DIGITAL StorageWorks
HSZ70 Array Controller
HSOF Version 7.0

EK–HSZ70–CG. A01

Digital Equipment Corporation
Maynard, Massachusetts

Configuration Manual

July 1997

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the 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

Planning Storagesets

Configuring Storagesets

Figures

Tables

Preface

This book describes the features of the HSZ70 array controller and configuration procedures for the controller and storagesets running HSOF Version 7.0.

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 you're carrying out the procedures in this book.

Electrostatic Discharge Precautions

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

- Always use an ESD wrist strap when servicing the controller or other components in the subsystem. Make sure that the strap contacts bare skin and fits snugly, and that its grounding lead is attached to a bus that is a verified earth ground.
- Before touching any circuit board or component, always touch a verifiable earth ground to discharge any static electricity that may be present in your clothing.
- Always keep circuit boards and components away from nonconducting material.
- Always keep clothing away from circuit boards and components.
- Always use antistatic bags and grounding mats for storing circuit boards or components during replacement procedures.
- Always keep the ESD cover over the program card when the card is in the controller. If you remove the card, put it in its original carrying case. Never touch the contacts or twist or bend the card while you're handling it.
- Do not touch the connector pins of a cable when it is attached to a component or host.

VHDCI Cable Precautions

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

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

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

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 *HSZ70 Array Controller HSOF Version 7.0 Configuration Manual* 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 avoid the hazard.

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 prior customers that the controller's behavior being discussed is different from prior software or hardware versions.

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

Required Tools

You'll need the following tools for servicing 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.
- An 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**
HSZ70 Array Controller HSOF Version 7.0 CLI Reference Manual	EK–CLI70–RM. A01
HSZ70 Array Controller HSOF Version 7.0 Configuration Manual	EK–HSZ70–CG. A01
HSZ70 Array Controller HSOF Version 7.0 Service Manual	EK–HSZ70–SV. A01
HSZ70 Family Array Controller Operating Software (HSOF) Version 7.0 Software Product Description	SPD *xx.xx*. 00
Getting Started–HSZ70 Solutions Software Version 7.0 for...*platform*	AA–R60*xx*–TE. A01
Polycenter Console Manager	See the *Getting Started* guide for the platform-specific order number
StorageWorks Array Controller HSZ70 Array Controller Operating Software HSOF Version 7.0 Release Notes	EK–HSZ70–RN. A01
StorageWorks Getting Started with Command Console, Version 2.1	AA–R0HJC–TE
DIGITAL StorageWorks Ultra SCSI RAID Cabinet Subsystem (SW600) Installation and User's Guide	EK–SW600–UG
DIGITAL StorageWorks Ultra SCSI RAID Enclosure (BA370-Series) User's Guide	EK–BA370–UG
The RAIDBOOK—A Source for RAID Technology	RAID Advisory Board

Revision History

This is a new document.

CHAPTER 1

Introduction

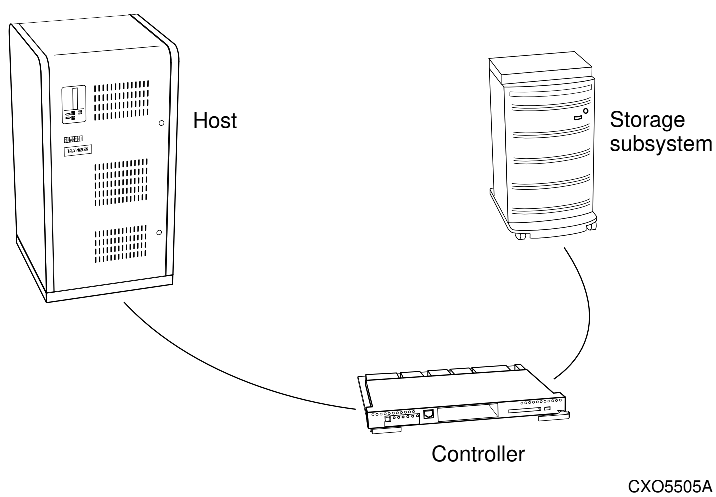
This chapter introduces the features and components of the HSZ70 controller.

Features of Your Controller

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

From the host's perspective, the controller is simply another SCSI device connected to one of its I/O buses. Consequently, the host sends its I/O requests to the controller just as it would to any SCSI device.

Figure 1–1 Bridging the Gap Between the Host and Its Storage Subsystem



From the subsystem's perspective, the controller receives the I/O requests from the host and directs them to the devices in the subsystem. Because the controller processes all of the I/O requests, in most cases it eliminates the host-based processing that is typically associated with reading and writing data to multiple storage devices.

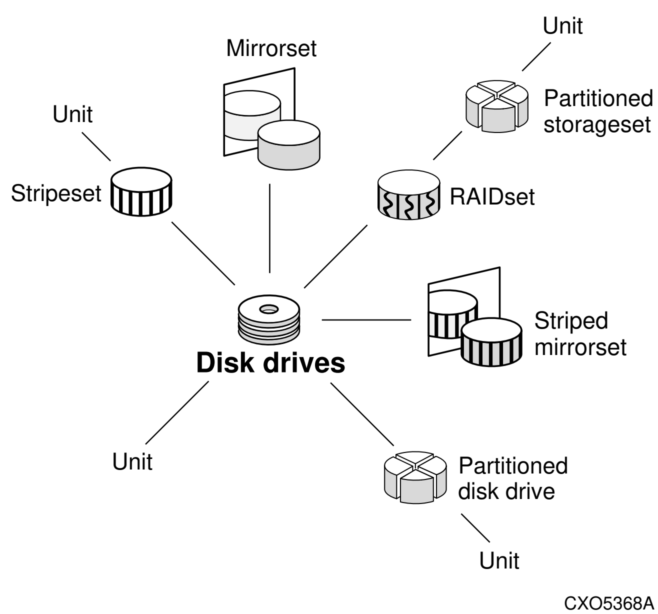
The controller does much more than simply manage I/O requests: it provides the ability to combine several ordinary disk drives into a single, high-performance storage unit called a storageset.

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

You 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 1–2.

- 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—just like 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 1–2 Units Created from Storagesets, Partitions, and Disk Drives



Of course, the controller also lets you add tape drives, loaders, and libraries to your subsystem to meet all of your storage requirements. For a complete discussion of RAID, refer to *The RAIDBOOK—A Source Book for Disk Array Technology.*

Table 1–1 summarizes the features of your controller.

Table 1–1 Summary of Controller Features

Feature	Supported
Host protocol	SCSI–2
Host bus interconnect	Wide Ultra Differential SCSI–2
Device protocol	SCSI–2
Device bus interconnect	Fast Wide Ultra Single-ended SCSI-2
Number of SCSI device ports	6
Number of SCSI device targets per port	12
Maximum number of SCSI devices (with two additional BA370 shelves)	72
RAID levels	0, 1, 0+1, 3/5
Cache size	64 or 128 MB
Mirrored write-back cache sizes	32 or 64 MB
Maximum number of host target ID numbers per controller	8
Program card updates	Yes
Device warm swap	Yes
Exercisers for testing disks	Yes
Tape drives, loaders, and libraries	Yes
Number of configuration entities (devices + storagesets + partitions + units)	191
Maximum number of RAID 5 storagesets	20
Maximum number of RAID 5 and RAID 1 storagesets	30 for dual controller configurations 20 for single controller configurations

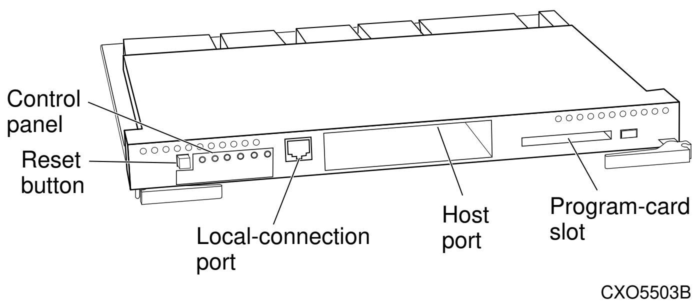
Table 1–1 Summary of Controller Features (Continued)

Feature	Supported
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	64 (63 if you're using the StorageWorks Command Console)
Maximum number of devices per unit	32
Maximum host port transfer speed	20 MHz
Largest device, storageset, or unit	120 GB

Controller Components

Take a few moments to familiarize yourself with the controller's components shown in Figure 1–3.

Figure 1–3 Key Controller Components



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. The reset button flashes green about once every second to indicate that the controller is operating normally. If an error occurs, one or more of the amber LED lights on the control panel will flash in a pattern that will help you to diagnose the problem. See the *HSZ70 Array Controller HSOF Version 7.0 Service Manual* for details about troubleshooting your controller.

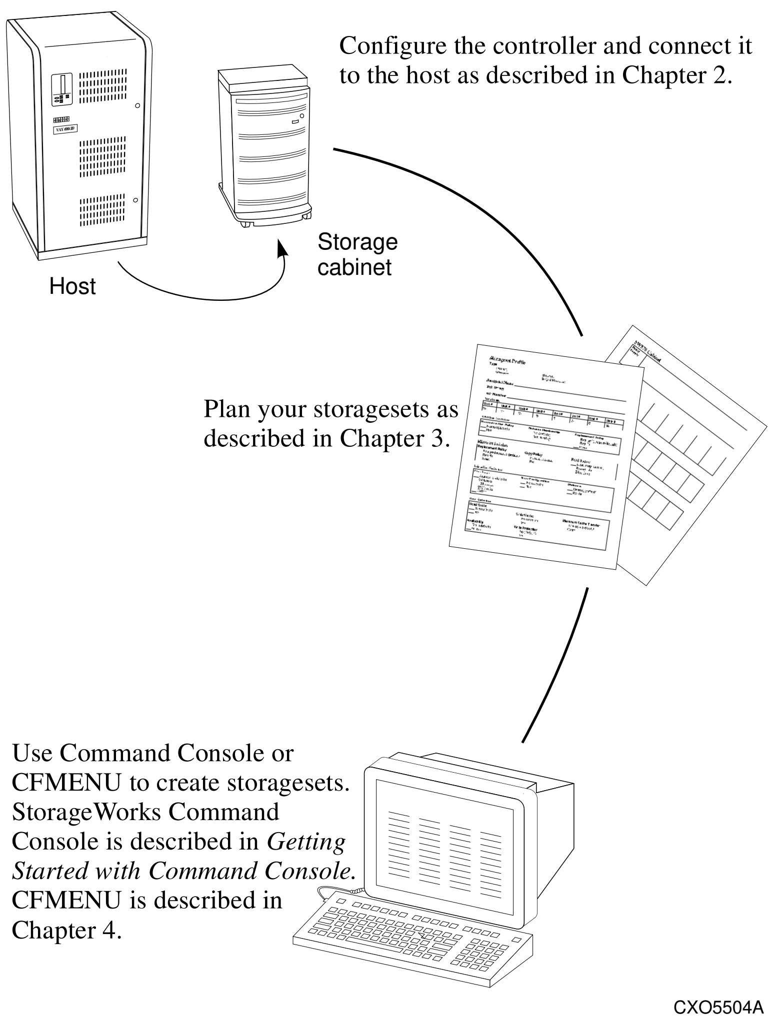
The host port and program-card slot are also located on the front panel, making it easy to update the controller software or to connect the controller to a different host.

The backplane enables two controllers to communicate with each other in dual-redundant configurations. It also contains device ports that enable the controller to communicate with the devices in your subsystem.

Key Steps for Configuring Your Subsystem

Figure 1–4 shows the key steps you will follow to set up and configure your subsystem and its controller. Each of these key steps are explained later in this book.

Figure 1–4 Overview of Configuring a Subsystem



CHAPTER 2

Configuring an HSZ70 Array Controller

This chapter contains information about configuring an HSZ70 array controller and the modules that support its operation in a StorageWorks subsystem.

Introduction

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

For the complete syntax and descriptions of the CLI commands used in the configuration procedure, see the *HSZ70 Array Controller HSOF Version 7.0 CLI Reference Manual*.

Configuration Rules

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

- Maximum 64 assignable, host-visible LUNs (maximum 63 assignable when using StorageWorks Command Console)
- Maximum 120 GB LUN capacity
- Maximum 72 physical devices
- Maximum 20 RAID-5 storagesets
- Maximum 30 RAID-5 and RAID-1 storagesets for dual controller configurations. Maximum 20 for single controller configurations.
- Maximum 45 RAID-5, RAID-1, and RAID-0 storagesets
- Maximum 8 partitions per storageset or individual disk
- Maximum 6 members per mirrorset
- Maximum 14 members per RAIDset or stripeset
- Maximum 32 physical device members total for a unit
- Maximum 1 external tape device per device port. If you have an external tape drive on a port, you cannot configure any other devices (disks or tapes) on that port.
- Maximum 1 internal (within an SBB) tape per device port. You can configure disks in the remaining slots on the port.

Configuring a Controller

You can use this procedure to configure your controller in a single, transparent failover, or multiple-bus failover configuration. Use the references in each step to locate details about the commands and concepts.

To configure a controller:

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

   See “Setting the PVA Module ID Switch,” page 2-6, for details about PVA switch settings.

2. Establish a local connection to the controller.

   See “Establishing a Local Connection to the Controller,” page 2-7, for details about creating a local connection.

3. Choose a single or failover configuration for the controller:

   a. If you are configuring a single controller, skip to step 7.

   b. If you are configuring dual redundant controllers in transparent failover mode, skip to step 4.

   c. If you are configuring dual-redundant controllers in multiple-bus (sometimes called host-assisted) failover mode, skip to step 5.

4. 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 2-9.

   Skip to step 6.

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

   SET MULTIBUS_FAILOVER COPY = THIS_CONTROLLER

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

See “Selecting a Failover Mode,” page 2-9, for details about failover modes.

Skip to step 7.

6. In a transparent failover configuration, if you want to enable mirrored write-back cache, enter this single command:

   SET THIS_CONTROLLER MIRRORED_CACHE

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

See “Selecting a Cache Mode,” page 2-10, for details about mirrored write-back cache.

7. Set the SCSI target IDs for the controller. Use the following syntax:

   SET THIS_CONTROLLER ID = *(n,n,n,n)*

   If you are configuring controllers in a dual-redundant configuration, issue this one command to set the ID numbers for both controllers.

   See “Setting the PVA Module ID Switch,” page 2-6, for details about valid SCSI target ID numbers.

8. In dual-redundant configurations, prefer SCSI target ID numbers to the controllers:

   a. If you’re configuring controllers in transparent failover mode, prefer some or all SCSI target ID numbers to “this controller.” Use the following syntax:

      SET THIS_CONTROLLER PREFERRED_ID = *(n,n)*

      where *n,n* is a subset of the target ID numbers you declared in step 7.

   b. If you’re configuring controllers in a multiple-bus failover configuration, prefer specific units to the controllers by issuing the following commands. Use the following syntax:

      SET *unit-number* **PREFERRED_PATH=THIS_CONTROLLER**

      SET *unit-number* **PREFERRED_PATH=OTHER_CONTROLLER**

See “Using Preferred ID Numbers,” page 2-19, for details about preferred SCSI target ID numbers.

9. Optional: Change the CLI prompt. Use the following syntax:

   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 the *HSZ70 Array Controller HSOF Version 7.0 CLI Reference Manual* for more information about using the SET OTHER_CONTROLLER PROMPT= command.

10. Optional: Set the maximum data-transfer rate. Use the following syntax:

    SET THIS_CONTROLLER TRANSFER_RATE_REQUESTED=*speed*

    If you’re configuring dual-redundant controllers, also set the transfer rate for the “other controller.” Set the same rate for both controllers. Use the following syntax:

    SET OTHER_CONTROLLER TRANSFER_RATE_REQUESTED=*speed*

    See “Setting the Maximum Data-Transfer Rate,” page 2-20, for an explanation of the transfer rate and how to set it.

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

    SET THIS_CONTROLLER CACHE_UPS

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

    SET OTHER_CONTROLLER CACHE_UPS

    See “Backing up Power with a UPS,” page 2-20, for more information.

12. Restart the controller, using the following syntax:

    RESTART THIS_CONTROLLER

    See the RESTART THIS_CONTROLLER command in the *HSZ70 Array Controller HSOF Version 7.0 CLI Reference Manual* for more information about using this command.

13. 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 the *HSZ70 Array Controller HSOF Version 7.0 CLI Reference Manual* for more information about using this command.

14. Connect the controller to the host.

    See “Connecting a Controller to the Host,” page 2-21, for information about how to complete the connection.

15. Plan and configure storagesets for your subsystem.

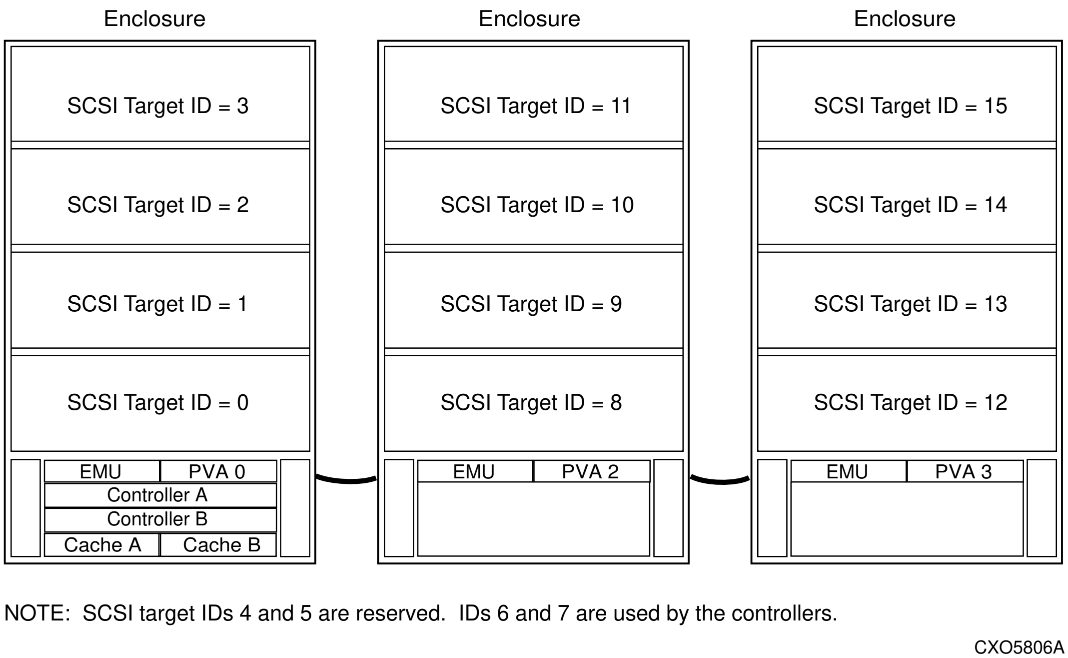
    See Chapter 3, “Planning Storagesets,” and Chapter 4, “Configuring Storagesets” for detailed information about planning and configuring storagesets.

Setting the PVA Module ID Switch

The PVA module provides unique addresses to extended subsystems. Each BA370 rack-mountable enclosure in an extended subsystem must have its own PVA ID. Use PVA ID 0 for the enclosure that contains the controllers. Use PVA IDs 2 and 3 for the additional enclosures. Figure 2–1 illustrates the PVA settings in an extended subsystem.

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

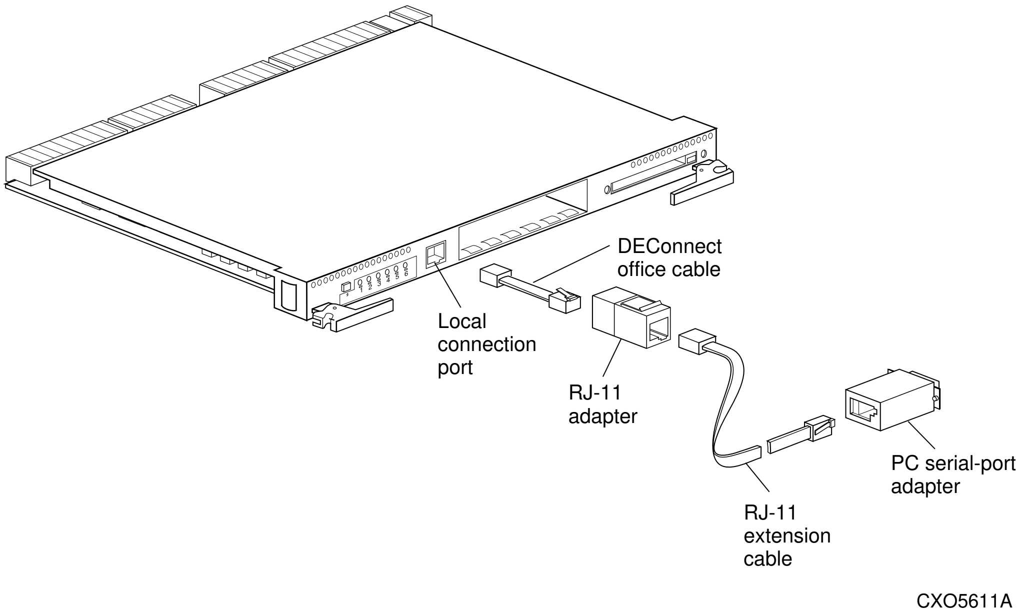
Figure 2–1 SCSI Target IDs and PVA Settings in an Extended Subsystem



Establishing a Local Connection to the Controller

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

Figure 2–2 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 the other controller fails.

You distribute the I/O load between the two controllers in a transparent failover configuration using the SET THIS_CONTROLLER ID= or SET OTHER_CONTROLLER ID= command to specify which controller handles which targets and units. 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 rack-mountable enclosure 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. By doing this, when you add devices, storagesets, and units to one controller's configuration, they are automatically added to the other controller's configuration.
- If you decide to configure your devices before setting the controllers for transparent failover, make sure you know which controller has the good configuration information before specifying SET FAILOVER COPY=. See the *HSZ70 Array Controller HSOF Version 7.0 CLI Reference Manual* for details about the SET FAILOVER COPY= command.

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

Using Multiple-Bus Failover

Multiple-bus (or host-assisted) failover is a dual-redundant configuration in which each controller has its own connection to the host. Thus, if one of the controllers loses contact with the host, the other controller can service all of the storage units through its host connection until the failed connection is restored. Of course, because both controllers service the same storage units, either controller can continue to service all of the units if the other controller fails.

Keep the following points in mind when considering using multiple-bus failover:

- The host distributes the I/O load between the controllers.
- The host must have two SCSI adapters as well as operating-system software to support the multiple-bus failover dual-redundant configuration.
- SCSI target ID number and LUN configurations on each controller must be indentical.
- Mount both controllers in the same BA370 rack-mountable enclosure and follow the steps in "Configuring a Controller," page 2-3.

Selecting a Cache Mode

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

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

Increasing Read Performance

Read Caching
This caching technique decreases 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.

When the controller receives a read request from the host, it reads the data from the disk drives, delivers it to the host, and also stores it in the cache memory. If the host requests the same data again, the controller can satisfy the read request from the cached data rather than re-reading it from the disk drives.

By default, read caching is enabled for all storage units.

Write-Through Caching
This caching technique also decreases 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.

When the controller receives a write request from the host, it stores the data in its cache memory, writes the data to the disk drives, then notifies the host when the write operation is complete. If the host requests the recently written data, the controller satisfies the read request from its cache memory rather than from the disk drives. This caching technique is called write-through caching because the write data passes through—and is stored in—the cache memory on its way to the target disk drives.

If read caching is enabled for a storage unit, write-through caching is also enabled. Also, because both caching techniques enhance the controller's read performance, if you disable read caching, write-through caching is automatically disabled.

By default, read caching—and, therefore, write-through caching—is enabled for all storage units.

Increasing Write Performance

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 storagesets. In either case, the controller will not provide write-back caching to a unit unless you ensure that the cache memory is non-volatile as described below.

Fault-Tolerance for Write-Back Caching

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

- Non-volatile memory (required for write-back caching)
- Mirrored caching (optional)
- Dynamic caching techniques (automatic)

Non-Volatile Memory
The controller cannot provide write-back caching to a unit unless its cache memory is non-volatile. In other words, 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 weren't backed up with backup power—the unwritten cache data would be lost during a power failure.

By default, the controller expects to use an ECB as its cache module's back-up power source. See the *HSZ70 Array Controller HSOF Version 7.0 Service Manual* for more information about the ECB. If the subsystem is backed up by a UPS see "Backing up Power with a UPS," page 2-20 for information about back-up power.

Note The controller executes multiple write operations to satisfy a single write request for a RAIDset or mirrorset. For this reason, a RAIDset or mirrorset requires non-volatile cache memory to ensure data redundancy until the write request is satisfied.

Regardless of the back-up power source you choose, the cache-memory power LED flashes about once every three seconds to indicate the cache module's memory array is receiving power from its primary power source.

Mirrored Caching

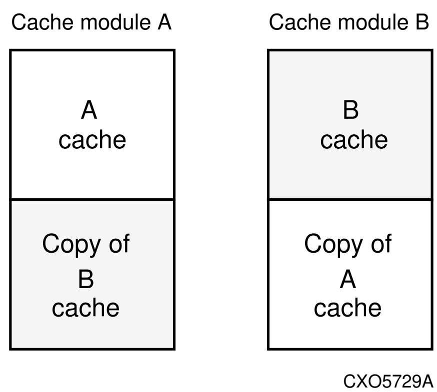
To further ensure the availability of unwritten cache data, you can use a portion of each cache module's memory to mirror the other cache module's write-back data in a dual-redundant configuration.

Figure 2–3 shows the principle of mirrored caching: half of cache A mirrors cache B's write-back data and vice versa. This arrangement ensures that the write-back data will be preserved if a cache module or any of its components fail.

Note When your controllers are configured to use mirrored write-back cache, the cache capacity is half of the total amount of cache in the configuration. If each cache module has 64 MB of cache for a total of 128 MB of cache in the configuration, the cache capacity is 64 MB.

See "Enabling Mirrored Write-Back Cache," page 2-19 for considerations when using mirrored write-back cache.

Figure 2–3 Mirrored Caching



Dynamic Caching Techniques

If the controller detects a full or partial failure of its cache module or ECB, it automatically reacts to preserve the cached write-back data. Then, 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 its ECB.

Table 2–1 shows the consequences of a full or partial failure of cache module A in a dual-redundant configuration. The consequences shown in this table are reciprocal for a failure of cache module B.

Table 2–1 Cache Policies and Cache Module Status

Cache Module Status		Cache Policy	
Cache A	**Cache B**	**Unmirrored Cache**	**Mirrored Cache**
Good	Good	**Data loss:** No. **Cache policy:** Both controllers support write-back caching. **Failover:** No.	**Data loss:** No. **Cache policy:** Both controllers support write-back caching. **Failover:** No.

Table 2–1 Cache Policies and Cache Module Status (Continued)

Cache Module Status		Cache Policy	
Cache A	**Cache B**	**Unmirrored Cache**	**Mirrored Cache**
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**.**	**Data loss:** No. Controller A recovers its lost write-back data from the mirrored copy on cache B. **Cache policy:** Both controllers support write-back caching. **Failover:** No.
SIMM 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 transparent failover, all units failover to controller B. In multiple-bus failover, only those units that use write-back caching, such as RAIDsets and mirrorsets, failover to controller B. All units with lost data become inoperative until you clear them with the CLEAR LOST_DATA command. Units that didn't lose data operate normally on controller B. In single 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.	**Data loss: No.** Controller A recovers all of its write-back data from the mirrored copy on cache B. **Cache policy:** Controller A supports write-through caching only; controller B supports write-back caching. **Failover:** In transparent failover, all units failover to controller B and operate normally. In multiple-bus failover, only those units that use write-back caching, such as RAIDsets and mirrorsets, failover to controller B.

Table 2–1 Cache Policies and Cache Module Status (Continued)

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

Table 2–2 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 2–2 Cache Policies and ECB Status

ECB Status		Cache Policy	
ECB Cache A	**ECB Cache B**	**Unmirrored Cache**	**Mirrored Cache**
Good	Good	**Data loss:** No. **Cache policy:** Both controllers continue to support write-back caching. **Failover:** No.	**Data loss:** No. **Cache policy:** Both controllers continue to support write-back caching. **Failover:** No.

Table 2–2 Cache Policies and ECB Status (Continued)

ECB Status		Cache Policy	
ECB Cache A	**ECB Cache B**	**Unmirrored Cache**	**Mirrored Cache**
Low	Good	**Data loss:** No. **Cache policy:** Controller A supports write-through caching only; controller B supports write-back caching. **Failover:** In transparent failover, all units failover to controller B and operate normally. In multiple-bus failover, only those units that use write-back caching, such as RAIDsets and mirrorsets, failover to controller B. In single configurations, the controller only provides write-through caching to its units.	**Data loss:** No. **Cache policy:** Both controllers continue to support write-back caching. **Failover:** No.
Failed	Good	**Data loss:** No. **Cache policy:** Controller A supports write-through caching only; controller B supports write-back caching. **Failover:** In transparent failover, all units failover to controller B and operate normally. In multiple-bus failover, only those units that use write-back caching, such as RAIDsets and mirrorsets, failover to controller B. In single configurations, RAIDsets and mirrorsets become inoperative.	**Data loss:** No. **Cache policy:** Both controllers continue to support write-back caching. **Failover:** No.
Low	Low	**Data loss:** No. **Cache policy:** Both controllers support write-through caching only. **Failover:** No.	**Data loss:** No. **Cache policy:** Both controllers support write-through caching only. **Failover:** No.

Table 2–2 Cache Policies and ECB Status (Continued)

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

Setting SCSI Target ID Numbers

Each BA370 rack-mountable enclosure in a configuration contains a range of SCSI target ID numbers. Each controller can access the range of target ID numbers you specify with the SET THIS_CONTROLLER ID= *(n,n,n,n)* command. The valid target ID numbers are 0–15, with the exception of ID numbers 4–7.

Using more than one target ID number allows the controller to present more units to the host. Enclose multiple ID numbers in parentheses and separate each by a comma.

Figure 2–1 on page 2-7 illustrates the SCSI target ID numbers and PVA settings for subsystems.

Device addressing which allows the controller to identify the location of devices based on a port-target-LUN (PTL) number is described in “Device PTL Addressing Convention within the Controller,” page 3-20.

Enabling Mirrored Write-Back Cache

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

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

For important considerations when replacing or upgrading SIMMs in a mirrored cache configuration, see the *HSZ70 Array Controller HSOF Version 7.0 Service Manual*.

Using Preferred ID Numbers

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

You can specify a total of eight preferred SCSI target ID numbers to a single controller. Use the ID numbers you defined with the SET THIS_CONTROLLER ID=command.

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

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

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

```
SET OTHER_CONTROLLER NOPREFERRED_ID
```

By declaring that it has no preferred IDs, the “other controller” won’t respond to any SCSI target IDs on the host SCSI bus. Instead, “this controller” will process all I/O during normal operation.

Changing the CLI Prompt

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

See the *HSZ70 Array Controller HSOF Version 7.0 CLI Reference Manual* for more information about the SET THIS_CONTROLLER command.

Setting the Maximum Data-Transfer Rate

You can set the data-transfer rate to 5 MHz, 10 MHz, and 20 MHz. Table 2–2 lists the maximum transfer rates for different lengths of SCSI bus cables. These lengths represent cable lengths plus enclosure-bus lengths.

Table 2–3 Maximum Data Transfer Rates for SCSI-bus Cables

Bus Type	Transfer Rate	Meters	Feet
single ended	5 MHz	6	19.7
single ended	10 MHz	3	9.8
differential	20 MHz	25	82.0

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 instead choose to use an uninterruptable power supply (UPS) to provide backup power in the event of a primary power failure. See the *HSZ70 Array Controller HSOF Version 7.0 CLI Reference Manual* for details about the SET THIS_CONTROLLER CACHE_UPS command. See Table 2–1 on page 2-14 and Table 2–2 on page 2-16 for information about cache policies.

Selecting a Host Compatibility Mode

For details about selecting a host compatibility mode, see the Getting Started Guide that came with your platform kit.

Connecting a Controller to the Host

The controller's configuration determines how you connect it to a host. Whether you install a new controller or move an old one to a new location, you should always configure it, or at least set its SCSI target IDs, before connecting it to a host. Failure to do so may adversely affect the host or cluster.

Follow one of these procedures to connect your controller to a host. Each procedure is described in detail in this section:

- Connecting a single (nonredundant) controller to the host.
- Connecting dual-redundant controllers to the host.
- Connecting multiple bus failover, dual-redundant controllers to the host.

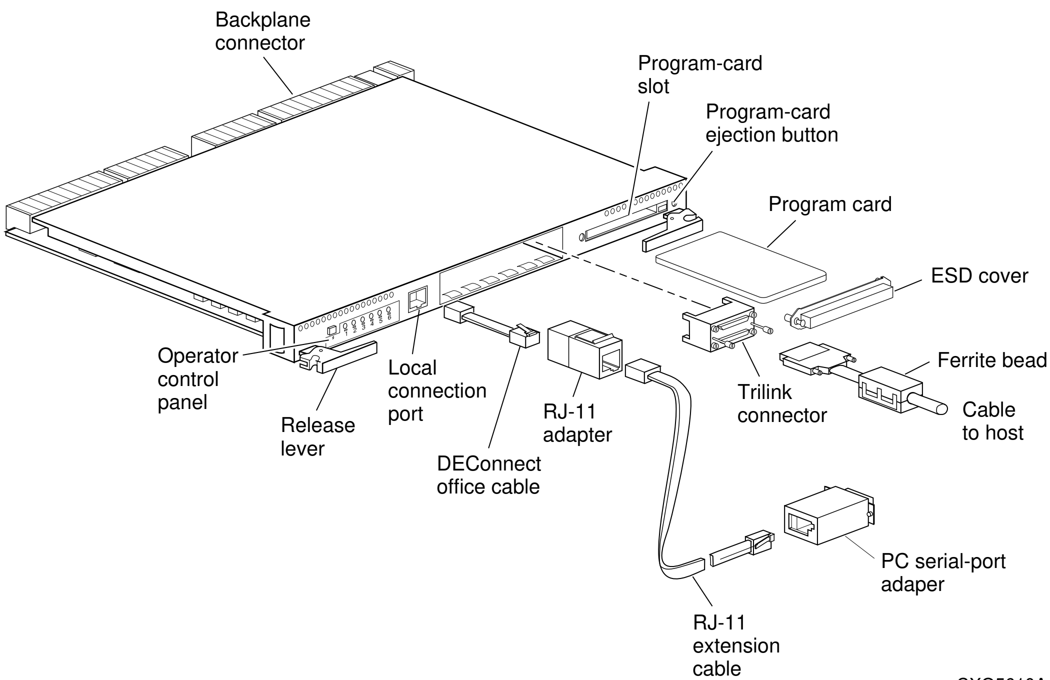
Connecting a Single Controller to the Host

To connect a single, nonredundant controller to the host:

1. Stop all I/O from the host to its devices on the bus to which you are connecting the controller.
2. Remove the trilink connector (12–39921–01) from the controller. This connector is a 68-pin Y-adapter that maintains bus continuity even when it is disconnected from the controller.
3. Connect the bus cable from the host to one of the connectors on the front of the trilink connector.
4. If you're connecting a host to a controller in a BA370 enclosure that will reside in an SW600 cabinet, snap the ferrite bead on the bus cable within one inch of the controller.

If you’re connecting a host to a controller in any other enclosure or cabinet, skip to step 5.

Figure 2–4 Connecting a Single Controller to Its Host



5. If the controller is at the end of the host bus, connect a terminator to the other connector on the front of the trilink connector. Otherwise, connect a cable that continues to the next device on the bus. (Be sure to install a terminator at the end of the bus.)
6. Reconnect the trilink connector to the host port on the controller. Do not disconnect the host cables from the trilink connector.
7. Route and tie the cables as desired.
8. 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.

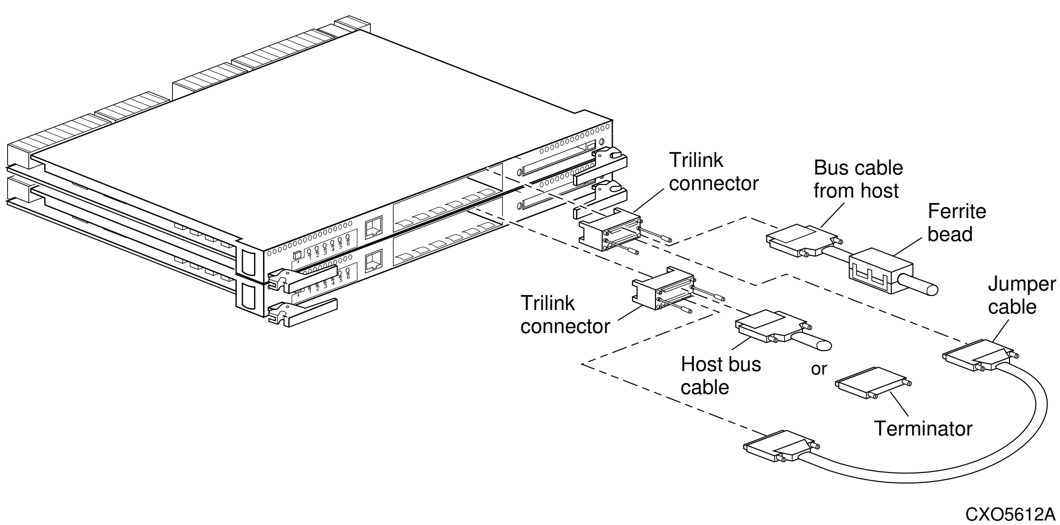
Connecting Dual-redundant Controllers to the Host

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. Remove the trilink connectors (12–39921–01) from both controllers. These connectors are 68-pin Y-adapters that maintain bus continuity even when they are disconnected from their controller.
3. Connect the bus cable from the host to one of the connectors on the front of one of the trilink connectors as shown in Figure 2–5.
4. If you're connecting a host to a controller in a BA370 enclosure that will reside in an SW600 cabinet, snap the ferrite bead on the bus cable within one inch of the controller.

   If you're connecting a host to a controller in any other enclosure or cabinet, skip to step 5.

Figure 2–5 Connecting Dual-Redundant Controllers to the Host



5. Connect the two trilink connectors with a jumper cable.
6. If the controllers are at the end of the bus, connect a terminator to the open connector on the front of the trilink connector. Otherwise, connect a cable that continues to the next device on the bus. (Be sure to install a terminator at the end of the bus.)

7. Reconnect the trilink connectors to the host ports on the controllers. Do not disconnect the host cables from the trilink connector.
8. Route and tie the cables as desired.
9. 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.

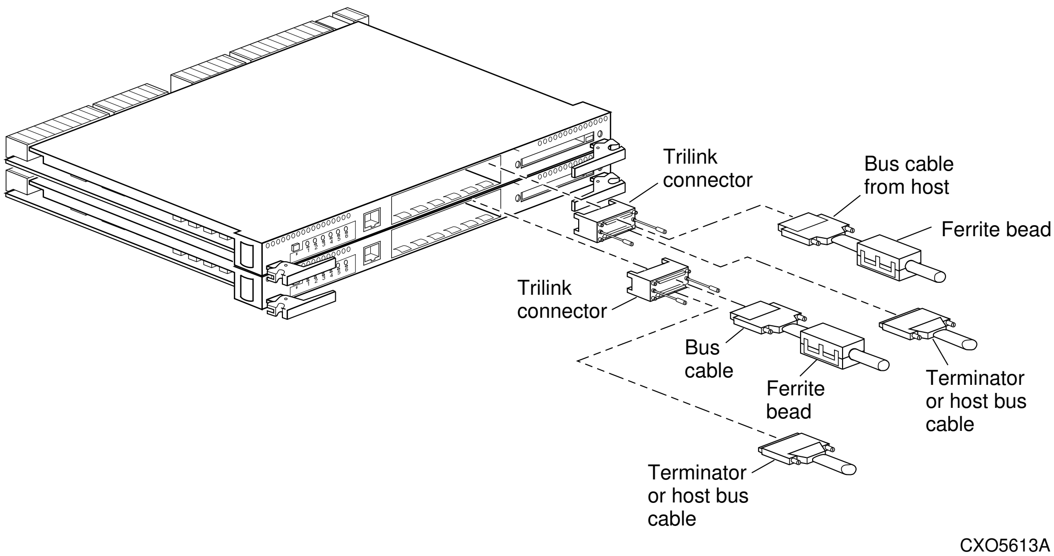
Connecting Multiple Bus Failover, Dual-Redundant Controllers to the Host

To connect a pair of multiple bus failover 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. Remove the trilink connectors (12–39921–01) from both controllers. These connectors are 68-pin Y-adapters that maintain bus continuity even when they are disconnected from their controller.
3. Connect a bus cable from the host to one of the connectors on the front of each trilink connector as shown in Figure 2–6.
4. If you're connecting a host to a controller in a BA370 enclosure that will reside in an SW600 cabinet, snap a ferrite bead on each bus cable within one inch of the controller.

   If you're connecting a host to a controller in any other enclosure or cabinet, skip to step 5.

Figure 2–6 Connecting Multiple Bus Failover, Dual-Redundant Controllers to the Host



5. If the controllers are at the end of the bus, connect a terminator to the open connectors on the front of each trilink connector. Otherwise, connect a cable that continues to the next device on each bus. (Be sure to install a terminator at the end of the bus.)
6. Reconnect the trilink connectors to host ports on the controllers. Do not disconnect the host cables from the trilink connectors.
7. Route and tie the cables as desired.
8. 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

Planning Storagesets

This chapter provides information to help you create a storageset profile for your subsystem. Each storageset type is described as well as the switches you can set for each type.

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 to make informed decisions about the switches that you can enable for each storageset or storage device that you configure in your subsystem.

Familiarize yourself with the kinds of information contained in a storageset profile, as shown in Figure 3–1.

Appendix B 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 3–1 A Typical Storageset Profile

Storageset Profile

Type

_X_ RAIDset ___ Stripeset
___ Mirrorset ___ Striped Mirrorset

Storageset Name accept default

Disk Drives DISK10300, DISK20300, DISK30300

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

Chunksize	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)	___ No (default)	_X_ 32 blocks (default)
___ No	_X_ Yes	___ 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 3-5, 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 the 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

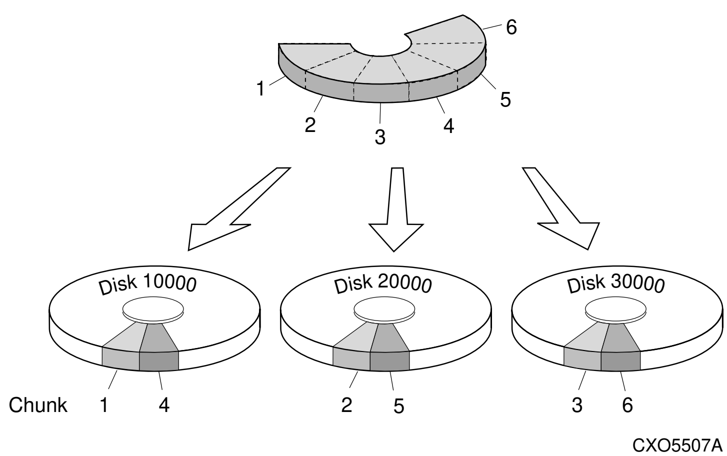
For a comprehensive discussion of RAID, refer to *The RAIDBOOK—A Source Book for Disk Array Technology.*

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 3–2 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 3-33, 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 *StorageWorks Array Controller HSZ70 Array Controller Operating Software HSOF Version 7.0 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 2 GB disk drives with 1 GB disk drives in the same storageset, you will waste 1 GB of capacity on each 2 GB member.

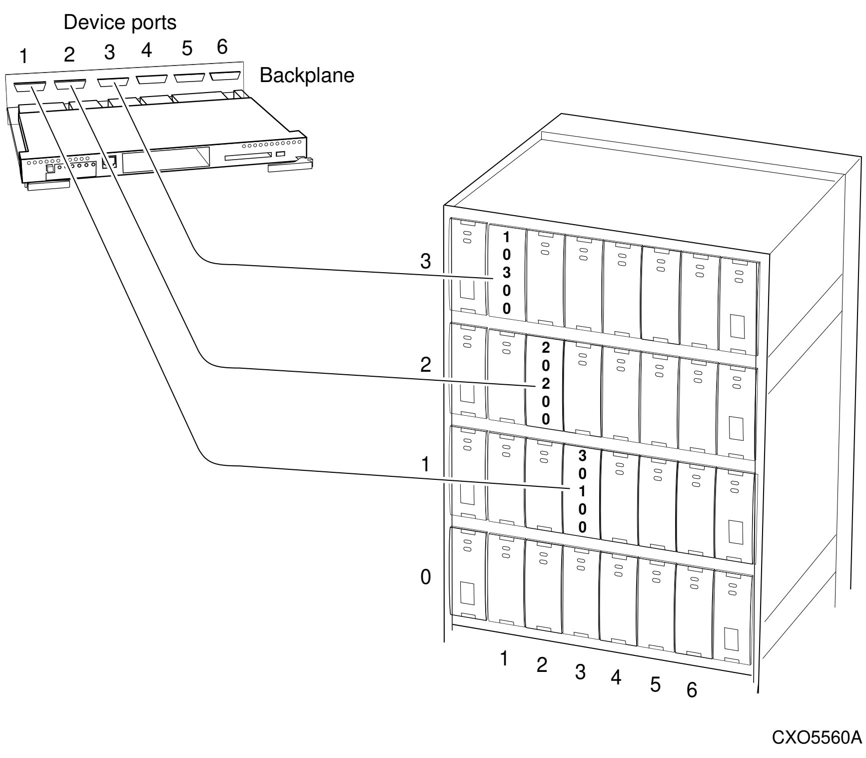
  If you need high performance and high availability, consider using a RAIDset, striped mirrorset, or a host-based shadow of a stripeset.
- Striping does not protect against data loss. In fact, because the failure of one member is equivalent to the failure of the entire stripeset, the likelihood of losing 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 l 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 4 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 3–3.
- Stripesets contain between two 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 3–3 Distribute Members across Ports

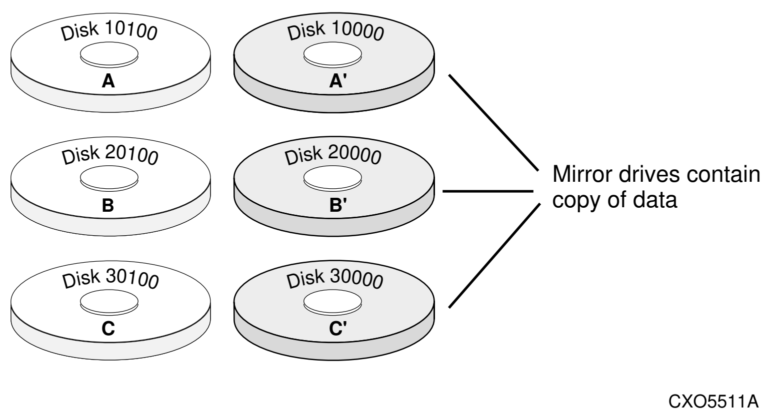


By spreading the traffic evenly across the buses, you will ensure that no bus handles the majority of data to the storageset.

Using Mirrorsets to Ensure Availability

Mirrorsets use redundancy to ensure availability, as illustrated in Figure 3–4. 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 3–4 Mirrorsets Maintain Two Copies of the Same Data



Considerations for Planning a Mirrorset

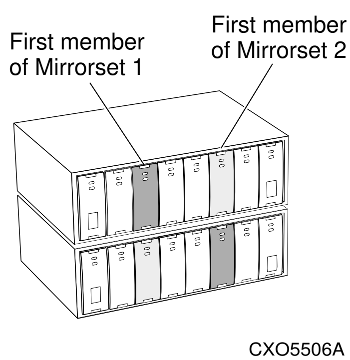
Keep these points in mind as you plan your mirrorsets:

- 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 6 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 3–5. (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 3–5 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 3-30, for more information about this switch.

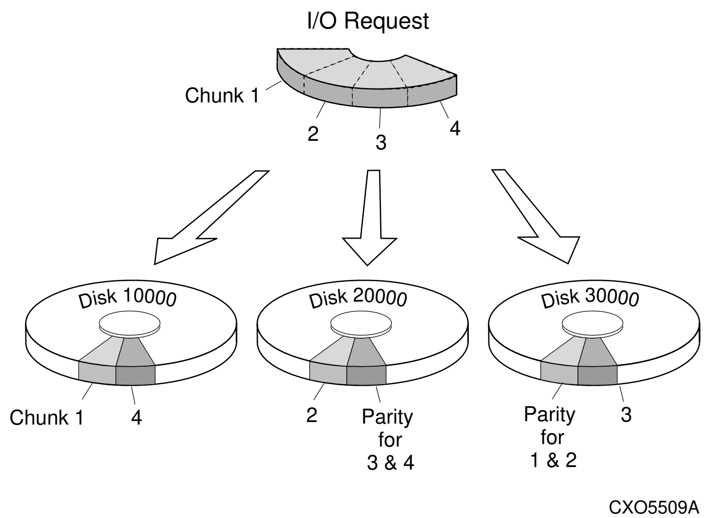
- Place mirrorsets and RAIDsets on different ports to minimize risk in the event of a single port bus failure.
- Mirrorset units are set to NOWRITEBACK_CACHE by default. To increase a unit’s performance, switch to WRITEBACK_CACHE.
- A storageset should only contain disk drives of the same capacity. The controller limits the capacity of each member to the capacity of the smallest member in the storageset. Thus, if you combine 2 GB disk drives with 1 GB disk drives in the same storageset, you waste 1 GB of capacity on each 2 GB member.
- Evenly distribute the members across the device ports to balance load and provide multiple paths as shown in Figure 3–3 on page 3-8.

- 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 3–6 illustrates the concept of RAIDsets and parity data.

Figure 3–6 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 3-33, for information about setting the chunk size.

Considerations for Planning a RAIDset

Keep these points in mind as you plan your RAIDsets:

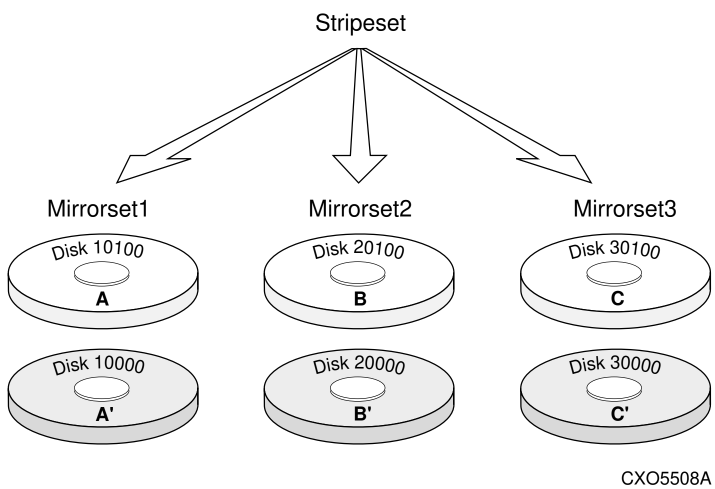
- Reporting methods and size limitations prevent certain operating systems from working with large RAIDsets. See the *StorageWorks Array Controller HSZ70 Array Controller Operating Software HSOF Version 7.0 Release Notes* or the Getting Started Guide that came with your platform kit for details about these restrictions.
- A write-back cache module is required for RAIDsets, but write-back cache need not be enabled for the RAIDset to function properly.
- Both write-back 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 3–3 on page 3-8.
- 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 2 GB disk drives with 1 GB disk drives in the same storageset, you’ll waste 1 GB of capacity on each 2 GB member.
- RAIDset units are set to NOWRITEBACK_CACHE by default. To increase a unit’s performance, switch to WRITEBACK_CACHE.
- Place 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 3–7 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 3-7, and "Considerations for Planning a Mirrorset," page 3-9.

Naming Storagesets and Units

Configure storagesets manually if you want to use your own naming scheme. StorageWorks Command Console and CFMENU name each storageset using a simple naming convention. Command Console and CFMENU also automatically provide unit-number prefixes and allow you to specify the actual number. See Chapter 4, “Configuring Storagesets,” for information about CFMENU. See the document *Getting Started with Command Console* for information about StorageWorks Command Console.

Assigning Unit Numbers for Host Access to Storagesets

A controller supports up to eight SCSI target IDs, each of which can present up to eight logical unit numbers (LUNs) to a host. This means that each controller or dual-redundant pair of controllers can present up to 64 storage units to a host. One of these is reserved for the StorageWorks Command Console interface, which leaves 63 available for use.

You will need to assign a unique unit number to each storageset, single disk unit, or storage device that you want your host to know about in your subsystem. A unit number is an alphanumeric tag that identifies each storage unit in your subsystem, such as D102 for a disk-based storage unit. 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: use D for disk drives or P for passthrough devices such as tape drives, loaders, and libraries. (If you use the CFMENU utility to configure your storagesets and devices, it will automatically supply a device letter for you.)
- A first number that indicates which controller accesses the storage unit during normal operation. Use one of the controller's SCSI target IDs (0 to 15).

  By carefully choosing the first number, you can establish preferred paths for all of your storage units in a dual-redundant configuration.
- A second number that is always zero.
- A third number that identifies the logical unit number (LUN) for the device or storage unit (0 to 7).

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

Table 3–2 Unit Numbering Examples

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

Note The host communicates with a logical unit based on its LUN address. The controller communicates with a device based on its PTL address.

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 3–8 on page 3-19. 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 3–8 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 “BA370 Rack-Mountable Enclosure Template,” page B-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 a shelf that supports 1 to 15 devices or "targets." Every device uses LUN 0, except some tape loaders, which use LUN 1.

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 for a single-controller configuration are 0 through 15 (excluding ID number 7). Valid target ID numbers for a dual-redundant controller configuration are 0 through 15 (excluding ID numbers 6 and 7).

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

- L—Designates the logical unit (LUN) of the device.

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

Figure 3–9 PTL Naming Convention

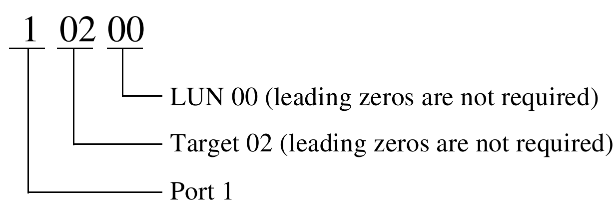
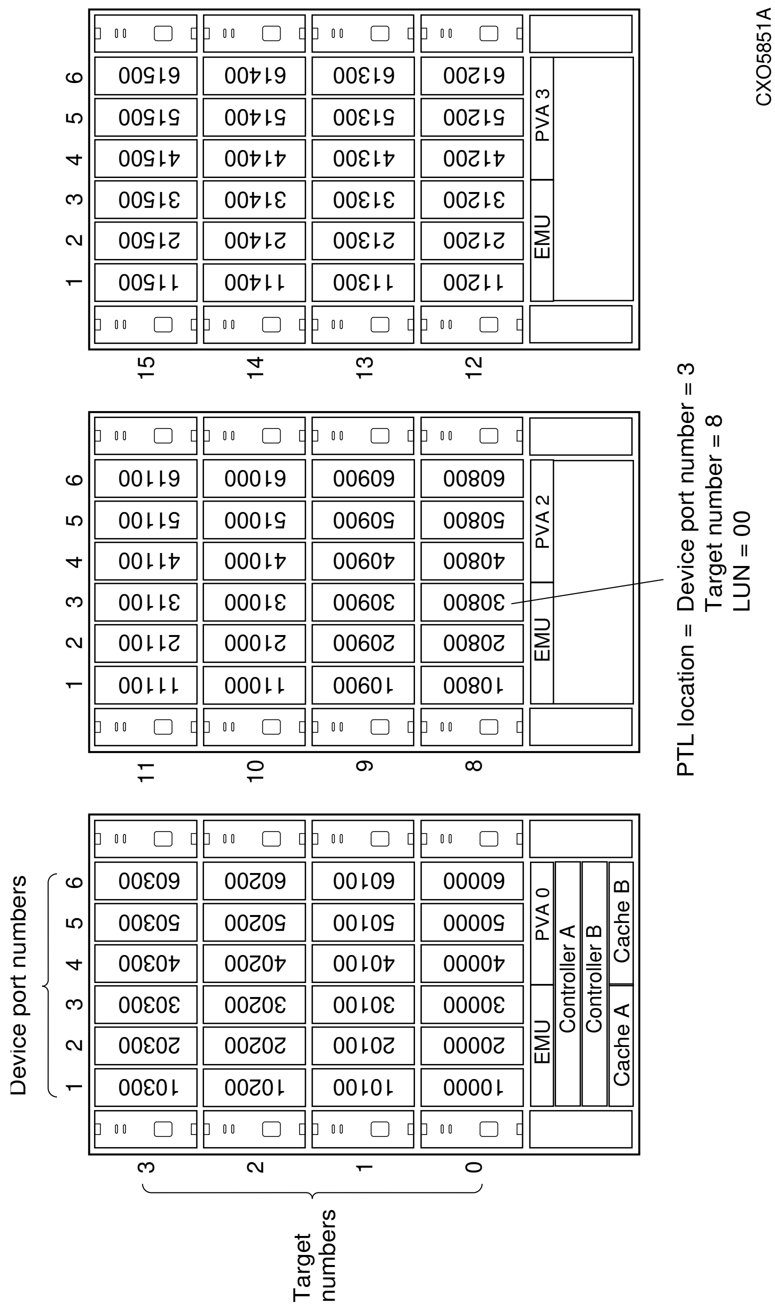


Figure 3–10 shows the addresses for each device in an extended configuration. Use this figure to help you work with the devices in your configuration.

Figure 3–10 PTL Addressing in an Extended Configuration

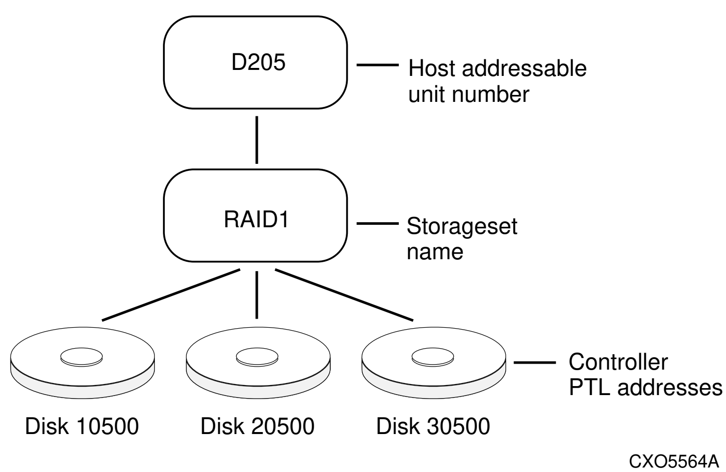


In Figure 3–10 the controller addresses DISK30800 through device port 3, target 08, LUN 00. This PTL location indicates the pathway the controller uses to address a disk or tape drive (device) in the subsystem. It also indicates the device name.

The controller uses the PTL location to name each device that you add to your subsystem with StorageWorks Command Console, the CONFIG utility, or CFMENU. (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; if it finds a tape drive at PTL 10200, it names it TAPE10200.

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 devices for the I/O request. (For example, the RAIDset “RAID1” might contain DISK10500, DISK20500, and DISK30500.) The controller generates the read or write request to the appropriate device using the PTL addressing convention. Figure 3–11 illustrates the concept of PTL addressing.

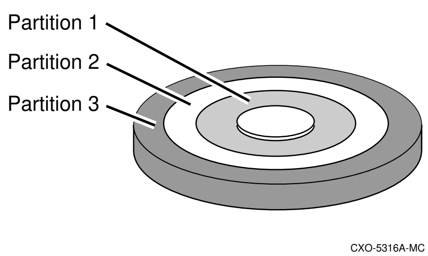
Figure 3–11 Locating Devices using PTLs



Planning Partitions

Use partitions to divide a storageset or disk drive into smaller pieces, each of which can be presented to the host as its own storage unit. Figure 3–12 shows the conceptual effects of partitioning a single-disk unit.

Figure 3–12 Partitioning a Single-Disk Unit



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

Defining a Partition

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

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

See “Partitioning a Storageset or Disk Drive,” page 4-14, for information on manually partitioning a storageset or single-disk unit.

Guidelines for Partitioning Storagesets and Disk Drives

Keep these points in mind as you plan your partitions:

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

Choosing Switches for Storagesets and Devices

Depending upon the kind of storageset or device you are configuring, you can enable the following kinds of 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 “Manually Configuring Storagesets,” page 4-2, indicate when and how to enable each switch.

Changing Switches

You can change the RAIDset, mirrorset, device, and unit switches at any time. See “Changing Switches for a Storageset or Device,” page 4-19, 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 kinds of 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:

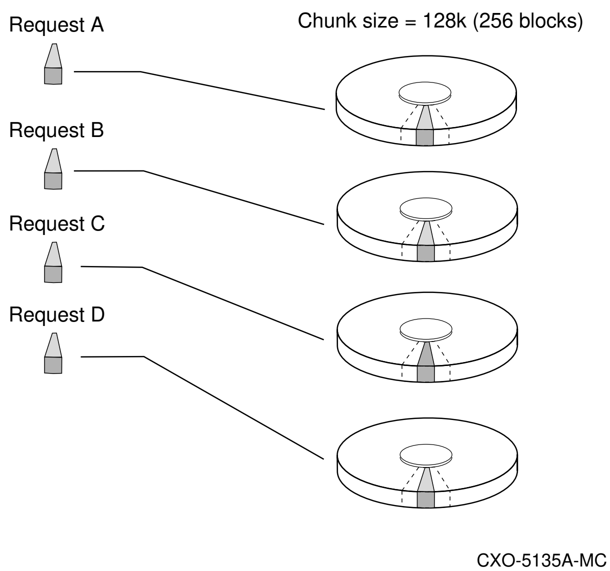
- CHUNKSIZE=DEFAULT lets the controller set the chunk size based on the number of disk drives (d) in a stripeset or RAIDset. If d < 9 then chunk size = 256. If d > 9 then chunk size = 128.
- CHUNKSIZE=*n* lets you specify a chunk size in blocks. The relationship between chunk size and request size determines 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 of 128K 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 3–13.

Figure 3–13 Chunk Size Larger than the Request Size



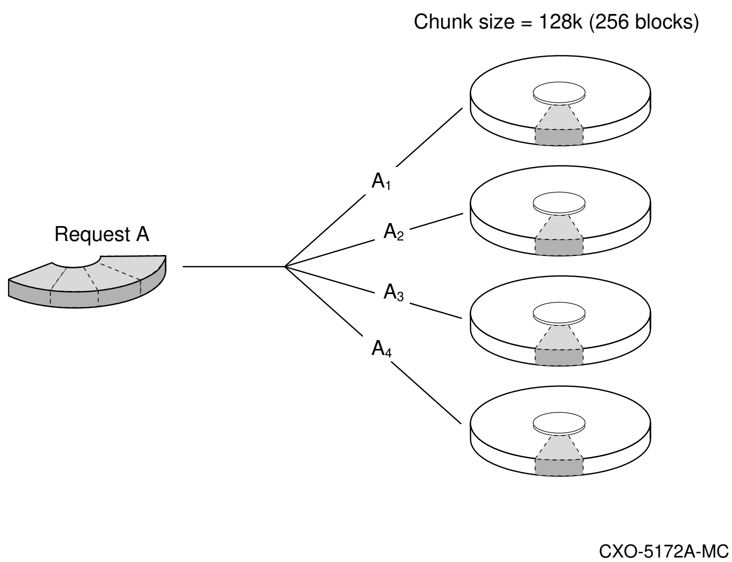
Applications such as interactive transaction processing, office automation, and file services for general timesharing tend to require high I/O request rates.

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

Increasing the Data Transfer Rate

A small chunk size relative to the average request size increases the data transfer rate by allowing multiple disk drives to participate in one I/O request. This concept is shown in Figure 3–14.

Figure 3–14 Chunk Size Smaller than the Request Size



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

Increasing Sequential Write Performance

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

Chunk size does not significantly affect sequential read performance.

Maximum Chunk Size for RAIDsets

Do not exceed the chunk sizes shown in Table 3–3 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 3–3 Maximum Chunk Sizes for a RAIDset

RAIDset Size	Max Chunk Size
3 members	1024 blocks

Table 3–3 Maximum Chunk Sizes for a RAIDset (Continued)

RAIDset Size	Max Chunk Size
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 3-38 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 *HSZ70 Array Controller HSOF Version 7.0 CLI Reference Manual* 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 3–4 for the listed storagesets and devices. See the *HSZ70 Array Controller HSOF Version 7.0 CLI Reference Manual* for a complete list of Unit switches.

Table 3–4 ADD UNIT Switches for Storagesets

Container Type \ **Switch**	ACCESS_ID=ALL ACCESS_ID=unit identification	PARTITION=partition-number	MAXIMUM_CACHED_ TRANSFER	PREFERRED_PATH NOPREFERRED_PATH	READ_CACHE NOREAD_CACHE	RUN NORUN	WRITE_PROTECT NOWRITE_PROTECT	WRITEBACK_CACHE NOWRITEBACK_CACHE
RAIDset	✓	✓	✓	✓	✓	✓	✓	✓
Stripeset	✓	✓	✓	✓	✓	✓	✓	✓
Mirrorset	✓	✓	✓	✓	✓	✓	✓	✓
NoTransportable Disk	✓	✓	✓	✓	✓	✓	✓	✓
Transportable Disk	✓	✓	✓	✓	✓	✓	✓	
Optical Disk	✓	✓	✓	✓	✓	✓	✓	✓
CD-ROM	✓	✓	✓	✓	✓	✓		
Partition	✓	✓	✓	✓	✓		✓	✓
Passthrough				✓				

Access Protection

Restrict or remove host access restrictions for a storage unit.

- ACCESS_ID=ALL (default) allows all hosts to access all units.
- ACCESS_ID=*nn* allows you to specify an access ID for a unit that corresponds to a SCSI host ID. Specifying an ID indicates that the unit can be accessed only by the host with the specified target ID number.

In a multiple host environment specifying access IDs prevents multiple hosts from accessing the same unit. Use this feature on a unit-by-unit basis to ensure units are accessed only by the host for which they are configured.

The access ID for a passthrough unit is always ALL; you cannot change the access ID for a passthrough unit.

You can define a unit's access ID with the ADD command when presenting the unit to the host, or you can specify an access ID for a unit already presented to the host, or change a unit's access ID, by using the SET command. See the *HSZ70 Array Controller HSOF Version 7.0 CLI Reference Manual* for more information about using the ADD and SET commands to define an access ID.

See the *StorageWorks Array Controller HSZ70 Array Controller Operating Software HSOF Version 7.0 Release Notes* for configuration rules and requirements that you should consider when setting up and working in a multiple host environment.

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.

Preferred Path for Multiple-Bus Failover Configurations

The PREFERRED PATH switch applies only to storage units in a multiple-bus failover configuration. See the discussion about assigning unit numbers in the section "Assigning Unit Numbers for Host Access to Storagesets," page 3-17 to find out how you can use unit numbers to establish preferred paths for storage units in a dual-redundant configuration.

Specify which controller accesses the storage unit. If one controller fails, the operational controller handles the I/O activity to all of the storage units regardless of their preferred paths:

- NOPREFERRED_PATH (default) allows either controller to access the storage unit.
- PREFERRED_PATH=THIS_CONTROLLER indicates that the controller to which you are connected handles all I/O activity to the storage unit. By establishing preferred paths, you can distribute the I/O load evenly between the two controllers by dividing the storage units into two equal groups—and assigning each group to its own controller.
- PREFERRED_PATH=OTHER_CONTROLLER indicates that the other controller—the one to which you are not connected—handles all I/O activity to the storage unit.

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.

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 (default) enables the controller to write new data to the storage unit.
- WRITE_PROTECT 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:

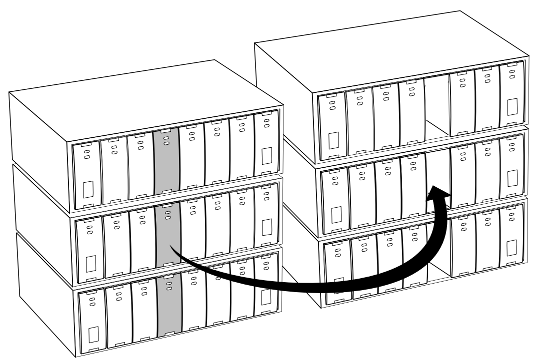
- WRITEBACK_CACHE enables write-back caching.
- NOWRITEBACK_CACHE (default) disables write-back caching.

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.

Moving Storagesets

You can move a storageset from one subsystem to another without destroying its data. You also can follow these steps to move a storageset to a new location within the same subsystem.

Figure 3–15 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 3–1

The following example moves unit D100 to another cabinet. D100 is the RAIDset RAID99 that comprises members 20000, 30000, and 40000.

SHOW RAID99

Name	Storageset	Uses	Used by
RAID99	raidset	disk10000 disk20000 disk30000	D100

```
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 3–2

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
```

The Next Step...

Turn to the section "Automatically Configuring Storagesets with CFMENU," page 4-21, if you want to configure your storage units automatically with CFMENU, a menu-based utility that simplifies the configuration process. This utility is especially helpful if you are configuring storagesets for the first time.

Turn to the section "Manually Configuring Storagesets," page 4-2, if you want to configure your storage units manually by issuing CLI commands from a local or remote connection. Configuring your storage units manually gives you more control when it comes to naming the storage units, because CFMENU automatically names them for you. It is a good idea to complete a profile for each storage unit or device that you want configured in your subsystem before you begin these procedures.

CHAPTER 4

Configuring Storagesets

The preferred method for configuring storagesets is to use StorageWorks Command Console. *Getting Started with Command Console* provides a description of the application and how to use it to configure storagesets. StorageWorks Command Console is a Graphical User Interface (GUI) for the set up and management of RAID storage subsystems.

This chapter describes how to configure storagesets either manually or by using the CFMENU utility.

Manually Configuring Storagesets

One method of configuring storagesets is manual configuration. This method allows you the most flexibility in defining and naming storagesets. See the *HSZ70 Array Controller HSOF Version 7.0 CLI Reference Manual* 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 Chapter 3, "Planning Storagesets," 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 4–1 contains the valid Initialize switches and values.

Table 4–1 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 4–2 contains the valid Unit switches and values.

Table 4–2 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	NOWRITEBACK_CACHE* WRITEBACK_CACHE
Write protection	NOWRITE_PROTECT* WRITE_PROTECT
Availability	RUN* NORUN
Access protection	ACCESS_ID=ALL* ACCESS_ID=*nn*

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 4–1
   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 WRITEBACK_CACHE

   SHOW STRIPE1

   SHOW D100

Configuring a Mirrorset

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

To configure a mirrorset:

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

   Use the following syntax to create a mirrorset:

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

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

Table 4–3 Mirrorset Switches

Mirrorset Switch	Value and Syntax
Replacement policy	POLICY=BEST_PERFORMANCE* POLICY=BEST_FIT NOPOLICY
Copy speed	COPY=NORMAL* COPY=FAST

Table 4–3 Mirrorset Switches (Continued)

Mirrorset Switch	Value and Syntax
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 4–4 contains the valid Initialize switches and values.

Table 4–4 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 4–5 contains the valid Unit switches and values.

Table 4–5 Unit Switches for Mirrorsets

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	NOWRITEBACK_CACHE* WRITEBACK_CACHE
Write protection	NOWRITE_PROTECT* WRITE_PROTECT

Table 4–5 Unit Switches for Mirrorsets (Continued)

Unit Switch	Value and Syntax
Availability	RUN* NORUN
Access protection	ACCESS_ID=ALL* ACCESS_ID=*nn*

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 4–2
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 D200 MIRR1 WRITEBACK_CACHE

SHOW MIRR1

SHOW D200

Configuring a RAIDset

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

To configure a RAIDset:

1. Create the RAIDset by adding its name to the controller’s list of storagesets and specifying the disk drives it contains. Optionally, you can append RAIDset switch values. If you do not specify switch values, 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 4–6 contains the valid RAIDset switches and values.

Table 4–6 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 4–7 contains the valid Initialize switches and values.

Table 4–7 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 4–8 contains the valid Unit switches and values.

Table 4–8 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
Write-back cache	NOWRITEBACK_CACHE* WRITEBACK_CACHE
Write protection	NOWRITE_PROTECT* WRITE_PROTECT
Availability	RUN* NORUN
Access protection	ACCESS_ID=ALL* ACCESS_ID=nn

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 4–3

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

```
ADD RAIDSET RAID1 DISK10000 DISK20000 DISK30000
INITIALIZE RAID1
ADD UNIT D300 RAID1 WRITEBACK_CACHE
SHOW RAID1
SHOW D300
```

Configuring a Striped Mirrorset

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

To configure a striped mirrorset:

1. Create—but do not initialize—at least two mirrorsets.
2. Create a stripeset and specify the mirrorsets it contains. Use the following 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 4–9 contains the valid Initialize switches and values.

Table 4–9 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 4–10 contains the valid Unit switches and values.

Table 4–10 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
Write-back cache	NOWRITEBACK_CACHE* WRITEBACK_CACHE

Table 4–10 Unit Switches for Striped Mirrorsets (Continued)

Unit Switch	Value and Syntax
Write protection	NOWRITE_PROTECT* WRITE_PROTECT
Availability	RUN* NORUN
Access protection	ACCESS_ID=ALL* ACCESS_ID=*nn*

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 4–4

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 WRITEBACK_CACHE
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 4-2. Optionally, you can append Device switch values. If you do not specify switch values, the default values (*) are applied.

Table 4–11 contains the valid Device switches and values.

Table 4–11 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=ASYNCHRO- NOUS

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 4–12 contains the valid Unit switches and values.

Table 4–12 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
Write-back cache	NOWRITEBACK_CACHE* WRITEBACK_CACHE
Write protection	NOWRITE_PROTECT* WRITE_PROTECT
Availability	RUN* NORUN
Access protection	ACCESS_ID=ALL* ACCESS_ID=*nn*

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 4–5
   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

Configuring a Tape Drive

The controller handles devices such as tape drives as passthrough devices. It uses a passthrough device to transport the host's commands to and from a SCSI device such as a tape drive. For this reason, configuring a tape drive involves creating a passthrough device to represent the tape drive. Note the following before configuring a tape drive:

- If the passthrough device has more than 160 bytes of sense data available for an event, the controller returns only the first 160 bytes from the passthrough device to the host.
- You must begin unit names for passthrough devices with the letter P.

To create a passthrough device to represent a tape drive:

1. Install the tape drive.
2. Create a passthrough device to logically represent the tape drive. Use the following syntax:

   ADD PASSTHROUGH *passthrough-name PTL-location*

3. Present the passthrough device to the host by giving it a unit number the host can recognize. Use the following syntax:

   ADD UNIT *unit-number passthrough-name*

4. Verify the configuration using the following syntax:

   SHOW *passthrough-name*

Example 4–6

The following example shows the commands you would use to create a passthrough device for controlling a tape drive:

ADD PASSTHROUGH TAPE10000 1 0 0

ADD UNIT P100 TAPE10000

SHOW *passthrough-name*

Configuring a Tape Loader

The controller handles devices such as tape loaders as passthrough devices. It uses a passthrough device to transport the host's commands to and from a SCSI device such as a tape loader. For this reason, configuring a tape loader involves creating a passthrough device to represent the loader. Note the following before configuring a tape loader:

- If the passthrough device has more than 160 bytes of sense data available for an event, the controller returns only the first 160 bytes from the passthrough device to the host.
- The controller supports only one tape loader per port. Connect only one tape loader or library to a controller device port at a time.
- You must begin unit names for passthrough devices with the letter P.

To create a passthrough device to represent a loader:

1. Install and configure the tape drive.
2. Install the tape loader.
3. Create a passthrough device at the loader's PTL location to logically represent the tape loader. Use the following syntax:

   ADD PASSTHROUGH *passthrough-name PTL-location*
4. Present the passthrough device to the host by giving it a unit number the host can recognize. Use the following syntax:

   ADD UNIT *unit-number passthrough-name*
5. Verify the configuration, using the following syntax:

   SHOW *passthrough-name*
6. Install and configure the host-based software that controls the loader. (This software is not provided with your array controller or its software.)

Example 4–7

The following example shows the commands you would use to create a passthrough device for controlling a tape loader:

ADD PASSTHROUGH LDR10300 1 3 0

ADD UNIT P130 LDR10300

SHOW PASSTHROUGH

Partitioning a Storageset or Disk Drive

See “Planning Partitions,” page 3-24, 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*

   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*

2. 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.

3. 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.

4. 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 4–13 contains the valid Unit switches and values for partitioned storagesets and disk drives.

Table 4–13 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
Write-back cache	NOWRITEBACK_CACHE* WRITEBACK_CACHE
Write protection	NOWRITE_PROTECT* WRITE_PROTECT
Access protection	ACCESS_ID=ALL* ACCESS_ID=*nn*

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

   SHOW *storageset-name*

Example 4–8

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	
2	1915 (0.98 MB)	1920	
3	1915 (0.98 MB)	3840	
4	2371 (1.21 MB)	5760	

.
.
.

```
ADD UNIT D1 RAID1 PARTITION=1 WRITEBACK_CACHE
ADD UNIT D2 RAID1 PARTITION=2 WRITEBACK_CACHE
ADD UNIT D3 RAID1 PARTITION=3 WRITEBACK_CACHE
ADD UNIT D4 RAID1 PARTITION=4 WRITEBACK_CACHE
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 4–9
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 4–10
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 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 4–11
   The following example shows the commands you would use to delete Stripe1, a three-member stripeset that comprises 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 3-26, 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 storageset. For example, the following command enables write protection for unit D100:

SET D100 WRITE_PROTECT

Automatically Configuring Storagesets with CFMENU

StorageWorks Command Console is the preferred method for automatically configuring storagesets and devices. See *Getting Started with Command Console*, for details about using it for configuration. CFMENU is an effective utility that simplifies the task of configuring storagesets and devices.

See “Changing Switches for a Storageset or Device,” page 4-19, if you want to modify an existing storageset. See “Configuring a Tape Drive,” page 4-12, if you want to configure a tape drive, or see “Configuring a Tape Loader,” page 4-13, if you want to configure a tape loader.

With CFMENU, you are free to think about what you want to do rather than how to get the controller to do it. Instead of issuing CLI commands, you choose from a menu of configuration tasks, such as adding a storageset or assigning a unit number.

Based on your choices, CFMENU prompts you for the information it needs to complete the task. It even prompts you to specify the switches you want to enable for a storageset or device.

Figure 4–1 CFMENU Main Menu

```
MAIN MENU:                   |Unconfig'd |   Config'd    Device       Product       Stor.set Stor.set Chnk  Trn  In- Re-        P W W
1. Add/delete devices        | Dev.PTLs  |     PTLs       Name          ID           Name    Typ/Sz  Size  sp. it'd duc  Unit  T P B
2. Add/delete mirrorsets     |---------- |    ------   --------- ----------------  --------- ------- ------ --- ---- --- -----  - - -
3. Add/delete stripesets     |           |miror:10200  DISK10200 RZ29B    (C) DEC     M1    MIR/1               Y   N         P
4. Add/delete raidsets/      |           |      20200  DISK20200 RZ29B    (C) DEC     M2    MIR/1               Y   N         P
   sparesets/failedsets      |           |      30200  DISK30200 RZ29B    (C) DEC     M3    MIR/2               Y   N         P
5. Partition Processing      |           |      50200  DISK50200 RZ29B    (C) DEC     "       "                 "   "
6. Initialize devices        |           |strps:40100  DISK40100 RZ29B    (C) DEC     S2    STR/3     16        Y        D*   P
   and/or storagesets        |           |      50100  DISK50100 RZ29B    (C) DEC     "       "       "         "
7. Add/delete units          |           |      60100  DISK60100 RZ29B    (C) DEC     "       "       "         "
8. Setup terminal            |           |                   M1                       S1    STR/3   32768       Y        D*
9. Exit CFMENU               |           |                   M2                       "       "       "         "
                             |           |                   M3                       "       "       "         "
                             |           |raid5:10300  DISK10300 RZ29B    (C) DEC     R2    RAD/3    1024       Y   N    D*   P
                             |           |      60000  DISK60000 RZ29B    (C) DEC     "       "       "         "   "
  D=Scroll down  U=Scroll up |           |      30300  DISK30300 RZ29B    (C) DEC     "       "       "         "   "
________________________________________________________________________________________________________________________________________
Enter menu choice (1,9) [9] ?
```

CFMENU uses columns of information to let you know what is going on during the configuration process. These columns are displayed on the Main Menu and other sub-menus and are continually updated to reflect the current configuration.

Table 4–14 lists the heading and contents for each column on the Main Menu.

Table 4–14 Interpreting CFMENU Columns

Column Heading	Information Displayed
Main menu	Shows the tasks you can accomplish with CFMENU.
Unconfig’d Dev.PTLs	Shows the PTL locations of devices that have not yet been added to the controller’s configuration. Use these devices to create single-disk units and storagesets, such as stripesets and RAIDsets.
Config’d PTLs	Shows the PTL locations of all devices that are used in—or are eligible to be used in—a storageset or a single-disk unit.
Device Name	Shows the names of all devices that are used in—or are eligible to be used in—a storageset or as a single-disk unit.
Product ID	Shows the model numbers of all devices that are used in—or are eligible to be used in—a storageset or as a single-disk unit.
Stor.set Name	Shows the name of all storagesets in the controller’s list of configured storagesets: by convention, Sn for stripesets, Mn for mirrorsets, and Rn for RAIDsets.
Stor.set Typ/Sz	Shows the types of storagesets and their number of members. For example, STR/5 is a stripeset that contains five disk drives; MIR/2 is a mirrorset that contains two disk drives.
Chnk Size	Shows the chunk size for stripesets and RAIDsets. This column is marked “unk” (unknown) until you initialize the storageset.
Trnsp.	Displays “Y” if you enabled the Transportable switch.
Init’d	Displays “Y” if you initialized the storageset.
Reduc	Displays “Y” if the storageset is in a reduced state due to the failure of one of its members.

Table 4–14 Interpreting CFMENU Columns (Continued)

Column Heading	Information Displayed
Unit	Shows the unit numbers for all storagesets or devices.
PT	Displays “P” if the unit is partitioned.
WP	Displays “Y” if the unit is write protected.
WB	Displays “Y” if you enabled write-back cache.

Considerations for Using CFMENU

Keep the following points in mind for using CFMENU:

- CFMENU names each storageset according to a simple naming convention: Mn for mirrorsets, Sn for stripesets, and Rn for RAIDsets, where n is a sequentially indexed number. CFMENU also automatically provides unit-number prefixes; you specify the actual number. (If you want to use your own naming scheme, configure your storagesets manually.)
- You can create and delete storagesets with CFMENU, however, you cannot modify them once you create them. See “Changing Switches for a Storageset or Device,” page 4-19 for the details about changing switches.
- If you are using dual-redundant controllers, you do not need to run CFMENU on both controllers simultaneously. The “other controller” automatically inherits the configuration you create with CFMENU.
- CFMENU cannot configure tape loaders. See “Configuring a Tape Drive,” page 4-12, and “Configuring a Tape Loader,” page 4-13.
- CFMENU cannot partition striped mirrorsets. See the procedure in “Partitioning a Storageset or Disk Drive,” page 4-14, to partition a striped mirrorset.

Adding Disk Drives with CFMENU

To add a disk drive or other storage device to your subsystem with CFMENU:

1. Install the new disk drives in your storage cabinet.

2. Start CFMENU using the following syntax:

   RUN CFMENU

3. From the Main Menu, choose task 1 to go to the Device Menu.
4. From the Device Menu, choose task 1 to add disk drives.
5. CFMENU presents—one at a time—the disk drives or devices that you may add to the subsystem. Type Y to add the disk drive, N to skip to the next one.
6. Set the disk drive NOTRANSPORTABLE. See “Initialize Switches,” page 3-33, for more information about this switch.
7. Return to the Main Menu and exit CFMENU.

Creating a Storageset with CFMENU

Creating a storageset or single-disk unit with CFMENU is as easy as choosing menu options and responding to prompts. Just remember to move through the Main Menu items from top to bottom for each storageset or single-disk unit you want to create. Press D to scroll down CFMENU’s columns; press U to scroll up.

To create a storageset or single-disk unit with CFMENU:

1. Start CFMENU using the following syntax:

   RUN CFMENU

2. Go to step 9 if you are configuring a single-disk unit, otherwise choose task 2, 3, or 4 depending on the kind of storageset you want to create.

   CFMENU displays a storageset menu that corresponds to your choice: Mirrorset Menu, Stripeset Menu, or RAIDset Menu.

3. From the storageset menu, choose task 1 to add a storageset to the controller’s list of available storagesets.
4. Enter the number of disk drives or members that you want to include in the storageset.
5. CFMENU presents—one at a time—the disk drives or members that you may include in the storageset. Type Y to include a member, N to skip to the next one.
6. When you reach the number of members specified in step 4, CFMENU prompts you for the switches you can apply to the storageset. Indicate your choice or press Return to accept the default value.

7. CFMENU displays a message that indicates the storageset's type and name, as well as the names of all its members. Press Return to create the storageset.
8. Return to the Main Menu and repeat step 2 through step 7 for each storageset or single-disk unit you want to create.
9. From the Main Menu, choose task 6 to go to the Initialization Menu.
10. From the Initialization Menu, choose task 1 to initialize the storageset (or the disk drive if you're creating a single-disk unit).
11. CFMENU prompts you for the Initialize switches you can apply to the storageset or single-disk unit. Indicate your choice or press Return to accept the default value. See "Initialize Switches," page 3-33, for more information about these switches.
12. Repeat step 10 and step 11 for each storageset or single-disk unit you want to initialize.
13. Return to the Main Menu.
14. From the Main Menu, choose task 7 to go to the Unit Menu.
15. From the Unit Menu, choose task 1 to assign a host-addressable unit number to the storageset or single-disk unit. See "Assigning Unit Numbers for Host Access to Storagesets," page 3-17, for more information about choosing unit numbers.
16. CFMENU prompts you for the Unit switches you can apply to the unit. Indicate your choice or press Return to accept the default value. See "Unit Switches," page 3-40, for more information about these switches.
17. Repeat step 15 and step 16 of this procedure for each storageset or single-disk unit to which you want to assign a unit number.
18. Return to the Main Menu and exit CFMENU.

Deleting a Storageset with CFMENU

To delete a storageset with CFMENU:

1. Start CFMENU using the following syntax:

   RUN CFMENU

2. From the Main Menu, choose the kind of storageset you want to delete.
3. From the storageset menu, choose task 2 to begin deleting storagesets.
4. CFMENU presents—one at a time—the storagesets that you may delete. Type Y to delete the storageset, N to skip to the next one.
5. Return to the Main Menu and exit.

Adding a Disk Drive to the Spareset with CFMENU

To add a disk drive to the spareset:

1. Start CFMENU using the following syntax:

   RUN CFMENU

2. From the Main Menu, choose task 4 to go to the RAIDset Menu.
3. From the RAIDset menu, choose task 4 to go to the Spareset/Failedset Menu.
4. From the Spareset/Failedset Menu, choose task 1 to add the disk drives to the spareset.
5. CFMENU presents—one at a time—the disk drives that you may add to the spareset. Type Y to add the disk drive, N to skip to the next one.
6. Return to the Main Menu and exit.

Partitioning a Storageset with CFMENU

To partition a storageset or single-disk unit:

1. Start CFMENU using the following syntax:

   RUN CFMENU

2. From the Main Menu, choose task 5 to go to the Partition Processing menu.
3. From the Partition Processing menu, choose task 1 to go to the Partition Menu.
4. CFMENU presents—one at a time—the storagesets that you may partition. Type Y to select the unit, N to skip to the next one.
5. From the Partition Menu, choose task 1 to partition the selected storageset.
6. Indicate the percentage of the unit that you want to dedicate to the first partition.
7. Repeat step 6 for each partition you want to create on the storageset.
8. Return to the Main Menu and exit.

CHAPTER 5

Periodic Procedures

This chapter describes procedures you might need to use when working with your HSZ70 array controller: formatting a disk, cloning data for backup, backing up your configuration, restarting your subsystem, and shutting down your subsystem.

Formatting Disk Drives

Use HSUTIL's FORMAT option to simultaneously format up to seven disk drives attached to a single controller or up to six disk drives attached to a dual-redundant pair of controllers.

Consider the following points before formatting disk drives with HSUTIL:

- HSUTIL cannot format disk drives configured as single-disk drive units or as members of a storageset, spareset, or failedset. If you want to format a disk drive that was previously configured as one of these things, you must first delete the unit number and storageset name associated with it.
- If the power fails or the bus is reset while HSUTIL is formatting a disk drive, the drive may become unusable. To minimize this possibility, DIGITAL recommends you use a reliable power source and suspend all non-HSUTIL activity to the bus that services the target disk drive.
- HSUTIL cannot control or affect the defect management for a disk drive.
- Do not invoke any CLI command or run any local program that might reference the target disk drive while HSUTIL is active. Also, do not reinitialize either controller in the dual-redundant configuration while using HSUTIL.

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

Caution Suspend all I/O to the buses that service the target disk drives before using HSUTIL to format the drives.

---

1. Start HSUTIL by issuing the following command:

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

   HSUTIL finds and displays all of the unattached disk drives configured on the controller.
3. Type the name of a disk drive you want to format.

4. Enter Y to enter another disk drive name or N to begin the formatting operation.
5. Read the cautionary information that HSUTIL displays, then confirm or cancel the formatting operation.

   The formatting operation will complete in approximately the estimated time.

Example 5–1

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

```
CLI> RUN HSUTIL

*** AVAILABLE FUNCTIONS ARE:
0. EXIT
1. FORMAT
2. DEVICE_CODE_LOAD_DISK
3. DEVICE_CODE_LOAD_TAPE
ENTER FUNCTION NUMBER (0:3) [0] ? 1

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

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

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

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

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

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

NOTE:
  IN ORDER TO MINIMIZE THE POSSIBILITY OF A SCSI BUS
  RESET, IT IS RECOMMENDED THAT YOU PREVENT NON-HSUTIL IO
  OPERATIONS TO ALL OTHER DEVICES ON THE SAME PORT AS THE
```

```
  DESTINATION DEVICE(S). IF A SCSI BUS RESET OCCURS, THE
  FORMAT MAY BE INCOMPLETE AND YOU MAY HAVE TO RE-INVOKE
  HSUTIL.

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

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

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

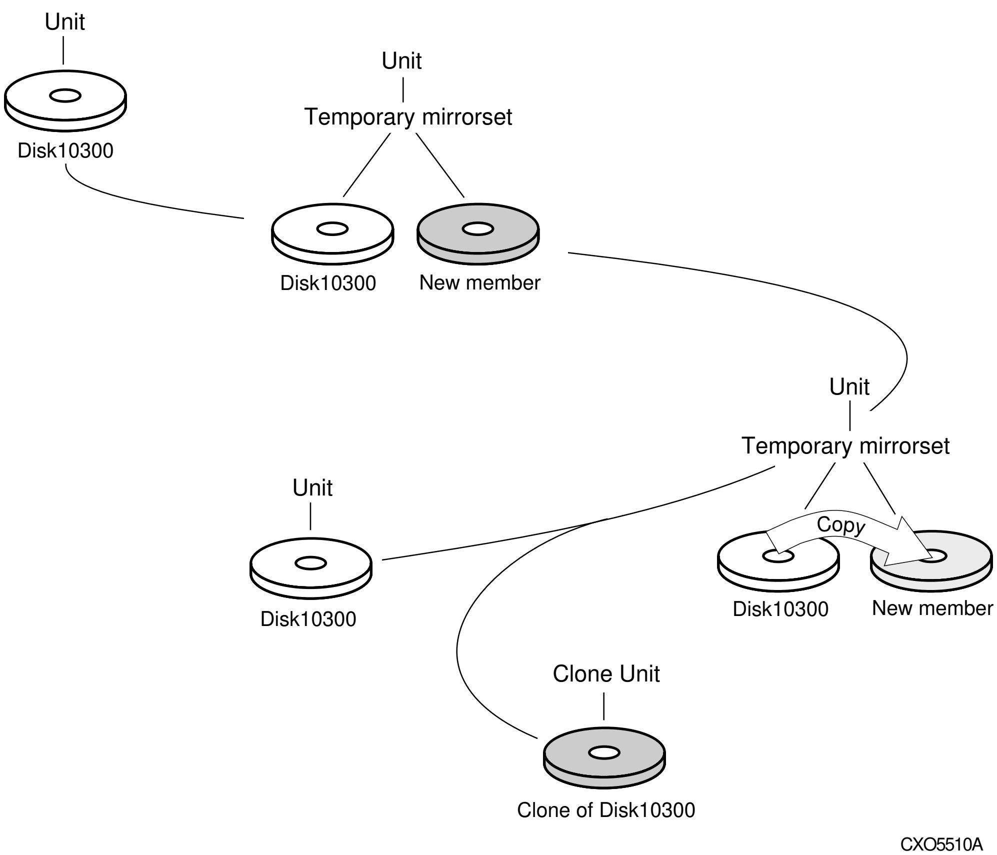
Cloning Data for Backup

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 5–1 to duplicate each member of a unit.

Figure 5–1 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 5–2

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

```
RUN CLONE

CLONE LOCAL PROGRAM INVOKED

UNITS AVAILABLE FOR CLONING:101
                            204

ENTER UNIT TO CLONE ? 204

CLONE WILL CREATE A NEW UNIT WHICH IS A COPY OF UNIT 204.

ENTER THE UNIT NUMBER WHICH YOU WANT ASSIGNED TO THE
NEW UNIT ? 205

THE NEW UNIT MAY BE ADDED USING ONE OF THE FOLLOWING
METHODS:

1. CLONE WILL PAUSE AFTER ALL MEMBERS HAVE BEEN COPIED. THE
USER MUST THEN PRESS RETURN TO CAUSE THE NEW UNIT TO BE
ADDED.

2. AFTER ALL MEMBERS HAVE BEEN COPIED, THE UNIT WILL BE ADDED
AUTOMATICALLY.

UNDER WHICH ABOVE METHOD SHOULD THE NEW UNIT BE
ADDED[]?1

DEVICES AVAILABLE FOR CLONE TARGETS:

DISK20200 (SIZE=832317)

DISK20400 (SIZE=832317)

DISK30100 (SIZE=832317)

USE AVAILABLE DEVICE DISK20200(SIZE=832317) FOR MEMBER
DISK10300(SIZE=832317) (Y,N) [Y] ? Y

MIRROR DISK10300 C_MA

SET C_MA NOPOLICY

SET C_MA MEMBERS=2

SET C_MA REPLACE=DISK220

DEVICES AVAILABLE FOR CLONE TARGETS:

DISK20400 (SIZE=832317)

DISK30100 (SIZE=832317)
```

USE AVAILABLE DEVICE DISK20400(SIZE=832317) FOR MEMBER DISK20000(SIZE=832317) (Y,N) [Y] ? Y

MIRROR DISK20000 C_MB

SET C_MB NOPOLICY

SET C_MB MEMBERS=2

SET C_MB REPLACE=DISK20400

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

.

.

COPY FROM DISK10300 TO DISK20200 IS 100% COMPLETE

COPY FROM DISK20000 TO DISK20400 IS 100% COMPLETE

PRESS RETURN WHEN YOU WANT THE NEW UNIT TO BE CREATED

REDUCE DISK20200 DISK20400

UNMIRROR DISK10300

UNMIRROR DISK20000

ADD MIRRORSET C_MA DISK20200

ADD MIRRORSET C_MB DISK20400

ADD STRIPESET C_ST1 C_MA C_MB

INIT C_ST1 NODESTROY CHUNK=128

ADD UNIT D205 C_ST1

D205 HAS BEEN CREATED. IT IS A CLONE OF D204.

CLONE - NORMAL TERMINATION

Backing Up Your Subsystem Configuration

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

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

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

You cannot use the SAVE_CONFIGURATION switch on TRANSPORTABLE disks.

Saving Subsystem Configuration Information to a Single Disk

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

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

INITIALIZE 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 indicating status and how many devices have copies of the configuration, as shown in the following example.

SHOW THIS_CONTROLLER FULL

```
Controller:
        HSZ70    (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
TAPE40300     passthrough tape             4    3    0        P1107
          DEC      TZS20     (C)DEC 01ab
        Switches:
       TRANSFER_RATE_REQUESTED = 20MHZ (synchronous 10.00 MHZ negotiated)
```

Shutting Down Your Subsystem

Follow these steps to shut down your StorageWorks subsystem:

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

   SHUTDOWN OTHER_CONTROLLER

   SHUTDOWN THIS_CONTROLLER

   Note This process can take up to five minutes to complete depending on the amount of data to be flushed from cache.

   ---

4. Turn off the power to the subsystem.
5. Unplug the subsystem’s power cord.
6. Disable the ECB by pressing its shut off button until its status light stops blinking—about two seconds.

Restarting Your Subsystem

Follow these steps to restart your subsystem:

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

   SHOW THIS_CONTROLLER

APPENDIX A

Controller Specifications

This appendix contains physical, electrical, and environmental specifications for the HSZ70 array controller.

Physical and Electrical Specifications for the Controller

Table A–1 lists the physical and electrical specifications for the controller and cache modules.

Table A–1 Controller Specifications

Hardware	Length	Width	Power	Current at +5 V	Current at +12 V
HSZ70 Array Controller module	12.5 inches	8.75 inches	23.27 W	4.63 A	10 mA
Write-back Cache, 64 MB or 128 MB (Battery charging)	12.5 inches	7.75 inches	2.48 W 8.72 W	400 mA 400 mA	40 mA 560 mA

Voltage measurements in Table A–1 are nominal measurements (at +5 and +12 B). This table does not included tolerances.

Environmental Specifications

The HSZ70 array controller is intended for installation in a Class A computer room environment.

The environmental specifications listed in Table A–2 are the same as for other DIGITAL storage devices.

Table A–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)

APPENDIX B

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
___ Tape Drive
___ Optical disk 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)	___ No (default)	___ 32 blocks (default)
___ No	___ Yes	___ Other:
Availability	**Write Protection**	
___ Run (default)	___ No (default)	
___ NoRun	___ Yes	

Storageset Profile

Type

___ RAIDset	___ Stripeset
___ Mirrorset	___ 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)	___ No (default)	___ 32 blocks (default)
___ No	___ Yes	___ Other:
Availability	**Write Protection**	
___ Run (default)	___ No (default)	
___ NoRun	___ Yes	

BA370 Rack-Mountable Enclosure Template

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

Glossary

This glossary defines terms pertaining to the HSZ70 array controller. It is not a comprehensive glossary of computer terms.

adapter A device that converts the protocol and hardware interface of one bus type into another without changing the function of the bus.

array controller *See* controller.

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.

backplane The electronic printed circuit board into which you plug subsystem devices—for example, the SBB or power supply.

BBR Bad Block Replacement. A replacement routine that substitutes defect-free disk blocks for those found to have defects. This process takes place in the controller, transparent to the host.

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.

cache memory A portion of memory used to accelerate read and write operations.

CDU Cable distribution unit. The power entry device for StorageWorks cabinets. The CDU provides the connections necessary to distribute power to the cabinet shelves and fans.

channel 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.

CLI Command line interpreter. The configuration interface to operate the controller software.

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.

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.

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.

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.

DILX Disk inline exerciser. The controller's diagnostic software used to test the data transfer capabilities of disk drives in a way that simulates a high level of user activity.

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.

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.

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.

DUART Dual universal asynchronous receiver and transmitter. An integrated circuit containing two serial, asynchronous transceiver circuits.

DWZZA A StorageWorks SCSI-bus-signal converter used to connect 8-bit single-ended devices to hosts with 16-bit differential SCSI adapters. This converter extends the range of a single-ended SCSI cable to the limit of a differential SCSI cable. *See also* SCSI bus signal converter.

DWZZB A StorageWorks SCSI bus signal converter used to connect a variety of 16-bit single-ended devices to hosts with 16-bit differential SCSI adapters. *See also* SCSI bus signal converter.

DWZZC The 16-bit SCSI table-top SCSI bus signal converter used to extend a differential SCSI bus, or connect a differential SCSI bus to a single ended SCSI bus. *See also* SCSI bus signal converter.

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.

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.

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.

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.

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.

flush The act of writing dirty data from cache to a storage media.

forced errors A data bit indicating a corresponding logical data block contains unrecoverable data.

FRU Field replaceable unit. A hardware component that can be replaced at the customer's location by DIGITAL service personnel or qualified customer service personnel.

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.

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™ Server.

hot disks A disk containing multiple hot spots. Hot disks occur when the workload is poorly distributed across storage devices and 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 also* 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.

HSOF Hierarchical Storage Operating Firmware. Software contained on a removable ROM program card that provides the operating system for the array controller.

initiator A SCSI device that requests an I/O process to be performed by another SCSI device, namely, the SCSI target. The controller is the initiator on the device bus. The host is the initiator on the host bus.

instance code A four-byte value displayed in most text error messages and issued by the controller when a subsystem error occurs. The instance code indicates when during software processing the error was detected.

I/O 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. *See also* I/O Module.

JBOD Just a bunch of disks. A term used to describe a group of single-device logical units.

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.

LRU
Least recently used. A cache term used to describe the block replacement policy for read cache.

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.

MB/s
Megabytes per second. A unit designation used to measure the SCSI bus bandwidth.

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. Non-transportable 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.

mirrored write-back caching
A method of caching data that maintains two copies of the cached data. The copy is available if either cache module fails.

mirrorset
See RAID level 1.

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.

non-redundant controller configuration
(1) A single controller configuration. (2) A controller configuration which 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—the contents of which survive loss of power.

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's connected to the controller that's serving your current CLI session. *See also* this controller.

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.

port (1) In general terms, a logical channel in a communications system. (2) The hardware and software used to connect a host controller to a communications bus, such as a SCSI bus or serial bus.

Regarding the controller, the port is: (1) the logical route for data in and out of a controller that can contain one or more channels, all of which contain the same type of data. (2) The hardware and software that connects a controller to a SCSI device.

primary cabinet The primary cabinet is the subsystem enclosure that contains the controllers, cache modules, external cache batteries, and the PVA module.

program card The PCMCIA card containing the controller's operating software.

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 is the only RAID level 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. *Contrast* 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. *Contrast* 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.

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.

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* DWZZA, DWZZB, DWZZC, and 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.

signal converter *See* SCSI bus signal converter.

single ended I/O module A 16-bit I/O module. *See also* I/O module.

single-ended SCSI bus An electrical connection where one wire carries the signal and another wire or shield is connected to electrical ground. Each signal's logic level is determined by the voltage of a single wire in relation to ground. This is in contrast to a differential connection where the second wire carries an inverted signal.

spareset A collection of disk drives made ready by the controller to replace failed members of a storageset.

storage array An integrated set of storage devices.

storage array subsystem *See* storage subsystem.

storageset (1) A group of devices configured with RAID techniques to operate as a single container. (2) Any collection of containers, such as stripesets, mirrorsets, striped mirrorsets, and RAIDsets.

storage subsystem The controllers, storage devices, shelves, cables, and power supplies used to form a mass storage subsystem.

storage unit The general term that refers to storagesets, single-disk units, and all other storage devices that are installed in your subsystem and accessed by the host. A storage unit can be any entity that is capable of storing data, whether it is a physical device or a group of physical devices.

StorageWorks A family of DIGITAL modular data storage products which 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.

tape A storage device supporting sequential access to variable sized data records.

Tape Inline Exerciser *See* TILX.

target (1) A SCSI device that performs an operation requested by an initiator. (2) Designates the target identification (ID) number of the device.

target ID number The address a bus initiator uses to connect with a bus target. Each bus target is assigned a unique target address.

this controller The controller that is serving your current CLI session through a local or remote terminal. *See also* other controller.

TILX Tape inline exerciser. The controller's diagnostic software to test the data transfer capabilities of tape drives in a way that simulates a high level of user activity.

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 HSZ term) or via the maintenance port through an intermediary host.

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.

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 there is an opportunity 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

D

E

F

H

I

T

U

V

W