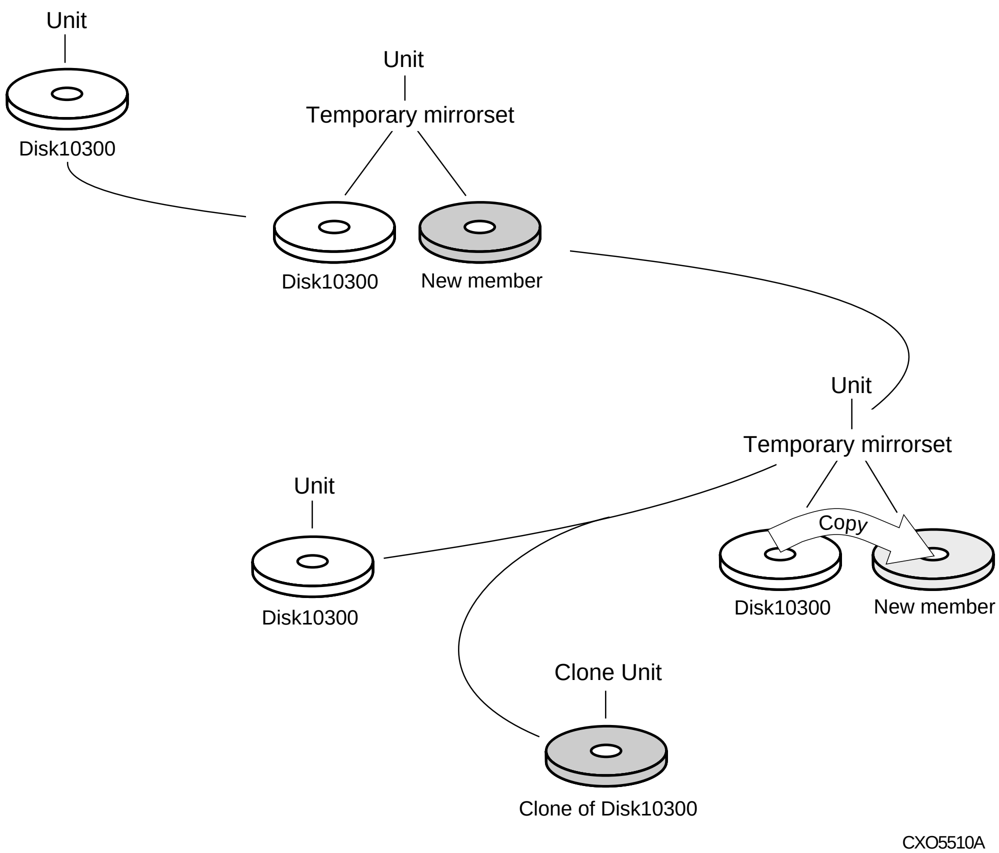
Cloning Data for Backup

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

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

After the CLONE utility copies the data from the members to the clones, it restores the unit to its original configuration and creates a clone unit you can backup. The CLONE utility uses steps shown in Figure 6–4 to duplicate each member of a unit.

Figure 6–4 CLONE Steps for Duplicating Unit Members



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

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

   RUN CLONE

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

6. When prompted, enter the disk drives you want to use for the clone units.
7. Back up the clone unit.

Example 6–1

This example shows the commands you would use to clone storage unit D98. The clone command terminates after it creates storage unit D99, a clone or copy of D98.

RUN CLONE

CLONE LOCAL PROGRAM INVOKED

UNITS AVAILABLE FOR CLONING:101

98

ENTER UNIT TO CLONE ? 98

CLONE WILL CREATE A NEW UNIT WHICH IS A COPY OF UNIT 98.

ENTER THE UNIT NUMBER WHICH YOU WANT ASSIGNED TO THE NEW UNIT ? 99

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)

DISK20300 (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:

DISK20300 (SIZE=832317)

DISK30100 (SIZE=832317)

USE AVAILABLE DEVICE DISK10400(SIZE=832317) FOR MEMBER DISK10000(SIZE=832317) (Y,N) [Y] ? Y

MIRROR DISK10000 C_MB

SET C_MB NOPOLICY

SET C_MB MEMBERS=2

SET C_MB REPLACE=DISK10400

COPY IN PROGRESS FOR EACH NEW MEMBER. PLEASE BE PATIENT...

.

.

COPY FROM DISK10300 TO DISK10200 IS 100% COMPLETE

COPY FROM DISK10000 TO DISK10400 IS 100% COMPLETE

PRESS RETURN WHEN YOU WANT THE NEW UNIT TO BE CREATED

REDUCE DISK10200 DISK10400

UNMIRROR DISK10300

UNMIRROR DISK10000

ADD MIRRORSET C_MA DISK10200

ADD MIRRORSET C_MB DISK10400

ADD STRIPESET C_ST1 C_MA C_MB

INIT C_ST1 NODESTROY CHUNK=128

ADD UNIT D105 C_ST1

D105 HAS BEEN CREATED. IT IS A CLONE OF D104.

CLONE - NORMAL TERMINATION

Backing Up Your Subsystem Configuration

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

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

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

You cannot use the SAVE_CONFIGURATION switch on TRANSPORTABLE disks.

Saving Subsystem Configuration Information to a Single Disk

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

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

INITIALIZE DISK*nnn* **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 DISK*nnn* **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:
        HSG80    (C) DEC ZG64100138 Firmware QBFB-0, Hardware CX02
        Configured for dual-redundancy with ZG64100209
            In dual-redundant configuration
        Device Port SCSI address 7
        Time: NOT SET
Host port:
        SCSI target(s) (1, 3, 11)
        Preferred target(s) (3, 11)
        TRANSFER_RATE_REQUESTED = 20MHZ
        Host Functionality Mode = A
        Command Console LUN is target 1, lun 5
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: 75524
        Configuration backup enabled on 3 devices
```

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

$ SHOW DEVICES FULL

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

APPENDIX A

System Profiles

This appendix contains device and storageset profiles you can use to create your system profiles. It also contains an enclosure template you can use to help keep track of the location of devices and storagesets in your shelves.

Device Profile

Type

___ Platter disk drive	___ Optical disk drive
___ Tape Drive	___ CD-ROM

Device Name____

Unit Number____

Device Switches

Transportability
___ No (default)
___ Yes

Initialize Switches

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

Unit Switches

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

Storageset Profile

Type

___ RAIDset
___ Mirrorset
___ Stripeset
___ Striped Mirrorset

Storageset Name ________________________________

Disk Drives ________________________________

Unit Number ________________________________

Partitions

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

RAIDset Switches

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

Mirrorset Switches

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

Initialize Switches

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

Unit Switches

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

Enclosure Template

Power Supply							**Power Supply**
Power Supply							**Power Supply**
Power Supply							**Power Supply**
Power Supply							**Power Supply**

APPENDIX B

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 in a dual-redundant configuration use the relative terms “this controller” and “other controller.” See “Typographical Conventions,” page xvi, for an explanation of these terms.

Using the CLI

You can access the CLI by connecting a maintenance terminal to the port in the front bezel of the controller (local connection) or by using HSZterm software (remote connection). See “Establishing a Local Connection to the Controller,” page 2–5 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 “Local-Connection Port Precautions,” page –xv, explains precautions you should observe when operating the CLI through a local connection port.

Command Overview

The CLI consists of six basic command types:

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

- Device Commands—Create and configure containers made from physical devices attached to the controller.
- Storageset Commands—Create and configure complex containers made from groups of device containers. There are four basic types of storagesets: stripesets, RAIDsets, striped-mirrorsets, and mirrorsets. Storageset commands group device containers together and allow them to be handled as single units.
- Logical Unit Commands—Create and optimize access to logical units made from any container type.
- Failover Commands—Configure the controllers to operate in transparent failover while also providing support for dual-redundant configurations.
- Diagnostic and Utility Commands—Perform general controller support functions.

Getting Help

Help on using the CLI is at your fingertips. For an overview of the CLI help system, enter help at the prompt. For help on a specific command or to determine what switches are available with a command, enter as much of the command as you know followed by a space and a question mark. For example, to get information on the switches used with the SET THIS_CONTROLLER command, enter:

```
SET THIS_CONTROLLER ?
```

See “HELP,” page B–57 for further information.

Entering CLI Commands

Use the following tips and techniques when entering CLI commands:

- Commands are not case sensitive.
- For most commands, you only need to enter enough of the command to make the command unique. For example, SHO is the same as entering SHOW.
- The controller processes each command in sequence. You can continue entering subsequent commands while the controller is processing prior commands. A controller experiencing heavy data I/O may respond slowly to CLI commands.

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. The following table lists the keys used to recall and edit commands.

Table B–1 Recall and Edit Command Keys

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

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. Parameters are represented in this manual 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.

Syntax

ADD DISK *container-name SCSI-port-location*

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 (_).

Tip It is common to name a disk drive DISK*ptl*, where *ptl* 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-location

Indicates the PTL address of the disk device. See the “Device PTL Addressing Convention within the Controller,” page 3–21 for an explanation of the PTL addressing naming format.

Note See the *HSG80 Array Controller ACS Version 8.0 Release Notes* to determine whether the disk drive you are planning to use is compatible with the controller.

Switches

NOTRANSPORTABLE (Default)

TRANSPORTABLE

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

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

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

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

Note DIGITAL recommends you avoid using transportable disks unless there is no other way to move the data.

TRANSFER_RATE_REQUESTED=ASYNCHRONOUS
TRANSFER_RATE_REQUESTED=20MHZ (Default)
TRANSFER_RATE_REQUESTED=10MHZ
TRANSFER_RATE_REQUESTED=5MHZ
Specifies the maximum data transfer rate at which the controller is to communicate with the disk drive. The user might need to limit the transfer rate to accommodate long cables between the controllers and the device.

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

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

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

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

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

ADD DISK DISK30200 3 2 0 NOTRANSPORTABLE TRANSFER_RATE_REQUESTED=10MHZ

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

INITIALIZE DISK20000

ADD UNIT D199 DISK20000

See also

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

ADD MIRRORSET

Names a mirrorset and adds it to the controller configuration.

Syntax

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

Parameters

mirrorset-name

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

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

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

---

disk-name1 [disk-nameN]

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

Switches

COPY=FAST

COPY=NORMAL (Default)

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

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

Specify COPY=NORMAL when operations performed by the controller should take priority over the copy operation. If you specify

COPY=NORMAL, creating the mirrored data has a minimal impact on performance.

POLICY=BEST_FIT
POLICY=BEST_PERFORMANCE (Default)
NOPOLICY
Sets the selection criteria the controller uses to choose a replacement disk from the spareset when a mirrorset member fails.

Specify POLICY=BEST_FIT to choose a replacement disk drive from the spareset that most closely matches the capacity of the mirrorset. If there is more than one disk drive in the spareset that meet the criteria, the controller selects the disk drive with the best performance.

Specify POLICY=BEST_PERFORMANCE to choose a replacement disk drive from the spareset with the best performance. The controller attempts to select a disk on a different port than existing mirrorset members. If there is more than one disk drive in the spareset matching the best performance criteria, the controller selects the disk drive that most closely matches the capacity of the mirrorset.

Specify NOPOLICY to prevent the controller from automatically replacing a failed disk device. The mirrorset operates in a reduced state until a POLICY=BEST_FIT or POLICY=BEST_PERFORMANCE is selected, or a member is manually placed in the mirrorset (see “SET,” page B–99).

READ_SOURCE=disk-name
READ_SOURCE=LEAST_BUSY (Default)
READ_SOURCE=ROUND_ROBIN
Selects the mirrorset member used by the controller to satisfy a read request.

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

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

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

Examples

To add DISK10000, DISK20100, and DISK30200 as a mirrorset with the name MIRR1, type:

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

The following example shows how to create a host-addressable unit after the mirrorset MIRR1 has been created:

```
INITIALIZE MIRR1
ADD UNIT D104 MIRR1
```

See also

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

ADD RAIDSET

Names a RAIDset and adds the RAIDset to the controller's configuration. DIGITAL RAIDsets are often referred to as RAID level 3/5 storagesets because they use the best characteristics of RAID level 3 and RAID level 5. The number of members in the storageset is determined by the number of containers specified by the *container-name* parameter in the command. The data capacity of the RAIDset is determined by the storage size of the smallest member.

Syntax
ADD RAIDSET *RAIDset-name container-name1 container-name2 [container-nameN]*

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

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

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

container-name1 container-name2 [container-nameN]
Identifies the disks making up the RAIDset. RAIDsets must include at least 3 disk drives and no more than 14.

Switches
POLICY=BEST_FIT
POLICY=BEST_PERFORMANCE (Default)
NOPOLICY
Set the selection criteria the controller uses to choose a replacement member from the spareset when a RAIDset member fails.

Specify POLICY=BEST_FIT to choose a replacement disk drive from the spareset most closely matching the sizes of the remaining members of the RAIDset. If more than one disk drive in the spareset is the correct size, the controller selects the disk drive giving the best performance.

Specify POLICY=BEST_PERFORMANCE to choose a replacement disk drive from the spareset resulting in the best performance of the RAIDset. The controller attempts to select a disk on a different port than existing RAIDset members. If there is more than one disk drive in the spareset matching the best performance criteria, the controller selects the disk drive that most closely matches the size of the remaining members of the RAIDset selected.

Specify NOPOLICY to prevent the controller from automatically replacing a failed disk device. This RAIDset operates in a reduced state until you select a POLICY=BEST_FIT or POLICY=BEST_PERFORMANCE, or manually place a member in the RAIDset. See “SET,” page B–107, for more information regarding this procedure.

RECONSTRUCT=FAST
RECONSTRUCT=NORMAL (Default)
Sets the speed at which the controller reconstructs data to a new RAIDset disk that replaces the failed disk.

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

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

REDUCED
NOREDUCED (Default)
*Permits the additi*on of a RAIDset missing a member. Specify the REDUCED switch when you add a reduced RAIDset (a RAIDset that is missing a member).

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

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

Examples

To create a RAIDset named RAID9 that contains disks DISK10000, DISK20100, and DISK30200, use the following commands:

```
ADD DISK DISK10000 1 0 0
ADD DISK DISK20100 2 1 0
ADD DISK DISK30200 3 2 0
ADD RAIDSET RAID9 DISK10000 DISK20100 DISK30200
```

This example shows how to create a RAIDset named RAID8 that contains disks DISK10000, DISK20100, and DISK30200 and uses the BEST_FIT switch to indicate the replacement policy. Enter the ADD RAIDSET command on one line.

```
ADD DISK DISK10000 1 0 0
ADD DISK DISK20100 2 1 0
ADD DISK DISK30200 3 2 0
ADD RAIDSET RAID8 DISK10000 DISK20100 DISK30200
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.

Example

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

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

See also

DELETE SPARESET
SET FAILEDSET
SHOW SPARESET
SHOW STORAGESETS

ADD STRIPESET

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

Syntax

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

Parameters

stripeset-name

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

container-name1 container-name2 [container-nameN]

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

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

Examples

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

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

To create a stripeset named STRIPE1 and then create a logical unit from it, type:

```
INITIALIZE STRIPE1
ADD UNIT D103 STRIPE1
```

This example shows how to create a two-member striped mirrorset (a stripeset whose members are mirrorsets), and how to create a logical unit from it. Because you can initialize the stripeset, you do not need to individually initialize the mirrorsets.

```
ADD DISK DISK10000 1 0 0
ADD DISK DISK20100 2 1 0
ADD DISK DISK30200 3 2 0
ADD DISK DISK40300 4 3 0
ADD MIRRORSET MR1 DISK10000 DISK20100
ADD MIRRORSET MR2 DISK30200 DISK40300
ADD STRIPESET STRIPE1 MR1 MR2
INITIALIZE STRIPE1
ADD UNIT D104 STRIPE1
```

See also

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

ADD UNIT

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

If you add a newly-created storageset or disk to your subsystem, you must initialize it before it can be added as a logical unit. If you are adding a storageset or disk that has data on it that you want to maintain, do not initialize it; it will be added as logical unit.

Syntax
ADD UNIT *unit-number container-name*

Parameters
unit-number
Assigns a number to the unit being created from a device, container, or partition in the subsystem. The host uses this number to indicate the source or target for every I/O request it sends to the controller. The *unit-number* is a host-addressable LUN. The *unit-number* is assigned to one of the host ports.

Unit numbers are 0-99 and are prefixed by one of the following:

D—assigns units to Port 1

D1—assigns units to Port 2

Adding unit D00 creates a logical unit and presents it as D00 to the host on port 1. Adding unit D100 creates a logical unit and presents it as D00 to the host on port 2. Units must be 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) that is used to create the unit.

A maximum of 48 devices can make up one unit.

Switches
The following table 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.

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

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

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

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 specified size. Accepted write block sizes are 1 through 1024.

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

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.

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

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 the host from writing data to the unit. However, the controller may still write to a write-protected RAIDset to complete a reconstruct operation and metadata, reconstruct data, and copy data may still be written to RAIDsets and mirrorsets.

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

WRITEBACK_CACHE (Default)
NOWRITEBACK_CACHE
Enable or disable the write-back data caching function of the controller. The controller’s write-back caching feature improves write

performance. WRITEBACK_CACHE is the default on transportable disks.

Specify WRITEBACK_CACHE for all new units you want to take advantage of the controller’s write-back caching feature. Specify NOWRITEBACK_CACHE for units you want to write data received from the host without being cached.

Caution Though there is built-in redundancy to protect data contained in cache, allowing data to be written to write-back cache may result in the loss of data if a catastrophic subsystem failure occurs.

Examples

This example shows how to create unit D102 from a single-disk drive named DISK10000 and sets the host’s access to the unit through “this controller.”

```
ADD DISK DISK10000 1 0 0
INITIALIZE DISK10000
ADD UNIT D102 DISK10000 PREFERRED_PATH=THIS_CONTROLLER
```

This example shows how to create unit D107 from a RAIDset named RAID9 and instructs the unit to take advantage of the controller’s write-back caching feature.

```
ADD DISK DISK10100 1 1 0
ADD DISK DISK20100 2 1 0
ADD DISK DISK30100 3 1 0
ADD DISK DISK40100 4 1 0
ADD RAIDSET RAID9 DISK10100 DISK20100 DISK30100 DISK40100
INITIALIZE RAID9
ADD UNIT D107 RAID9 WRITEBACK_CACHE
```

See also

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

CLEAR_ERRORS CLI

Stops the display of current or previous error messages at the CLI prompt. This command does not clear the error conditions, it only stops the display of errors at the CLI prompt.

After the cause of the error condition has been corrected, issue the CLEAR_ERRORS CLI command to clear the error message.

Enter this command to clear errors existing from previous error conditions.

Syntax

CLEAR_ERRORS CLI

Example

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

```
ALL NVPM COMPONENTS INITIALIZED TO THEIR DEFAULT SETTINGS
CLEAR_ERRORS CLI
```

See also

CLEAR_ERRORS INVALID_CACHE
CLEAR_ERRORS LOST_DATA
CLEAR_ERRORS UNKNOWN
CLEAR_ERRORS UNWRITEABLE_DATA

CLEAR_ERRORS *controller* INVALID_CACHE

Clears an invalid cache error and allows the controller and cache to resume operation. If the error is due to an incorrectly-mirrored configuration, the controller indicates mirrored mode status after the error is cleared.

Use this command for the following situations:

- When the controller or cache modules have been replaced, resulting in mismatched data between the controllers.
- When the controller or cache module is replaced while data is still in cache and not properly flushed with the SHUTDOWN or SET NOFAILOVER COPY= commands.

Syntax
CLEAR_ERRORS *controller* INVALID_CACHE

Spell out INVALID_CACHE when using this command.

Parameters
controller
Identifies which controller is to receive the CLEAR_ERRORS command. You must specify THIS_CONTROLLER or OTHER_CONTROLLER.

data-retention-policy
DESTROY_UNFLUSHED_DATA
NODESTROY_UNFLUSHED_DATA (Default)
Instructs the controller on how to handle write-back cached data.

Specify NODESTROY_UNFLUSHED_DATA (default) to retain the cached data and discard controller information. Specify 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 to retain the controller information and discard the cached data. Specify DESTROY_UNFLUSHED_DATA in the following situations:

- If the cache module has been replaced
- Any other reason not listed above

Caution Specifying the DESTROY_UNFLUSHED_DATA switch destroys data remaining in cache, which can result in data loss.

Examples

This example shows how to clear an invalid cache error on "this controller" after you have replaced a controller module. Enter the command on one line.

```
CLEAR_ERRORS THIS_CONTROLLER INVALID_CACHE
NODESTROY_UNFLUSHED_DATA
```

This example shows how to clear an invalid cache error on the "other controller" after a cache module has been replaced. Enter this command on the same line. Enter the command on one line.

```
CLEAR_ERRORS OTHER_CONTROLLER INVALID_CACHE
DESTROY_UNFLUSHED_DATA
```

See also

CLEAR_ERRORS CLI
CLEAR_ERRORS LOST_DATA
CLEAR_ERRORS UNKNOWN
CLEAR_ERRORS UNWRITEABLE_DATA

CLEAR_ERRORS *unit-number* LOST_DATA

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

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

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

Syntax
CLEAR_ERRORS *unit-number* LOST_DATA

Spell out LOST_DATA when using this command.

Caution This command may cause data loss.

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

Example
The following command will clear the lost data error on disk unit number D103:

```
CLEAR_ERRORS D103 LOST_DATA
```

See also
CLEAR_ERRORS CLI
CLEAR_ERRORS INVALID_CACHE
CLEAR_ERRORS UNKNOWN
CLEAR_ERRORS UNWRITEABLE_DATA

CLEAR_ERRORS UNKNOWN

Clears unknown errors from a device the controller identifies as unknown or not correctly configured.

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

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

Syntax

CLEAR_ERRORS *device-name* UNKNOWN

Spell out UNKNOWN when using this command.

Parameters

device-name

Identifies the device with the unknown error.

Example

To force the controller to recognize a previously unknown device named DISK30000, enter this command:

```
CLEAR_ERRORS DISK30000 UNKNOWN
```

See also

CLEAR_ERRORS CLI
CLEAR_ERRORS INVALID_CACHE
CLEAR_ERRORS UNKNOWN
CLEAR_ERRORS UNWRITEABLE_DATA

CLEAR_ERRORS *unit-number* **UNWRITEABLE_DATA**

Clears an unwriteable data error on a unit. It affects all partitions on the same container.

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

Caution This command causes data loss.

Syntax

CLEAR_ERRORS *unit-number* UNWRITEABLE_DATA

Spell out UNWRITEABLE_DATA when using this command.

Parameters

unit-number

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

Example

Use the following command to clear the unwriteable data error on disk unit D103:

```
CLEAR_ERRORS D103 UNWRITEABLE_DATA
```

See also

CLEAR_ERRORS CLI
CLEAR_ERRORS INVALID_CACHE
CLEAR_ERRORS LOST_DATA
CLEAR_ERRORS UNKNOWN
RETRY_ERRORS UNWRITEABLE_DATA

CONFIGURATION 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.

Use the INITIALIZE SAVE_CONFIGURATION command to set up the location of the configuration file on disk.

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.

CONFIGURATION RESET

Erases the entire configuration on “this controller,” restores the controller’s default configuration, and shuts down the controller. The default configuration is stored on devices that were initialized with the SAVE_CONFIGURATION switch on the INITIALIZE command.

Specify the CONFIGURATION RESET command on “this controller” in nofailover mode only. Enter this command to ensure that 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.

CREATE_PARTITION

Divides a non-transportable disk drive or single-disk 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 into a maximum of eight partitions, each of which can be separately presented to the host. Initialize disks and storagesets before creating partitions.

Syntax
CREATE_PARTITION *container-name SIZE=percent*

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

SIZE=percent
SIZE=LARGEST
Specifies the size of the partition to be created as a percentage of the total container's storageset size.

To create a partition, specify a percentage of the container's total capacity. The entire container is then divided into segments equal to the percentage specified. For example, if SIZE=20, the container is divided into five (1.0/0.2=5) equal segments. The resulting partition is slightly smaller than the size specified because metadata also occupies some of the partition's allocated space.

Specify LARGEST in the following situations:

- To have the controller create the largest partition possible from unused space on the disk or storageset.
- To create the last partition on a container. Because the remaining space is not equal to an exact percentage value, specifying LARGEST allows you to optimize use of the remaining space.

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_PARTITOIN RAID9 SIZE=LARGEST
ADD UNIT D101 RAID9 PARTITION=1
ADD UNIT D102 RAID9 PARTITION=2
ADD UNIT D103 RAID9 PARTITION=3
ADD UNIT D104 RAID9 PARTITION=4
```

See also

ADD UNIT
DELETE *unit-number*
DESTROY PARTITION
SHOW

DELETE *container-name*

Deletes a container belonging to the controller's configuration. You cannot delete a container in use by a higher-level container. For example, you cannot delete a disk belonging to a RAIDset, or a RAIDset belonging to a unit; you must first delete the higher-level container or containers.

Note This command does not delete 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 initialized or used for anything, the container can be reassembled using its original disks.

Syntax
DELETE *container-name*

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

Examples
To delete a disk drive named DISK10000, type:

```
DELETE DISK10000
```

To delete a stripeset named STRIPE1, enter:

```
DELETE STRIPE1
```

To delete a RAIDset named RAID9, use:

```
DELETE RAID9
```

See also
DELETE FAILEDSET
DELETE SPARESET
UNMIRROR

DELETE FAILEDSET

Removes a disk drive from the failedset. The failedset contains disk drives removed by the controller from RAIDsets and mirrorsets because they failed or were manually removed using the SET command. Enter the DELETE FAILEDSET command before physically removing failed disks from the storage shelf for testing, repair, or replacement.

You should consider all disk drives in the failedset defective. Repair or replace disks found in the failedset.

Syntax

DELETE FAILEDSET *disk-name*

Parameter

disk-name

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

Example

To delete DISK20200 from the failedset, use the following command:

```
DELETE FAILEDSET DISK20200
```

See also

SET FAILEDSET
SHOW FAILEDSET

DELETE SPARESET

Removes a disk drive from the spareset.

Syntax
DELETE SPARESET *disk-name*

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

Example
This command will remove DISK20300 from the spareset:

```
DELETE SPARESET DISK20300
```

See also
ADD SPARESET
SHOW SPARESET

DELETE *unit-number*

Deletes a logical unit from the controller configuration. The host cannot address deleted units. If the unit's write-back caching feature is enabled, the controller flushes the cached data to the unit's devices before deleting the unit.

Before using the DELETE *unit-number* command, clear any errors with the CLEAR_ERRORS UNWRITEABLE_DATA or CLEAR_ERRORS LOST_DATA commands.

Syntax
DELETE *unit-number*

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

Example
To delete disk unit number D103, enter:

```
DELETE D103
```

See also
ADD UNIT
CLEAR_ERRORS LOST_DATA
CLEAR_ERRORS UNWRITEABLE_DATA
DESTROY_PARTITION

DESTROY_PARTITION

Marks the area reserved for a partition as available. The freed area is then consolidated with any adjacent free areas.

Caution Data contained on a partition is lost when you enter the DESTROY_PARTITION command.

You cannot destroy a partition that has been assigned a unit number. First enter the DELETE *unit-number* command to delete the unit using the partition.

Syntax

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

Parameters

container-name

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

partition-number

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

Example

The following example shows how to delete the unit for partition 2 on RAIDset RAID9 and destroy the partition:

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

See also

CREATE_PARTITION
DELETE *unit-number*
SHOW

DIRECTORY

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

Syntax
DIRECTORY

Example
The example below shows how to display a directory listing:

DIRECTORY

```
HSUTILV70Z   D
CHVSN V70Z   D
CLCP  V70Z   D
CLONE V70Z   D
CONFIGV70Z   D
DILX  V70Z   D
DIRECTV70Z   D
DSTAT V70Z   D
FMU   V70Z   D
VTDPY V70Z   D
```

Note CHVSN and DSTAT are not user utilities. They may be used by DIGITAL authorized service personnel only.

See also
RUN

HELP

Displays a brief explanation of how to use the question mark (?) to obtain help on any command or CLI function. You must precede the question mark with a space.

Syntax
HELP

Example
To display information regarding the HELP command, type:

HELP

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

SET THIS_CONTROLLER ?

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

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

SET ?

```
Your options are:

EMU
FAILEDSET
FAILOVER
NOFAILOVER
OTHER_CONTROLLER
THIS_CONTROLLER
Unit number or mirrorset or raidset or device name
```

INITIALIZE

Initializes or destroys metadata on a container. During initialization, a small amount of disk space is reserved for controller metadata and is made inaccessible to the host. Disks made transportable do not contain controller metadata.

Syntax
INITIALIZE *container-name*

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

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

Use the INITIALIZE command when:

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

Do not use the INITIALIZE command when:

- Creating a unit from the same disks previously initialized, such as when a RAIDset is moved
- Creating a storageset from existing members
- Adding a RAIDset with the REDUCED switch

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

Switches

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. The CHUNKSIZE switch does not apply to mirrorsets.

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.

Tip Accept the default chunk size setting for most applications. Do not change the default setting unless you are fully aware of the impact to the storageset’s performance.

See “Chunk Size,” page 3–34 for information regarding recommended chunk size settings for your application.

DESTROY (Default)

NODESTROY

Note The DESTROY and NODESTROY switches are only valid with mirrorsets and striped mirrorsets.

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

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

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

SAVE_CONFIGURATION
NOSAVE_CONFIGURATION (Default)
Instructs the controller whether to save the controller's configuration to the container being initialized. See also the INITIAL_CONFIGURATION parameter of "SET," page B–85.

Save configuration is intended only for use with single controller applications. 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 6–16 for more information regarding SAVE_CONFIGURATION.

Examples
To initialize container DISK10000 and save a copy of the controller configuration on it, enter the following commands:

```
ADD DISK DISK10000 1 0 0
INITIALIZE DISK10000 SAVE_CONFIGURATION
```

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

SHOW DEVICES FULL

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

This example shows how to initialize stripeset STRIPE1 with the default chunk size. The chunk size is not specified, so the controller initializes the unit with the default chunk size.

ADD DISK DISK10100 1 1 0
ADD DISK DISK20100 2 1 0
ADD DISK DISK30100 3 1 0
ADD STRIPESET STRIPE1 DISK10100 DISK20100 DISK30100
INITIALIZE STRIPE1

LOCATE

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

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

Syntax
LOCATE *parameter*

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

ALL
Causes the 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 3–21 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 green 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 green device fault LED on device DISK10000 to flash:

```
LOCATE DISK10000
LOCATE CANCEL
```

This example shows how to cause the device fault LEDs on all of the devices assigned to disk unit number D102 to flash:

```
LOCATE D102
```

This example shows how to cause the device fault LEDs on all configured disk devices to flash:

```
LOCATE DISKS
```

This example shows how to turn off the flashing device fault LEDs on all devices:

```
LOCATE CANCEL
```

MIRROR

Creates a one-member mirrorset from a single disk. This command is used only on disks configured as units or members of a stripeset, then enter the ADD MIRRORSET command to create a mirrorset from disk drives not already members of higher level containers.

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

Syntax
MIRROR *disk-name mirrorset-name*

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

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

Switches
COPY=FAST
COPY=NORMAL (Default)
Sets the speed at which the controller copies data to a new member from normal mirrorset members when data is being mirrored to the storageset’s disk drives.

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

Specify COPY=NORMAL when operations performed by the controller should take priority over the copy operation. If you specify

COPY=NORMAL creating the mirrored data has a minimal impact on performance.

POLICY=BEST_FIT
POLICY=BEST_PERFORMANCE
NOPOLICY (Default)
Sets the selection criteria the controller uses to choose a replacement member from the spareset when a mirrorset member fails.

Specify POLICY=BEST_FIT to choose a replacement disk drive from the spareset most closely matching the individual capacities of the remaining members. If there is more than one disk drive in the spareset most closely matching the capacity of the remaining members, the controller selects the disk drive that has the best performance.

Specify POLICY=BEST_PERFORMANCE to choose a replacement disk drive from the spareset resulting in the best performance. The controller attempts to select a disk on a different port than existing members. If there is more than one disk drive in the spareset matching the best performance criteria, the controller selects the disk drive most closely matching the individual capacities of the remaining members.

Specify NOPOLICY to prevent the controller from automatically replacing a failed disk device. This causes the mirrorset to operate in a reduced state until a POLICY=BEST_FIT or POLICY=BEST_PERFORMANCE is selected, or a member is manually replaced in the mirrorset. See “SET,” page B–85.

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. The system sets all disk units in the cabinet to write through caching, and begins to run down the units after the time interval you specify. When all units in the cabinet are successfully run down, the cabinet power turns off.

Syntax
POWEROFF

Switch
SECONDS=nn
Specifies the time interval, in seconds, after which the system is to begin running down all units.

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.

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 disk(s)
- Decreases the nominal number of members of the mirrorset(s) by the number of disk(s) removed from the mirroset(s).

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

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

Note An error is displayed if you attempt to reduce a mirrorset so that there would not be any normal member remaining.

Syntax

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

Parameters

disk-name1 disk-name2 disk-name3...

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

Example

This example shows how to remove DISK20100, DISK20200, and DISK40200 from their respective mirrorsets:

SHOW STRIPE1

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

SHOW MIRRORSETS

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

REDUCE DISK20100 DISK20500 DISK40200

SHOW MIRRORSETS

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

See also

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

RENAME

Renames a 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 “Command Syntax,” page B–5, 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
```

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 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 "Write-Back Caching," page 1–14 for considerations when implementing write-back cache. Do not perform any hardware changes until the controller flushes the cache.

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

Specify NOIMMEDIATE_SHUTDOWN to instruct the controller not to restart without checking for online devices or before all data has been flushed from write-back cache to the devices.

Examples
This example shows how to restart “this controller”:

```
RESTART THIS_CONTROLLER
```

This example shows how to restart the “other controller”:

```
RESTART OTHER_CONTROLLER
```

See also
SELFTEST *controller*
SHUTDOWN *controller*

RETRY_ERRORS UNWRITEABLE_DATA

Causes the controller to attempt to write previously unwriteable data from the write-back cache to the devices. If a container fails, preventing the data in write-back cache to be written to the container, an unwriteable data error is reported. If possible, correct the condition that caused the unwriteable data and try the write operation again. No data is lost if the retry fails.

Syntax

RETRY_ERRORS *unit-number* UNWRITEABLE_DATA

Parameter

unit-number

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

Example

This example shows how to retry writing the cached data previously marked unwriteable to disk unit D103:

```
RETRY_ERRORS D103 UNWRITEABLE_DATA
```

See also

CLEAR_ERRORS 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 DIGITAL authorized service personnel only.
- CLCP—A utility used to load updated software code or patches. See “Using CLCP to List, Install, and Delete Software Patches,” page 6–4 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 6–12 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 Several Disk Drives at a Time,” page 3–44 for more information regarding this utility.
- DILX—A utility used to test and verify the controller’s operation with attached storage devices under a high or low I/O load. Run DILX (disk inline exerciser) only when there is no activity on the controller. The total I/O load is handled by the controller, bypassing the host.

  The DILX utility has two modes, an autoconfigure mode, and a standard mode.

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

The autoconfigure mode is the most thorough mode and allows you to:

- Automatically test all of the disk units configured

- Automatically perform thorough tests on all units with writes enabled

  The standard mode is more flexible and allows you to:

  - Test disks you select
  - Perform tests in read-only mode or write-only mode
  - Provide run time and performance summary options
  - Can be run in read-only mode

- DIRECT—A command used to display a list of all executable diagnostic or utility programs.
- DSTAT—This is not a user utility. This utility may be used by DIGITAL 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.
- 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 Chapter 4, “Troubleshooting,” 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 "Write-Back Caching," page 1–14 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 “Write-Back Caching,” page 1–14 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

The following table lists the switches available with this command. Descriptions of the switches follow the table.

Table B–2 SET *controller* Switches

Switch	Values
CACHE_FLUSH_TIMER	1–65535 sec, 10 (default)
CACHE_UPS NOCACHE_UPS	None
INITIAL_CONFIGURATION	None
NODE_ID	assigned during manufacturing
PORT_1_ ALPA PORT_2_ALPA	0-EF (hexadecimal value)
PORT_1_TOPOLOGY PORT_2_TOPOLOGY	LOOP_HARD LOOP_SOFT OFFLINE
PROMPT	1–16 characters
TERMINAL_PARITY NOTERMINAL_PARITY	odd, even
TERMINAL_SPEED	4800, 9600, 19200
TIME	dd–mmm–yyy:hh:mm:ss

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

CACHE_UPS
NOCACHE_UPS (Default)
Specifies whether the controller should perform regular battery condition checks. When changed, you must restart both controllers in order for the new setting to take effect.

Specify CACHE_UPS if your storage subsystem power is supported by an uninterruptable power supply (UPS). The controller does not check the condition of the cache batteries and ignores the battery’s state. This causes RAIDsets and mirrorsets to always be available, regardless of the condition of the cache batteries.

Caution Setting CACHE_UPS without having a UPS or similar backup system in place may result in data loss if power is interrupted.

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

Setting the CACHE_UPS switch for either controller sets the CACHE_UPS switch for both controllers.

INITIAL_CONFIGURATION
This switch is being replaced by the CONFIGURATION RESET command. See "CONFIGURATION RESET," page B–41 for details.

PORT_1_ALPA=
PORT_2_ALPA=
Specifies the hexadecimal arbitrated loop physical address (ALPA) for the host ports. Use this switch only when LOOP_HARD is specified for PORT_1_TOPOLOGY or PORT_2_TOPOLOGY. The range of addresses allowed is 0-EF (hexadecimal). The default value is 69.

PORT_1_TOPOLOGY=LOOP_HARD
PORT_1_TOPOLOGY=LOOP_SOFT
PORT_1_TOPOLOGY=OFFLINE
PORT_2_TOPOLOGY=LOOP_HARD
PORT_2_TOPOLOGY=LOOP_SOFT
PORT_2_TOPOLOGY=OFFLINE
Indicates whether the user or controller selects the ALPA for a host port, or whether the port is to be set offline (not used.) LOOP_HARD allows you to pick the ALPA. LOOP_SOFT requests the controller to pick the ALPA.

OFFLINE sets the host port offline. Specify OFFLINE for a port when it will not be used.

NODE_ID=nnnn-nnnn-nnnn-nnnn checksum
Sets the subsystem worldwide name (node ID). If a situation occurs that requires you to reset the subsystem worldwide ID (node ID), use

the name and check sum that appear on the sticker on the frame into which your controller is inserted.

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

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.

TERMINAL_PARITY=ODD
TERMINAL_PARITY=EVEN
NOTERMINAL_PARITY (Default)
Specifies the parity with which data is transmitted and received. When changed, the new setting for this switch takes effect immediately.

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

TIME=dd–mmm–yyyy:hh:mm:ss
Sets the date and time. The time is set on both controllers in a dual-redundant configuration.When changed, the new value entered for this switch takes effect immediately.

Examples
This example shows how to change the other controller's CLI prompt:

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

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
TRANSFER_RATE_REQUESTED=ASYNCHRONOUS
TRANSFER_RATE_REQUESTED=20MHZ (Default)
TRANSFER_RATE_REQUESTED=10MHZ
TRANSFER_RATE_REQUESTED=5MHZ
Specifies the maximum data transfer rate for the controller to use in communicating with the device. You may need to limit the transfer rate to accommodate long cables between the controllers and the device.

TRANSPORTABLE
NOTRANSPORTABLE (Default)
Indicates whether a disk can be accessed exclusively by StorageWorks controllers.

Set the TRANSPORTABLE switch for disks only.

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

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

Note DIGITAL recommends you avoid specifying TRANSPORTABLE unless transportability of the device or media is imperative and there is no other way to accomplish moving the data.

Examples

This example shows how to set the data transfer rate of DISK20000 to 5MHz:

```
SET DISK20000 TRANSFER_RATE_REQUESTED=5MHZ
```

This example shows how to set DISK10300 to transportable:

```
SET DISK10300 TRANSPORTABLE
```

See also

ADD DISK
SHOW DISKS

SET EMU

Sets operating parameters for the environmental monitoring unit (EMU).

Syntax

SET EMU

Switches

The SENSOR and FANSPEED switches control both the master and slave EMU settings. The EMU within the primary cabinet (master) instructs the EMUs within the other cabinets to operate at the same SENSOR and FANSPEED settings to which the master EMU is set.

SENSOR_1_SETPOINT=nn
SENSOR_2_SETPOINT=nn
SENSOR_3_SETPOINT=nn
SENSOR_x_SETPOINT=35 (Default)
Sets the acceptable temperatures (in Celsius) at which the subsystem operates. Sensor 1 and Sensor 2 set the maximum operating temperature for the primary subsystem cabinet. Sensor 3 sets the maximum operating temperature for the EMU unit. The allowable range for the setpoint is 0°C (32°F) to 49°C (120°F). The EMU determines the default setpoint for all three sensors.

The following table lists the valid EMU set-point temperatures in both Fahrenheit and Celsius.

Table B–3 EMU Set Point Temperatures

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

If any of the setpoints assigned to a slave EMU do not match the corresponding setpoints assigned to the master EMU, the slave EMU settings change to match the corresponding master EMU settings.

Refer to the enclosure documentation for detailed information about setting the EMU temperature set points.

FANSPEED=HIGH
FANSPEED=AUTOMATIC (Default)
Sets the speed at which the fan operates.

Select FANSPEED=HIGH to force the fans in all connected cabinets to operate at high speed continuously.

Select FANSPEED=AUTOMATIC to allow the EMU to control the fan speed for the fans in all connected cabinets.

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

Examples

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

```
SET EMU SENSOR_2_SETPOINT=34
```

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

```
SET EMU FANSPEED=HIGH
```

See also

SHOW

SET FAILEDSET

Changes the automatic replacement policy for the failedset.

Syntax
SET FAILEDSET *replacement policy*

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

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

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

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

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

These steps use DISK10000 as an example.

1. Delete all containers to which the disk belongs.
2. Make the disk transportable.

   SET DISK10000 TRANSPORTABLE.

3. Initialize the disk.

   INIT DISK10000

4. Delete the disk.

   DELETE DISK10000

5. Move DISK10000 to the failedset’s vacant slot.

Example

This example shows how to enable the automatic spare feature:

SET FAILEDSET AUTOSPARE

This example shows how to disable the automatic spare feature:

SET FAILEDSET NOAUTOSPARE

See also

SHOW FAILEDSET

SET FAILOVER

Configures both controllers to operate in a dual-redundant, transparent failover, configuration. This allows both controllers to access the storage devices, providing controller fault-tolerant data processing. If one of the two controllers fails, the devices and any cache attached to the failed controller become available to and accessible through the other controller.

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 NOFAILOVER

SET *mirrorset-name*

Changes the characteristics of a mirrorset, including the addition and removal of members.

Syntax

SET *mirrorset-name*

Parameter

mirrorset-name

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

Switches

COPY=FAST

COPY=NORMAL (Default)

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

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

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

MEMBERSHIP=*number-of-members*

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

Note No other switches can be set when you specify the MEMBERSHIP switch.

If you increase the number of members and specify a replacement policy with the POLICY= switch, the controller automatically adds disk drives from the spareset to the mirrorset until the new number of members is reached, or there are no more suitable disk drives in the spareset.

If you increase the number of members and the NOPOLICY switch is specified, the REPLACE=*disk-name* switch must be specified to bring the mirrorset up to the new nominal number of members.

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

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

Note No other switches can be set when the REMOVE= switch is specified.

If the mirrorset won't have a normal or normalizing member remaining after you remove the disk drive, the controller reports an error and no action is taken. A normal or normalizing member is a mirrorset member whose contents are the same as all other normal members.

For each reduced mirrorset, there must be at least one remaining normal member after the reduction.

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

Note Normalizing members exist only when you first create a mirrorset or when you clear lost data on a mirrored unit. The controller recognizes the member as normal, and all other original mirrorset members as “normalizing.” New data that is written to the mirrorset is written to all members. The controller copies the data existing before the mirrorset was created on the normal member to the normalizing members. The controller recognizes the normalizing members as normal when the mirrorset member’s blocks are all the same.

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

- The replacement policy is set to NOPOLICY
- The mirrorset is missing at least one member

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

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

Note Do not specify any other switches when the REPLACE= switch is specified.

POLICY=BEST_FIT
POLICY=BEST_PERFORMANCE (Default)
NOPOLICY
Sets the selection criteria the controller uses to choose a replacement disk from the spareset when a mirrorset member fails.

Specify POLICY=BEST_FIT to choose a replacement disk drive from the spareset that most closely matches the capacity of the mirrorset. If there is more than one disk drive in the spareset that meet the criteria, the controller selects the disk drive with the best performance.

Specify POLICY=BEST_PERFORMANCE to choose a replacement disk drive from the spareset with the best performance. The controller attempts to select a disk on a different port than existing mirrorset members. If there is more than one disk drive in the spareset matching the best performance criteria, the controller selects the disk drive that most closely matches the capacity of the mirrorset.

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

READ_SOURCE=disk-name
READ_SOURCE=LEAST_BUSY (Default)
READ_SOURCE=ROUND_ROBIN
Selects the mirrorset member used by the controller to satisfy a read request.

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

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

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

Examples
This example shows how to change the replacement policy of mirrorset MIRR1 to BEST_FIT:

```
SET MIRR1 POLICY=BEST_FIT
```

This example shows how to remove member DISK30000 from mirrorset MIRR1 created above. If the mirrorset has a replacement policy and an acceptable disk drive is in the spareset, the controller automatically adds the spare disk drive to the mirrorset.

```
SET MIRR1 REMOVE=DISK30000
```

This example shows how to add disk DISK30200 to the mirrorset MIRR1:

```
SET MIRR1 REPLACE=DISK30200
```

A copy operation begins immediately on DISK30200.

See also

ADD MIRRORSET
MIRROR
REDUCE
SHOW MIRRORSET
UNMIRROR

SET NOFAILOVER

Reconfigures both controllers to operate in a non-dual-redundant (non-failover) configuration. Immediately after entering this command, remove one controller from the shelf because the sharing of devices is not supported by nonredundant controllers.

Note SET NOFAILOVER and SET NOMULTIBUS_FAILOVER have the same effect. Either command exits from transparent or multiple bus failover mode.

It is recommended that both controllers be present when this command is carried out. Otherwise, the controllers become misconfigured with each other, requiring additional steps later to allow the “other controller” to be configured for failover.

This command affects both controllers, regardless of the controller on which the command is carried out. All units accessed through the “other controller” failover to “this controller” and the “other controller” is shut down. No configuration information is lost when the SET NOFAILOVER command is carried out.

Syntax
SET NOFAILOVER

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

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

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

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

Caution Unflushed data cannot be recovered after it is destroyed.

Example

This example shows how to terminate failover mode between two controllers in a dual-redundant configuration:

```
SET NOFAILOVER
```

See also

SET FAILOVER

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 most closely matching the sizes of the remaining members of the RAIDset. If more than one disk drive in the spareset is the correct size, the controller selects the disk drive 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 the disk drive most closely matching the size of the remaining members of the RAIDset selected.

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

RECONSTRUCT=FAST
RECONSTRUCT=NORMAL (Default)

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

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

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

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

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

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

Note Do not specify other switches when you use the REMOVE= switch.

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

- The replacement policy is set to NOPOLICY.
- The disk member is not in any configuration, including a spareset.

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

Note Do not specify other switches when you use the REPLACE= switch.

Examples

This example shows how to change the replacement policy for RAIDset RAID9 to BEST_FIT:

```
SET RAID9 POLICY=BEST_FIT
```

This example shows how to remove member DISK10000 from the RAID9 RAIDset:

```
SET RAID9 REMOVE=DISK10000
```

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

This example shows how to add disk DISK20100 to the reduced RAIDset, RAID9:

```
SET RAID9 REPLACE=DISK20100
```

Reconstruction immediately begins on DISK20100.

See also

ADD RAIDSET
SHOW RAIDSETS

SET *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

The following table 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 B–4 SET UNIT Switches for Existing Containers

Container Type / **Switch**	PARTITION=partition-number	MAXIMUM_CACHED_ TRANSFER	READ_CACHE NOREAD_CACHE	READAHEAD_CACHE NOREADAHEAD_CACHE	WRITE_PROTECT NOWRITE_PROTECT	WRITEBACK_CACHE NOWRITEBACK_CACHE
RAIDset	✓	✓	✓	✓	✓	✓
Stripeset	✓	✓	✓	✓	✓	✓
Mirrorset	✓	✓	✓	✓	✓	✓
NoTransportable Disk	✓	✓	✓	✓	✓	✓
Transportable Disk		✓	✓	✓	✓	

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

MAXIMUM_CACHED_TRANSFER=n
MAXIMUM_CACHED_TRANSFER=32 (Default)
Sets the largest number of write blocks to be cached by the controller. The controller does not cache any transfers over the set size. Accepted values are 1 through 1024.

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

READ_CACHE (Default)
NOREAD_CACHE
Switches enable or disable the read-cache function for the unit.

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

RUN (Default)
NORUN
Controls the disk drive's operation and availability to the host. Specify RUN to make a unit available to the host. Specify NORUN to make a unit unavailable to the host and to cause any data in cache to be flushed to one or more drives. NORUN spins down the devices making up a unit. The drives making up the unit spin down after the data has been completely flushed.

Note Do not specify the RUN and NORUN switches for partitioned units.

WRITE_PROTECT
NOWRITE_PROTECT (Default)
Assigns to the unit's a write-protect policy.

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

Specify NOWRITE_PROTECT to write data to the unit, overwriting existing data.

WRITEBACK_CACHE (Default)
NOWRITEBACK_CACHE
Specifies whether or not the unit is to take advantage of the controller's write-back caching feature.

Specify WRITEBACK_CACHE for all new RAIDsets, mirrorsets, and units you want to take advantage of write-back caching.

Specify NOWRITEBACK_CACHE for units you want to receive data directly from the host without being cached.

Caution Specifying NOWRITEBACK_CACHE may result in data loss 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-type*
SHOW *device-name*
SHOW *storageset-type*
SHOW *storageset-name*
SHOW EMU
SHOW FAILEDSET
SHOW UNITS
SHOW *unit-number*

Parameters

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

device-name

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

EMU

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

storageset-type

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

- STORAGESETS—Shows all storagesets configured with the controller
- RAIDSETS—Shows all RAIDsets configured to the controller

- STRIPESETS—Shows all stripesets configured to the controller
- MIRRORSETS—Shows all mirrorsets configured to the controller
- SPARESET—Show the spareset configured to the controller
- FAILEDSET—Shows the failedset configured to the controller

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

UNITS
Displays information for all units configured to the controller.

In addition to the unit name you defined for the unit, the information includes the unique 128-bit subsystem unit ID. This ID consists of the controller node ID plus a 64-bit unit ID generated by the subsystem. You name the units, however, the subsystem identifies them internally using this identifier.

A unit on controller 1234 5678 9ABC EF00 would have an ID like the following:

1234 5678 9ABC EF00 0001 0001 3056 00D2

Each single disk unit or storage device in your subsystem is assigned a unique unit ID number. The controller constructs a unit ID number for each device you add to the subsystem. The ID number consists of the controller’s worldwide node ID and a unique, internally generated serial stamp. You cannot set or change unit ID numbers.

Unit ID numbers stay with the unit when you move the unit from one slot to another in the enclosure.

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

THIS_CONTROLLER
OTHER_CONTROLLER
Specifies the controller to be displayed.

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           100 D100
DISK10100     disk           1 1 0             D101
```

This example shows a full listing of devices attached to the controller:

SHOW DEVICES FULL

```
Name          Type                      Port Targ  Lun        Used by
-----------------------------------------------------------------------------
DISK10300 disk                         1   3 0        R0

        Switches:
          NOTRANSPORTABLE
          TRANSFER_RATE_REQUESTED = ASYNCHRONOUS (ASYNCHRONOUS negotiated)
        Size: 8378028 blocks
DISK20100     disk                         2   01 0        S0

        Switches:
          NOTRANSPORTABLE
          TRANSFER_RATE_REQUESTED = ASYNCHRONOUS (ASYNCHRONOUS negotiated)
        Size: 8377528 blocks
        Configuration being backed up on this container
```

This example shows how to display a complete listing of the mirrorset named MIRR1:

SHOW MIRR1

```
Name          Storageset                    Uses            Used by
-----------------------------------------------------------------------------
MIRR1         mirrorset                     DISK50300       S0
                                            DISK60300
        Switches:
          POLICY (for replacement) = BEST_PERFORMANCE
          COPY (priority) = NORMAL
          READ_SOURCE = LEAST_BUSY
          MEMBERSHIP = 2, 2 members present
        State:
          NORMAL
          DISK60300 (member  0) is NORMAL
          DISK50300 (member  1) is NORMAL
        Size: 17769177 blocks
```

This example shows the full information for a controller:

SHOW THIS_CONTROLLER FULL

```
Controller:

HSG80 (C) DEC ZG74100120 Software R052G-0, Hardware 0000
NODE_ID = 5000-1FE1-FF00-00B0
ALLOCATION_CLASS = 0SCSI_VERSION = SCSI-2
Not configured for dual-redundancy
Device Port SCSI address 7
TIME: NOT SET
Host PORT_1:
Reported PORT_ID = 5000-1FE1-FF00-00B1
PORT_1_PROFILE = PLDA
PORT_1_TOPOLOGY = LOOP_SOFT (loop up)
PORT_1_AL_PA = 01 (negotiated)

Host PORT_2:
Reported PORT_ID = 5000-1FE1-FF00-00B2
PORT_2_PROFILE = PLDA
PORT_2_TOPOLOGY = LOOP_SOFT (loop up)
PORT_2_AL_PA = 02 (negotiated)
Cache:
256 megabyte write cache, version 0012
Cache is GOOD
No unflushed data in cache
CACHE_FLUSH_TIMER = DEFAULT (10 seconds)
NOCACHE_UPS
Mirrored Cache:
Not enabled
Battery:
MORE THAN 50% CHARGED
Standby capacity: Less than one hour
Time to full charge: 31 hours
Expires: 23-AUG-1957
WARNING: BATTERY AT END OF LIFE WITHIN ONE WEEK, REPLACE BATTERY SOON!
Extended information:
Terminal speed 19200 baud, eight bit, no parity, 1 stop bit
Operation control: 00000001 Security state code: 33506
Configuration backup disabled
Other controller not responding - RESET signal NOT asserted - NINDY ON Temperature
within optimum limit.
```

This example shows how to display the current settings for the EMU:

SHOW EMU

```
EMU CABINET SETTINGS
    SENSOR_1_SETPOINT   35  DEGREES C
    SENSOR_2_SETPOINT   35  DEGREES C
    SENSOR_3_SETPOINT   35  DEGREES C
    FANSPEED AUTOMATIC
```

SHUTDOWN *controller*

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

Syntax
SHUTDOWN *controller*

Parameter
controller
Indicates which controller is to shut down. Specify OTHER_CONTROLLER or THIS_CONTROLLER.

Switches
IGNORE_ERRORS
NOIGNORE_ERRORS (Default)
Controls the reaction of the controller based on the status of write-back cache.

Caution The IGNORE_ERRORS switch causes the controller to keep unflushed data in the write-back cache until it restarts and is able to write the data to devices. Do not perform any hardware changes until the controller flushes the cache.

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

Specify NOIGNORE_ERRORS to instruct the controller to stop operation if the data within write-back cache cannot be written to the devices.

IMMEDIATE_SHUTDOWN
NOIMMEDIATE_SHUTDOWN (Default)
Instructs the controller when to shutdown.

Caution The IMMEDIATE_SHUTDOWN switch causes the controller to keep unflushed data in the write-back cache until it restarts and is able to write the data to devices. Do not perform any hardware changes until the controller flushes the cache.

Specify IMMEDIATE_SHUTDOWN to cause the controller to shutdown immediately without checking for online devices or before flushing data from the write-back cache to devices.

Specify NOIMMEDIATE_SHUTDOWN to cause the controller not to shutdown without checking for online devices or before all data has been flushed from the write-back cache to devices.

Examples
This example shows how to shut down "this controller":

```
SHUTDOWN THIS_CONTROLLER
```

This example shows how to shut down the other controller, even if it cannot write all of the write-back cached data to the units:

```
SHUTDOWN OTHER_CONTROLLER IGNORE_ERRORS
```

See also
RESTART *controller*
SELFTEST *controller*

UNMIRROR

Converts a one-member mirrorset back to a non-mirrored disk drive and deletes its mirrorset from the list of known mirrorsets. This command can be used on mirrorsets already members of higher-level containers (stripesets or units).

The UNMIRROR command is not valid for disk drives having a capacity greater than the capacity of the existing mirrorset. If a mirrorset is comprised of disk drives with different capacities, the mirrorset capacity is limited to the size of the smallest member; larger members contain unused capacity. If a member with unused capacity is the last remaining member of a mirrorset, the UNMIRROR command cannot be used to change the disk drive back to a single-disk unit. This change would cause a change in the reported disk capacity, possibly corrupting user data.

Syntax

UNMIRROR *disk-name*

Parameter

disk-name

Specifies the name of the normal mirrorset member to be removed from a mirror storageset.

Example

This example shows how to convert DISK10300 back to a single device:

```
UNMIRROR DISK10300
```

See also

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

APPENDIX C

LED Codes

This appendix shows and describes the LED codes that you may encounter while servicing the controller, cache module, and external cache battery.

Operator Control Panel LED Codes

Use Table C–1 to interpret solid OCP patterns and Table C–2 to interpret flashing OCP patterns. Use this legend for both of these tables:

■ = reset button on
□ = reset button off
● = LED on
❍ = LED off

Note If the reset button is flashing and an LED is lit continuously, either the devices on that LED's bus don't match the controller's configuration, or an error has occurred in one of the devices on that bus. Also, a single LED that is lit indicates a failure of the drive on that port.

Solid OCP Patterns

Table C–1 Solid OCP Patterns

Pattern	Error	Repair Action
■●●●●●●	DAEMON diagnostic failed hard in non-fault tolerant mode DAEMON diagnostic detected critical hardware component failure; controller can no longer operate	Verify that cache module is present. If the error persists, replace controller.
■●●●●●❍	Recursive bugcheck detected Last Failure event (bugcheck) occurred repeatedly	Replace controller
■●●●●❍●	NVPM structure revision greater than image's NVPM structure revision number is greater than the one that can be handled by the software version attempting to be executed	Replace program card with one that contains the latest software version
■●●●●❍❍	NVPM write loop hang Attempt to write data to NVPM failed	Replace controller
■●●●❍●●	NVPM read loop hang Attempt to read data from NVPM failed	Replace controller
■●●●❍●❍	An unexpected NMI occurred during Last Failure processing Last Failure processing interrupted by a Non-Maskable Interrupt (NMI)	Reset controller
■●●●❍❍●	NVPM configuration inconsistent Device configuration within the NVPM is inconsistent	Reset controller

Table C–1 Solid OCP Patterns (Continued)

Pattern	Error	Repair Action
■●●●❍❍❍	Controller operation terminated Last Failure event required termination of controller operation (e.g. SHUT DOWN VIA CLI)	Reset controller
■●●❍●●●	Software-induced controller reset expected Software-induced reset failed	Replace controller
■●●❍●●❍	Hardware-induced controller reset expected Automatic hardware reset failed	Replace controller
■●●❍●❍●	An unexpected bugcheck occurred during Last Failure processing Last Failure Processing interrupted by another Last Failure event	Reset controller
■●●❍❍●●	NVPM structure revision too low NVPM structure revision number is less than the one that can be handled by the software version attempting to be executed	Verify that the program card contains the latest software version. If the error persists, replace controller.
■●●❍❍●❍	Code load program card write failure Attempt to update program card failed	Replace card
■●●❍❍❍●	ILF$INIT unable to allocate memory Attempt to allocate memory by ILF$INIT failed	Replace controller
■●●❍❍❍❍	An unexpected bugcheck occurred before subsystem initialization completed An unexpected Last Failure occurred during initialization	Reinsert controller. If that does not correct the problem, reset the controller. If the error persists, try resetting the controller again, and replace it if no change occurs.

Table C–1 Solid OCP Patterns (Continued)

Pattern	Error	Repair Action
■●❍●●●●	Memory module has illegal DIMM configuration	Verify that DIMMs are installed as shown in Figure 1-3 on page 1-7.
■●❍●●●❍	Multiple cabinets have the same SCSI ID More than one cabinet have the same SCSI ID	Reconfigure PVA ID to uniquely-identify each cabinet in the subsystem. The cabinet with the controllers must be set to PVA ID 0; additional cabinets must use PVA IDs 2 and 3. If error continues after PVA settings are unique, replace each PVA module one at a time. Check cabinet if problem remains.
■●❍●●❍●	All master cabinet SCSI buses are not set to ID 0	Set PVA ID to 0 for the cabinet with the controllers. If problem persists, try the following repair actions: 1. Replace the PVA module 2. Replace the EMU 3. Remove all devices 4. Replace the cabinet
■●❍●●❍❍	Cabinet IO termination power out of range Faulty or missing IO module causes cabinet IO termination power to be out of range	Ensure that all of the cabinet's device SCSI buses have an I/O module. If problem persists, replace the failed I/O module.
■●❍●❍●●	Jumpers not terminators found on backplane One or more SCSI bus terminators are either missing from the backplane or broken	Ensure that cabinet's SCSI bus terminators are installed and that there are no jumpers. Replace the failed terminator if the problem continues.
■●❍●❍●❍	All cabinet IO modules are not of the same type Cabinet I/O modules are a combination of single-sided and differential	Ensure that the I/O modules in an extended subsystem are either all single-ended or all differential, not both.

Table C–1 Solid OCP Patterns (Continued)

Pattern	Error	Repair Action
■●❍●❍❍●	EMU protocol version incompatible The microcode in the EMU and the software in the controller are not compatible	Upgrade either the EMU microcode or the software (refer to the Release Notes that accompanied the controller’s software).
nlmlmmm	An unexpected Machine Fault/NMI occurred during Last Failure processing A machine fault was detected while a Non-Maskable Interrupt was processing	Reset the controller
nlmmlll	Memory module has insufficient usable memory	Replace indicated DIMM(s) (this indication is only provided when Fault LED logging is enabled).
nlmmllm	Indicated memory module is missing Controller is unable to detect a particular memory module	Insert memory module (cache board)
■❍❍❍❍❍❍	No program card detected or kill asserted by other controller Controller unable to read program card	Ensure that program card is properly seated while resetting the controller. If the error persists, try the card with another controller; or replace the card. Otherwise, replace the controller that reported the error.
ommmmmm	Catastrophic controller or power failure	Check power. If good, reset controller. If problem persists, reseat controller module and reset controller. If problem is still evident, replace controller module.

Flashing OCP Patterns

Table C–2 Flashing OCP Patterns

Pattern	Error	Repair Action
nmmmmml	Program card EDC error	Replace program card
nmmmlmm	Timer zero on the processor is bad	Replace controller
nmmmlml	Timer one on the processor is bad	Replace controller
nmmmllm	Processor Guarded Memory Unit (GMU) is bad	Replace controller
nmmlmll	Nonvolatile Journal Memory (JSRAM) structure is bad because of a memory error or an incorrect upgrade procedure	First, verify correct upgrade (see Release Notes). If error continues, replace controller
nmmllml	One or more bits in the diagnostic registers did not match the expected reset value	Press the reset button to restart the controller. If this does not correct the error, replace the controller
nmmlllm	Memory error in the JSRAM	Replace controller
nmmllll	Wrong image found on program card	Replace program card or replace controller if needed
nmlmmmm	Controller Module memory is bad	Replace controller
nmlmmlm	Controller Module memory addressing is malfunctioning	Replace controller
nmlmmll	Controller Module memory parity is not working	Replace controller
nmlmlmm	Controller Module memory controller timer has failed	Replace controller
nmllmml	The Controller Module memory controller interrupt handler has failed	Replace controller
nmllllm	During the diagnostic memory test, the Controller Module memory controller caused an unexpected Non-Maskable Interrupt (NMI)	Replace controller
nlmmlmm	The card’s code image changed when the contents were copied to memory	Replace controller
nllmmmm	The JSRAM battery is bad	Replace controller
nllmmlm	First-half diagnostics of the Time of Year Clock failed	Replace controller

Table C–2 Flashing OCP Patterns (Continued)

Pattern	Error	Repair Action
nllmmll	Second-half diagnostics of the Time of Year Clock failed	Replace controller
nllmlml	The processor bus-to-device bus bridge chip is bad	Replace controller
nlllmll	There is an unnecessary interrupt pending	Replace controller
nllllmm	There was an unexpected fault during initialization	Replace controller
nllllml	There was an unexpected maskable interrupt during initialization	Replace controller
nllllllm	There was an unexpected NMI during initialization	Replace controller
nllllll	An invalid process ran during initialization	Replace controller

APPENDIX D

Event Codes

This appendix describes the event codes that the fault-management software generates for spontaneous events and last-failure events.

Instance Codes

This table contains the instance codes that can be issued by the controller’s fault-management software.

Table D–1 Instance Codes

Instance Code	Description	Template
01010302	An unrecoverable hardware detected fault occurred.	01
0102030A	An unrecoverable software inconsistency was detected or an intentional restart or shutdown of controller operation was requested.	01
01032002	Nonvolatile parameter memory component EDC check failed; content of the component reset to default settings.	11
02020064	Disk Bad Block Replacement attempt completed for a write within the user data area of the disk. Note that due to the way Bad Block Replacement is performed on SCSI disk drives, information on the actual replacement blocks is not available to the controller and is therefore not included in the event report.	51
02032001	Journal SRAM backup battery failure; detected during system restart. The Memory Address field contains the starting physical address of the Journal SRAM.	12
02042001	Journal SRAM backup battery failure; detected during periodic check. The Memory Address field contains the starting physical address of the Journal SRAM.	12
02052301	A processor interrupt was generated by the CACHEA0 Memory Controller with an indication that the CACHE backup battery has failed or is low (needs charging). The Memory Address field contains the starting physical address of the CACHEA0 memory.	12
02072201	The CACHEAO Memory Controller failed testing performed by the Cache Diagnostics. The Memory Address field contains the starting physical address of the CACHEA0 memory.	14
02082201	The CACHEA1 Memory Controller failed testing performed by the Cache Diagnostics. The Memory Address field contains the starting physical address of the CACHEA1 memory.	14
02090064	A data compare error was detected during the execution of a compare modified READ or WRITE command.	51
020B2201	Failed read test of a write-back metadata page residing in cache. Dirty write-back cached data exists and cannot be flushed to media. The dirty data is lost. The Memory Address field contains the starting physical address of the CACHEA0 memory.	14
020C2201	Cache Diagnostics have declared the cache bad during testing. The Memory Address field contains the starting physical address of the CACHEA0 memory.	14

Table D–1 Instance Codes (Continued)

Instance Code	Description	Template
020D2401	The wrong write cache module is configured. The serial numbers do not match. Either the existing or the expected cache contains dirty write-back cached data. Note that in this instance the Memory Address, Byte Count, FX Chip Register, Memory Controller register, and Diagnostic register fields are undefined.	14
020E2401	The write cache module is missing. A cache is expected to be configured and contains dirty write-back cached data. Note that in this instance the Memory Address, Byte Count, FX Chip Register, Memory Controller register, and Diagnostic register fields are undefined.	14
02102401	The write cache modules are not configured properly for a dual-redundant configuration. One of the cache modules is not the same size to perform cache failover of dirty write-back cached data. Note that in this instance the Memory Address, Byte Count, FX Chip Register, Memory Controller register, and Diagnostic register fields are undefined.	14
02110064	Disk Bad Block Replacement attempt completed for a read within the user data area of the disk. Note that due to the way Bad Block Replacement is performed on SCSI disk drives, information on the actual replacement blocks is not available to the controller and is therefore not included in the event report.	51
021A0064	Disk Bad Block Replacement attempt completed for a write of controller metadata to a location outside the user data area of the disk. Note that due to the way Bad Block Replacement is performed on SCSI disk drives, information on the actual replacement blocks is not available to the controller and is therefore not included in the event report.	41
021B0064	Disk Bad Block Replacement attempt completed for a read of controller metadata from a location outside the user data area of the disk. Note that due to the way Bad Block Replacement is performed on SCSI disk drives, information on the actual replacement blocks is not available to the controller and is therefore not included in the event report.	41
021D0064	Unable to lock the other controller’s cache in a write-cache failover attempt. Either a latent error could not be cleared on the cache or the other controller did not release its cache. Note that in this instance the Memory Address, Byte Count, FX Chip register, Memory Controller register, and Diagnostic register fields are undefined.	14
021E0064	The device specified in the Device Locator field has been added to the RAIDset associated with the logical unit.The RAIDset is now in Reconstructing state.	51
02280064	The device specified in the Device Locator field has been added to the Mirrorset associated with the logical unit. The new Mirrorset member is now in Copying state.	51

Table D–1 Instance Codes (Continued)

Instance Code	Description	Template
022C0064	The device specified in the Device Locator has transitioned from Copying or Normalizing state to Normal state.	51
022E0064	The device specified in the Device Locator field has been converted to a Mirrorset associated with the logical unit.	51
022F0064	The mirrored device specified in the Device Locator field has been converted to a single device associated with the logical unit.	51
02383A01	The CACHEB0 Memory Controller, which resides on the other cache module failed testing performed by the Cache Diagnostics. This is the mirrored cache Memory Controller. The Memory Address field contains the starting physical address of the CACHEB0 memory.	14
02392201	Both the CACHEB0 Memory Controller and CACHEB1 Memory Controller, which resides on the other cache module, failed testing performed by the Cache Diagnostics. Data cannot be accessed in the primary cache or the mirror cache. The Memory Address field contains the starting physical address of the CACHEA0 memory.	14
023E2401	Metadata residing in the controller and on the two cache modules disagree as to the mirror node. Note that in this instance the Memory Address, Byte Count, FX Chip register, Memory Controller register, and Diagnostic register fields are undefined.	14
023F2301	The cache backup battery covering the mirror cache is insufficiently charged. The Memory Address field contains the starting physical address of the CACHEB1 memory.	12
02402301	The cache backup battery covering the mirror cache has been declared bad. Either it failed testing performed by the Cache Diagnostics during system startup or it was low (insufficiently charged) for longer than the expected duration. The Memory Address field contains the starting physical address of the CACHEB1 memory.	12
02412401	Mirrored cache writes have been disabled. Either the primary or the mirror cache has been declared bad or data invalid and will not be used. Note that in this instance the Memory Address, Byte Count, FX Chip register, Memory Controller register, and Diagnostic register fields are undefined.	14
02422464	Cache failover attempt failed because the other cache was illegally configured with DIMMs. Note that in this instance the Memory Address, Byte Count, FX Chip register, Memory Controller register, and Diagnostic register fields are undefined.	14
02492401	The write cache module which is the mirror for the primary cache is unexpectedly not present (missing). A cache is expected to be configured and it may contain dirty write cached data. Note that in this instance, the Memory Address, Byte Count, FX Chip register, Memory Controller register, and Diagnostic register fields are undefined.	14

Table D–1 Instance Codes (Continued)

Instance Code	Description	Template
024A2401	Mirroring is enabled and the primary write cache module is expectedly not present (missing). A cache is expected to be configured and it may contain dirty write cached data. Note that in this instance, the Memory Address, Byte Count, FX Chip register, Memory Controller register, and Diagnostic register fields are undefined.	14
024B2401	Write-back caching has been disabled either due to a cache or battery-related problem. The exact nature of the problem is reported by other instance codes. Note that in this instance the Memory Address, Byte Count, FX Chip register, Memory Controller register, and Diagnostic register fields are undefined.	14
024F2401	This cache module is populated with DIMMs incorrectly. Cache metadata resident in the cache module indicates that unflushed write cache data exists for a cache size different than what is found present. Note that in this instance the Memory Address, Byte Count, FX Chip register, Memory Controller register, and Diagnostic register fields are undefined.	14
0251000A	This command failed because the target unit is not online to the controller. The Information field of the Device Sense Data contains the block number of the first block in error.	51
0252000A	The last block of data returned contains a forced error. A forced error occurs when a disk block is successfully reassigned, but the data in that block is lost. Re-writing the disk block will clear the forced error condition. The Information field of the Device Sense Data contains the block number of the first block in error.	51
0253000A	The data supplied from the host for a data compare operation differs from the data on the disk in the specified block. The Information field of the Device Sense Data contains the block number of the first block in error.	51
0254000A	The command failed due to a host data transfer failure. The Information field of the Device Sense Data contains the block number of the first block in error.	51
0255000A	The controller was unable to successfully transfer data to target unit. The Information field of the Device Sense Data contains the block number of the first block in error.	51
0256000A	The write operation failed because the unit is Data Safety Write Protected. The Information field of the Device Sense Data contains the block number of the first block in error.	51
0257000A	An attempt to reassign a bad disk block failed. The contents of the disk block is lost. The Information field of the Device Sense Data contains the block number of the first block in error.	51

Table D–1 Instance Codes (Continued)

Instance Code	Description	Template
0258000A	This command was aborted prior to completion. The Information field of the Device Sense Data contains the block number of the first block in error.	51
0259000A	The write operation failed because the unit is hardware write protected. The Information field of the Device Sense Data contains the block number of the first block in error.	51
025A000A	The command failed because the unit became inoperative prior to command completion. The Information field of the Device Sense Data contains the block number of the first block in error.	51
025B000A	The command failed because the unit became unknown to the controller prior to command completion. The Information field of the Device Sense Data contains the block number of the first block in error.	51
025C000A	The command failed because of a unit media format error. The Information field of the Device Sense Data contains the block number of the first block in error.	51
025D000A	The command failed for an unknown reason. The Information field of the Device Sense Data contains the block number of the first block in error	51
025F2201	Memory diagnostics performed during controller initialization detected an excessive number (512 pages or more) of memory errors detected on the primary cache memory. Diagnostics have not declared the cache failed, due to the isolated bad memory regions, but this is a warning to replace the cache as soon as possible in case of further degradation. The software performed the necessary error recovery as appropriate. Note that in this instance the Memory Address and Byte Count fields are undefined.	14
02603A01	Memory diagnostics performed during controller initialization detected an excessive number (512 pages or more) of memory errors detected on mirrored cache memory. Diagnostics has not declared the cache failed, due to the isolated bad memory regions, but this is a warning to replace the cache as soon as possible in case of further degradation. The software performed the necessary error recovery as appropriate. Note that in this instance the Memory Address, Byte Count fields are undefined.	14
02613801	Memory diagnostics performed during controller initialization detected that the DIMM in location 1 failed on the cache module. Note that in this instance the Byte Count field in undefined.	14
02623801	Memory diagnostics performed during controller initialization detected that the DIMM in location 2 failed on the cache module. Note that in this instance the Byte Count field in undefined.	14
02633801	Memory diagnostics performed during controller initialization detected that the DIMM in location 3 failed on the cache module. Note that in this instance the Byte Count field in undefined.	14

Table D–1 Instance Codes (Continued)

Instance Code	Description	Template
02643801	Memory diagnostics performed during controller initialization detected that the DIMM in location 4 failed on the cache module. Note that in this instance the Byte Count field in undefined.	14
02653C01	Memory diagnostics performed during controller initialization detected that the DIMM in location 3 on the other controller’s cache module (on mirrored cache) failed. Mirroring has been disabled. Note that in this instance the Byte Count field is undefined.	14
02663C01	Memory diagnostics performed during controller initialization detected that the DIMM in location 4 on the other controller’s cache module (on mirrored cache) failed. Mirroring has been disabled. Note that in this instance the Byte Count field is undefined.	14
02675201	The device specified in the Device Locator field has been removed from the RAIDset associated with the logical unit. The removed device is now in the Failedset. The RAIDset is now in Reduced state.	51
0268530A	The device specified in the Device Locator field failed to be added to the RAIDset associated with the logical unit. The device will remain in the Spareset.	51
02695401	The device specified in the Device Locator field failed to be added to the RAIDset associated with the logical unit. The failed device has been moved to the Failedset.	51
026A5001	The RAIDset associated with the logical unit has gone inoperative.	51
026B0064	The RAIDset associated with the logical unit has transitioned from Normal state to Reconstructing state.	51
026C0064	The RAIDset associated with the logical unit has transitioned from Reconstructing state to Normal state.	51
026D5201	The device specified in the Device Locator field has been removed from the Mirrorset associated with the logical unit. The removed device is now in the Failedset.	51
026E0001	The device specified in the Device Locator field has been reduced from the Mirrorset associated with the logical unit. The nominal number of members in the mirrorset has been decreased by one. The reduced device is now available for use.	51
026F530A	The device specified in the Device Locator field failed to be added to the mirrorset associated with the logical unit. The device will remain in the spareset.	51
02705401	The device specified in the Device Locator field failed to be added to the mirrorset associated with the logical unit. The failed device has been moved to the Failedset.	51

Table D–1 Instance Codes (Continued)

Instance Code	Description	Template
02710064	The mirrorset associated with the logical unit has had its nominal membership changed. The new nominal number of members for the mirrorset is specified in the Device Sense Data Information field.	51
02725101	The Mirrorset associated with the logical unit has gone inoperative.	51
02730001	The device specified in the Device Locator field had a read error which has been repaired with data from another mirrorset member.	51
02745A0A	The device specified in the Device Locator field had a read error. Attempts to repair the error with data from another mirrorset member failed due to lack of alternate error-free data source.	51
02755601	The device specified in the Device Locator field had a read error. Attempts to repair the error with data from another mirrorset member failed due to a write error on the original device. The original device will be removed from the mirrorset.	51
02773D01	The mirrored cache is not being used because the data in the mirrored cache is inconsistent with the data in the primary cache. The primary cache contains valid data, so the controller is caching solely from the primary cache. The mirrored cache is declared "failed", but this is not due to a hardware fault, only inconsistent data. Mirrored writes have been disabled until this condition is cleared. NOte that in this instance the Memory Address, Byte Count, FX Chip register, Memory Controller register, and Diagnostic register fields are undefined.	14
02782301	The cache backup battery is not present. The Memory Address field contains the starting physical address of the CACHEA0 memory.	12
02792301	The cache backup battery covering the mirror cache is not present. The Memory Address field contains the starting physical address of the CACHEB1 memory.	12
027A2201	The CACHEB0 Memory Controller failed Cache Diagnostics testing performed on the other cache during a cache failover attempt. The Memory Address field contains the starting physical address of the CACHEB0 memory.	14
027B2201	The CACHEB1 Memory Controller failed Cache Diagnostics testing performed on the other cache during a cache failover attempt. The Memory Address field contains the starting physical address of the CACHEB1 memory.	14
027C2201	The CACHEB0 and CACHEB1 Memory Controllers failed Cache Diagnostics testing performed on the other cache during a cache failover attempt. The Memory Address field contains the starting physical address of the CACHEB0 memory.	14

Table D–1 Instance Codes (Continued)

Instance Code	Description	Template
027D5B01	The Mirrorset associated with the logical unit has gone inoperative due to a disaster tolerance failsafe locked condition.	51
027E5B01	The command failed because the disaster tolerance mirrorset went failsafe locked prior to command completion. The Information field of the Device Sense Data contains the block number of the first block in error.	51
027F2301	The CACHE backup battery has been declared bad. The battery did not become fully charged within the expected duration. The Memory Address field contains the starting physical address of the CACHEA0 memory.	12
02805B01	The command failed because the disaster tolerance mirrorset is failsafe locked. The Information field of the Device Sense Data contains the block number of the first block in error.	51
02815B01	The command failed because the disaster tolerance mirrorset is failsafe locked. The Information field of the Device Sense Data contains the block number of the first block in error.	51
02825C64	The Mirrorset associated with the logical unit has just had a membership change such that disaster tolerance failsafe error mode can now be enabled if desired.	51
03010101	No command control structures available for disk operation. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41
03022002	SCSI interface chip command timeout during disk operation. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41
03034002	Byte transfer timeout during disk operation. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41
03044402	SCSI bus errors during disk operation. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41
03052002	Device port SCSI chip reported gross error during disk operation. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41
03062002	Non-SCSI bus parity error during disk operation. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41
03070101	Source driver programming error encountered during disk operation. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41

Table D–1 Instance Codes (Continued)

Instance Code	Description	Template
03080101	Miscellaneous SCSI Port Driver coding error detected during disk operation. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41
03094002	An unrecoverable disk drive error was encountered while performing work related to disk unit operations.	51
030C4002	A Drive failed because a Test Unit Ready command or a Read Capacity command failed.	51
030D000A	Drive was failed by a Mode Select command received from the host.	51
030E4002	Drive failed due to a deferred error reported by drive.	51
030F4002	Unrecovered Read or Write error.	51
03104002	No response from one or more drives.	51
0311430A	Nonvolatile memory and drive metadata indicate conflicting drive configurations.	51
0312430A	The Synchronous Transfer Value differs between drives in the same storageset.	51
03134002	Maximum number of errors for this data transfer operation exceeded.	51
03144002	Drive reported recovered error without transferring all data.	51
03154002	Data returned from drive is invalid.	51
03164002	Request Sense command to drive failed.	51
03170064	Illegal command for pass through mode.	51
03180064	Data transfer request error.	51
03194002	Premature completion of a drive command.	51
031A4002	Command timeout.	51
031B0101	Watchdog timer timeout.	51
031C4002	Disconnect timeout.	51
031D4002	Unexpected bus phase.	51
031E4002	Disconnect expected.	51
031F4002	ID Message not sent by drive.	51
03204002	Synchronous negotiation error.	51
03214002	The drive unexpectedly disconnected from the SCSI bus.	51
03224002	Unexpected message.	51
03234002	Unexpected Tag message.	51
03244002	Channel busy.	51
03254002	Message Reject received on a valid message.	51
0326450A	The disk device reported Vendor Unique SCSI Sense Data.	51

Table D–1 Instance Codes (Continued)

Instance Code	Description	Template
03270101	A disk related error code was reported which was unknown to the Fault Management firmware. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41
0328450A	The disk device reported standard SCSI Sense Data.	51
03324002	SCSI bus selection timeout.	
03330002	Device power on reset.	
03344002	Target assertion of REQ after WAIT DISCONNECT.	
03354002	During device initialization a Test Unit Ready command or a Read Capacity command to the device failed.	
03364002	During device initialization the device reported a deferred error.	
03374002	During device initialization the maximum number of errors for a data transfer operation was exceeded.	
03384002	Request Sense command to the device failed.	
03394002	Command timeout.	
033A4002	Disconnect timeout.	
033B4002	Unexpected bus phase.	
033C4002	The device unexpectedly disconnected from the SCSI bus.	
033D4002	Unexpected message.	
033E4002	Message Reject received on a valid message.	
033F0101	No command control structures available for passthrough device operation.	
03402002	Device port SCSI chip reported gross error.	
03410101	Miscellaneous SCSI Port Driver coding error.	
03420101	A passthrough device related internal error code was reported which is not recognized by the Fault Management firmware.	
03434002	During device initialization the device reported unexpected standard SCSI Sense Data.	
03C80101	No command control structures available for operation to a device which is unknown to the controller. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41
03C92002	SCSI interface chip command timeout during operation to a device which is unknown to the controller. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41

Table D–1 Instance Codes (Continued)

Instance Code	Description	Template
03CA4002	Byte transfer timeout during operation to a device which is unknown to the controller. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41
03CB0101	Miscellaneous SCSI Port Driver coding error detected during operation to a device which is unknown to the controller. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41
03CC0101	An error code was reported which was unknown to the Fault Management software. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41
03CD2002	Device port SCSI chip reported gross error during operation to a device which is unknown to the controller. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41
03CE2002	Non-SCSI bus parity error during operation to a device which is unknown to the controller. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41
03CF0101	Source driver programming error encountered during operation to a device which is unknown to the controller. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41
03D04002	A failure occurred while attempting a SCSI Test Unit Ready or Read Capacity command to a device. The device type is unknown to the controller. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41
03D14002	The identification of a device does not match the configuration information. The actual device type is unknown to the controller. Note that in this instance the Associated Additional Sense Code, and Associated Additional Sense Code Qualifier fields are undefined.	41
03D24402	SCSI bus errors during device operation. The device type is unknown to the controller. Note that in this instance the Associated Additional Sense Code and Associated Additional Sense Code Qualifier fields are undefined.	41

Table D–1 Instance Codes (Continued)

Instance Code	Description	Template
03D3450A	During device initialization, the device reported the SCSI Sense Key NO SENSE. This indicates that there is no specific sense key information to be reported for the designated logical unit. This would be the case for a successful command or a command that received CHECK CONDITION or COMMAND TERMINATED status because one of the FM, EOM, or ILI bits is set to one in the sense data flags field.	41
03D4450A	During device initialization, the device reported the SCSI Sense Key RECOVERED ERROR. This indicates the last command completed successfully with some recovery action performed by the target.	41
03D5450A	During device initialization, the device reported the SCSI Sense Key NOT READY. This indicates that the logical unit addressed cannot be accessed. Operator intervention may be required to correct this condition.	41
03D6450A	During device initialization, the device reported the SCSI Sense Key MEDIUM ERROR. This indicates that the command terminated with a non-recovered error condition that was probably caused by a flaw in the medium or an error in the recorded data. This sense key may also be returned if the target is unable to distinguish between a flaw in the medium and a specific hardware failure (HARDWARE ERROR sense key).	41
03D7450A	During device initialization, the device reported the SCSI Sense Key HARDWARE ERROR. This indicates that the target detected a non-recoverable hardware failure (for example, controller failure, device failure, parity error, etc.) while performing the command or during a self test.	41
03D8450A	During device initialization, the device reported the SCSI Sense Key ILLEGAL REQUEST. Indicates that there was an illegal parameter in the command descriptor block or in the additional parameters supplied as data for some commands (FORMAT UNIT, SEARCH DATA, etc.). If the target detects an invalid parameter in the command descriptor block, then it shall terminate the command without altering the medium. If the target detects an invalid parameter in the additional parameters supplied as data, then the target may have already altered the medium. This sense key may also indicate that an invalid IDENTIFY message was received.	41
03D9450A	During device initialization, the device reported the SCSI Sense Key UNIT ATTENTION. This indicates that the removable medium may have been changed or the target has been reset.	41
03DA450A	During device initialization, the device reported the SCSI Sense Key DATA PROTECT. This indicates that a command that reads or writes the medium was attempted on a block that is protected from this operation. The read or write operation is not performed.	41

Table D–1 Instance Codes (Continued)

Instance Code	Description	Template
03DB450A	During device initialization, the device reported the SCSI Sense Key BLANK CHECK. This indicates that a write-once device encountered blank medium or format-defined end-of-data indication while reading or a write-once device encountered a non-blank medium while writing.	41
03DC450A	During device initialization, the device reported a SCSI Vendor Specific Sense Key. This sense key is available for reporting vendor specific conditions.	41
03DD450A	During device initialization, the device reported the SCSI Sense Key COPY ABORTED. This indicates a COPY, COMPARE, or COPY AND VERIFY command was aborted due to an error condition on the source device, the destination device, or both.	41
03DE450A	During device initialization, the device reported the SCSI Sense Key ABORTED COMMAND. This indicates the target aborted the command. The initiator may be able to recover by trying the command again.	41
03DF450A	During device initialization, the device reported the SCSI Sense Key EQUAL. This indicates a SEARCH DATA command has satisfied an equal comparison.	41
03E0450A	During device initialization, the device reported the SCSI Sense Key VOLUME OVERFLOW. This indicates a buffered peripheral device has reached the end-of-partition and data may remain in the buffer that has not been written to the medium. A RECOVER BUFFERED DATA command(s) may be issued to read the unwritten data from the buffer.	41
03E1450A	During device initialization, the device reported the SCSI Sense Key MISCOMPARE. This indicates the source data did not match the data read from the medium.	41
03E2450A	During device initialization, the device reported a reserved SCSI Sense Key.	41
03EE0064	The EMU for the cabinet indicated by the Associated Port field has become available. Note that the Associated Target, Associated Additional Sense Code, and the Associated Additional Sense Code Qualifier fields are undefined.	41
03EF8301	The EMU for the cabinet indicated by the Associated Port field has become unavailable. Note that the Associated Target, Associated Additional Sense Code, and the Associated Additional Sense Code Qualifier fields are undefined.	41

Table D–1 Instance Codes (Continued)

<table>
<tr><th>Instance Code</th><th>Description</th><th>Template</th></tr>
<tr><td>03F10502</td><td>The SWAP interrupt from the device port indicated by the Associated Port field can not be cleared. All SWAP interrupts from all ports will be disabled until corrective action is taken. When SWAP interrupts are disabled, both controller front panel button presses and removal/insertion of devices are not detected by the controller. Note that in this instance the Associated Target, Associated Additional Sense Code, and Associated Additional Sense Code Qualifier fields are undefined.</td><td>41</td></tr>
<tr><td>03F20064</td><td>The SWAP interrupts have been cleared and re-enabled for all device ports. Note that in this instance the Associated Port, Associated Target, Associated Additional Sense Code, and Associated Additional Sense Code Qualifier fields are undefined.</td><td>41</td></tr>
<tr><td>03F30064</td><td>An asynchronous SWAP interrupt was detected by the controller for the device port indicated by the Associated Port field. Possible reasons for this occurrence include:
■ device insertion or removal
■ shelf power failure
■ SWAP interrupts reenabled
Note that in this instance the Associated Target, Associated Additional Sense Code, and Associated Additional Sense Code Qualifier fields are undefined.</td><td>41</td></tr>
<tr><td>03F40064</td><td>Device services had to reset the port to clear a bad condition. Note that in this instance the Associated Target, Associated Additional Sense Code, and Associated Additional Sense Code Qualifier fields are undefined.</td><td>41</td></tr>
<tr><td>03F60402</td><td>The controller shelf is reporting a problem. This could mean one or both of the following:
■ If the shelf is using dual power supplies, one power supply has failed.
■ One of the shelf cooling fans has failed.
Note that in this instance the Associated Target, Associated Additional Sense Code, and Associated Additional Sense Code Qualifier fields are undefined.</td><td>41</td></tr>
<tr><td>03F70401</td><td>The shelf indicated by the Associated Port field is reporting a problem. This could mean one or both of the following:
■ If the shelf is using dual power supplies, one power supply has failed.
■ One of the shelf cooling fans has failed.
Note that in this instance the Associated Target, Associated Additional Sense Code, and Associated Additional Sense Code Qualifier fields are undefined.</td><td>41</td></tr>
</table>

Table D–1 Instance Codes (Continued)

Instance Code	Description	Template
03F80701	The EMU has detected one or more bad power supplies. Note that in this instance the Associated Target, Associated Additional Sense Code, and Associated Additional Sense Code Qualifier fields are undefined.	41
03F90601	The EMU has detected one or more bad fans. Note that in this instance the Associated Target, Associated Additional Sense Code, and Associated Additional Sense Code Qualifier fields are undefined.	41
03FA0D01	The EMU has detected an elevated temperature condition. Note that in this instance the Associated Target, Associated Additional Sense Code, and Associated Additional Sense Code Qualifier fields are undefined.	41
03FB0E01	The EMU has detected an external air sense fault. Note that in this instance the Associated Target, Associated Additional Sense Code, and Associated Additional Sense Code Qualifier fields are undefined.	41
03FC0F01	The EMU-detected power supply fault is now fixed. Note that in this instance the Associated Target, Associated Additional Sense Code, and Associated Additional Sense Code Qualifier fields are undefined.	41
03FD0F01	The EMU-detected bad-fan fault is now fixed. Note that in this instance the Associated Target, Associated Additional Sense Code, and Associated Additional Sense Code Qualifier fields are undefined.	41
03FE0F01	The EMU-detected elevated temperature fault is now fixed. Note that in this instance the Associated Target, Associated Additional Sense Code, and Associated Additional Sense Code Qualifier fields are undefined.	41
03FF0F01	The EMU-detected external air sense fault is now fixed. Note that in this instance the Associated Target, Associated Additional Sense Code, and Associated Additional Sense Code Qualifier fields are undefined.	41
07030B0A	Failover Control detected a receive packet sequence number mismatch. The controllers are out of synchronization with each other and are unable to communicate. Note that in this instance the Last Failure Code and Last Failure Parameters fields are undefined.	05
07040B0A	Failover Control detected a transmit packet sequence number mismatch. The controllers are out of synchronization with each other and are unable to communicate. Note that in this instance the Last Failure Code and Last Failure Parameters fields are undefined.	05
07050064	Failover Control received a Last Gasp message from the other controller. The other controller is expected to restart itself within a given time period. If it does not, it will be held reset with the “Kill” line.	05
07060C01	Failover Control detected that both controllers are acting as SCSI ID 6. Since ids are determined by hardware, it is unknown which controller is the real SCSI ID 6. Note that in this instance the Last Failure Code and Last Failure Parameters fields are undefined.	05

Table D–1 Instance Codes (Continued)

Instance Code	Description	Template
07070C01	Failover Control detected that both controllers are acting as SCSI ID 7. Since ids are determined by hardware, it is unknown which controller is the real SCSI ID 7. Note that in this instance the Last Failure Code and Last Failure Parameters fields are undefined.	05
07080B0A	Failover Control was unable to send keepalive communication to the other controller. It is assumed that the other controller is hung or not started. Note that in this instance the Last Failure Code and Last Failure Parameters fields are undefined.	05
0C00370A	Memory System Error Analysis is indicated in the information preserved during a previous last failure but no error conditions are indicated in the available Memory Controller registers. The Quadrant 0 Memory Controller (CACHEA0) registers content is supplied.	14
0C103E02	The Quadrant 0 Memory Controller (CACHEA0) detected an Address Parity error.	14
0C113E02	The Quadrant 1 Memory Controller (CACHEA1) detected an Address Parity error.	14
0C123E02	The Quadrant 2 Memory Controller (CACHEB0) detected an Address Parity error.	14
0C133E02	The Quadrant 3 Memory Controller (CACHEB1) detected an Address Parity error.	14
0C203E02	The Quadrant 0 Memory Controller (CACHEA0) detected a Data Parity error.	14
0C213E02	The Quadrant 1 Memory Controller (CACHEA1) detected a Data Parity error.	14
0C223E02	The Quadrant 2 Memory Controller (CACHEB0) detected a Data Parity error.	14
0C233E02	The Quadrant 3 Memory Controller (CACHEB1) detected a Data Parity error.	14
0C303F02	The Quadrant 0 Memory Controller (CACHEA0) detected a Multibit ECC error.	14
0C313F02	The Quadrant 1 Memory Controller (CACHEA1) detected a Multibit ECC error.	14
0C323F02	The Quadrant 2 Memory Controller (CACHEB0) detected a Multibit ECC error.	14
0C333F02	The Quadrant 3 Memory Controller (CACHEB1) detected a Multibit ECC error.	14
0C403E02	The Quadrant 0 Memory Controller (CACHEA0) detected a Firewall error.	14
0C413E02	The Quadrant 1 Memory Controller (CACHEA1) detected a Firewall error.	14
0C423E02	The Quadrant 2 Memory Controller (CACHEB0) detected a Firewall error.	14
0C433E02	The Quadrant 3 Memory Controller (CACHEB1) detected a Firewall error.	14
82042002	A spurious interrupt was detected during the execution of a Subsystem Built-In Self Test.	13
82052002	An unrecoverable error was detected during execution of the HOST PORT Subsystem Test. The system will not be able to communicate with the host.	13

Table D–1 Instance Codes (Continued)

Instance Code	Description	Template
82062002	An unrecoverable error was detected during execution of the UART/ DUART Subsystem Test. This will cause the console to be unusable. This will cause failover communications to fail.	13
82072002	An unrecoverable error was detected during execution of the FX Subsystem Test.	13
820A2002	An unrecoverable error was detected during execution of the PCI9060ES Test.	13
820B2002	An unrecoverable error was detected during execution of the Device Port Subsystem Built-In Self Test. One or more of the device ports on the controller module has failed; some/all of the attached storage is no longer accessible via this controller.	13

Last Failure Codes

This table contains the last failure codes can be issued by the controller's fault-management software.

Table D–2 Last Failure Codes

Code	Description
01000100	Memory allocation failure during executive initialization.
01010100	An interrupt without any handler was triggered.
01020100	Entry on timer queue was not of type AQ or BQ.
01030100	Memory allocation for a facility lock failed.
01040100	Memory initialization called with invalid memory type.
01082004	The core diagnostics reported a fault. ■ Last Failure Parameter[0] contains the error code value (same as blinking OCP LEDs error code). ■ Last Failure Parameter[1] contains the address of the fault. ■ Last Failure Parameter[2] contains the actual data value. ■ Last Failure Parameter[3] contains the expected data value.
01090105	■ An NMI occurred during EXEC$BUGCHECK processing. ■ Last Failure Parameter[0] contains the executive flags value. ■ Last Failure Parameter[1] contains the RIP from the NMI stack. ■ Last Failure Parameter[2] contains the read diagnostic register 0 value. ■ Last Failure Parameter[3] contains the FX Chip CSR value. ■ Last Failure Parameter[4] contains the SIP last failure code value
010D0110	The System Information structure within the System Information Page has been reset to default settings. The only known cause for this event is an I960 processor hang caused by a reference to a memory region that is not implemented. When such a hang occurs, controller modules equipped with inactivity watchdog timer circuitry will spontaneously reboot after the watchdog timer expires (within seconds of the hang). Controller modules not so equipped will just hang as indicated by the green LED on the OCP remaining in a steady state.
010E0110	All structures contained in the System Information Page (SIP) and the Last Failure entries have been reset to their default settings. This is a normal occurrence for the first boot following manufacture of the controller module and during the transition from one software version to another if and only if the format of the SIP is different between the two versions. If this event is reported at any other time, follow the recommended repair action associated with this Last Failure code.

Table D–2 Last Failure Codes (Continued)

Code	Description
010F0110	All structures contained in the System Information Page and the Last Failure entries have been reset to their default settings as the result of certain controller manufacturing configuration activities. If this event is reported at any other time, follow the recommended repair action associated with this Last Failure code.
01100100	Non-maskable interrupt entered but no Non-maskable interrupt pending. This is typically caused by an indirect call to address 0.
01110106	A bugcheck occurred during EXEC$BUGCHECK processing. ■ Last Failure Parameter [0] contains the executive flags value. ■ Last Failure Parameter [1] contains the RIP from the bugcheck call stack. ■ Last Failure Parameter [2] contains the first SIP last failure parameter value. ■ Last Failure Parameter [3] contains the second SIP last failure parameter value. ■ Last Failure Parameter [4] contains the SIP last failure code value. ■ Last Failure Parameter [5] contains the EXEC$BUGCHECK call last failure code value.
01150106	A bugcheck occurred before subsystem initialization completed. ■ Last Failure Parameter [0] contains the executive flags value. ■ Last Failure Parameter [1] contains the RIP from the bugcheck call stack. ■ Last Failure Parameter [2] contains the first SIP last failure parameter value. ■ Last Failure Parameter [3] contains the second SIP last failure parameter value. ■ Last Failure Parameter [4] contains the SIP last failure code value. ■ Last Failure Parameter [5] contains the EXEC$BUGCHECK call last failure code value.
01160108	The I960 reported a machine fault (parity error). ■ Last Failure Parameter [0] contains the RESERVED value. ■ Last Failure Parameter [1] contains the access type value. ■ Last Failure Parameter [2] contains the access address value. ■ Last Failure Parameter [3] contains the number of faults value. ■ Last Failure Parameter [4] contains the PC value. ■ Last Failure Parameter [5] contains the AC value. ■ Last Failure Parameter [6] contains the fault type and subtype values. ■ Last Failure Parameter [7] contains the RIP value.

Table D–2 Last Failure Codes (Continued)

Code	Description
01170108	The I960 reported a machine fault (parity error) while an NMI was being processed. ■ Last Failure Parameter [0] contains the RESERVED value. ■ Last Failure Parameter [1] contains the access type value. ■ Last Failure Parameter [2] contains the access address value. ■ Last Failure Parameter [3] contains the number of faults value. ■ Last Failure Parameter [4] contains the PC value. ■ Last Failure Parameter [5] contains the AC value. ■ Last Failure Parameter [6] contains the fault type and subtype values. ■ Last Failure Parameter [7] contains the RIP value.
01180105	A machine fault (parity error) occurred during EXEC$BUGCHECK processing. ■ Last Failure Parameter [0] contains the executive flags value. ■ Last Failure Parameter [1] contains the RIP from the machine fault stack. ■ Last Failure Parameter [2] contains the read diagnostic register 0 value. ■ Last Failure Parameter [3] contains the FX Chip CSR value. ■ Last Failure Parameter [4] contains the SIP last failure code value.
011B0108	The I960 reported a machine fault (nonparity error). ■ Last Failure Parameter [0] contains the Fault Data (2) value. ■ Last Failure Parameter [1] contains the Fault Data (1) value. ■ Last Failure Parameter [2] contains the Fault Data (0) value. ■ Last Failure Parameter [3] contains the Number of Faults value. ■ Last Failure Parameter [4] contains the PC value. ■ Last Failure Parameter [5] contains the AC value. ■ Last Failure Parameter [6] contains the Fault Flags, Type and Subtype values. ■ Last Failure Parameter [7] contains the RIP value (actual).
011C0011	Controller execution terminated via display of solid fault code in OCP LEDs. Note that upon receipt of this Last Failure in a last gasp message the other controller in a dual controller configuration will inhibit assertion of the KILL line. ■ Last Failure Parameter [0] contains the OCP LED solid fault code value.
018000A0	A powerfail interrupt occurred.
018600A0	A processor interrupt was generated with an indication that the other controller in a dual controller configuration asserted the KILL line to disable this controller.
018700A0	A processor interrupt was generated with an indication that the (//) RESET button on the controller module was depressed.
018800A0	A processor interrupt was generated with an indication that the program card was removed.

Table D–2 Last Failure Codes (Continued)

Code	Description
018900A0	A processor interrupt was generated with an indication that the controller inactivity watch dog timer expired.
018E2582	A NMI interrupt was generated with an indication that a memory system problem occurred. ■ Last Failure Parameter [0] contains the memory controller register address which encountered the error. ■ Last Failure Parameter [1] contains the memory controller's Command Status Register value.
018F2087	A NMI interrupt was generated with an indication that a controller system problem occurred. ■ Last Failure Parameter [0] contains the value of read diagnostic register 0. ■ Last Failure Parameter [1] contains the value of read diagnostic register 1. ■ Last Failure Parameter [2] contains PCI status. Bits 31:24 hold PCFX PSCR status and bits 15:08 hold PLX PSCR status. ■ Last Failure Parameter [3] contains the PCFX PDAL control/status register. ■ Last Failure Parameter [4] contains the IBUS address of error register. ■ Last Failure Parameter [5] contains the previous PDAL address of error register. ■ Last Failure Parameter [6] contains the current PDAL address of error register.
01902080	The PCI bus on the controller will not allow a master to initiate a transfer. Unable to provide further diagnosis of the problem.
02010100	Initialization code was unable to allocate enough memory to set up the send data descriptors.
02040100	Unable to allocate memory necessary for data buffers.
02050100	Unable to allocate memory for the Free Buffer Array.
02080100	A call to EXEC$ALLOCATE_MEM_ZEROED failed to return memory when populating the disk read DWD stack.
02090100	A call to EXEC$ALLOCATE_MEM_ZEROED failed to return memory when populating the disk write DWD stack.
020C0100	A call to EXEC$ALLOCATE_MEM_ZEROED failed to return memory when populating the miscellaneous DWD stack.
02100100	A call to EXEC$ALLOCATE_MEM_ZEROED failed to return memory when creating the device services state table.
02170100	Unable to allocate memory for the Free Node Array.

Table D–2 Last Failure Codes (Continued)

Code	Description
021D0100	Unable to allocate memory for the Free Buffer Array.
021F0100	Unable to allocate memory for WARPs and RMDs.
02210100	Invalid parameters in CACHE$OFFER_META call.
02220100	No buffer found for CACHE$MARK_META_DIRTY call.
02270104	A callback from DS on a transfer request has returned a bad or illegal DWD status. ■ Last Failure Parameter [0] contains the DWD Status. ■ Last Failure Parameter [1] contains the DWD address. ■ Last Failure Parameter [2] contains the PUB address. ■ Last Failure Parameter [3] contains the Device Port.
022C0100	A READ_LONG operation was requested for a Local Buffer Transfer. READ_LONG is not supported for Local Buffer Transfers.
022D0100	A WRITE_LONG operation was requested for a Local Buffer Transfer. WRTE_LONG is not supported for Local Buffer Transfers.
023A2084	A processor interrupt was generated by the controller’s XOR engine (FX), indicating an unrecoverable error condition. ■ Last Failure Parameter [0] contains the FX Control and Status Register (CSR). ■ Last Failure Parameter [1] contains the FX DMA Indirect List Pointer register (DILP). ■ Last Failure Parameter [2] contains the FX DMA Page Address register (DADDR). ■ Last Failure Parameter [3] contains the FX DMA Command and control register (DCMD).
02440100	The logical unit mapping type was detected invalid in va_set_disk_geometry()
02530102	An invalid status was returned from CACHE$LOOKUP_LOCK(). ■ Last Failure Parameter[0] contains the DD address. ■ Last Failure Parameter[1] contains the invalid status.
02560102	An invalid status was returned from CACHE$LOOKUP_LOCK(). ■ Last Failure Parameter[0] contains the DD address. ■ Last Failure Parameter[1] contains the invalid status.
02570102	An invalid status was returned from VA$XFER() during an operation. ■ Last Failure Parameter[0] contains the DD address. ■ Last Failure Parameter[1] contains the invalid status.

Table D–2 Last Failure Codes (Continued)

Code	Description
025A0102	An invalid status was returned from CACHE$LOOKUP_LOCK(). ■ Last Failure Parameter[0] contains the DD address. ■ Last Failure Parameter[1] contains the invalid status.
02620102	An invalid status was returned from CACHE$LOOKUP_LOCK(). ■ Last Failure Parameter[0] contains the DD address. ■ Last Failure Parameter[1] contains the invalid status.
02690102	An invalid status was returned from CACHE$OFFER_WRITE_DATA(). ■ Last Failure Parameter[0] contains the DD address. ■ Last Failure Parameter[1] contains the invalid status.
027B0102	An invalid status was returned from VA$XFER() in a complex ACCESS operation. ■ Last Failure Parameter[0] contains the DD address. ■ Last Failure Parameter[1] contains the invalid status.
027D0100	Unable to allocate memory for a Failover Control Block.
027E0100	Unable to allocate memory for a Failover Control Block.
027F0100	Unable to allocate memory for a Failover Control Block.
02800100	Unable to allocate memory for a Failover Control Block.
02840100	Unable to allocate memory for the XNode Array.
02860100	Unable to allocate memory for the Fault Management Event Information Packet used by the Cache Manager in generating error logs to the host.
02880100	Invalid FOC Message in cmfoc_snd_cmd.
028A0100	Invalid return status from DIAG$CACHE_MEMORY_TEST.
028B0100	Invalid return status from DIAG$CACHE_MEMORY_TEST.
028C0100	Invalid error status given to cache_fail.
028E0100	Invalid DCA state detected in init_crashover.
02910100	Invalid metadata combination detected in build_raid_node.
02920100	Unable to handle that many bad dirty pages (exceeded MAX_BAD_DIRTY). Cache memory is bad.

Table D–2 Last Failure Codes (Continued)

Code	Description
02930100	There was no free or freeable buffer to convert bad metadata or to borrow a buffer during failover of bad dirty.
02940100	A free Device Correlation Array entry could not be found during write-back cache failover.
02950100	Invalid DCA state detected in start_crashover.
02960100	Invalid DCA state detected in start_failover.
02970100	Invalid DCA state detected in init_failover.
02990100	A free RAID Correlation Array entry could not be found during write-back cache failover.
029A0100	Invalid cache buffer metadata detected while scanning the Buffer Metadata Array. Found a page containing dirty data but the corresponding Device Correlation Array entry does exist.
029D0100	Invalid metadata combination detected in build_bad_raid_node.
029F0100	The Cache Manager software has insufficient resources to handle a buffer request pending.
02A00100	VA change state is trying to change device affinity and the cache has data for this device.
02A10100	Pubs not one when transportable
02A20100	Pubs not one when transportable
02A30100	No available data buffers. If the cache module exists then this is true after testing the whole cache. Otherwise there were no buffers allocated from BUFFER memory on the controller module.
02A40100	A call to EXEC$ALLOCATE_MEM_ZEROED failed to return memory when allocating VAXDs.
02A50100	A call to EXEC$ALLOCATE_MEM_ZEROED failed to return memory when allocating DILPs.
02A60100	A call to EXEC$ALLOCATE_MEM_ZEROED failed to return memory when allocating Change State Work Items.
02A70100	A call to EXEC$ALLOCATE_MEM_ZEROED failed to return memory when allocating VA Request Items.

Table D–2 Last Failure Codes (Continued)

Code	Description
02A90100	Too many pending FOC$SEND requests by the Cache Manager. Code is not designed to handle more than one FOC$SEND to be pending because there’s no reason to expect more than one pending.
02AA0100	An invalid call was made to CACHE$DEALLOCATE_CLD. Either that device had dirty data or it was bound to a RAIDset.
02AB0100	An invalid call was made to CACHE$DEALLOCATE_SLD. A RAIDset member either had dirty data or write-back already turned on.
02AC0100	An invalid call was made to CACHE$DEALLOCATE_SLD. The RAIDset still has data (strip nodes).
02AD0180	The FX detected a compare error for data that was identical. This error has always previously occurred due to a hardware problem.
02AE0100	The mirrorset member count and individual member states are inconsistent. Discovered during a mirrorset write or erase.
02AF0102	An invalid status was returned from VA$XFER() in a write operation. ■ Last Failure Parameter[0] contains the DD address. ■ Last Failure Parameter[1] contains the invalid status.
02B00102	An invalid status was returned from VA$XFER () in an erase operation. ■ Last Failure Parameter [0] contains the DD address. ■ Last Failure Parameter [1] contains the invalid status.
02B10100	A mirrorset read operation was received and the round robin selection algorithm found no normal members in the mirrorset. Internal inconsistency.
02B20102	An invalid status was returned from CACHE$LOCK_READ during a mirror copy operation. ■ Last Failure Parameter[0] contains the DD address. ■ Last Failure Parameter[1] contains the invalid status.
02B30100	CACHE$CHANGE_MIRROR_MODE invoked illegally (cache bad, dirty data still resident in the cache.)
02B90100	Invalid code loop count attempting to find the Cache ID Blocks.
02BD0100	A mirrorset metadata online operation found no normal members in the mirrorset. Internal inconsistency.

Table D–2 Last Failure Codes (Continued)

Code	Description
02BE0100	No free pages in the other cache. In performing mirror cache failover, a bad page was found, and an attempt was made to recover the data from the good copy (primary/ mirror), but no free good page was found on the other cache to copy the data to.
02BF0100	Report_error routine encountered an unexpected failure status returned from DIAG$LOCK_AND_TEST_CACHE_B.
02C00100	Copy_buff_on_this routine expected the given page to be marked bad and it wasn't.
02C10100	Copy_buff_on_other routine expected the given page to be marked bad and it wasn't.
02C30100	CACHE$CREATE_MIRROR was invoked by C_SWAP under unexpected conditions (e.g., other controller not dead, bad lock state).
02C60100	Mirroring transfer found CLD with writeback state OFF.
02C70100	Bad BBR offsets for active shadowset, detected on write.
02C80100	Bad BBR offsets for active shadowset, detected on read.
02C90100	Illegal call made to CACHE$PURGE_META when the storageset was not quiesced.
02CA0100	Illegal call made to VA$RAID5_META_READ when another read (of metadata) is already in progress on the same strip.
02CB0000	A restore of the configuration has been done. This cleans up and restarts with the new configuration.
02CC0100	On an attempt, which is not allowed to fail, to allocate a cache node, no freeable cache node was found.
02D00100	Not all alter_device requests from VA_SAVE_CONFIG completed within the timeout interval.
02D30100	The controller has insufficient memory to allocate enough data structures used to manage metadata operations.
02D60100	An invalid storage set type was specified for metadata initialization.
02D90100	Bad CLD pointer passed setwb routine.
02DA0100	A fatal logic error occurred while trying to restart a stalled data transfer stream.
02DB0100	A call to EXEC$ALLOCATE_MEM_ZEROED failed to return memory when populating the disk read PCX DWD stack.

Table D–2 Last Failure Codes (Continued)

Code	Description
02DC0100	A call to EXEC$ALLOCATE_MEM_ZEROED failed to return memory when populating the disk write PCX DWD stack.
02DD0101	The VA state change deadman timer expired, and at least one VSI was still interlocked. ■ Last Failure Parameter [0] contains the nv_index.
02DE0100	An attempt to allocate memory for a null pub failed to get the memory.
02DF0101	License identified in Last Failure Parameter [0] was not forced valid.
02E11016	While attempting to restore saved configuration information, data for two unrelated controllers was found. The restore code is unable to determine which disk contains the correct information. The Port/Target/LUN information for the two disks is contained in the parameter list. Remove the disk containing the incorrect information, reboot the controller, and issue the SET THIS_CONTROLLER INITIAL_CONFIGURATION command. When the controller reboots, the proper configuration will be loaded. ■ Last Failure Parameter [0] contains the first disk port. ■ Last Failure Parameter [1] contains the first disk target. ■ Last Failure Parameter [2] contains the first disk LUN. ■ Last Failure Parameter [3] contains the second disk port. ■ Last Failure Parameter [4] contains the second disk target. ■ Last Failure Parameter [5] contains the second disk LUN.
02E20100	An attempt to allocate a va_cs_work item from the S_va_free_cs_work_queue failed.
02E30100	An attempt to allocate a free VAR failed.
02E40100	An attempt to allocate a free VAR failed.
O2E50100	An attempt to allocate a free VAR failed.
02E60100	An attempt to allocate a free VAR failed.
02E70100	An attempt to allocate a free VAR failed.
02E80100	An attempt to allocate a free VAR failed.
02E90100	An attempt to allocate a free VAR failed.
02EA0100	An attempt to allocate a free VAR failed.
02EB0100	An attempt to allocate a free metadata WARP failed.

Table D–2 Last Failure Codes (Continued)

Code	Description
02EC0101	An online request was received for a unit when both controllers had dirty data for the unit. The crash is to allow the surviving controller to copy over all of the dirty data. Last Failure Parameter [0] contains the nv_index of the unit.
02ED0100	On an attempt, which is not allowed to fail, to allocate a BDB, no freeable BDB was found.
02EE0102	A CLD is already allocated when it should be free. ■ Last Failure Parameter [0] contains the requesting entity. ■ Last Failure Parameter [1] contains the CLD index.
02EF0102	A CLD is free when it should be allocated. ■ Last Failure Parameter [0] contains the requesting entity. ■ Last Failure Parameter [1] contains the CLD index.
02F00100	The controller has insufficient free resources for the configuration restore process to obtain a facility lock.
02F10102	The configuration restore process encountered an unexpected non-volatile parameter store format. The process cannot restore from this version. ■ Last Failure Parameter [0] contains the version found. ■ Last Failure Parameter [1] contains the expected version.
02F20100	The controller has insufficient free resources for the configuration restore process to release a facility lock.
02F34083	A device read operation failed during the configuration restore operation. The controller is crashed to prevent possible loss of saved configuration information on other functioning devices. ■ Last Failure Parameter [0] contains the disk port. ■ Last Failure Parameter [1] contains the disk target. ■ Last failure Parameter [2] contains the disk LUN.
02F44083	The calculated error detection code on the saved configuration information is bad. The controller is crashed to prevent destruction of other copies of the saved configuration information. Remove the device with the bad information and retry the operation. ■ Last Failure Parameter [0] contains the disk port. ■ Last Failure Parameter [1] contains the disk target. ■ Last Failure Parameter [2] contains the disk LUN.

Table D–2 Last Failure Codes (Continued)

Code	Description
02F54083	The device saved configuration information selected for the restore process is from an unsupported controller type. Remove the device with the unsupported information and retry the operation. ■ Last Failure Parameter [0] contains the disk port. ■ Last Failure Parameter [1] contains the disk target. ■ Last Failure Parameter [2] contains the disk LUN.
02F60103	An invalid modification to the no_interlock VSI flag was attempted. ■ Last Failure Parameter [0] contains the nv_index of the config on which the problem was found. ■ Last Failure Parameter [1] contains modification flag. ■ Last Failure Parameter [2] contains the current value of the no_interlock flag. If the modification flag is 1, then an attempt was being made to set the no_interlock flag, and the no_interlock flag was not clear at the time. If the modification flag is 0, then an attempt was being made to clear the no_interlock flag, and the no_interlock flag was not set (== 1) at the time.
02F70100	During boot testing, one or more device ports (SCSI) were found to be bad. Due to a problem in the SYMBIOS 770 chip, the diagnostic may occasionally fail the port even though the hardware is OKAY. A reboot should clear up the problem. If the port is actually broken, logic to detect a loop that repeatedly causes the same bugcheck will cause a halt.
02F80103	An attempt was made to bring a unit online when the cache manager says that a member CLD was not in the appropriate state. ■ Last Failure Parameter [0] contains the nv_index of the config on which the problem was found. ■ Last Failure Parameter [1] contains the map type of that config. ■ Last Failure Parameter [2] contains the value from CACHE$CHECK_CID that was not acceptable.
02F90100	A call to EXEC$ALLOCATE_MEM_ZEROED failed to return memory when allocating structures for read ahead caching.
02FA0100	A read ahead caching data structure (RADD) is inconsistent.

Table D–2 Last Failure Codes (Continued)

Code	Description
02FB2084	A processor interrupt was generated by the controller’s XOR engine (FX), indicating an unrecoverable error condition. ■ Last Failure Parameter [0] contains the FX Control and Status Register (CSR). ■ Last Failure Parameter [1] contains the FX DMA Indirect List Pointer register (DILP). ■ Last Failure Parameter [2] contains the FX DMA Page Address register (DADDR). ■ Last Failure Parameter [3] contains the FX DMA Command and control register (DCMD).
02FC0180	The FX detected a compare error for data that was identical. This error has always previously occurred due to a hardware problem.
03010100	Failed request for port-specific scripts memory allocation.
03020101	Invalid SCSI direct-access device opcode in misc command DWD. ■ Last Failure Parameter [0] contains the SCSI command opcode.
03040101	Invalid SCSI CDROM device opcode in misc command DWD. ■ Last Failure Parameter [0] contains the SCSI command opcode.
03060101	Invalid SCSI device type in PUB. ■ Last Failure Parameter [0] contains the SCSI device type.
03070101	Invalid CDB Group Code detected during create of misc cmd DWD ■ Last Failure Parameter [0] contains the SCSI command opcode.
03080101	Invalid SCSI OPTICAL MEMORY device opcode in misc command DWD. ■ Last Failure Parameter [0] contains the SCSI command opcode.
03090101	Failed request for allocation of pci miscellaneous block Last Failure Parameter [0] contains the failed dwd command class.
030A0100	Error DWD not found in port in_proc_q.

Table D–2 Last Failure Codes (Continued)

Code	Description
030B0188	A dip error was detected when pcb_busy was set. ■ Last Failure Parameter [0] contains the PCB port_ptr value. ■ Last Failure Parameter [1] contains the new info NULL-SSTAT0-DSTAT-ISTAT. ■ Last Failure Parameter [2] contains the PCB copy of the device port DBC register. ■ Last Failure Parameter [3] contains the PCB copy of the device port DNAD register. ■ Last Failure Parameter [4] contains the PCB copy of the device port DSP register. ■ Last Failure Parameter [5] contains the PCB copy of the device port DSPS register. ■ Last Failure Parameter [6] contains the PCB copies of the device port SSTAT2/SSTAT1/SSTAT0/DSTAT registers. ■ Last Failure Parameter [7] contains the PCB copies of the device port LCRC/RESERVED/ISTAT/DFIFO registers.
031E0100	Can’t find in_error dwd on in-process queue.
031F0100	Either DWD_ptr is null or bad value in dsps.
03280100	SCSI CDB contains an invalid group code for a transfer command.
03290100	The required Event Information Packet (EIP) or Device Work Descriptor (DWD) were not supplied to the Device Services error logging code.
032B0100	A Device Work Descriptor (DWD) was supplied with a NULL Physical Unit Block (PUB) pointer.
03320101	An invalid code was passed to the error recovery thread in the error_stat field of the PCB. ■ Last Failure Parameter[0] contains the PCB error_stat code.

Table D–2 Last Failure Codes (Continued)

Code	Description
03330188	A parity error was detected by a device port while sending data out onto the SCSI bus. ■ Last Failure Parameter [0] contains the PCB port_ptr value. ■ Last Failure Parameter [1] contains the PCB copy of the device port TEMP register. ■ Last Failure Parameter [2] contains the PCB copy of the device port DBC register. ■ Last Failure Parameter [3] contains the PCB copy of the device port DNAD register. ■ Last Failure Parameter [4] contains the PCB copy of the device port DSP register. ■ Last Failure Parameter [5] contains the PCB copy of the device port DSPS register. ■ Last Failure Parameter [6] contains the PCB copies of the device port SSTAT2/ SSTAT1/SSTAT0/DSTAT registers. ■ Last Failure Parameter [7] contains the PCB copies of the device port LCRC/ RESERVED/ISTAT/DFIFO registers.
03350188	The TEA (bus fault) signal was asserted into a device port. ■ Last Failure Parameter [0] contains the PCB port_ptr value. ■ Last Failure Parameter [1] contains the PCB copy of the device port TEMP register. ■ Last Failure Parameter [2] contains the PCB copy of the device port DBC register. ■ Last Failure Parameter [3] contains the PCB copy of the device port DNAD register. ■ Last Failure Parameter [4] contains the PCB copy of the device port DSP register. ■ Last Failure Parameter [5] contains the PCB copy of the device port DSPS register. ■ Last Failure Parameter [6] contains the PCB copies of the device port SSTAT2/ SSTAT1/SSTAT0/DSTAT registers. ■ Last Failure Parameter [7] contains the PCB copies of the device port LCRC/ RESERVED/ISTAT/DFIFO registers.

Table D–2 Last Failure Codes (Continued)

Code	Description
03370108	A device port detected an illegal script instruction. ■ Last Failure Parameter [0] contains the PCB port_ptr value. ■ Last Failure Parameter [1] contains the PCB copy of the device port TEMP register. ■ Last Failure Parameter [2] contains the PCB copy of the device port DBC register. ■ Last Failure Parameter [3] contains the PCB copy of the device port DNAD register. ■ Last Failure Parameter [4] contains the PCB copy of the device port DSP register. ■ Last Failure Parameter [5] contains the PCB copy of the device port DSPS register. ■ Last Failure Parameter [6] contains the PCB copies of the device port SSTAT2/SSTAT1/SSTAT0/DSTAT registers. ■ Last Failure Parameter [7] contains the PCB copies of the device port LCRC/RESERVED/ISTAT/DFIFO registers.
03380188	A device port’s DSTAT register contains multiple asserted bits, or an invalidily asserted bit, or both. ■ Last Failure Parameter [0] contains the PCB port_ptr value. ■ Last Failure Parameter [1] contains the PCB copy of the device port TEMP register. ■ Last Failure Parameter [2] contains the PCB copy of the device port DBC register. ■ Last Failure Parameter [3] contains the PCB copy of the device port DNAD register. ■ Last Failure Parameter [4] contains the PCB copy of the device port DSP register. ■ Last Failure Parameter [5] contains the PCB copy of the device port DSPS register. ■ Last Failure Parameter [6] contains the PCB copies of the device port SSTAT2/SSTAT1/SSTAT0/DSTAT registers. ■ Last Failure Parameter [7] contains the PCB copies of the device port LCRC/RESERVED/ISTAT/DFIFO registers.

Table D–2 Last Failure Codes (Continued)

Code	Description
03390108	An unknown interrupt code was found in a device port's DSPS register. ■ Last Failure Parameter [0] contains the PCB port_ptr value. ■ Last Failure Parameter [1] contains the PCB copy of the device port TEMP register. ■ Last Failure Parameter [2] contains the PCB copy of the device port DBC register. ■ Last Failure Parameter [3] contains the PCB copy of the device port DNAD register. ■ Last Failure Parameter [4] contains the PCB copy of the device port DSP register. ■ Last Failure Parameter [5] contains the PCB copy of the device port DSPS register. ■ Last Failure Parameter [6] contains the PCB copies of the device port SSTAT2/SSTAT1/SSTAT0/DSTAT registers. ■ Last Failure Parameter [7] contains the PCB copies of the device port LCRC/RESERVED/ISTAT/DFIFO registers.
033C0101	An invalid code was seen by the error recovery thread in the er_funct_step field of the PCB. ■ Last Failure Parameter [0] contains the PCB er_funct_step code.
033E0108	An attempt was made to restart a device port at the SDP DBD. ■ Last Failure Parameter [0] contains the PCB port_ptr value. ■ Last Failure Parameter [1] contains the PCB copy of the device port TEMP register. ■ Last Failure Parameter [2] contains the PCB copy of the device port DBC register. ■ Last Failure Parameter [3] contains the PCB copy of the device port DNAD register. ■ Last Failure Parameter [4] contains the PCB copy of the device port DSP register. ■ Last Failure Parameter [5] contains the PCB copy of the device port DSPS register. ■ Last Failure Parameter [6] contains the PCB copies of the device port SSTAT2/SSTAT1/SSTAT0/DSTAT registers. ■ Last Failure Parameter [7] contains the PCB copies of the device port LCRC/RESERVED/ISTAT/DFIFO registers.

Table D–2 Last Failure Codes (Continued)

Code	Description
033F0108	An EDC error was detected on a read of a soft-sectored device path not yet implemented. ■ Last Failure Parameter [0] contains the PCB port_ptr value. ■ Last Failure Parameter [1] contains the PCB copy of the device port TEMP register. ■ Last Failure Parameter [2] contains the PCB copy of the device port DBC register. ■ Last Failure Parameter [3] contains the PCB copy of the device port DNAD register. ■ Last Failure Parameter [4] contains the PCB copy of the device port DSP register. ■ Last Failure Parameter [5] contains the PCB copy of the device port DSPS register. ■ Last Failure Parameter [6] contains the PCB copies of the device port SSTAT2/ SSTAT1/SSTAT0/DSTAT registers. ■ Last Failure Parameter [7] contains the PCB copies of the device port LCRC/ RESERVED/ISTAT/DFIFO registers.
03410101	Invalid SCSI device type in PUB. ■ Last Failure Parameter [0] contains the PUB SCSI device type.
03450188	A Master Data Parity Error was detected by a port. ■ Last Failure Parameter [0] contains the PCB port_ptr value. ■ Last Failure Parameter [1] contains the PCB copies of the device port DCMD/ DBC registers. ■ Last Failure Parameter [2] contains the PCB copy of the device port DNAD register. ■ Last Failure Parameter [3] contains the PCB copy of the device port DSP register. ■ Last Failure Parameter [4] contains the PCB copy of the device port DSPS register. ■ Last Failure Parameter [5] contains the PCB copies of the device port DSTAT/ SSTAT0/SSTAT1/SSTAT2 registers. ■ Last Failure Parameter [6] contains the PCB copies of the device port DFIFO/ ISTAT/SBCL/RESERVED registers. ■ Last Failure Parameter [7] contains the PCB copies of the device port SIST0/ SIST1/SXFER/SCNTL3 registers.
03470100	Insufficient memory available for target block allocation.
03480100	Insufficient memory available for device port info block allocation.
03490100	Insufficient memory available for autoconfig buffer allocation.

Table D–2 Last Failure Codes (Continued)

Code	Description
034A0100	Insufficient memory available for PUB allocation.
034B0100	Insufficient memory available for DS init buffer allocation.
034C0100	Insufficient memory available for static structure allocation.
034D0100	DS init DWDs exhausted.
034E2080	Diagnostics report all device ports are broken.
034F0100	Insufficient memory available for reselect target block allocation.
03500100	Insufficient memory available for command disk allocation.
03520100	A failure resulted when an attempt was made to allocate a DWD for use by DS CDI.
035A0100	Invalid SCSI message byte passed to DS.
035B0100	Insufficient DWD resources available for SCSI message passthrough.
03640100	Processing run_switch disabled for LOGDISK associated with the other controller.
03650100	Processing pub unblock for LOGDISK associated with the other controller.
03660100	No memory available to allocate pub to tell the other controller of reset to one if its LUNs
03670100	No memory available to allocate pub to tell the other controller of a BDR to one if its LUNs
036F0101	Either send_sdtr or send_wdtr flag set in a non-miscellaneous DWD. Last Failure Parameter [0] contains the invalid command class type.
03780181	In ds_get_resume_addr, the buffer address is non-longword aligned for FX access. Last Failure Parameter [0] contains the re-entry dbd address value.
03820100	Failed request for mapping table memory allocation.
03830100	Failed request pci 875 block memory allocation.
03850101	ds_alloc_mem called with invalid memory type Last Failure Parameter [0] contains the invalid memory type.
03860100	ds_alloc_mem was unable to get requested memory allocated: NULL pointer returned.
038C0100	Insufficient memory available for completion dwd array allocation.
03980100	Failed to allocate expandable EMU static work structures.

Table D–2 Last Failure Codes (Continued)

Code	Description
03990100	Failed to allocate expandable EMU work entry.
039A0100	Failed to allocate expandable EMU FOC work entry.
039B0100	EMU request work queue corrupted.
039C0100	EMU response work queue corrupted.
039D0100	EMU work queue corrupted.
039E0100	EMU foc request work queue corrupted.
039F0100	EMU foc response work queue corrupted.
03A08093	A configuration or hardware error was reported by the EMU. ■ Last Failure Parameter [0] contains the solid OCP pattern which identifies the type of problem encountered. ■ Last Failure Parameter [1] contains the cabinet ID reporting the problem. ■ Last Failure Parameter [2] contains the SCSI Port number where the problem exists (if port-specific).
03A18192	The EMU reported terminator power out of range. ■ Last Failure Parameter [0] contains the cabinet ID reporting the problem. ■ Last Failure Parameter [1] contains a bit mask indicating the SCSI Port number(s) where the problem exists. Bit 0 set indicates SCSI Port 0, Bit 1 set indicates SCSI Port 1, etc.
04010101	The requester id component of the instance code passed to FM$REPORT_EVENT is larger than the maximum allowed for this environment. ■ Last Failure Parameter[0] contains the instance code value.
04020102	The requester’s error table index passed to FM$REPORT_EVENT is larger than the maximum allowed for this requester. ■ Last Failure Parameter[0] contains the instance code value. ■ Last Failure Parameter[1] contains the requester error table index value.
04030102	The USB index supplied in the Event Information Packet (EIP) is larger than the maximum number of USBs. ■ Last Failure Parameter[0] contains the instance code value. ■ Last Failure Parameter[1] contains the USB index value.

Table D–2 Last Failure Codes (Continued)

Code	Description
04040103	The event log format found in V_fm_template_table is not supported by the Fault Manager. The bad format was discovered while trying to fill in a supplied Event Information Packet (EIP). ■ Last Failure Parameter[0] contains the instance code value. ■ Last Failure Parameter[1] contains the format code value. ■ Last Failure Parameter[2] contains the requester error table index value.
04050100	The Fault Manager could not allocate memory for its Event Information Packet (EIP) buffers.
040A0100	The caller of FM$CANCEL_SCSI_DE_NOTIFICATION passed an address of a deferred error notification routine which doesn’t match the address of any routines for which deferred error notification is enabled.
040E0100	FM$ENABLE_DE_NOTIFICATION was called to enable deferred error notification but the specified routine was already enabled to receive deferred error notification.
040F0102	The Event Information Packet (EIP)->generic.mscp1.flgs field of the EIP passed to FM$REPORT_EVENT contains an invalid flag. ■ Last Failure Parameter[0] contains the instance code value. ■ Last Failure Parameter[1] contains the value supplied in the Event Information Packet (EIP)->generic.mscp1.flgs field.
04100101	Unexpected template type found during fmu_display_errlog processing. ■ Last Failure Parameter[0] contains the unexpected template value.
04110101	Unexpected instance code found during fmu_memerr_report processing. ■ Last Failure Parameter[0] contains the unexpected instance code value.
04120101	CLIB$SDD_FAO call failed. ■ Last Failure Parameter[0] contains the failure status code value.
04140103	The template value found in the eip is not supported by the Fault Manager. The bad template value was discovered while trying to build an esd. ■ Last Failure Parameter [0] contains the instance code value. ■ Last Failure Parameter [1] contains the template code value. ■ Last Failure Parameter [2] contains the requester error table index value.
05010100	In recursive_nonconflict could not get enough memory for scanning the keyword tables for configuration name conflicts.

Table D–2 Last Failure Codes (Continued)

Code	Description
06010100	The DUART was unable to allocate enough memory to establish a connection to the CLI.
06020100	A port other than terminal port A was referred to by a set terminal characteristics command. This is illegal.
06030100	A DUP question or default question message type was passed to the DUART driver, but the pointer to the input area to receive the response to the question was NULL.
06040100	Attempted to detach unattached maintenance terminal.
06050100	Attempted output to unattached maintenance terminal.
06060100	Attempted input from output only maintenance terminal service.
06070100	The DUART was unable to allocate enough memory for its input buffers
06080000	Controller was forced to restart due to entry of a CNTRL-K character on the maintenance terminal.
07010100	All available slots in the FOC notify table are filled.
07020100	FOC$CANCEL_NOTIFY() was called to disable notification for a rtn that did not have notification enabled.
07030100	Unable to start the Failover Control Timer before main loop.
07040100	Unable to restart the Failover Control Timer.
07050100	Unable to allocate flush buffer.
07060100	Unable to allocate active receive fcb.
07070100	The other controller killed this, but could not assert the kill line because nindy on or in debug. So it killed this now.
07080000	The other controller crashed, so this one must crash too.
07090100	A call to EXEC$ALLOCATE_MEM_ZEROED failed to return memory when allocating VA Request Items.
08010101	A remote state change was received from the FOC thread that NVFOC does not recognize. ■ Last Failure Parameter[0] contains the unrecognized state value.
08020100	No memory could be allocated for a NVFOC information packet.

Table D–2 Last Failure Codes (Continued)

Code	Description
08030101	Work received on the S_nvfoc_bque did not have a NVFOC work id. ■ Last Failure Parameter[0] contains the id type value that was received on the NVFOC work queue.
08040101	Unknown work value received by the S_nvfoc_bque. ■ Last Failure Parameter[0] contains the unknown work value.
08060100	A really write command was received when the NV memory was not locked.
08070100	A write to NV memory was received while not locked.
08080000	The other controller requested this controller to restart.
08090010	The other controller requested this controller to shutdown.
080A0000	The other controller requested this controller to selftest.
080B0100	Could not get enough memory to build a FCB to send to the remote routines on the other controller.
080C0100	Could not get enough memory for FCBs to receive information from the other controller.
080D0100	Could not get enough memory to build a FCB to reply to a request from the other controller.
080E0101	An out-of-range receiver ID was received by the NVFOC communication utility (master send to slave send ACK). Last Failure Parameter[0] contains the bad id value.
080F0101	An out-of-range receiver ID was received by the NVFOC communication utility (received by master). Last Failure Parameter[0] contains the bad id value.
08100101	A call to NVFOC$TRANSACTION had a from field (id) that was out of range for the NVFOC communication utility. ■ Last Failure Parameter [0] contains the bad id value.
08110101	NVFOC tried to defer more than one FOC send. Last Failure Parameter[0] contains the master ID of the connection that had the multiple delays.
08140100	Could not allocate memory to build a workblock to queue to the NVFOC thread.
08160100	A request to clear the remote configuration was received but the memory was not locked.
08170100	A request to read the next configuration was received but the memory was not locked.

Table D–2 Last Failure Codes (Continued)

Code	Description
08180100	Could not get enough memory for FLS FCBs to receive information from the other controller.
08190100	An unlock command was received when the NV memory was not locked.
081A0100	Unable to allocate memory for remote work.
081B0101	Bad remote work received on remote work queue. ■ Last Failure Parameter[0] contains the id type value that was received on the NVFOC remote work queue.
081C0101	Bad member management work received. ■ Last Failure Parameter[0] contains the bad member management value that was detected.
081D0000	In order to go into mirrored cache mode, the controllers must be restarted.
081E0000	In order to go into nonmirrored cache mode, the controllers must be restarted.
081F0000	An FLM$INSUFFICIENT_RESOURCES error was returned from a FLM lock or unlock call.
08200000	Expected restart so the write_instance may recover from a configuration mismatch.
08210100	Unable to allocate memory to setup NVFOC lock/unlock notification routines.
09010100	Unable to acquire memory to initialize the FLM structures.
09640101	Work that was not FLM work was found on the FLM queue. Bad format is detected or the formatted string overflows the output buffer. ■ Last Failure Parameter [0] contains the work found.
09650101	Work that was not FLM work was found on the FLM queue. ■ Last Failure Parameter [0] contains the structure found.
09670101	Local FLM detected an invalid facility to act upon. ■ Last Failure Parameter [0] contains the faciltiy found.
09680101	Remote FLM detected an error and requested the local controller to restart. ■ Last Failure Parameter [0] contains the reason for the request.
09C80101	Remote FLM detected an invalid facility to act upon. ■ Last Failure Parameter [0] contains the facility found.

Table D–2 Last Failure Codes (Continued)

Code	Description
09C90101	Remote FLM detected an invalid work type. ■ Last Failure Parameter [0] contains the work type found.
09CA0101	Remote FLM detected an invalid work type. ■ Last Failure Parameter [0] contains the work type found.
09CB0012	Remote FLM detected that the other controller has a facility lock manager at an incompatible revision level with this controller. ■ Last Failure Parameter [0] contains the controller's FLM revision. ■ Last Failure Parameter [1] contains the other controller's FLM revision.
0A020100	ILF$CACHE_READY unable to allocate necessary DWDs.
0A030100	ILF$CACHE_READY buffers_obtained > non-zero stack entry count.
0A040100	ILF$CACHE_READY DWD overrun.
0A050100	ILF$CACHE_READY DWD underrun.
0A060100	ILF$CACHE_READY found buffer marked for other controller.
0A070100	CACHE$FIND_LOG_BUFFERS returned continuation handle > 0.
0A080100	Not processing a bugcheck.
0A090100	No active DWD.
0A0A0100	Current entry pointer is not properly aligned.
0A0B0100	Next entry pointer is not properly aligned.
0A0C0100	Next entry was partially loaded.
0A0E0100	Active DWD is not a DISK WRITE DWD as expected.
0A0F0100	New active DWD is not a DISK WRITE DWD as expected.
0A100100	Data buffer pointer is not properly aligned.
0A120100	Data buffer pointer is not properly aligned.
0A130100	Data buffer pointer is not properly aligned.
0A140100	New entry pointer is not properly aligned.
0A150100	New entry record type is out of range.

Table D–2 Last Failure Codes (Continued)

Code	Description
0A190102	ilf_depopulate_DWD_to_cache first page guard check failed. ■ Last Failure Parameter [0] contains the DWD address value ■ Last Failure Parameter [1] contains the buffer address value.
0A1C0102	ILF$LOG_ENTRY page guard check failed. ■ Last Failure Parameter [0] contains the DWD address value ■ Last Failure Parameter [1] contains the buffer address value.
0A1D0102	ILF$LOG_ENTRY page guard check failed. ■ Last Failure Parameter [0] contains the DWD address value ■ Last Failure Parameter [1] contains the buffer address value.
0A1E0102	ILF$LOG_ENTRY page guard check failed. ■ Last Failure Parameter [0] contains the DWD address value ■ Last Failure Parameter [1] contains the buffer address value.
0A1F0100	ilf_rebind_cache_buffs_to_DWDs found duplicate buffer for current DWD.
0A200101	Unknown bugcheck code passed to ilf_cache_interface_crash. ■ Last Failure Parameter [0] contains the unknown bugcheck code value.
0A210100	ilf_rebind_cache_buffs_to_DWDs found buffer type not IDX_ILF.
0A220100	ilf_rebind_cache_buffs_to_DWDs found buffer DBD index too big.
0A240100	ilf_check_handle_array_edc found ihiea EDC bad.
0A250100	ilf_get_next_handle found no free ihiea entry.
0A260100	ilf_remove_handle could not find specified handle.
0A270100	ilf_depopulate_DWD_to_cache could not find handle for first buffer.
0A280100	ilf_depopulate_DWD_to_cache buffer handle does not match current handle.
0A290100	ilf_rebind_cache_buffs_to_DWDs could not find handle for DWD being rebound.
0A2B0100	ILF$CACHE_READY cache manager did not return multiple of DWD DBDs worth of buffers.
0A2C0100	ilf_rebind_cache_buffs_to_DWDs page guard check failed.
0A2D0100	ilf_populate_DWD_from_cache buffer stack entry zero or not page aligned.
0A2E0100	ilf_populate_DWD_from_cache returned buffer type not IDX_ILF.

Table D–2 Last Failure Codes (Continued)

Code	Description
0A2F0100	ilf_rebind_cache_buffs_to_DWDs buffer stack entry not page aligned.
0A300100	ilf_depopulate_DWD_to_cache buffer stack entry zero or not page aligned.
0A310100	ilf_distribute_cache_DWDs active handle count not as expected.
0A320102	ILF$LOG_ENTRY, page guard check failed. ■ Last Failure Parameter [0] contains the DWD address value. ■ Last Failure Parameter [1] contains the buffer address value.
0B010010	Due to an operator request, the controllers non-volatile configuration information has been reset to its initial state.
0B020100	The controller has insufficient free memory to allocate a Configuration Manager work item needed to perform the requested configuration reset.
0B030100	The controller has insufficient free memory to allocate a Configuration Manager work item needed to perform the requested configuration restore.
0B040100	The controller has insufficient free memory to allocate a Configuration Manager WWL work item needed to perform the requested World-Wide LUN ID change.
0B050100	More requests to WWL$NOTIFY have been made than can be supported.
0B060100	A call to WWL$UPDATE resulted in the need for another World-Wide LUN ID slot, and no free slots were available.
20010100	The action for work on the CLI queue should be CLI_CONNECT, CLI_COMMAND_IN or CLI_PROMPT. If it isn’t one of these three, this bugcheck will result.
20020100	The FAO returned a non-successful response. This will only happen if a bad format is detected or the formatted string overflows the output buffer.
20030100	The type of work received on the CLI work queue wasn’t of type CLI.
20060100	A work item of an unknown type was placed on the CLI’s SCSI Virtual Terminal thread’s work queue by the CLI.
20080000	This controller requested this controller to restart.
20090010	This controller requested this controller to shutdown.
200A0000	This controller requested this controller to selftest.

Table D–2 Last Failure Codes (Continued)

Code	Description
200B0100	Could not get enough memory for FCBs to receive information from the other controller.
200D0101	After many calls to DS$PORT_BLOCKED, we never got a FALSE status back (which signals that nothing is blocked). ■ Last Failure Parameter[0] contains the port number (1 - n) that we were waiting on to be unblocked.
200E0101	While traversing the structure of a unit, a config_info node was discovered with an unrecognized structure type. ■ Last Failure Parameter[0] contains the structure type number that was unrecognized.
200F0101	A config_info node was discovered with an unrecognized structure type. ■ Last Failure Parameter[0] contains the structure type number that was unrecognized.
20100101	A config_node of type VA_MA_DEVICE had an unrecognized SCSI device type. ■ Last Failure Parameter[0] contains the SCSI device type number that was unrecognized.
20110100	An attempt to allocate memory so the CLI prompt messages could be deleted failed.
20120101	While traversing the structure of a unit, a config_info node was discovered with an unrecognized structure type. ■ Last Failure Parameter[0] contains the structure type number that was unrecognized.
20130101	While traversing the structure of a unit, the device was of an unrecognized type. ■ Last Failure Parameter[0] contains the SCSI device type that was unrecognized.
20150100	On failover, both controllers must be restarted for failover to take effect. This is how this controller is restarted in COPY=OTHER.
20160000	In order to go into mirrored cache mode, the controllers must be restarted.
20160100	Unable to allocate resources needed for the CLI local program.
20170000	In order to go into unmirrored cache mode, the controllers must be restarted.
20190010	A cache state of a unit remains WRITE_CACHE_UNWRITTEN_DATA. The unit is not ONLINE, thus this state would only be valid for a very short period of time.
201A0100	An attempt to allocate memory so a CLI prompt message could be reformatted failed.

Table D–2 Last Failure Codes (Continued)

Code	Description
201B0100	Insufficient resources to get memory to lock CLI.
201C0100	Insufficient resources to get memory to unlock CLI.
201D0100	With “set failover copy=other”, the controller which is having the configuration copied to will automatically be restarted via this bugcheck.
201E0101	CLI$ALLOCATE_STRUCT() was called by a process which it does not support ■ Last Failure Parameter [0] contains pscb address.
201F0101	CLI$DEALLOCATE_ALL_STRUCT() was called by a process which it does not support. ■ Last Failure Parameter [0] contains pscb address.
20200100	CLI$ALLOCATE_STRUCT() could not obtain memory for a new nvfoc_rw_remote_nvmem structure.
20220020	This controller requested this subsystem to poweroff.
20640000	Nindy was turned on.
20650000	Nindy was turned off.
20692010	To enter dual-redundant mode, both controllers must be of the same type.
206A0000	Controller restart forced by DEBUG CRASH REBOOT command.
206B0010	Controller restart forced by DEBUG CRASH NOREBOOT command.
206C0020	Controller was forced to restart in order for new controller code image to take effect.
206D0000	Controller code load was not completed because the controller could not rundown all units.
43000100	Encountered an unexpected structure type on hp_work_q.
43030100	Unable to allocate the necessary number of large Sense Data buckets in HPP_init().
43100100	Encountered a NULL completion routine pointer in a DD.
43130100	Could not allocate a large sense bucket.
43160100	A sense data bucket of unknown type (neither LARGE or SMALL) was passed to deallocate_SDB().
43170100	Call to VA$ENABLE_NOTIFICATION() failed due to INSUFFICIENT_RESOURCES.

Table D–2 Last Failure Codes (Continued)

Code	Description
43190100	Unable to allocate necessary memory in HPP_int().
431A0100	Unable to allocate necessary timer memory in HPP_int().
43210101	HPP detected unknown error indicated by HPT. ■ Last Failure Parameter [0] contains the error value.
43220100	Unable to obtain Free CSR in HPP().
44640100	Not enough abort requests in the system.
44650100	Exceeded the number of SEST abort retries.
44660100	Unable to allocate enough abort requests for Fibre Channel Host Port Transport software layer.
44670100	Unable to allocate enough command HTBs for Fibre Channel Host Port Transport software layer.
44680100	Unable to allocate enough FC HTBs for Fibre Channel Host Port Transport software layer.
44690100	Unable to allocate enough work requests for Fibre Channel Host Port Transport software layer.
446A0100	Unable to allocate enough HTBs for Fibre Channel Host Port Transport software layer.
446B0100	Unable to allocate enough TIS structures for Fibre Channel Host Port Transport software layer.
446C0100	Unable to allocate enough MFSs for Fibre Channel Host Port Transport software layer.
446D0100	Unable to allocate enough Tachyon headers for Fibre Channel Host port Transport software layer.
446E0100	Unable to allocate enough EDB structures for Fibre Channel Host Port Tansport software layer.
446F0100	Unable to allocate enough LSFS structures for Fibre Channel Host Port Transport software layer.
44700100	Unable to allocate enough TPS structures for Fibre Channel Host Port Transport software layer.
44720101	An illegal status was returned to the FLOGI command error handler. ■ Last Failure Parameter [0] contains error value.

Table D–2 Last Failure Codes (Continued)

Code	Description
44730101	An illegal completion message was returned by the Tachyon to I960. ■ Last Failure Parameter [0] contains the completion message type.
44740101	The Host Port Transport process handler received an illegal timer. ■ Last Failure Parameter [0] contains the timer pointer. type.
44750100	The Host Port Transport work handler received an illegal work request.
44760100	The Host Port Transport ran out of work requests.
44770102	An illegal script return value was received by the Host Port Transport init script handler. ■ Last Failure Parameter [0] contains the init function. ■ Last Failure Parameter [1] contains return value. The Host Port Transport ran out of work requests.
44780102	An illegal script return value was received by the Host Port Transport send script handler. ■ Last Failure Parameter [0] contains the send function. ■ Last Failure Parameter [1] contains return value. The Host Port Transport ran out of work requests.
44790102	An illegal script return value was received by the Host Port Transport response script handler. ■ Last Failure Parameter [0] contains the rsp function. ■ Last Failure Parameter [1] contains return value. The Host Port Transport ran out of work requests.
447A0102	An illegal script return value was received by the Host Port Transport error script handler. ■ Last Failure Parameter [0] contains the error function. ■ Last Failure Parameter [1] contains return value. The Host Port Transport ran out of work requests.
447B0100	The Host Port Transport response script handler received a response before a command was sent.

Table D–2 Last Failure Codes (Continued)

Code	Description
447C0101	Unhandled command HTB status. ■ Last Failure Parameter [0] contains the status value. The Host Port Transport ran out of work requests.
447D0100	The Host Port Transport ran out of command HTBs.
447E0100	The Host Port Transport memory for LOGI parameters.
447F0100	The Host Port Transport memory for LOGI parameters.
44800101	An illegal status was returned to the name service command error handler. ■ Last Failure Parameter [0] contains error value.
44810101	An illegal status was returned to the PLOGI command error handler. ■ Last Failure Parameter [0] contains error value.
44820101	An illegal abort type was given to the Host Port Transport abort handler. ■ Last Failure Parameter [0] contains abort type.
44830101	An illegal failover request was given to the Host Port Transport request handler. ■ Last Failure Parameter [0] contains failover request.
44840101	An illegal failover response was given to the Host Port Transport failover response handler. ■ Last Failure Parameter [0] contains failover response.
44850100	The Host Port Transport failover control had a bad send count.
44860100	Unable to allocate enough ESD structures for Fibre Channel Host Port Transport software layer.
44870101	An illegal abort type was given to the Host Port Transport abort handler. ■ Last Failure Parameter [0] contains abort type.
44892091	Host Port Hardware diagnostic field at system initialization. ■ Last Failure Parameter [0] contains failed port number.
64000100	Insufficient buffer memory to allocate data structures needed to propagate SCSI Mode Select changes to other controller.
64010100	During an initialization of LUN specific mode pages, an unexpected device type was encountered.

Table D–2 Last Failure Codes (Continued)

Code	Description
80010100	An HTB was not available to issue an I/O when it should have been.
80030100	DILX tried to release a facility that wasn't reserved by DILX.
80040100	DILX tried to change the unit state from MAINTENANCE_MODE to NORMAL but was rejected because of insufficient resources.
80050100	DILX tried to change the usb unit state from MAINTENANCE_MODE to NORMAL but DILX never received notification of a successful state change.
80060100	DILX tried to switch the unit state from MAINTENANCE_MODE to NORMAL but was not successful.
80070100	DILX aborted all cmds via va$d_abort() but the HTBS haven't been returned.
80090100	DILX received an end msg which corresponds to an op code not supported by DILX.
800A0100	DILX was not able to restart his timer.
800B0100	DILX tried to issue an I/O for an opcode not supported.
800C0100	DILX tried to issue a oneshot I/O for an opcode not supported.
800D0100	A DILX device control block contains an unsupported unit_state.
800F0100	A DILX cmd completed with a sense key that DILX does not support.
80100100	DILX could not compare buffers because no memory was available from EXEC$ALLOCATE_MEM_ZEROED.
80110100	While DILX was deallocating his deferred error buffers, at least one could not be found.
80120100	DILX expected an Event Information Packet (EIP) to be on the receive EIP queue but no EIPs were there.
80130100	DILX was asked to fill a data buffer with an unsupported data pattern.
80140100	DILX could not process an unsupported answer in dx$reuse_params().
80150100	A deferred error was received with an unsupported template.
83020100	An unsupported message type or terminal request was received by the CONFIG virtual terminal code from the CLI.
83030100	Not all alter_device requests from the CONFIG utility completed within the timeout interval.

Table D–2 Last Failure Codes (Continued)

Code	Description
84010100	An unsupported message type or terminal request was received by the CLONE virtual terminal code from the CLI.
85010100	HSUTIL tried to release a facility that wasn’t reserved by HSUTIL.
85020100	HSUTIL tried to change the unit state from MAINTENANCE_MODE to NORMAL but was rejected because of insufficient resources.
85030100	HSUTIL tried to change the usb unit state from MAINTENANCE_MODE to NORMAL but HSUTIL never received notification of a successful state change.
85040100	HSUTIL tried to switch the unit state from MAINTENANCE_MODE to NORMAL but was not successful.
86000020	Controller was forced to restart in order for new code load or patch to take effect.
86010010	The controller code load function is about to update the program card. This requires controller activity to cease. This code is used to inform the other controller this controller will stop responding to inter-controller communications during card update. An automatic restart of the controller at the end of the program card update will cause normal controller activity to resume.
86020011	The EMU firmware returned a bad status when told to prepare for a code load. Last Failure Parameter [0] contains the value of the bad status.

Recommended Repair Action Codes

This table contains the repair action codes assigned to each significant event in the system.

Table D–3 Recommended Repair Action Codes

Code	Description
00	No action necessary.
01	An unrecoverable hardware detected fault occurred or an unrecoverable software inconsistency was detected. Proceed with controller support avenues. Contact Digital Services.
03	Follow the recommended repair action contained as indicated in the Last Failure Code.
04	Two possible problem sources are indicated: ■ In the case of a shelf with dual power supplies, one of the power supplies has failed. Follow repair action 07 for the power supply with the Power LED out. ■ One of the shelf blowers has failed. Follow repair action 06.
05	Four possible problem sources are indicated: ■ Total power supply failure on a shelf. Follow repair action 09. ■ A device inserted into a shelf that has a broken internal SBB connector. Follow repair action 0A. ■ A standalone device is connected to the controller with an incorrect cable. Follow repair action 08. ■ A controller hardware failure. Follow repair action 20.
06	Determine which blower has failed and replace it.
07	Replace power supply.
08	Replace the cable. Refer to the specific device documentation.
09	Determine power failure cause.
0A	Determine which SBB has a failed connector and replace it.

Table D–3 Recommended Repair Action Codes (Continued)

Code	Description
0B	The other controller in a dual-redundant configuration has been reset with the “Kill” line by the controller that reported the event. To restart the “Killed” controller enter the CLI command RESTART OTHER on the “Surviving” controller and then depress the (//) RESET button on the “Killed” controller. If the other controller is repeatedly being “Killed” for the same or a similar reason, follow repair action 20.
0C	Both controllers in a dual-redundant configuration are attempting to use the same SCSI ID (either 6 or 7 as indicated in the event report). Note that the other controller of the dual-redundant pair has been reset with the “Kill” line by the controller that reported the event. Two possible problem sources are indicated: ■ A controller hardware failure. ■ A controller backplane failure. First, follow repair action 20 for the “Killed” controller. If the problem persists follow repair action 20 for the “Surviving” controller. If the problem still persists replace the controller backplane.
0D	The Environmental Monitor Unit has detected an elevated temperature condition. Check the shelf and its components for the cause of the fault.
0E	The Environmental Monitor Unit has detected an external air-sense fault. Check components outside of the shelf for the cause of the fault.
0F	An environmental fault previously detected by the Environmental Monitor Unit is now fixed. This event report is notification that the repair was successful.
10	Restore on-disk configuration information to original state.
20	Replace the controller module.
22	Replace the indicated cache module or the appropriate memory DIMMs on the indicated cache module.
23	Replace the indicated write cache battery. CAUTION: BATTERY REPLACEMENT MAY CAUSE INJURY.

Table D–3 Recommended Repair Action Codes (Continued)

Code	Description
24	Check for the following invalid write cache configurations: ■ If the wrong write cache module is installed, replace with the matching module or clear the invalid cache error via the CLI. Refer to Appendix B, “CLI Commands” for more information. ■ If the write cache module is missing, reseat cache if it is actually present, or add the missing cache module, or clear the invalid cache error via the CLI. Refer to Appendix B, “CLI Commands” for more details. ■ If in a dual-redundant configuration and one of the write cache modules is missing, match write cache boards with both controllers. ■ If in a dual-redundant configuration and both caches are not of the same type (both write cache), replace a cache module to assure both are compatible. ■ If in a dual-redundant configuration and both write caches are not of the same size, replace a cache module to assure both are compatible. ■ If the cache module is populated with memory DIMMs in an illegal configuration, reconfigure according to guidelines.
25	An unrecoverable Memory System failure occurred. Upon restart the controller will generate one or more Memory System Failure Event Sense Data Responses; follow the repair action(s) contained therein.
37	The Memory System Failure translator could not determine the failure cause. Follow repair action 01.
38	Replace the indicated cache memory DIMM.
39	Check that the cache memory DIMMs are properly configured.
3A	This error applies to this controller’s mirrored cache. Since the mirrored cache is physically located on the other controller’s cache module, replace the other controller’s cache module, or the appropriate memory DIMMs on the other controller’s cache module.
3C	This error applies to this controller’s mirrored cache. Since the mirrored cache is physically located on the other controller’s cache module, replace the indicated cache memory DIMM on the other controller’s cache module.

Table D–3 Recommended Repair Action Codes (Continued)

Code	Description
3D	Either the primary cache or the mirrored cache has inconsistent data. Check for the following conditions to determine appropriate means to restore mirrored copies. ■ If the mirrored cache is reported as inconsistent and a previous FRU Utility warmswap of the mirrored cache module was unsuccessful, retry the procedure via the FRU Utility, by removing the module and re-inserting the same or a new module. ■ Otherwise, enter the CLI command SHUTDOWN THIS to clear the inconsistency upon reboot.
3E	Replace the indicated cache module.
3F	No action necessary, cache diagnostics will determine whether the indicated cache module is faulty.
40	If the Sense Data FRU field is non-zero, follow repair action 41. Otherwise, replace the appropriate FRU associated with the device’s SCSI interface or the entire device.
41	Consult the device’s maintenance manual for guidance on replacing the indicated device FRU.
43	Update the configuration data to correct the problem.
44	Replace the SCSI cable for the failing SCSI bus. If the problem persists, replace the controller backplane, drive backplane, or controller module.
45	Interpreting the device supplied Sense Data is beyond the scope of the controller’s firmware. See the device’s service manual to determine the appropriate repair action, if any.
50	The RAIDset is inoperative for one or more of the following reasons: ■ More than one member malfunctioned. Perform repair action 55. ■ More than one member is missing. Perform repair action 58. ■ Before reconstruction of a previously replaced member completes another member becomes missing or malfunctions. Perform repair action 59. ■ The members have been moved around and the consistency checks show mismatched members. Perform repair action 58.
51	The mirrorset is inoperative for one or more of the following reasons: ■ The last NORMAL member has malfunctioned. Perform repair actions 55 and 59. ■ The last NORMAL member is missing. Perform repair action 58. ■ The members have been moved around and the consistency checks show mismatched members. Perform repair action 58.

Table D–3 Recommended Repair Action Codes (Continued)

Code	Description
52	The indicated Storageset member was removed for one of the following reasons: ■ The member malfunctioned. Perform repair action 56. ■ By operator command. Perform repair action 57.
53	The STORAGESET may be in a state that prevents the adding of a replacement member, check the state of the STORAGESET and its associated UNIT and resolve the problems found before adding the replacement member.
54	The device may be in a state that prevents adding it as a replacement member or may not be large enough for the STORAGESET. Use another device for the ADD action and perform repair action 57 for the device that failed to be added.
55	Perform the repair actions indicated in any and all event reports found for the devices that are members of the STORAGESET.
56	Perform the repair actions indicated in any and all event reports found for the member device that was removed from the STORAGESET. Then perform repair action 57.
57	Delete the device from the FAILEDSET and redeploy, perhaps by adding it to the SPARESET so it will be available to be used to replace another failing device.
58	Install the physical devices that are members of the STORAGESET in the proper Port, Target, and LUN locations.
59	Delete the STORAGESET, recreate it with the appropriate ADD, INITIALIZE, and ADD UNIT commands and reload its contents from backup storage.
5A	Restore the MIRRORSET data from backup storage.
5B	The mirrorset is inoperative due to a disaster tolerance failsafe locked condition, as a result of the loss of all local or remote NORMAL/NORMALIZING members while ERROR_MODE=FAILSAFE was enabled. To clear the failsafe locked condition, enter the CLI command SET unit-number ERROR_MODE=NORMAL.
5C	The mirrorset has at least one local NORMAL/NORMALIZING member and one remote NORMAL/NORMALIZING member. Failsafe error mode can now be enabled by entering the CLI command SET unit-number ERROR_MODE=FAILSAFE.

Table D–3 Recommended Repair Action Codes (Continued)

Code	Description
80	An EMU fault has occurred.
81	The EMU reported terminator power out of range. Replace the indicated I/O module(s).
83	An EMU (Environmental Monitoring Unit) has become unavailable. ■ This EMU Unit (and associated cabinet) may have been removed from the subsystem; no action is required. ■ The cabinet has lost power; restore power to the cabinet. ■ The EM- to-EMU communications bus cable has been disconnected or broken; replace or reconnect the cable to reestablish communications. ■ The specified EMU is broken; replace the EMU module. ■ The EMU in cabinet 0 is broken; replace the EMU module.

Component Identifier Codes

This table lists the component identifier codes.

Table D–4 Component Identifier Codes

Code	Description
01	Executive Services
02	Value Added Services
03	Device Services
04	Fault Manager
05	Common Library Routines
06	Dual Universal Asynchronous Receiver/Transmitter Services
07	Failover Control
08	Nonvolatile Parameter Memory Failover Control
09	Facility Lock Manager
0A	Integrated Logging Facility
0B	Configuration Manager Process
0C	Memory Controller Event Analyzer
20	Command Line Interpreter
43	Host Port Protocol Layer
44	Host Port Transport Layer
64	SCSI Host Value Added Services
80	Disk Inline Exercise (DILX)
82	Subsystem Built-In Self Tests (BIST)
83	Device Configuration Utilities (CONFIG/CFMENU)
84	Clone Unit Utility (CLONE)
85	Format and Device Code Load Utility (HSUTIL)
86	Code Load/Code Patch Utility (CLCP)

Event Threshold Codes

This table lists the classifications for event notification and recovery threshold values.

Table D–5 Event Notification/Recovery Threshold Classifications

Threshold Value	Classification	Description
01	IMMEDIATE	Failure or potential failure of a component critical to proper controller operation is indicated; immediate attention is required.
02	HARD	Failure of a component that affects controller performance or precludes access to a device connected to the controller is indicated.
0A	SOFT	An unexpected condition detected by a controller firmware component (e.g., protocol violations, host buffer access errors, internal inconsistencies, uninterpreted device errors, etc.) or an intentional restart or shutdown of controller operation is indicated.
64	INFORMATIONAL	An event having little or no effect on proper controller or device operation is indicated.

ASC/ASCQ Codes

This table lists SCSI ASC and ASCQ codes.

Table D–6 ASC and ASCQ Codes

ASC Code	ASCQ Code	Description
04	80	Logical unit is disaster tolerant failsafe locked (inoperative).
3F	85	Test Unit Ready or Read Capacity Command failed
3F	87	Drive failed by a Host Mode Select command.
3F	88	Drive failed due to a deferred error reported by drive.
3F	90	Unrecovered Read/Write error.
3F	C0	No response from one or more drives.
3F	C2	NV memory and drive metadata indicate conflicting drive configurations.
3F	D2	Synchronous Transfer Value differences between drives.
80	00	Forced error on Read
82	01	No Command control structures available.
84	04	Command failed - SCSI ID verification failed.
85	05	Data returned from drive is invalid.
89	00	Request Sense command to drive failed.
8A	00	Illegal command for pass through mode.
8C	04	Data transfer request error.
8F	00	Premature completion of a drive command.
93	00	Drive returned vendor unique sense data.
A0	00	Last failure event report.
A0	01	Nonvolatile parameter memory component event report.
A0	02	Backup battery failure event report.
A0	03	Subsystem built-in self test failure event report.
A0	04	Memory system failure event report.

Table D–6 ASC and ASCQ Codes (Continued)

ASC Code	ASCQ Code	Description
A0	05	Failover event report.
A0	07	RAID membership event report.
A0	0A	Disaster Tolerance failsafe error mode can now be enabled.
A1	00	Shelf OK is not properly asserted.
A1	01	Unable to clear SWAP interrupt. Interrupt disabled.
A1	02	Swap interrupt re-enabled.
A1	03	Asynchronous SWAP detected.
A1	04	Controller shelf OK is not properly asserted.
A1	0A	EMU fault: Power Supplies not OK.
A1	0B	EMU fault: Fans not OK.
A1	0C	EMU fault: Temperature not OK.
A1	0D	EMU fault: External Air Sense not OK.
A1	10	Power supply fault is now fixed.
A1	11	Fans fault is now fixed.
A1	12	Temperature fault is now fixed.
A1	13	External Air Sense fault is now fixed.
A1	14	EMU and cabinet now available.
A1	15	EMU and cabinet now unavailable.
B0	00	Command timeout.
B0	01	Watchdog timer timeout.
D0	01	Disconnect timeout.
D0	02	Chip command timeout.
D0	03	Byte transfer timeout.
D1	00	Bus errors.
D1	02	Unexpected bus phase.

Table D–6 ASC and ASCQ Codes (Continued)

ASC Code	ASCQ Code	Description
D1	03	Disconnect expected.
D1	04	ID Message not sent.
D1	05	Synchronous negotiation error.
D1	07	Unexpected disconnect.
D1	08	Unexpected message.
D1	09	Unexpected Tag message.
D1	0A	Channel busy.
D1	0B	Device initialization failure. Device sense data available.
D2	00	Miscellaneous SCSI driver error.
D2	03	Device services had to reset the bus.
D3	00	Drive SCSI chip reported gross error.
D4	00	Non-SCSI bus parity error.
D5	02	Message Reject received on a valid message.
D7	00	Source driver programming error.
E0	03	Fault Manager detected an unknown error code.
E0	06	Maximum number of errors for this I/O exceeded.
E0	07	Drive reported recovered error without transferring all data.

Templates

Figure D–1 Passthrough Device Reset Event Sense Data Response Format

off \ bit	7	6	5	4	3	2	1	0
0	Valid	Error Code						
1	Segment							
2	FM	EOM	ILI	Rsvd	Sense Key			
3-6	Information							
7	Additional Sense Length							
8-11	Instance Code							
12	Additional Sense Code (ASC)							
13	Additional Sense Code Qualifier (ASCQ)							
14	Field Replaceable Unit Code							
15	SKSV	Sense Key Specific						
16	Sense Key Specific							
17	Sense Key Specific							

CXO-5093A-MC

Figure D–2 Controller Level Sense Data Format

off \ bit	7	6	5	4	3	2	1	0
0	Unusd	Error Code						
1	Unused							
2	Unused				Sense Key			
3-6	Unused							
7	Additional Sense Length							
8-11	Unused							
12	Additional Sense Code (ASC)							
13	Additional Sense Code Qualifier (ASCQ)							
14	Unused							
15–17	Unused							
18–31	Reserved							
32-35	Instance Code							
36	Template							
37	TM0	TM1	Rsvd	Rsvd	Rsvd	Rsvd	Rsvd	HCE
38–53	Reserved							
54–69	Controller Board Serial Number							
70–73	Controller Software Revision Level							
74-75	Reserved							
76	LUN Status							
77–159	Template Dependent Information							

CXO6173A

Figure D–3 Device Level Sense Data Format

off \ bit	7	6	5	4	3	2	1	0
0	Valid	Error Code						
1	Segment							
2	FM	EOM	ILI	Rsvd	Sense Key			
3-6	Information							
7	Additional Sense Length							
8-11	Command Specific Information							
12	Additional Sense Code (ASC)							
13	Additional Sense Code Qualifier (ASCQ)							
14	Field Replaceable Unit Code							
15	SKSV	Sense Key Specific						
16								
17								
18–19	Reserved							
20	Total Number of Errors							
21	Total Retry Count							
22	Most Recent ASC							
23	Most Recent ASCQ							
24	Least Recent ASC							
25	Least Recent ASCQ							
26	Port							
27	Target							
28	Logical Unit Number							
29–31	Reserved							
32–35	Instance Code							
36	Template							
37	TM0	TM1	Rsvd	Rsvd	Rsvd	Rsvd	Rsvd	HCE
38	Reserved							
39	Command Opcode							
40	DSSD	IDSD	FBW	MSBD	UWEUO	BUFMODE		
41	Operation Code							

off \ bit	7	6	5	4	3	2	1	0
42	LUN			Reserved				
43–46	Logical Block Address							
47	Reserved							
48–49	Transfer, Parameter, or Allocation Length							
50	Control							
51	Host ID							
52–53	Reserved							
54–69	Controller Board Serial Number							
70–73	Controller Software Revision Level							
74–75	Reserved							
76	LUN Status							
77-78	Reserved							
79-82	Device Firmware Revision Level							
83–98	Device Product ID							
99–100	Reserved							
101	Device Type							
102–103	Reserved							
104	Valid	Error Code						
105	Segment							
106	FM	EOM	ILI	Rsvd	Sense Key			
107–110	Information							
111	Additional Sense Length							
112–115	Command Specific Information							
116	Additional Sense Code (ASC)							
117	Additional Sense Code Qualifier (ASCQ)							
118	Field Replaceable Unit Code							
119	SKSV	Sense Key Specific						
120								
121								
122–159	Reserved							

CXO6174A

Figure D–4 Template 01 - Last Failure Event Sense Data Response Format

off \ bit	7	6	5	4	3	2	1	0
0	Unusd	Error Code						
1	Unused							
2	Unused				Sense Key			
3-6	Unused							
7	Additional Sense Length							
8-11	Unused							
12	Additional Sense Code (ASC)							
13	Additional Sense Code Qualifier (ASCQ)							
14	Unused							
15–17	Unused							
18–31	Reserved							
32-35	Instance Code							
36	Template							
37	Template Flags							
38–53	Reserved							
54–69	Controller Board Serial Number							
70–73	Controller Software Revision Level							
74–75	Reserved							
76	LUN Status							
77–103	Reserved							
104-107	Last Failure Code							
108–111	Last Failure Parameter[0]							
112–115	Last Failure Parameter[1]							
116–119	Last Failure Parameter[2]							
120–123	Last Failure Parameter[3]							
124–127	Last Failure Parameter[4]							
128–131	Last Failure Parameter[5]							
132–135	Last Failure Parameter[6]							
136–139	Last Failure Parameter[7]							
140-159	Reserved							

CXO6175A

Figure D–5 Template 05 - Failover Event Sense Data Response Format

off \ bit	7	6	5	4	3	2	1	0
0	Unusd	Error Code						
1	Unused							
2	Unused				Sense Key			
3–6	Unused							
7	Additional Sense Length							
8–11	Unused							
12	Additional Sense Code (ASC)							
13	Additional Sense Code Qualifier (ASCQ)							
14	Unused							
15–17	Unused							
18–31	Reserved							
32–35	Instance Code							
36	Template							
37	Template Flags							
38–53	Reserved							
54–69	Controller Board Serial Number							
70–73	Controller Software Revision Level							
74–75	Reserved							
76	LUN Status							
77-103	Reserved							
104–107	Last Failure Code							
108–111	Last Failure Parameter[0]							
112–115	Last Failure Parameter[1]							
116–119	Last Failure Parameter[2]							
120–123	Last Failure Parameter[3]							
124–127	Last Failure Parameter[4]							
128–131	Last Failure Parameter[5]							
132–135	Last Failure Parameter[6]							
136–139	Last Failure Parameter[7]							
140–159	Reserved							

CXO6177A

Figure D–6 Template 11 - Nonvolatile Parameter Memory Component Event Sense Data Response

off \ bit	7	6	5	4	3	2	1	0
0	Unusd	Error Code						
1	Unused							
2	Unused				Sense Key			
3-6	Unused							
7	Additional Sense Length							
8-11	Unused							
12	Additional Sense Code (ASC)							
13	Additional Sense Code Qualifier (ASCQ)							
14	Unused							
15–17	Unused							
18–31	Reserved							
32-35	Instance Code							
36	Template							
37	Template Flags							
38–53	Reserved							
54–69	Controller Board Serial Number							
70–73	Controller Software Revision Level							
74–75	Reserved							
76	LUN Status							
77-103	Reserved							
104–107	Memory Address							
108–111	Byte Count							
112–114	Number of Times Written							
115	Undefined							
116–159	Reserved							

CXO6178A

Figure D–7 Template 12 - Backup Battery Failure Event Sense Data Response Format

off \ bit	7	6	5	4	3	2	1	0
0	Unusd	Error Code						
1	Unused							
2	Unused				Sense Key			
3-6	Unused							
7	Additional Sense Length							
8-11	Unused							
12	Additional Sense Code (ASC)							
13	Additional Sense Code Qualifier (ASCQ)							
14	Unused							
15–17	Unused							
18–31	Reserved							
32-35	Instance Code							
36	Template							
37	Template Flags							
38–53	Reserved							
54–69	Controller Board Serial Number							
70–73	Controller Software Revision Level							
74-75	Reserved							
76	LUN Status							
77–103	Reserved							
104–107	Memory Address							
108–159	Reserved							

CXO6179A

Figure D–8 Template 13 - Subsystem Built-In Self Test Failure Event Sense Data Response Format

off \ bit	7	6	5	4	3	2	1	0
0	Unusd	Error Code						
1	Unused							
2	Unused				Sense Key			
3-6	Unused							
7	Additional Sense Length							
8-11	Unused							
12	Additional Sense Code (ASC)							
13	Additional Sense Code Qualifier (ASCQ)							
14	Unused							
15–17	Unused							
18–31	Reserved							
32-35	Instance Code							
36	Template							
37	Template Flags							
38–53	Reserved							
54–69	Controller Board Serial Number							
70–73	Controller Software Revision Level							
74–75	Reserved							
76	LUN Status							
77-103	Reserved							
104–105	Undefined							
106	Header Type							
107	Header Flags							
108	TE							
109	Test Number							
110	Test Command							
111	Test Flags							
112–113	Error Code							
114–115	Return Code							
116–119	Address of Error							
120–123	Expected Error Data							
124–127	Actual Error Data							
128–131	Extra Status 1							
132–135	Extra Status 2							
136–139	Extra Status 3							
140-159	Reserved							

CXO6180A

Figure D–9 Template 14 - Memory System Failure Event Sense Data Response Format

off \ bit	7	6	5	4	3	2	1	0
0	Unusd	Error Code						
1	Unused							
2	Unused				Sense Key			
3-6	Unused							
7	Additional Sense Length							
8-11	Unused							
12	Additional Sense Code (ASC)							
13	Additional Sense Code Qualifier (ASCQ)							
14	Unused							
15-17	Unused							
18-19	Reserved							
20-23	Reserved or RDR2 (TM1)							
24-27	Reserved or RDEAR (TM1)							
28-31	Reserved							
32–35	Instance Code							
36	Template							
37	Template Flags							
38-39	Reserved							
40-43	Reserved or FXPSCR (TM1)							
44-47	Reserved or FXCSR (TM1)							
48-51	Reserved or FXCCSR (TM1)							
52-53	Reserved							
54-69	Controller Board Serial Number							
70-73	Controller Software Revision Level							

off \ bit	7	6	5	4	3	2	1	0
74-75	Reserved							
76	LUN Status							
77-79	Reserved							
80-83	Reserved or FXPAEC(TM1)							
84-87	Reserved or FXCAEC(TM1)							
88-91	Reserved or FXPAEP(TM1)							
92-95	Reserved or CHC (TM0) or FXCAEP(TM1)							
96-99	Reserved or CMC (TM0) or CFW(TM1)							
100-103	Reserved or DSR2 (TM0) or RRR(TM1)							
104-107	Memory Address							
108-111	Byte Count							
112-115	DSR or PSR(TM1)							
116-119	CSR or CSR(TM1)							
120-123	DCSR or EAR(TM1)							
124-127	DER or EDR1(TM1)							
128-131	EAR or EDR0(TM1)							
132-135	EDR or ICR TM1)							
136-139	ERR or IMR(TM1)							
140-143	RSR or DIO(TM1)							
144-147	RDR0							
148-151	RDR1							
152-155	WDR0							
156-159	WDR1							

CXO6181A

Figure D–10 Template 41 - Device Services Non-Transfer Error Event Sense Data Response Format

off \ bit	7	6	5	4	3	2	1	0
0	Unusd	Error Code						
1	Unused							
2	Unused				Sense Key			
3-6	Unused							
7	Additional Sense Length							
8-11	Unused							
12	Additional Sense Code (ASC)							
13	Additional Sense Code Qualifier (ASCQ)							
14	Unused							
15–17	Unused							
18–31	Reserved							
32-35	Instance Code							
36	Template							
37	Template Flags							
38–53	Reserved							
54–69	Controller Board Serial Number							
70-73	Controller Software Revision Level							
74-75	Reserved							
76	LUN Status							
77–103	Reserved							
104	Associated Port							
105	Associated Target							
106	Associated Additional Sense Code							
107	Associated Additional Sense Code Qualifier							
108–159	Reserved							

CXO6182A

Figure D–11 Template 51 - Disk Transfer Error Event Sense Data Response Format

off \ bit	7	6	5	4	3	2	1	0
0–17	Standard Sense Data							
18–19	Reserved							
20	Total Number of Errors							
21	Total Retry Count							
22–25	ASC/ASCQ Stack							
26–28	Device Locator							
29–31	Reserved							
32–35	Instance Code							
36	Template							
37	Template Flags							
38	Reserved							
39	Command Opcode							
40	Sense Data Qualifier							
41–50	Original CDB							
51	Host ID							
52–53	Reserved							
54–69	Controller Board Serial Number							
70–73	Controller Software Revision Level							
74–75	Reserved							
76	LUN Status							
77–78	Reserved							
79-82	Device Firmware Revision Level							
83–98	Device Product ID							
99–100	Reserved							
101	Device Type							
102–103	Reserved							
104–121	Device Sense Data							
122–159	Reserved							

CXO6183A

APPENDIX E

Controller Specifications

This appendix contains physical, electrical, and environmental specifications for the HSG80 array controller.

Physical and Electrical Specifications for the Controller

Table E–1 lists the physical and electrical specifications for the controller and cache modules.

Table E–1 Controller Specifications

Hardware	Length	Width	Power	Current at +5 V	Current at +12 V
HSG80 Array Controller module	12.5 inches	8.75 inches	23.27 W	6.04A	
Write-back Cache, 128 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 E–1 are nominal measurements (at +5 and +12 B). This table does not included tolerances.

Environmental Specifications

The HSG80 array controller is intended for installation in a Class A computer room environment.

The environmental specifications listed in Table E–2 are the same as for other DIGITAL storage devices.

Table E–2 StorageWorks Environmental Specifications

Condition	Optimum Operating Environment
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)
Condition	**Maximum Operating Environment (Range)**
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)
Condition	**Maximum Nonoperating Environment (Range)**
Temperature	-40° to +66°C (-40° to +151°F) (During transportation and associated short-term storage)
Relative Humidity	8% to 95% in original shipping container (noncondensing); otherwise, 50% (noncondensing)
Altitude	From -300 m (-1000 ft) to +3600 m (+12,000 ft) Mean Sea Level (MSL)

Glossary

This glossary defines terms pertaining to the HSG80 Fibre Channel array controller. It is not a comprehensive glossary of computer terms.

8B/10B A type of byte encoding and decoding to reduce errors in data transmission patented by the IBM Corporation. This process of encoding and decoding data for transmission has been adopted by ANSI.

adapter A device that converts the protocol and hardware interface of one bus type into another without changing the function of the bus.

ACS *See* array controller software.

AL_PA *See* arbitrated loop physical address.

alias address An AL_PA value recognized by an Arbitrated Loop port in addition to its assigned AL_PA.

ANSI Pronounced “ann-see.” Acronym for the American National Standards Institute. An organization who develops standards used voluntarily by many manufacturers within the USA. ANSI is not a government agency.

arbitrate A process of selecting one L_Port from a collection of several ports that request use of the arbitrated loop concurrently.

arbitrated loop A loop type of topology where two or more ports can be interconnected, but only two ports at a time can communicate.

arbitrated loop physical address Abbreviated AL_PA. A one-byte value used to identify a port in an Arbitrated Loop topology. The AL_PA value corresponds to bits 7:0 of the 24-bit Native Address Indentifier.

array controller *See* controller.

array controller software Abbreviated ACS. Software contained on a removable ROM program card that provides the operating system for the array controller.

asynchronous Pertaining to events that are scheduled as the result of a signal asking for the event; pertaining to that which is without any specified time relation. *See also* synchronous.

autospare A controller feature that automatically replaces a failed disk drive. To aid the controller in automatically replacing failed disk drives, you can enable the AUTOSPARE switch for the failedset causing physically replaced disk drives to be automatically placed into the spareset. Also called “autonewspare.”

bad block A data block that contains a physical defect.

bad block replacement Abbreviated BBR. A replacement routine that substitutes defect-free disk blocks for those found to have defects. This process takes place in the controller, transparent to the host.

backplane The electronic printed circuit board into which you plug subsystem devices—for example, the SBB or power supply.

BBR *See* bad block replacement.

BIST *See* built-in self-test.

bit A single binary digit having a value of either 0 or 1. A bit is the smallest unit of data a computer can process.

block Also called a sector. The smallest collection of consecutive bytes addressable on a disk drive. In integrated storage elements, a block contains 512 bytes of data, error codes, flags, and the block’s address header.

bootstrapping A method used to bring a system or device into a defined state by means of its own action. For example, a machine routine whose first few instructions are enough to bring the rest of the routine into the computer from an input device.

built-in self-test A diagnostic test performed by the array controller software on the controller’s policy processor.

byte	A binary character string made up of 8 bits operated on as a unit.
cache memory	A portion of memory used to accelerate read and write operations.
CCITT	Acronym for Consultive Committee International Telephone and Telegraph. An international association that sets worldwide communication standards, recently renamed International Telecommunications Union (ITU).
CDU	Cable distribution unit. The power entry device for StorageWorks cabinets. The CDU provides the connections necessary to distribute power to the cabinet shelves and fans.
channel	An interface which allows high speed transfer of large amounts of data. Another term for a SCSI bus. *See also* SCSI.
chunk	A block of data written by the host.
chunk size	The number of data blocks, assigned by a system administrator, written to the primary RAIDset or stripeset member before the remaining data blocks are written to the next RAIDset or stripeset member.
CLCP	An abbreviation for code-load code-patch utility.
CLI	*See* command line interpreter.
coax	*See* coaxial cable.
coaxial cable	A two-conductor wire in which one conductor completely wraps the other with the two separated by insulation.
cold swap	A method of device replacement that requires the entire subsystem to be turned off before the device can be replaced. *See also* hot swap and warm swap.
command line interpreter	The configuration interface to operate the controller software.
configuration file	A file that contains a representation of a storage subsystem's configuration.

container 1) Any entity that is capable of storing data, whether it is a physical device or a group of physical devices. (2) A virtual, internal controller structure representing either a single disk or a group of disk drives linked as a storageset. Stripesets and mirrorsets are examples of storageset containers the controller uses to create units.

controller A hardware device that, with proprietary software, facilitates communications between a host and one or more devices organized in an array. HS family controllers are examples of array controllers.

copying A state in which data to be copied to the mirrorset is inconsistent with other members of the mirrorset. *See also* normalizing.

copying member Any member that joins the mirrorset after the mirrorset is created is regarded as a copying member. Once all the data from the normal member (or members) is copied to a normalizing or copying member, the copying member then becomes a normal member. *See also* normalizing member.

CSR An acronym for control and status register.

DAEMON Pronounced “demon.” A program usually associated with a UNIX systems that performs a utility (housekeeping or maintenance) function without being requested or even known of by the user. A daemon is a diagnostic and execution monitor.

data center cabinet A generic reference to large DIGITAL subsystem cabinets, such as the SW600-series and 800-series cabinets in which StorageWorks components can be mounted.

data striping The process of segmenting logically sequential data, such as a single file, so that segments can be written to multiple physical devices (usually disk drives) in a round-robin fashion. This technique is useful if the processor is capable of reading or writing data faster than a single disk can supply or accept the data. While data is being transferred from the first disk, the second disk can locate the next segment.

device *See* node and peripheral device.

differential I/O module A 16-bit I/O module with SCSI bus converter circuitry for extending a differential SCSI bus. *See also* I/O module.

differential SCSI bus A bus in which a signal's level is determined by the potential difference between two wires. A differential bus is more robust and less subject to electrical noise than is a single-ended bus.

DIMM Dual inline Memory Module.

dirty data The write-back cached data that has not been written to storage media, even though the host operation processing the data has completed.

DMA Direct Memory Access.

DOC DWZZA-On-a-Chip. An NCR53C120 SCSI bus extender chip used to connect a SCSI bus in an expansion cabinet to the corresponding SCSI bus in another cabinet.

driver A hardware device or a program that controls or regulates another device. For example, a device driver is a driver developed for a specific device that allows a computer to operate with the device, such as a printer or a disk drive.

dual-redundant configuration A controller configuration consisting of two active controllers operating as a single controller. If one controller fails, the other controller assumes control of the failing controller's devices.

dual-simplex A communications protocol that allows simultaneous transmission in both directions in a link, usually with no flow control.

DUART Dual universal asynchronous receiver and transmitter. An integrated circuit containing two serial, asynchronous transceiver circuits.

ECB External cache battery. The unit that supplies backup power to the cache module in the event the primary power source fails or is interrupted.

ECC Error checking and correction.

EDC Error detection code.

EIA The abbreviation for Electronic Industries Association. EIA is a standards organization specializing in the electrical and functional characteristics of interface equipment. Same as Electronic Industries Association.

EMU Environmental monitoring unit. A unit that provides increased protection against catastrophic failures. Some subsystem enclosures include an EMU which works with the controller to detect conditions such as failed power supplies, failed blowers, elevated temperatures, and external air sense faults. The EMU also controls certain cabinet hardware including DOC chips, alarms, and fan speeds.

ESD Electrostatic discharge. The discharge of potentially harmful static electrical voltage as a result of improper grounding.

extended subsystem A subsystem in which two cabinets are connected to the primary cabinet.

external cache battery *See* ECB.

F_Port A port in a fabric where an N_Port or NL_Port may attach.

fabric A group of interconnections between ports that includes a fabric element.

failedset A group of failed mirrorset or RAIDset devices automatically created by the controller.

failover The process that takes place when one controller in a dual-redundant configuration assumes the workload of a failed companion controller. Failover continues until the failed controller is repaired or replaced.

FC–AL The Fibre Channel Arbitrated Loop standard.

FC–ATM ATM AAL5 over Fibre Channel

FC–FG Fibre Channel Fabric Generic Requirements

FG–FP Fibre Channel Framing Protocol (HIPPI on FC)

FC-GS-1 Fibre Channel Generic Services-1

FC–GS-2 Fibre Channel Generic Services-2

FC–IG Fibre Channel Implementation Guide

FC–LE Fibre Channel Link Encapsulation (ISO 8802.2)

FC–PH The Fibre Channel Physical and Signaling standard.

FC–SB Fibre Channel Single Byte Command Code Set

FC–SW Fibre Channel Switched Topology and Switch Controls

FCC Federal Communications Commission. The federal agency responsible for establishing standards and approving electronic devices within the United States.

FCC Class A This certification label appears on electronic devices that can only be used in a commercial environment within the United States.

FCC Class B This certification label appears on electronic devices that can be used in either a home or a commercial environment within the United States.

FCP The mapping of SCSI-3 operations to Fibre Channel.

FDDI Fiber Distributed Data Interface. An ANSI standard for 100 megabaud transmission over fiber optic cable.

FD SCSI The fast, narrow, differential SCSI bus with an 8-bit data transfer rate of 10 MB/s. *See also* FWD SCSI and SCSI.

fiber A fiber or optical strand. Spelled *fibre* in Fibre Channel.

fiber optic cable A transmission medium designed to transmit digital signals in the form of pulses of light. Fiber optic cable is noted for its properties of electrical isolation and resistance to electrostatic contamination.

FL_Port A port in a fabric where N_Port or an NL_Port may be connected.

flush The act of writing dirty data from cache to a storage media.

FMU Fault management utility.

forced errors A data bit indicating a corresponding logical data block contains unrecoverable data.

frame An invisible unit used to transfer information in Fibre Channel.

FRU	Field replaceable unit. A hardware component that can be replaced at the customer’s location by DIGITAL service personnel or qualified customer service personnel.
full duplex (n)	A communications system in which there is a capability for 2-way transmission and acceptance between two sites at the same time.
full duplex (adj)	Pertaining to a communications method in which data can be transmitted and received at the same time.
FWD SCSI	A fast, wide, differential SCSI bus with a maximum 16-bit data transfer rate of 20 MB/s. *See also* SCSI and FD SCSI.
GLM	Gigabit link module
giga	A prefix indicating a billion (10^9) units, as in gigabaud or gigabyte.
gigabaud	An encoded bit transmission rate of one billion (10^9) bits per second.
gigabyte	A value normally associated with a disk drives storage capacity, meaning a billion (10^9) bytes. The decimal value 1024 is usually used for one thousand.
half-duplex (adj)	Pertaining to a communications system in which data can be either transmitted or received but only in one direction at one time.
hard address	The AL_PA which an NL_Port attempts to acquire during loop initialization.
HIPPI–FC	Fibre Channel over HIPPI
host	The primary or controlling computer to which a storage subsystem is attached.
host adapter	A device that connects a host system to a SCSI bus. The host adapter usually performs the lowest layers of the SCSI protocol. This function may be logically and physically integrated into the host system.
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. The supported modes are A, Normal (including DIGITAL

UNIX$^{®}$, OpenVMS, Sun$^{®}$, and Hewlett-Packard$^{®}$ HP–UX); B, IBM AIX$^{®}$; C, Proprietary; and D, Microsoft Windows NT^{TM} Server.

hot disks A disk containing multiple hot spots. Hot disks occur when the workload is poorly distributed across storage devices which prevents optimum subsystem performance. *See also* hot spots.

hot spots A portion of a disk drive frequently accessed by the host. Because the data being accessed is concentrated in one area, rather than spread across an array of disks providing parallel access, I/O performance is significantly reduced. *See als*o hot disks.

hot swap A method of device replacement that allows normal I/O activity on a device's bus to remain active during device removal and insertion. The device being removed or inserted is the only device that cannot perform operations during this process. *See also* cold swap and warm swap.

IBR Initial Boot Record.

ILF Illegal function.

INIT Initialize input and output.

initiator A SCSI device that requests an I/O process to be performed by another SCSI device, namely, the SCSI target. The controller is the initiator on the device bus. The host is the initiator on the host bus.

instance code A four-byte value displayed in most text error messages and issued by the controller when a subsystem error occurs. The instance code indicates when during software processing the error was detected.

interface A set of protocols used between components, such as cables, connectors, and signal levels.

I/O Refers to input and output functions.

I/O driver The set of code in the kernel that handles the physical I/O to a device. This is implemented as a fork process. Same as driver.

I/O interface *See* interface.

I/O module A 16-bit SBB shelf device that integrates the SBB shelf with either an 8-bit single ended, 16-bit single-ended, or 16-bit differential SCSI bus.

I/O operation The process of requesting a transfer of data from a peripheral device to memory (or visa versa), the actual transfer of the data, and the processing and overlaying activity to make both of those happen.

IPI Intelligent Peripheral Interface. An ANSI standard for controlling peripheral devices by a host computer.

IPI-3 Disk Intelligent Peripheral Interface Level 3 for Disk

IPI-3 Tape Intelligent Peripheral Interface Level 3 for Tape

JBOD Just a bunch of disks. A term used to describe a group of single-device logical units.

kernel The most privileged processor access mode.

LBN Logical Block Number.

L_port A node or fabric port capable of performing arbitrated loop functions and protocols. NL_Ports and FL_Ports are loop-capable ports.

LED Light Emitting Diode.

link A connection between two Fibre Channel ports consisting of a transmit fibre and a receive fibre.

logical block number *See* LBN.

local connection A connection to the subsystem using either its serial maintenance port or the host's SCSI bus. A local connection enables you to connect to one subsystem controller within the physical range of the serial or host SCSI cable.

local terminal A terminal plugged into the EIA-423 maintenance port located on the front bezel of the controller. *See also* maintenance terminal.

logical bus A single-ended bus connected to a differential bus by a SCSI bus signal converter.

logical unit A physical or virtual device addressable through a target ID number. LUNs use their target’s bus connection to communicate on the SCSI bus.

logical unit number A value that identifies a specific logical unit belonging to a SCSI target ID number. A number associated with a physical device unit during a task’s I/O operations. Each task in the system must establish its own correspondence between logical unit numbers and physical devices.

logon Also called login. A procedure whereby a participant, either a person or network connection, is identified as being an authorized network participant.

loop *See* arbitrated loop.

loop_ID A seven-bit value numbered contiguously from zero to 126-decimal and represent the 127 legal AL_PA values on a loop (not all of the 256 hex values are allowed as AL_PA values per FC-AL.

loop tenancy The period of time between the following events: when a port wins loop arbitration and when the port returns to a monitoring state.

L_Port A node or fabric port capable of performing Arbitrated Loop functions and protocols. NL_Ports and FL_Ports are loop-capable ports.

LRU Least recently used. A cache term used to describe the block replacement policy for read cache.

Mbps Approximately one million (10^6) bits per second—that is, megabits per second.

MBps Approximately one million (10^6) bytes per second—that is, megabytes per second.

maintenance terminal An EIA-423-compatible terminal used with the controller. This terminal is used to identify the controller, enable host paths, enter configuration information, and check the controller’s status. The maintenance terminal is not required for normal operations.
See also local terminal.

member A container that is a storage element in a RAID array.

metadata The data written to a disk for the purposes of controller administration. Metadata improves error detection and media defect management for the disk drive. It is also used to support storageset configuration and partitioning. Nontransportable disks also contain metadata to indicate they are uniquely configured for StorageWorks environments. Metadata can be thought of as “data about data.”

mirroring The act of creating an exact copy or image of data.

mirrorset *See* RAID level 1.

MIST Module Integrity Self-Test.

N_port A port attached to a node for use with point-to-point topology or fabric topology.

NL_port A port attached to a node for use in all three topologies.

network A data communication, a configuration in which two or more terminals or devices are connected to enable information transfer.

node In data communications, the point at which one or more functional units connect transmission lines.

Non-L_Port A Node of Fabric port that is not capable of performing the Arbitrated Loop functions and protocols. N_Ports and F_Ports loop-capable ports.

non-participating mode A mode within an L_Port that inhibits the port from participating in loop activities. L_Ports in this mode continue to retransmit received transmission words but are not permitted to arbitrate or originate frames. An L_Port in non-participating mode may or may not have an AL_PA. *See also* participating mode.

nominal membership The desired number of mirrorset members when the mirrorset is fully populated with active devices. If a member is removed from a mirrorset, the actual number of members may fall below the “nominal” membership.

node In data communications, the point at which one or more functional units connect transmission lines. In fibre channel, a device that has at least one N_Port or NL_Port.

nonredundant controller configuration (1) A single controller configuration. (2) A controller configuration that does not include a second controller.

normal member A mirrorset member that, block-for-block, contains the same data as other normal members within the mirrorset. Read requests from the host are always satisfied by normal members.

normalizing Normalizing is a state in which, block-for-block, data written by the host to a mirrorset member is consistent with the data on other normal and normalizing members. The normalizing state exists only after a mirrorset is initialized. Therefore, no customer data is on the mirrorset.

normalizing member A mirrorset member whose contents is the same as all other normal and normalizing members for data that has been written since the mirrorset was created or lost cache data was cleared. A normalizing member is created by a normal member when either all of the normal members fail or all of the normal members are removed from the mirrorset. *See also* copying member.

NVM Non-Volatile Memory. A type of memory where the contents survive power loss. Also sometimes referred to as NVMEM.

OCP Operator control panel. The control or indicator panel associated with a device. The OCP is usually mounted on the device and is accessible to the operator.

other controller The controller in a dual-redundant pair that is connected to the controller serving your current CLI session. *See also* this controller.

outbound fiber One fiber in a link that carries information away from a port.

parallel data transmission A data communication technique in which more than one code element (for example, bit) of each byte is sent or received simultaneously.

parity A method of checking if binary numbers or characters are correct by counting the ONE bits. In odd parity, the total number of ONE bits must be odd; in even parity, the total number of ONE bits must be even.

parity bit A binary digit added to a group of bits that checks to see if errors exist in the transmission.

parity check A method of detecting errors when data is sent over a communications line. With even parity, the number of ones in a set of binary data should be even. With odd parity, the number of ones should be odd.

participating mode A mode within an L_Port that allows the port to participate in loop activities. A port must have a valid AL_PA to be in participating mode.

PCM Polycenter Console Manager.

PCMCIA Personal Computer Memory Card Industry Association. An international association formed to promote a common standard for PC card-based peripherals to be plugged into notebook computers. The card commonly known as a PCMCIA card is about the size of a credit card.

parity A method of checking if binary numbers or characters are correct by counting the ONE bits. In odd parity, the total number of ONE bits must be odd; in even parity, the total number of ONE bits must be even. Parity information can be used to correct corrupted data. RAIDsets use parity to improve the availability of data.

parity bit A binary digit added to a group of bits that checks to see if there are errors in the transmission.

parity RAID *See* RAIDset.

partition A logical division of a container, represented to the host as a logical unit.

peripheral device Any unit, distinct from the CPU and physical memory, that can provide the system with input or accept any output from it. Terminals, printers, tape drives, and disks are peripheral devices.

point-to-point connection A network configuration in which a connection is established between two, and only two, terminal installations. The connection may include switching facilities.

port (1) In general terms, a logical channel in a communications system. (2) The hardware and software used to connect a host controller to a communications bus, such as a SCSI bus or serial bus.

Regarding the controller, the port is (1) the logical route for data in and out of a controller that can contain one or more channels, all of which contain the same type of data. (2) The hardware and software that connects a controller to a SCSI device.

port_name A 64-bit unique identifier assigned to each Fibre Channel port. The Port_Name is communicated during the logon and port discovery process.

preferred address The AL_PA which an NL_Port attempts to acquire first during initialization.

primary cabinet The primary cabinet is the subsystem enclosure that contains the controllers, cache modules, external cache batteries, and the PVA module.

private NL_Port An NL_Port which does not attempt login with the fabric and only communicates with NL_Ports on the same loop.

public NL_Port An NL_Port that attempts login with the fabric and can observe the rules of either public or private loop behavior. A public NL_Port may communicate with both private and public NL_Ports.

program card The PCMCIA card containing the controller's operating software.

protocol The conventions or rules for the format and timing of messages sent and received.

PTL Port-Target-LUN. The controller's method of locating a device on the controller's device bus.

PVA module Power Verification and Addressing module.

quiesce The act of rendering bus activity inactive or dormant. For example, "quiesce the SCSI bus operations during a device warm-swap."

RAID Redundant Array of Independent Disks. Represents multiple levels of storage access developed to improve performance or availability or both.

RAID level 0 A RAID storageset that stripes data across an array of disk drives. A single logical disk spans multiple physical disks, allowing parallel data processing for increased I/O performance. While the performance characteristics of RAID level 0 is excellent, this RAID level is the only one that does not provide redundancy. Raid level 0 storagesets are sometimes referred to as stripesets.

RAID level 0+1 A RAID storageset that stripes data across an array of disks (RAID level 0) and mirrors the striped data (RAID level 1) to provide high I/O performance and high availability. This RAID level is alternatively called a striped mirrorset. Raid level 0+1 storagesets are sometimes referred to as striped mirrorsets.

RAID level 1 A RAID storageset of two or more physical disks that maintains a complete and independent copy of the entire virtual disk’s data. This type of storageset has the advantage of being highly reliable and extremely tolerant of device failure. Raid level 1 storagesets are sometimes referred to as mirrorsets.

RAID level 3 A RAID storageset that transfers data parallel across the array’s disk drives a byte at a time, causing individual blocks of data to be spread over several disks serving as one enormous virtual disk. A separate redundant check disk for the entire array stores parity on a dedicated disk drive within the storageset. *See also* RAID level 5.

RAID Level 5 A RAID storageset that, unlike RAID level 3, stores the parity information across all of the disk drives within the storageset. *See also* RAID level 3.

RAID level 3/5 A DIGITAL-developed RAID storageset that stripes data and parity across three or more members in a disk array. A RAIDset combines the best characteristics of RAID level 3 and RAID level 5. A RAIDset is the best choice for most applications with small to medium I/O requests, unless the application is write intensive. A RAIDset is sometimes called parity RAID. Raid level 3/5 storagesets are sometimes referred to as RAIDsets.

RAIDset *See* RAID level 3/5.

RAM Random access memory.

read ahead caching A caching technique for improving performance of synchronous sequential reads by prefetching data from disk.

read caching A cache management method used to decrease the subsystem's response time to a read request by allowing the controller to satisfy the request from the cache memory rather than from the disk drives.

reconstruction The process of regenerating the contents of a failed member's data. The reconstruct process writes the data to a spareset disk and then incorporates the spareset disk into the mirrorset, striped mirrorset, or RAIDset from which the failed member came. *See also* regeneration.

reduced Indicates that a mirrorset or RAIDset is missing one member because the member has failed or has been physically removed.

redundancy The provision of multiple interchangeable components to perform a single function in order to cope with failures and errors. A RAIDset is considered to be redundant when user data is recorded directly to one member and all of the other members include associated parity information.

regeneration (1) The process of calculating missing data from redundant data. (2) The process of recreating a portion of the data from a failing or failed drive using the data and parity information from the other members within the storageset. The regeneration of an entire RAIDset member is called reconstruction. *See also* reconstruction.

request rate The rate at which requests are arriving at a servicing entity.

RFI Radio frequency interference. The disturbance of a signal by an unwanted radio signal or frequency.

replacement policy The policy specified by a switch with the SET FAILEDSET command indicating whether a failed disk from a mirrorset or RAIDset is to be automatically replaced with a disk from the spareset. The two switch choices are AUTOSPARE and NOAUTOSPARE.

SBB StorageWorks building block. (1) A modular carrier plus the interface required to mount the carrier into a standard StorageWorks shelf. (2) any device conforming to shelf mechanical and electrical standards installed in a 3.5-inch or 5.25-inch carrier, whether it is a storage device or power supply.

SCSI
Small computer system interface. (1) An ANSI interface standard defining the physical and electrical parameters of a parallel I/O bus used to connect initiators to devices. (2) a processor-independent standard protocol for system-level interfacing between a computer and intelligent devices including hard drives, floppy disks, CD-ROMs, printers, scanners, and others.

SCSI-A cable
A 50-conductor (25 twisted-pair) cable generally used for single-ended, SCSI-bus connections.

SCSI bus signal converter
Sometimes referred to as an adapter. (1) A device used to interface between the subsystem and a peripheral device unable to be mounted directly into the SBB shelf of the subsystem. (2) a device used to connect a differential SCSI bus to a single-ended SCSI bus. (3) A device used to extend the length of a differential or single-ended SCSI bus. *See also* I/O module.

SCSI device
(1) A host computer adapter, a peripheral controller, or an intelligent peripheral that can be attached to the SCSI bus. (2) Any physical unit that can communicate on a SCSI bus.

SCSI device ID number
A bit-significant representation of the SCSI address referring to one of the signal lines, numbered 0 through 7 for an 8-bit bus, or 0 through 15 for a 16-bit bus. *See also* target ID number.

SCSI ID number
The representation of the SCSI address that refers to one of the signal lines numbered 0 through 15.

SCSI-P cable
A 68-conductor (34 twisted-pair) cable generally used for differential bus connections.

SCSI port
(1) Software: The channel controlling communications to and from a specific SCSI bus in the system. (2) Hardware: The name of the logical socket at the back of the system unit to which a SCSI device is connected.

serial transmission
A method transmission in which each bit of information is sent sequentially on a single channel rather than simultaneously as in parallel transmission.

service rate The rate at which an entity is able to service requests For example, the rate at which an Arbitrated Loop is able to service arbitrated requests.

signal converter *See* SCSI bus signal converter.

SIMM Single Inline Memory Module.

single ended I/O module A 16-bit I/O module. *See also* I/O module.

single-ended SCSI bus An electrical connection where one wire carries the signal and another wire or shield is connected to electrical ground. Each signal's logic level is determined by the voltage of a single wire in relation to ground. This is in contrast to a differential connection where the second wire carries an inverted signal.

spareset A collection of disk drives made ready by the controller to replace failed members of a storageset.

storage array An integrated set of storage devices.

storage array subsystem *See* storage subsystem.

storageset (1) A group of devices configured with RAID techniques to operate as a single container. (2) Any collection of containers, such as stripesets, mirrorsets, striped mirrorsets, and RAIDsets.

storage subsystem The controllers, storage devices, shelves, cables, and power supplies used to form a mass storage subsystem.

storage unit The general term that refers to storagesets, single-disk units, and all other storage devices that are installed in your subsystem and accessed by the host. A storage unit can be any entity that is capable of storing data, whether it is a physical device or a group of physical devices.

StorageWorks A family of DIGITAL modular data storage products that allow customers to design and configure their own storage subsystems. Components include power, packaging, cabling, devices, controllers, and software. Customers can integrate devices and array controllers in StorageWorks enclosures to form storage subsystems.

StorageWorks systems include integrated SBBs and array controllers to form storage subsystems. System-level enclosures to house the shelves and standard mounting devices for SBBs are also included.

stripe The data divided into blocks and written across two or more member disks in an array.

striped mirrorset *See* RAID level 0+1.

stripeset *See* RAID level 0.

stripe size The stripe capacity as determined by *n*–1 times the chunksize, where *n* is the number of RAIDset members.

striping The technique used to divide data into segments, also called chunks. The segments are striped, or distributed, across members of the stripeset. This technique helps to distribute hot spots across the array of physical devices to prevent hot spots and hot disks.

Each stripeset member receives an equal share of the I/O request load, improving performance.

surviving controller The controller in a dual-redundant configuration pair that serves its companion's devices when the companion controller fails.

switch A method that controls the flow of functions and operations in software.

synchronous Pertaining to a method of data transmission which allows each event to operate in relation to a timing signal. *See also* asynchronous.

tape A storage device supporting sequential access to variable sized data records.

target (1) A SCSI device that performs an operation requested by an initiator. (2) Designates the target identification (ID) number of the device.

this controller The controller that is serving your current CLI session through a local or remote terminal. *See also* other controller.

topology An interconnection scheme that allows multiple Fibre Channel ports to communicate with each other. For example, point-to-point, Arbitrated Loop, and switched fabric are all Fibre Channel topologies.

transfer data rate The speed at which data may be exchanged with the central processor, expressed in thousands of bytes per second.

ULP Upper Layer Protocol.

ULP process A function executing within a Fibre Channel node which conforms to the Upper Layer Protocol (ULP) requirements when interacting with other ULP processes.

Ultra-SCSI 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 eight units on each target. *See also* target and target ID number.

unwritten cached data Sometimes called unflushed data. *See* dirty data.

UPS Uninterruptible power supply. A battery-powered power supply guaranteed to provide power to an electrical device in the event of an unexpected interruption to the primary power supply. Uninterruptible power supplies are usually rated by the amount of voltage supplied and the length of time the voltage is supplied.

VHDCI Very high-density-cable interface. A 68-pin interface. Required for Ultra-SCSI connections.

virtual terminal A software path from an operator terminal on the host to the controller’s CLI interface, sometimes called a host console. The path can be established via the host port on the controller (using HSZterm) or via the maintenance port through an intermediary host.

VTDPY An abbreviation for Virtual Terminal Display Utility.

warm swap A device replacement method that allows the complete system remains online during device removal or insertion. The system bus may be halted, or quiesced, for a brief period of time during the warm-swap procedure.

Worldwide name A unique 64-bit number assigned to a subsystem by the Institute of Electrical and Electronics Engineers (IEEE) and set by DIGITAL manufacturing prior to shipping. This name is referred to as the node ID within the CLI.

write-back caching A cache management method used to decrease the subsystem’s response time to write requests by allowing the controller to declare the write operation “complete” as soon as the data reaches its cache memory. The controller performs the slower operation of writing the data to the disk drives at a later time.

write-through caching A cache management method used to decrease the subsystem’s response time to a read. This method allows the controller to satisfy the request from the cache memory rather than from the disk drives.

write hole The period of time in a RAID level 1 or RAID level 5 write operation when an opportunity emerges for undetectable RAIDset data corruption. Write holes occur under conditions such as power outages, where the writing of multiple members can be abruptly interrupted. A battery backed-up cache design eliminates the write hole because data is preserved in cache and unsuccessful write operations can be retried.

write-through cache A cache management technique for retaining host write requests in read cache. When the host requests a write operation, the controller writes data directly to the storage device. This technique allows the controller to complete some read requests from the cache, greatly improving the response time to retrieve data. The operation is complete only after the data to be written is received by the target storage device.

This cache management method may update, invalidate, or delete data from the cache memory accordingly, to ensure that the cache contains the most current data.

Index

F

G

H

I

J

L

M

N

O

P

Q

R

S