Compaq StorageWorks

HSG80 Array Controller ACS Version 8.5

Configuration Guide

First Edition (October 1999)
Part Number: EK–HSG85-CG. A01 / 165144–001
Compaq Computer Corporation

Notice

HSG80 Array Controller ACS V8.5 Configuration Guide
First Edition October 1999
Part Number EK–HSG85-CG. A01 / 165144–001

Contents

Chapter 1
Planning a Subsystem

Chapter 3

Configuration Procedures for Fabric Subsystems

Chapter 4
Configuration Procedures for Loop Subsystems

Figures

Tables

About This Guide

This book describes:

- things to consider while planning a configuration
- configuration procedures

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

Conventions

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

Special Notices

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

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

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

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

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

Related Publications

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

Table 1 Relevant Publications

Document Title	Part Number
StorageWorks Solutions Software Kit Overview	EK–SOLSR–AA / 166314-001
Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 Maintenance and Service Guide	EK–HSG84–SV. B01 / 118620–002
Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide	EK–HSG85–RG / 165145–001
Compaq StorageWorks Command Console (HSG80) User's Guide	AA–RFA2D–TE / 387405–004
Compaq StorageWorks UltraSCSI RAID Enclosure (BA370-Series) User's Guide	EK–BA370–UG / 387403–001
Compaq StorageWorks Fibre Channel Storage Switch Service Guide	AA–RHBZA–TE / 135268–001

Table 1 Relevant Publications (Continued)

Document Title	Part Number
Compaq StorageWorks Fibre Channel Storage Switch User's Guide	AA–RHBYA–TE / 135267–001
Compaq StorageWorks Fibre Channel Switch Quick Setup Guide	AA–RHC0A–TE / 135269–001
Compaq StorageWorks Fibre Channel Arbitrated Loop Hub (DS-DHGGA-CA) User's Guide	EK–DHGGA–UG
Compaq StorageWorks Data Replication Manager HSG80 ACS Version 8.5P Operations Guide	EK–HSG84–DT / 128519–002
Compaq StorageWorks KGPSA PCI-to-Fibre Channel Host Adapter	EK–KGPSA–UG
The RAIDBOOK—A Source Book for Disk Array Technology	RAID Advisory Board: I.S.B.N 1-879936-90-9
Compaq StorageWorks RA8000/ESA12000 Storage Subsystem User's Guide	EK–SMCPR–UG / 387404–001
Compaq StorageWorks RA8000/ESA12000 HSG80 Solution Software V8.5 Installation Reference Guides:	
for OpenVMS	AA–RH4BB–TE / 387401–002
for TRU64 UNIX	AA–RFAUB–TE / 387389–002
for HP-UX	AA–RFBED–TE / 387374–004
for IBM AIX	AA–RJ25B–TE / 152848–002
for Novell NetWare	AA–RFB9B–TE / 387376–002
for SILICON GRAPHICS IRIX	AA–RFBKC–TE / 387399–003
for Sun Solaris	AA–RFBQE–TE / 387384–005
for Windows NT Server - Alpha	AA–RFB4A–TE / 388206–001
for Windows NT Server - Intel	AA–RFA9C–TE / 387387–003

Revision History

First Release: October 1999

Chapter **1**

Planning a Subsystem

The concepts in this chapter will help you plan the configuration of the subsystem. This chapter is one of two planning chapters; the other planning chapter is Chapter 2, which presents information about what types of storagesets to choose. When you have planned both storage and subsystem, Chapter 3 presents a configuration flowchart and a sample configuration procedure for fabric subsystems; Chapter 4 presents a configuration flowchart and a sample configuration procedure for arbitrated loop subsystems.

This chapter frequently references the command line interface (CLI). For the complete syntax and descriptions of the CLI commands, see the *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide*.

Terminology

The terms A, B, “this controller”, and “other controller” are used to distinguish one controller from another in a two-controller (also called dual-redundant) subsystem. These terms are described in the following sections.

Controller Designations A and B

Controllers and cache modules are designated either A or B depending on their location in the enclosure, as shown in Figure 1–1.

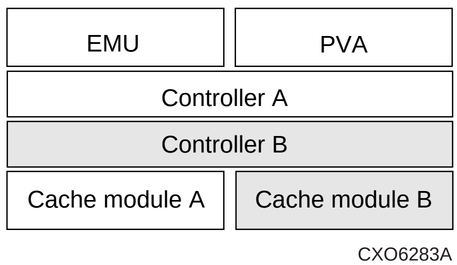


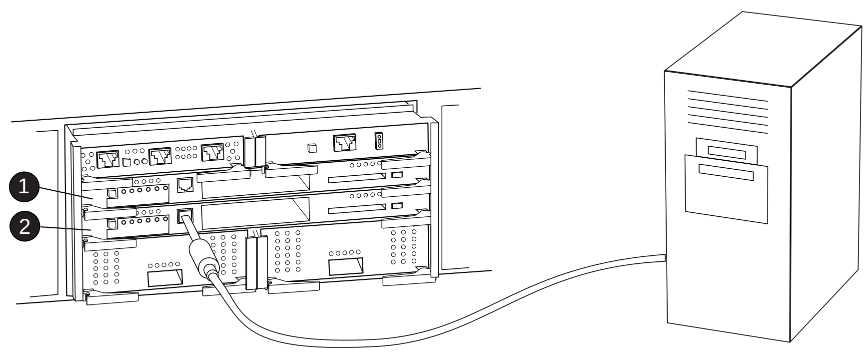
Figure 1–1. Location of controllers and cache modules

Controller Designations “This Controller” and “Other Controller”

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

- “**this controller**”—the controller that is the focus of the CLI session. “This controller” is the controller to which the maintenance terminal is attached and through which the CLI commands are being entered. “This controller”can be shortened to “this” in CLI commands.
- “**other controller**”—the controller that is not the focus of the CLI session and through which CLI commands are not being entered. “Other controller” can be shortened to “other” in CLI commands.

Figure 1–2 shows the relationship between “this controller” and “other controller.”



CXO6468D

❶ Other controller ❷ This controller

Figure 1–2. “This controller” and “other controller”

Selecting a Failover Mode

Failover is a way to keep the storage array available to the host in the event of one controller becoming unresponsive. A controller can become unresponsive due to a hardware failure (such as a failure of a host bus adapter or of the controller) or to a failure of the link between host and controller.Failover keeps the storage array available to the host(s) by allowing the surviving controller to take over total control of the subsystem.

There are two failover modes:

- **transparent**, which is handled by the surviving controller and is transparent (invisible) to the host(s)
- **multiple-bus**, which is handled by the host(s)

Either mode of failover can work with either topology (loop or fabric).

Transparent Failover Mode

Transparent failover mode has the following characteristics:

- hosts do not know failover has taken place
- units are divided between host ports 1 and 2

In transparent failover mode, host port 1 of controller A and host port 1 of controller B must be on the same Fibre Channel link. Likewise, host port 2 of controller A and host port 2 of controller B must be on the same Fibre Channel link. Depending on operating system restrictions and requirements, the port 1 link and the port 2 links can be separate links, or they can be the same link.

At any one time, only one controller has an active port 1and only on controller has an active port 2. The other ports are in standby mode. In normal operation, controller A's port 1 is active and controller B's port 2 is active. A representative configuration is shown in Figure 1–3. The active and standby ports share port identity, enabling the standby port to take over for the active one. If one controller fails, its companion controller (known as the surviving controller) takes over control by making both its host ports active, as shown in Figure 1–4.

Units are divided between the host ports:

- Units 0-99 are on host port 1 of both controllers (but accessible only through the active port)
- Units 100-199 are on host port 2 of both controllers (but accessible only through the active port)

The limitation of transparent failover is that it only compensates for a controller failure, and not for failures of either the Fibre Channel link or host Fibre Channel adapters.

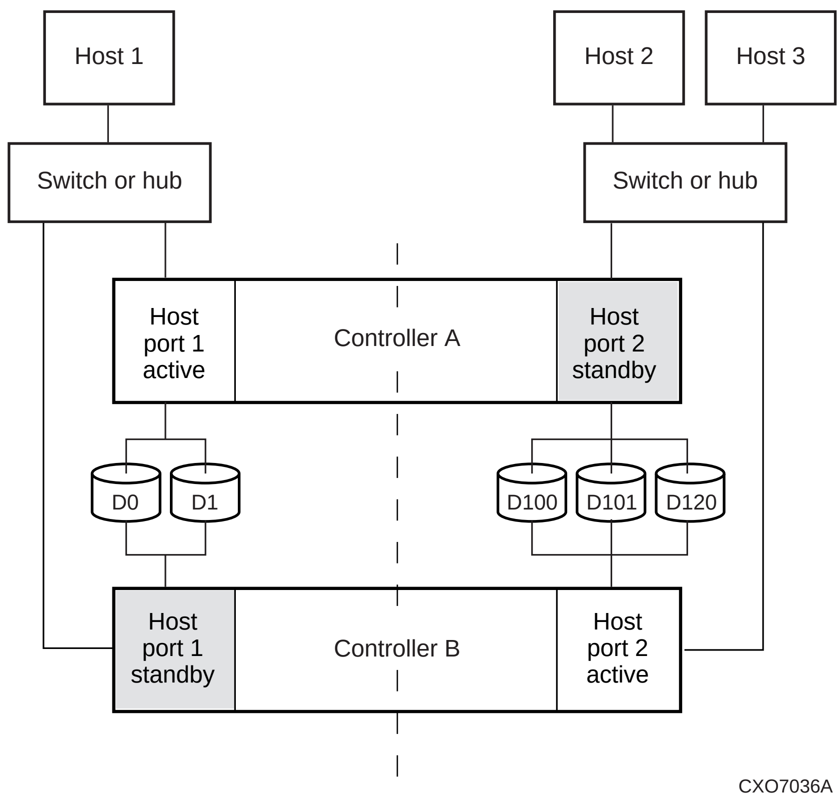


Figure 1–3. Transparent failover - normal operation

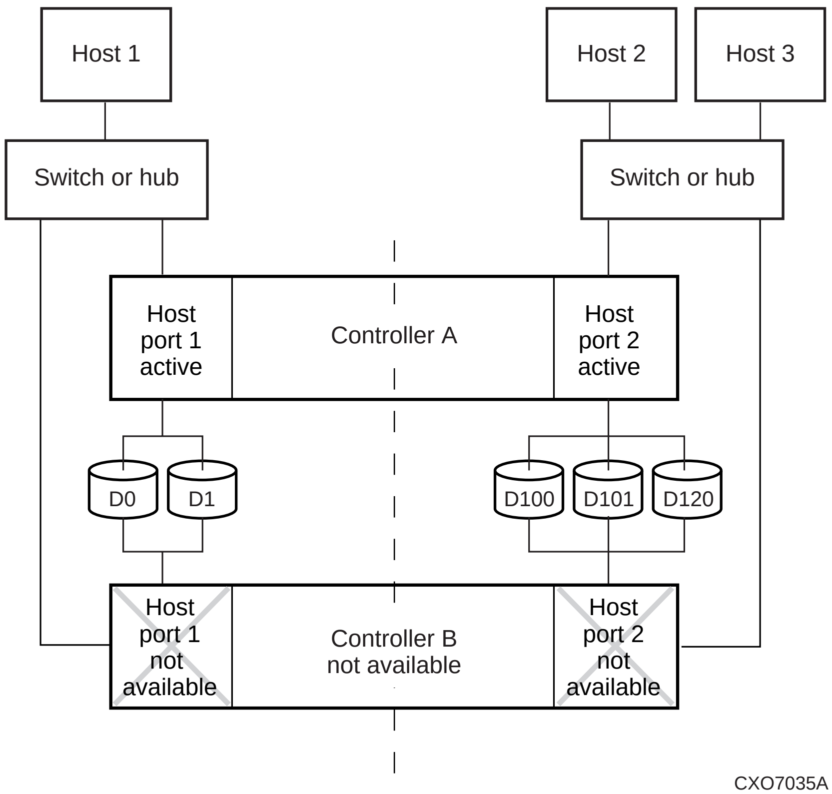


Figure 1–4. Transparent failover - after failover from controller B to controller A

Multiple-Bus Failover Mode

Multiple-bus failover mode has the following characteristics:

- host control the failover process by moving the unit(s) from one controller to another
- all units (0 through 199) are visible at all host ports, but accessible only through one controller at any specific time
- each host has two or more paths to the units

Each host must have special software to control failover. With this software, the host sees the same units visible through two (or more) paths. When one path fails, the host can issue commands to move the units from one path to another. A typical multiple-bus failover configuration is shown in Figure 1–5.

In multiple-bus failover mode, you can specify which units are normally serviced by a specific controller of a controller pair. This process is called preferring or preferment. Units can be preferred to one controller or the other by the *PREFERRED_PATH* switch of the ADD (or SET) UNIT command. For example, use the following command to prefer unit D101 to ‘this controller’:

```
SET D101 PREFERRED_PATH=THIS_CONTROLLER
```

NOTE: This is a temporary, initial preference, which can be overridden by the host(s).

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

- multiple-bus failover can compensate for a failure in any of the following:
  - ❑ controller
  - ❑ switch or hub
  - ❑ Fibre Channel link
  - ❑ host Fibre Channel adapter
- a host can re-distribute the I/O load between the controllers
- all hosts must have operating-system software that supports multiple-bus failover mode

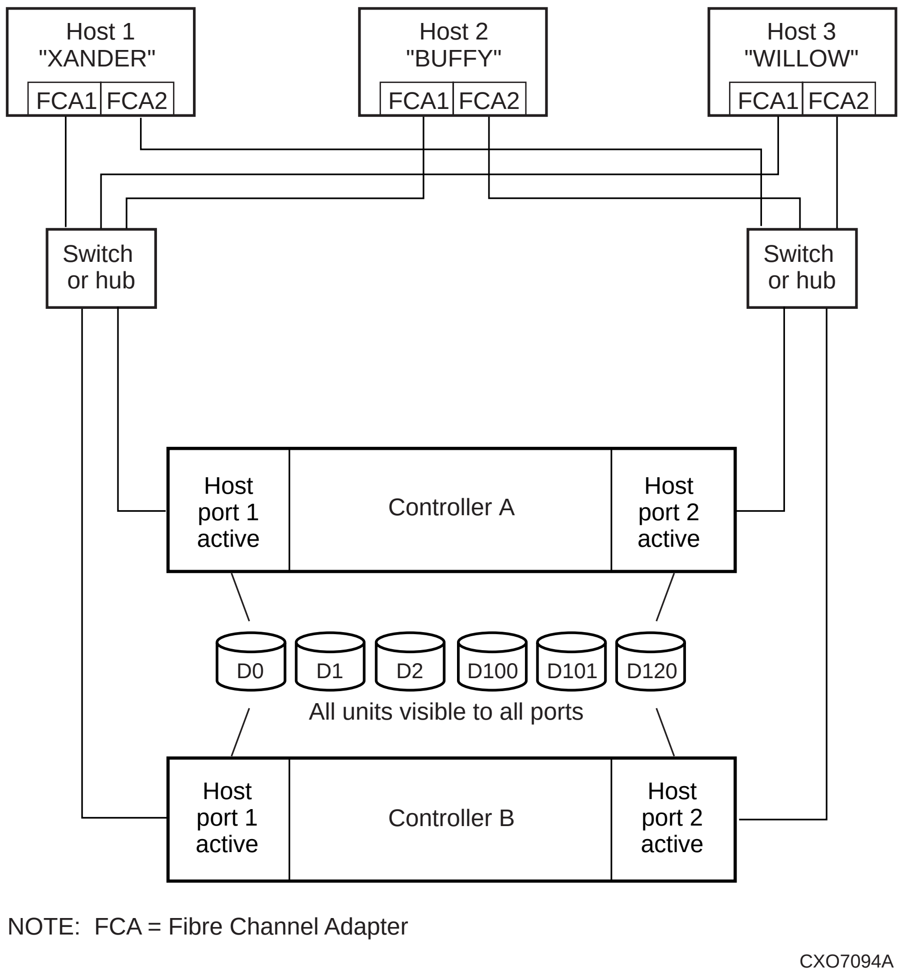


Figure 1–5. Typical multiple-bus configuration

Selecting a Cache Mode

The cache module supports read, read-ahead, write-through, and write-back caching techniques. Cache technique is selected separately for each unit. For example, you can enable only read and write-through caching for some units while enabling only write-back caching for other units.

Read Caching

When the controller receives a read request from the host, it reads the data from the disk drives, delivers it to the host, and stores the data in its cache module. Subsequent reads for the same data will take the data from cache rather than accessing the data from the disks. This process is called read caching.

Read caching can give improved response time to many of the host's read requests. By default, read caching is enabled for all units.

Read-Ahead Caching

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

Write-Back Caching

Write-back caching improves the subsystem's response time to write requests by allowing the controller to declare the write operation complete as soon as the data reaches cache memory. The controller performs the slower operation of writing the data to the disk drives at a later time.

By default, write-back caching is enabled for all units, but only if there is a backup power source for the cache modules (either batteries or an unterruptible power supply).

Write-Through Caching

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

Enabling Mirrored Caching

In mirrored caching, half of each controller’s cache mirrors the companion controller’s cache, as shown in Figure 1–6.

The total memory available for cached data is reduced by half, but the level of protection if greater.

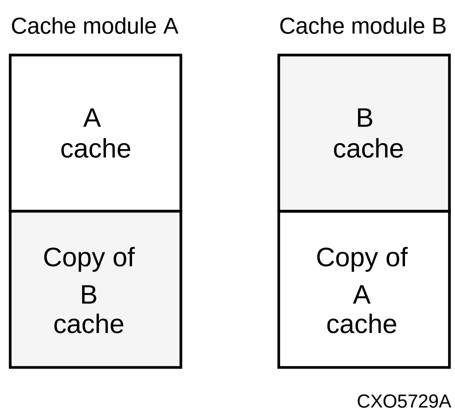


Figure 1–6. Mirrored caching

Before enabling mirrored caching, make sure the following conditions are met:

- both controllers support the same size cache
- diagnostics indicates that both caches are good
- 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

The Command Console LUN

The GUI interface, StorageWorks Command Console (SWCC), needs to see a unit (which from the host's point of view is a logical unit number, or LUN) in order to communicate with the controller. When a subsystem is new and no storage units have yet been configured, a fake LUN needs to be created. This is called the Command Console LUN, or CCL.

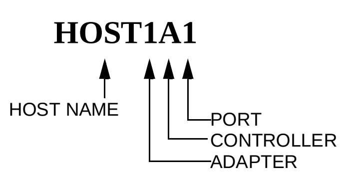
To see the state of the CCL, use the SHOW "this controller"/ "other controller" command. Because the CCL is not an actual LUN, the SHOW UNITS command will not display the CCL location.

Connections

The term "connection" applies to every path between a Fibre Channel adapter in a host computer and an active host port on a controller.

Naming Connections

It is highly recommended to give connections names that has meaning in the context of your particular configuration. One system that works well is to name each connection after its host, its adapter, its controller, and its controller host port, as follows:



Examples:

A connection from the first adapter in host SPGS that goes to port 1 of controller A would be called SPGS1A1.

A connection from the third adapter in host LONDON that goes to port 2 of controller B would be called LONDON3B2.

NOTE: Connection names are can have a maximum of 9 characters.

Numbers of Connections

How many connections result from cabling one adapter into a switch or hub depends on failover mode and how many links the configuration has:

- If a controller pair is in transparent failover mode and the port 1 link is separate from the port 2 link (that is, ports 1 of both controllers are on one loop or fabric, and port 2 of both controllers are on another), each adapter will have one connection, as shown in Figure 1–7.
- If a controller pair is in transparent failover mode and ports 1 and ports 2 are on the same link (that is, all ports are on the same loop or fabric), each adapter will have two connection, as shown in Figure 1–8.
- If a controller pair is in multiple-bus failover mode, each adapter has four connections, as shown in Figure 1–9.

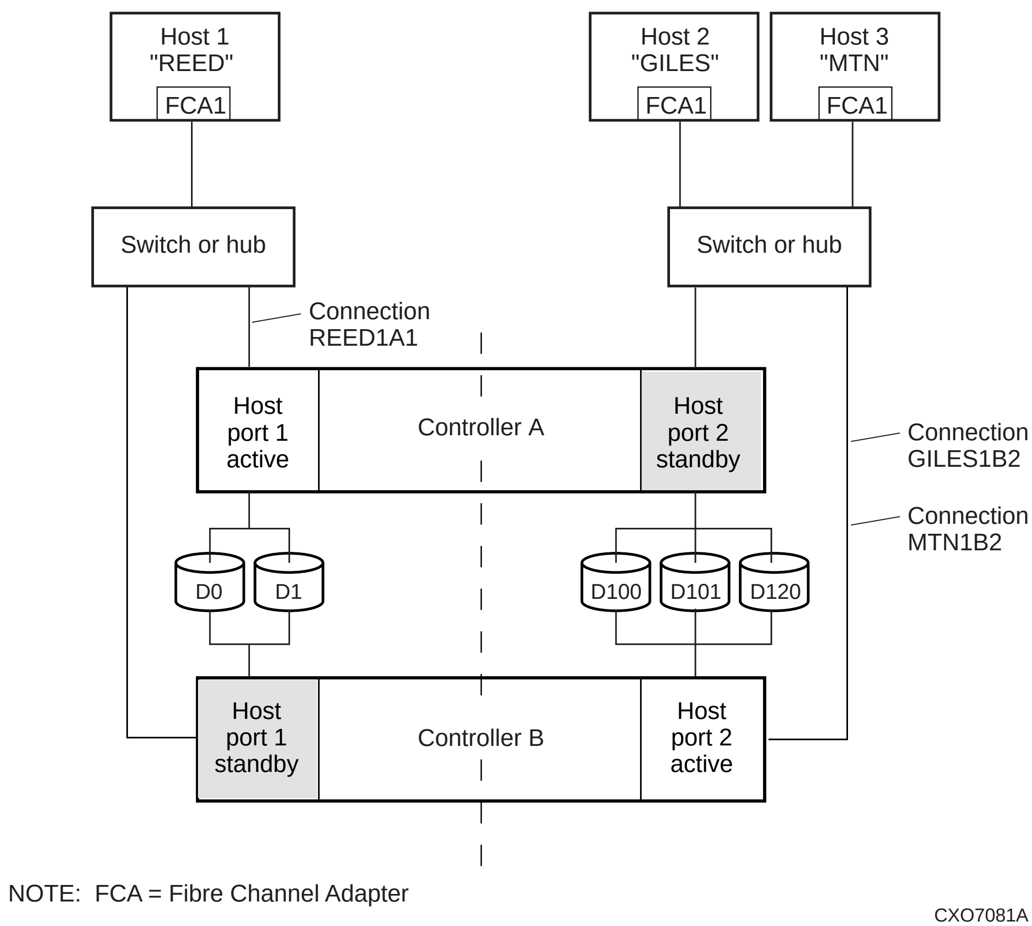


Figure 1–7. Connections in separate-link, transparent failover mode configurations

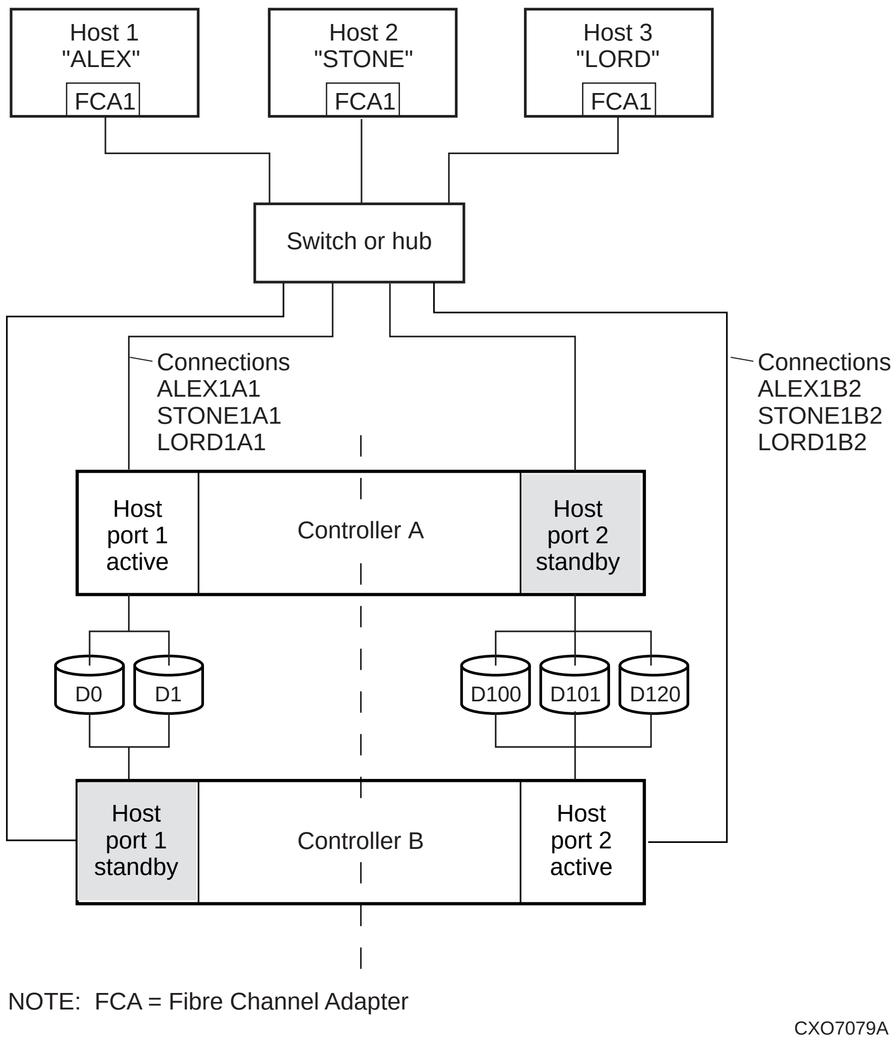


Figure 1–8. Connections in single-link, transparent failover mode configurations

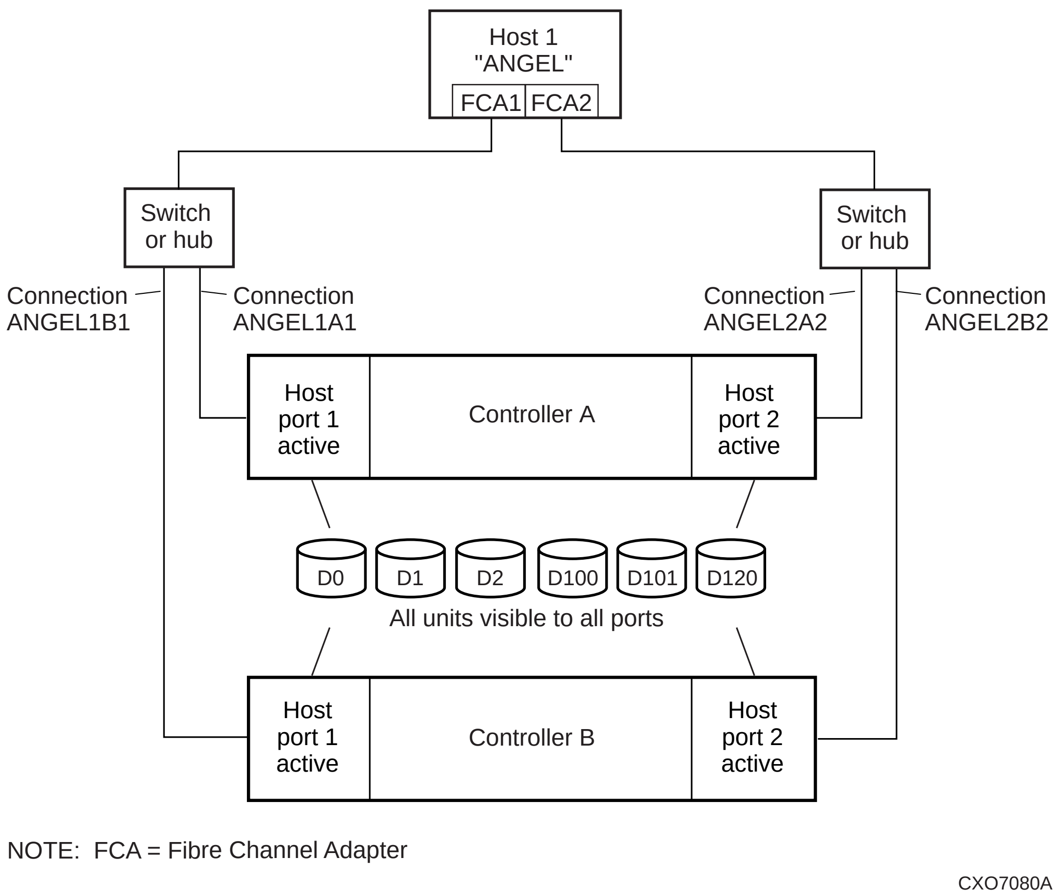


Figure 1–9. Connections in multiple-bus failover mode

Assigning Unit Numbers

The unit number is the designation by which the controller keeps track of the unit. Unit number is a number from 0 - 199 prefixed by a D, which stands for disk drive. A unit can be presented as different LUNs to different connections. The interaction of a unit and a connection is determined by several factors:

- failover mode of the controller pair
- the *ENABLE_ACCESS_PATH* and *PREFERRED_PATH* switches in the ADD UNIT (or SET *unit*) commands
- the *UNIT_OFFSET* switch in the ADD CONNECTIONS (or SET *connections*) commands
- which controller port the connection is attached to
- the *SCSI_VERSION* switch of the SET *this_controller/other_controller* command

The considerations for assigning unit numbers are discussed in the following sections.

Matching Units to Host Connections in Transparent Failover Mode

In transparent failover mode, the ADD UNIT command creates a unit for host connection(s) to access and assigns it to either port 1 of both controllers or to port 2 of both controllers.

Unit numbers are assigned to ports as follows:

- 0 - 99 are assigned to host port 1 of both controllers
- 100 - 199 are assigned to host port 2 of both controllers

For example, unit D2 is on port 1 and unit D102 is on port 2.

What LUN number a host connection assigns to a unit is a function of the *UNIT_OFFSET* switch of the ADD (or SET) CONNECTIONS command. The relationship of offset, LUN number, and unit number is as follows:

LUN number = unit number – offset

- ❑ LUN number is relative to the host (what the host sees the unit as)
- ❑ unit number is relative to the controller (what the controller sees the unit as)

If no value is specified for offset, then connections on port 1 have a default offset of 0 and connections on port 2 have a default offset of 100.

For example, if all host connections use the default offset values, unit D2 will be presented to a port 1 host connection as LUN 2 (unit number of 2 minus offset of 0). Unit 102 will be presented to a port 2 host connection as LUN 2 (unit number of 102 minus offset of 100).

Figure 1–10 shows how units are presented as different LUNs, depending on the offset of the host. In this illustration, host connection 1 and host connection 2 would need to be on host port 1; host connection 3 would need to be on host port 2.

Controller units	Host connection 1 Offset: 0	Host connection 2 Offset: 20	Host connection 3 Offset: 100
D0	LUN 0		
D1	LUN 1		
D2	LUN 2		
D3	LUN 3		
D20	LUN 20	LUN 0	
D21	LUN 21	LUN 1	
D100			LUN 0
D101			LUN 1
D102			LUN 2
D130			LUN 30
D131			LUN 31

CXO6455B

Figure 1–10. LUN presentation to hosts, as determined by offset

Offsets other than the default values can be specified. For example, unit D17 would be visible to a host connection on port 1 that had an offset of 10 as LUN 7 (unit number of 17 minus offset of 10). The unit would not be visible at all to a host connection with a unit offset of 18 or greater, because that offset is not within the units range (unit number of 17 minus offset of 18 is a negative number).

Similarly, unit D127 would be visible to a host connection on port 2 that had an offset of 120 as LUN 7 (unit number of 127 minus offset of 120). The unit would not be visible at all to a host connection with a unit offset of 128 or greater, because that offset is not within the units range (unit number of 127 minus offset of 128 is a negative number).

An additional factor to consider when assigning unit numbers and offsets is SCSI version. If the *SCSI_VERSION* switch of the SET *this_controller/other_controller* command is set to SCSI-3, the Command Console LUN (CCL) is presented as LUN 0 to every connection, superseding any unit assignments. The interaction between SCSI version and unit numbers is explained further in "Assigning Unit Numbers Depending on SCSI_VERSION" on page 1–19.

In addition, the access path to the host connection must be enabled for the connection to access the unit. See "Restricting Host Access in Transparent Failover Mode," page 1–20

Matching Units to Host Connections in Multiple-bus Failover Mode

In multiple-bus failover mode, the ADD UNIT command creates a unit for host connections to access. All unit numbers (0 through 199) are potentially visible on all four controller ports, but are accessible only to those host connections for which access path is enabled and which have offsets in the unit's range.

What LUN number a host connection assigns to a unit is a function of the *UNIT_OFFSET* switch of the ADD (or SET) CONNECTIONS command. The default offset is 0. The relationship of offset, LUN number, and unit number is as follows:

LUN number = unit number – offset

- ❏ LUN number is relative to the host (what the host sees the unit as)
- ❏ unit number is relative to the controller (what the controller sees the unit as)

For example, unit D17 would be visible to a host connection with an offset of 0 as LUN 17 (unit number of 17 minus offset of 0). The same unit would be visible to a host connection with an offset of 10 as LUN 7 (unit number of 17 minus offset of 10). The unit would not be visible at all to a host connection with a unit offset of 18 or greater, because that offset is not within the units range (unit number of 17 minus offset of 18 is a negative number).

In addition, the access path to the host connection must be enabled for the connection to access the unit. This is done through the *ENABLE_ACCESS_PATH* switch of the ADD UNIT (or SET *unit*) command.

Which controller of a dual-redundant pair initially accesses the unit is determined by the *PREFERRED_PATH* switch of the ADD UNIT (or SET *unit*) command. Initially *PREFERRED_PATH* determines which controller presents the unit as Ready. The other controller presents the unit as Not Ready. Hosts can issue a SCSI Start Unit command to move the unit from one controller to the other.

Assigning Unit Numbers Depending on *SCSI_VERSION*

The *SCSI_VERSION* switch of the SET *this/other controller* command determines how the Command Console LUN (CCL) is presented. There are two choices: SCSI-2 and SCSI-3. The choice for *SCSI_VERSION* effects how certain unit numbers and certain host connection offsets interact.

The CCL in SCSI-3 mode

If *SCSI_VERSION* is set to SCSI-3, the CCL is presented as LUN 0 to all connections. The CCL supersedes any other unit assignment. Therefore, in SCSI-3 mode, a unit that would normally be presented to a connection as LUN 0 is not visible to that connection at all.

The following are recommendations for assigning host connection offsets and unit numbers in SCSI-3 mode:

- Offsets should be divisible by 10 (for consistency and simplicity)
- Unit numbers should not be assigned at connection offsets (to avoid being masked by the CCL at LUN 0)

For example, if a host connection has an offset of 20 and SCSI-3 mode is selected, the connection will see LUNs as follows:

LUN 0 - CCL

LUN 1 - unit 21

LUN 2 - unit 22, etc.

In this example, if a unit 20 is defined, it will be superseded by the CCL and invisible to the connection.

The CCL in SCSI-2 Mode

Some operating systems expect or require a disk unit to be at LUN 0. In this case, it is necessary to specify SCSI-2 mode.

If *SCSI_VERSION* is set to SCSI-2 mode, the CCL floats, moving to the first available LUN location, depending on the configuration.

Recommendations for assigning host connection offsets and unit numbers in SCSI-2 mode are as follows:

- Offsets should be divisible by 10 (for consistency and simplicity)

- Unit numbers should be assigned at connection offsets (so that every host connection has a unit presented at LUN 0)

Table 1-2 summarizes the recommendations for unit assignments based on the *SCSI_VERSION* switch.

Table 1–1 Unit Assignments and *SCSI_VERSION*

SCSI_VERSION	Offset	Unit Asignment	What the connection sees LUN 0 as
SCSI-2	Divisible by 10	At offsets	Unit whose number matches offset
SCSI-3	Divisible by 10	Not at offsets	CCL

Restricting Host Access (Selective Storage Presentation)

In a subsystem that is attached to more than one host or if the host(s) have more than one adapter, it is possible to reserve certain units for the exclusive use by a certain host connections. For a controller pair, the method used to restrict host access depends on which failover mode the controllers are in (transparent or multiple-bus). For a single controller, the methods are the same as for a controller pair in transparent failover.

Restricting Host Access in Transparent Failover Mode

There are three methods used to restrict host access to storage units in transparent failover mode:

1. using separate Fibre Channel links (either loop or fabric)
2. enabling the access path of selected host connections on a shared loop or fabric
3. setting offsets

These techniques also work for a single controller.

Restricting Host Access by Separate Links

In transparent failover mode, host port 1 of controller A and host port 1 of controller B share a common Fibre Channel link; likewise, host port 2 of controller A and host port 2 of controller B share a common Fibre Channel link. If the host 1 link is separate from the host 2 link, the simplest way of limiting host access is to have one host or set of hosts on the port 1 link, and another host or set of hosts on the port 2 link. Each host can then see only units assigned to its respective controller port. This separation of host buses is shown in Figure 1–11.

NOTE: This method applies only if the host 1 link and host 2 link are separate links.

NOTE: It is highly recommended that you still give access to only specific connections. That way, if new connections are added, they will not have automatic access to all units. See “Restricting Host Access by Disabling Access Paths,” page 1–22

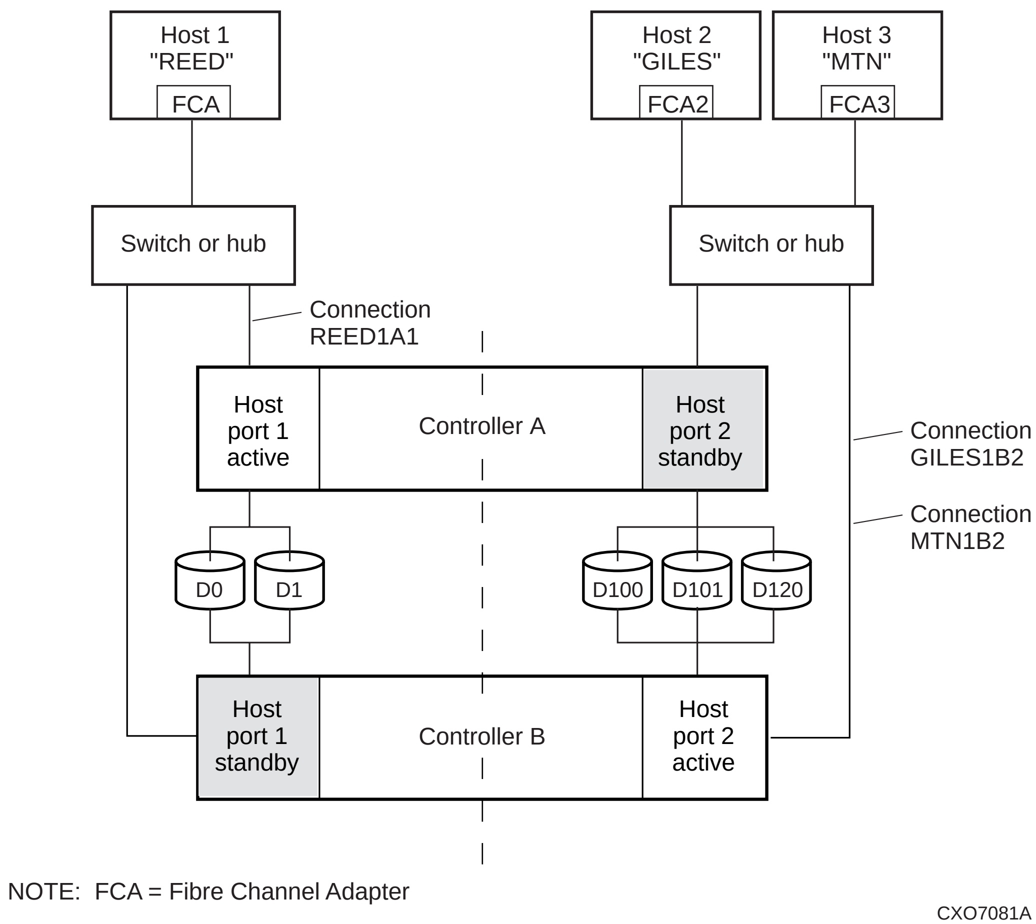


Figure 1–11. Limiting host access In transparent failover mode

Restricting Host Access by Disabling Access Paths

If more than one host is on a link (that is, attached to the same port), host access can be limited by enabling the access of certain host connections and disabling the access of others. This is done through the *ENABLE_ACCESS_PATH* and *DISABLE_ACCESS_PATH* switches of the ADD UNIT or SET *unit* commands. Access path is a unit switch, meaning it must be specified for each unit. Default access is for the unit to be accessible to all hosts.

Example:

In Figure 1–11, restricting the access of unit D101 to host 3, the host named MTN, can be done by enabling only the connection to host 3. Enter the following commands:

```
SET D101 DISABLE_ACCESS_PATH=ALL
SET D101 ENABLE_ACCESS_PATH=MTN1B2
```

CAUTION: If the storage subsystem has more than one host connection, access path must be specified carefully to avoid giving undesirable host connections access to the unit. The default condition for a unit is that access paths to all host connections are enabled. To restrict host access to a set of host connections, specify *DISABLE_ACCESS_PATH=ALL* for the unit, then specify the set of host connections that are to have access to the unit.

Enabling the access path to a particular host connection does not override previously enabled access paths. All access paths previously enabled are still valid; the new host connection is simply added to the list of connections that can access the unit.

The procedure of restricting access by enabling all access paths then disabling selected paths is particularly not recommended because of the potential data/security breach that occurs when a new host connection is added.

Restricting Host Access by Offsets

Offsets establish the start of the range of units that a host connection can access.

Example:

In Figure 1–11, assume both host connections on port 2 (connections GILES1B2 and MTN1B2) initially have the default port 2 offset of 100. Setting the offset of connection MTN1B2 to 120 will present unit D120 to host MTN as LUN 0.

```
SET MTN1B2 UNIT_OFFSET=120
```

Host MTN cannot see units lower than its offset, so it cannot access units D100 and D101. However, host GILES can still access D120 as LUN 20 if the operating system permits. To restrict access of D120 to only host MTN, enable only host MTN's access, as follows:

```
SET D120 DISABLE_ACCESS_PATH=ALL
SET D120 ENABLE_ACCESS_PATH=MTN1B2
```

NOTE: It is highly recommended that you still give access to only specific connections, even if there is just one connection on the link. That way, if new connections are added, they will not have automatic access to all units.

Restricting Host Access in Multiple-Bus Failover Mode

In multiple-bus mode, the units assigned to any port are visible to all ports.

There are two ways to limit host access in multiple-bus failover mode:

- Enabling the access path of selected host connections
- Setting offsets

Restricting Host Access by Disabling Access Paths

Host access can be limited by enabling the access of certain host connections and disabling the access of others. This is done through the *ENABLE_ACCESS_PATH* and *DISABLE_ACCESS_PATH* switches of the ADD UNIT or SET *unit* commands. Access path is a unit switch, meaning it must be specified for each unit. Default access is for the unit to be accessible to all hosts. Also, it is important to remember that at least two paths between the unit and the host must be enabled in order for multiple-bus failover to work.

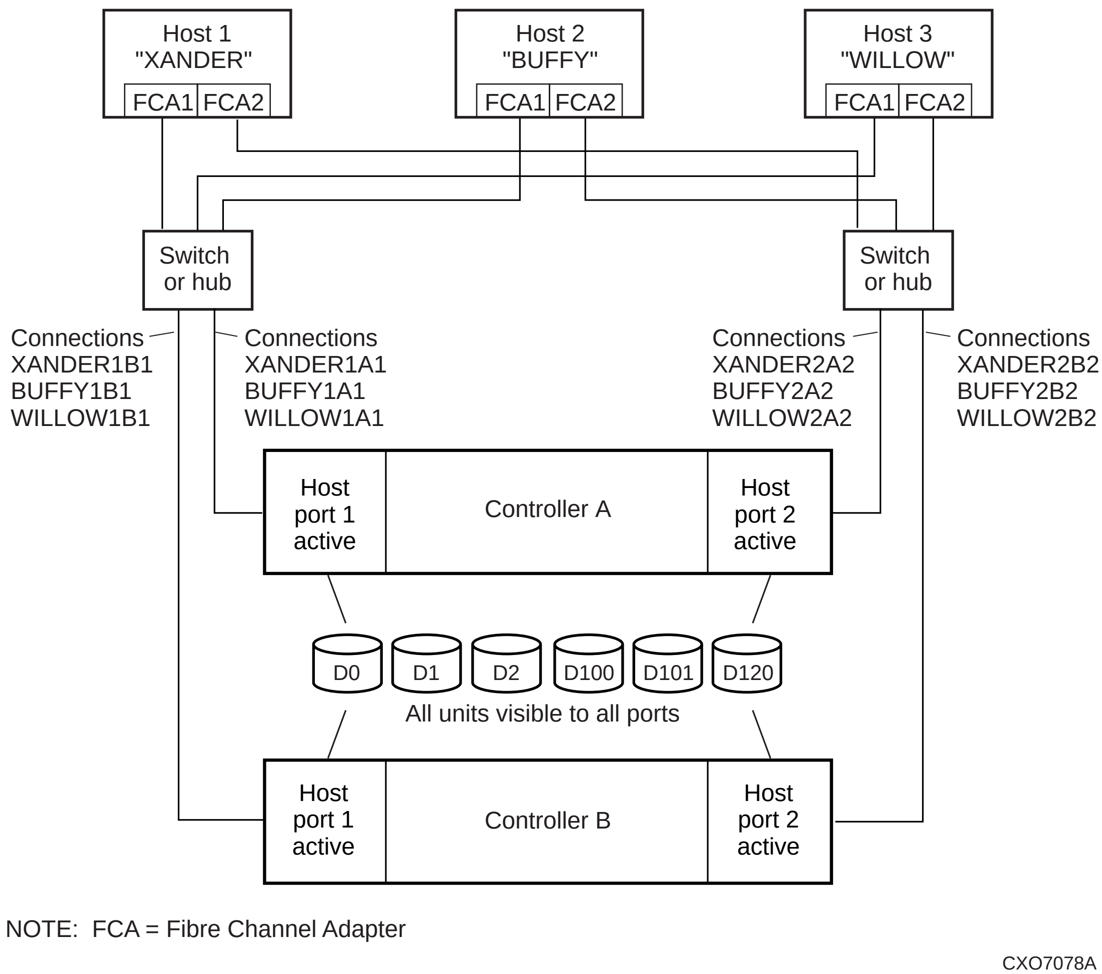


Figure 1–12. Limiting host access in multiple-bus failover mode

Example:

Figure 1–12 shows a representative multiple-bus failover configuration. Restricting the access of unit D101 to host WILLOW can be done by enabling only the connections to host WILLOW. At least two connections must be enabled for multiple-bus failover to work. For most operating systems, it is desirable to have all connections to the host enabled. To enable all connections for host WILLOW, enter the following commands:

```
SET D101 DISABLE_ACCESS_PATH=ALL
SET D101 ENABLE_ACCESS_PATH=(WILLOW1A1,WILLOW1B1,WILLOW12A2,WILLOW2B2)
```

To enable only two connections for host WILLOW (if it is a restriction of the operating system), select two connections that use different adapters, different switches or hubs, and different controllers:

```
SET D101 DISABLE_ACCESS_PATH=ALL
SET D101 ENABLE_ACCESS_PATH=(WILLOW1A1,WILLOW2B2)
```

or

```
SET D101 DISABLE_ACCESS_PATH=ALL
SET D101 ENABLE_ACCESS_PATH=(WILLOW1B1,WILLOW2A2)
```

CAUTION: If the storage subsystem has more than one host connection, access path must be specified carefully to avoid giving undesirable host connections access to the unit. The default condition for a unit is that access paths to all host connections are enabled. To restrict host access to a set of host connections, specify *DISABLE_ACCESS_PATH=ALL* when the unit is added, then use the SET *unit* command to specify the set of host connections that are to have access to the unit.

Enabling the access path to a particular host connection does not override previously enabled access paths. All access paths previously enabled are still valid; the new host connection is simply added to the list of connections that can access the unit.

The procedure of restricting access by enabling all access paths then disabling selected paths is particularly not recommended because of the potential data/security breach that occurs when a new host connection is added.

Restricting Host Access by Offsets

Offsets establish the start of the range of units that a host connection can access. However, depending on the operating system, hosts that have lower offsets may be able to access the units in the specified range.

NOTE: All host connections to the same host computer must be set to the same offset.

Example:

In Figure 1–12, assume all host connections initially have the default offset of 0. Giving all connections to host WILLOW an offset of 120 will present unit D120 to host WILLOW as LUN 0. Enter the following commands:

```
SET WILLOW1A1 UNIT_OFFSET=120
SET WILLOW1B1 UNIT_OFFSET=120
SET WILLOW2A2 UNIT_OFFSET=120
SET WILLOW2B2 UNIT_OFFSET=120
```

Host WILLOW cannot see units lower than its offset, so it cannot access any other units. However, the other two hosts can still access D120 as LUN 20 if their operating system permits. To restrict access of D120 to only host WILLOW, enable only host WILLOW's access, as follows:

```
SET D102 DISABLE_ACCESS_PATH=ALL
SET D102 ENABLE_ACCESS_PATH=(WILLOW1A1,WILLOW1B1,WILLOW12A2,WILLOW2B2)
```

NOTE: It is highly recommended that you always give access to only specific connections. That way, if new connections are added, they will not have automatic access to all units. See "Restricting Host Access by Disabling Access Paths," page 1–24

Worldwide Names (Node IDs and Port IDs)

A worldwide name—also called a node ID—is a unique, 64-bit number assigned to a subsystem prior to shipping. The node ID belongs to the subsystem itself and never changes.

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

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

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

In multiple-bus failover mode, each of the host ports has its own port ID:

- Controller B, port 1—worldwide name + 1

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

Use the CLI command, SHOW *this_controller/other_controller* to display the subsystem's worldwide name.

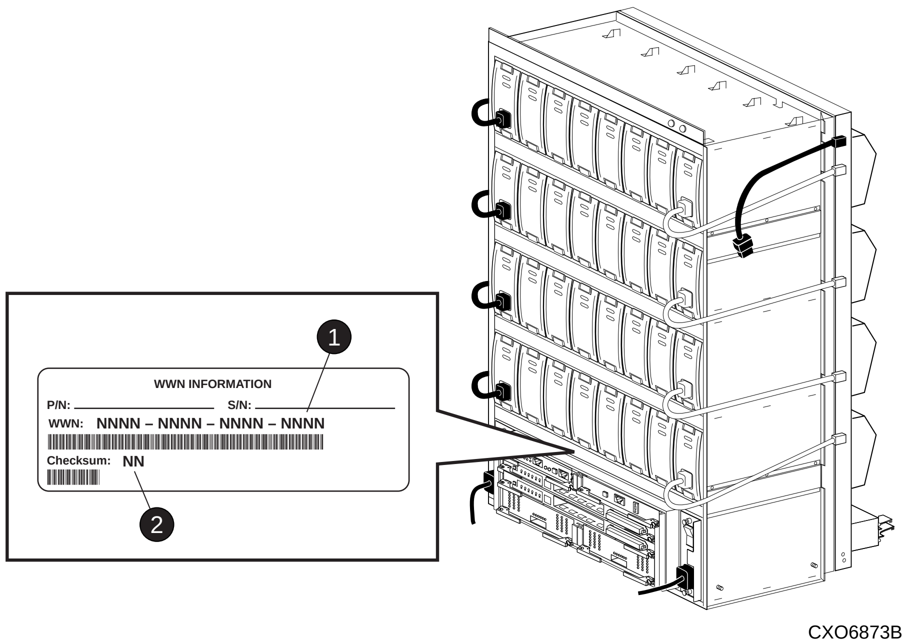
Restoring Worldwide Names (Node IDs)

If a situation occurs that requires you to restore the worldwide name, you can restore it using the worldwide name and check sum printed on the sticker on the frame into which the controller is inserted, as shown in Figure 1–13.

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

Unit World Wide Names (LUN IDs)

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



❶ Node ID (Worldwide name) ❷ Checksum

Figure 1–13. Placement of the worldwide name label in a subsystem

Chapter 2

Planning Storage

This chapter provides information to help you plan the storage configuration of your subsystem. Use the guidelines found in this section to plan the various types of storage containers needed.

The following information is contained in this chapter:

Where to Start....

Containers are defined as individual disk drives (JBOD), storageset types (mirrorsets, stripesets, and so on), and/or partitioned drives. The following is a structure you may follow to plan your storage configuration. The references stated in each step may be used to locate details about specific commands and/or concepts. See Appendix A to locate the blank templates for keeping track of the containers being configured.

1. Review configuration rules. See “Configuration Rules,” page 2–3.
2. Familiarize yourself with the current physical layout of the devices and their addressing scheme. See “Device PTL Addressing Convention,” page 2–3.
3. Determine your storage requirements. Use the questions in “Determining Storage Requirements,” page 2–5 to help you.
4. Choose the type of storage container(s) you need to use in your subsystem. See “Choosing a Container Type,” page 2–6 for a comparison and description of each type of storageset.
5. Create a storageset profile (described in “Creating a Storageset Profile,” page 2–7). Fill out the storageset profile while you read the sections that pertain to your chosen storage type:
   - “Storageset Planning Considerations,” page 2–10
   - “Mirrorset Planning Considerations,” page 2–13
   - “Partition Planning Considerations,” page 2–19
   - “Striped Mirrorset Planning Considerations,” page 2–18
6. Decide which switches you will need for your subsystem. General information on switches is detailed in “Storageset Switches,” page 2–22
   - Determine what unit switches you want for your units (“Unit Switches,” page 2–27)
   - Determine what initialization switches you want for your planned storage containers (“Initialization Switches,” page 2–23).
7. Create a storage map (“Storage Maps,” page 2–28).
8. Configure the storage you have now planned using one of the following methods:
   - Use the StorageWorks Command Console (SWCC) graphical user interface (GUI). See the SWCC documentation for details regarding the use of the Command Console to configure your storage.

- Use the Command Line Interpreter (CLI) commands by way of a terminal or PC connected to the maintenance port of the controller. This method allows you flexibility in defining and naming your storage containers. The *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide* contains the CLI command details.

Configuration Rules

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

- Maximum 128 assignable, host-accessible units
- Maximum 512-GB unit capacity
- Maximum 20 RAID-5 storagesets
- Maximum 30 RAID-5 and RAID-1 storagesets
- Maximum 45 RAID-5, RAID-1, and RAID-0 storagesets
- Maximum 8 partitions per storageset or individual disk
- Maximum 6 members per mirrorset
- Maximum 14 members per RAID-5 storageset
- Maximum 24 members per stripeset
- Maximum 48 physical devices per striped mirrorset

Device PTL Addressing Convention

The controller has six SCSI device ports, each of which connects to a SCSI bus. In dual-controller subsystems, these device buses are shared between the two controllers. (The StorageWorks Command Console GUI calls the device ports “channels.”) The standard BA370 enclosure provides a maximum of 4 SCSI target IDs for each device port. If more target IDs are needed, expansion enclosures can be added to the subsystem.

The controller identifies devices based on a Port-Target-LUN (PTL) numbering scheme, shown in Figure 2–1. The physical location of a device in its enclosure determines its PTL.

- P—Designates the controller's SCSI device port number (1 through 6).
- T—Designates the target identification (ID) number of the device. Valid target ID numbers for a single-controller configuration and dual-redundant controller configuration are 0 - 3 and 8 - 15.

- L—Designates the logical unit (LUN) of the device. For disk devices the LUN is always 0.

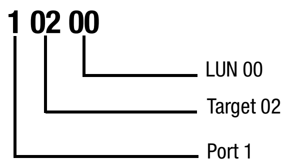


Figure 2–1. PTL naming convention

The controller operates with BA370 enclosures that are assigned ID numbers 0, 2, and 3. These ID numbers are set through the PVA module. Enclosure ID number 1, which assigns devices to targets 4 through 7, is not supported. Figure 2–2 shows the addresses for each device in an extended configuration.

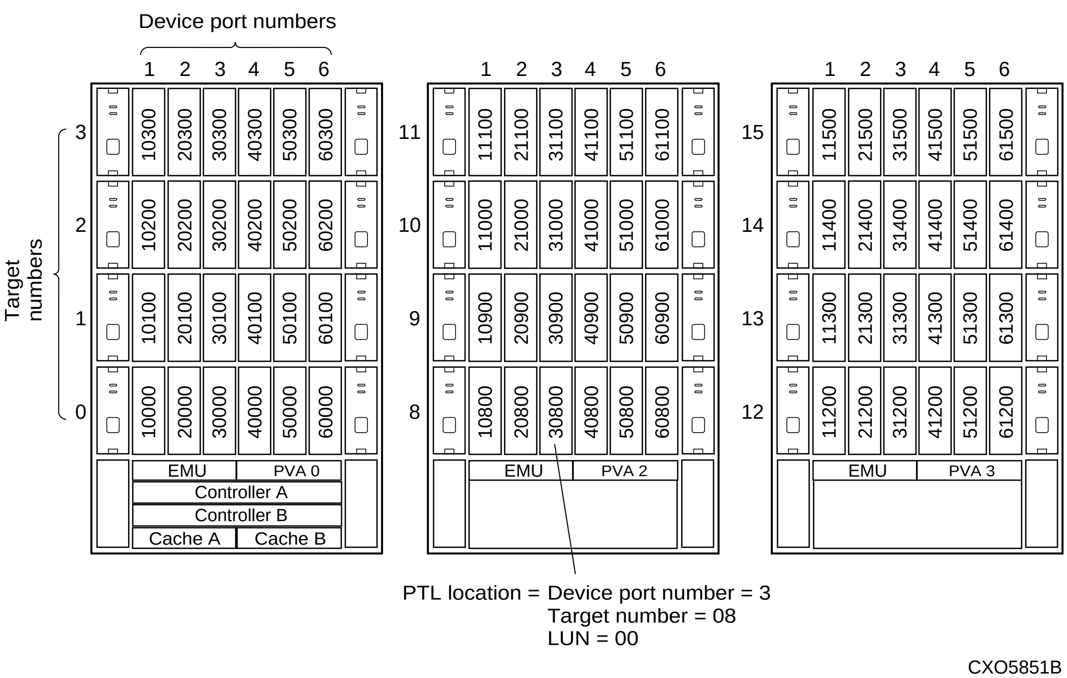


Figure 2–2. PTL addressing in a configuration

Example

In Figure 2–2, the controller addresses DISK30800 through device port 3, target 08, LUN 00. This PTL location indicates the pathway the controller uses to address a disk drive (device) in the subsystem. It also indicates the device name.

When the controller receives an I/O request, it identifies the storageset unit number for the request, then correlates the unit number to the storageset name. From the storageset name, the controller locates the appropriate device(s) for the I/O request. (For example, the RAIDset “RAID1” might contain DISK10000, DISK20000, and DISK30000.) The controller generates the read or write request to the appropriate device(s) using the PTL addressing convention. Figure 2–3 illustrates the concept of mapping unit numbers to PTL addresses.

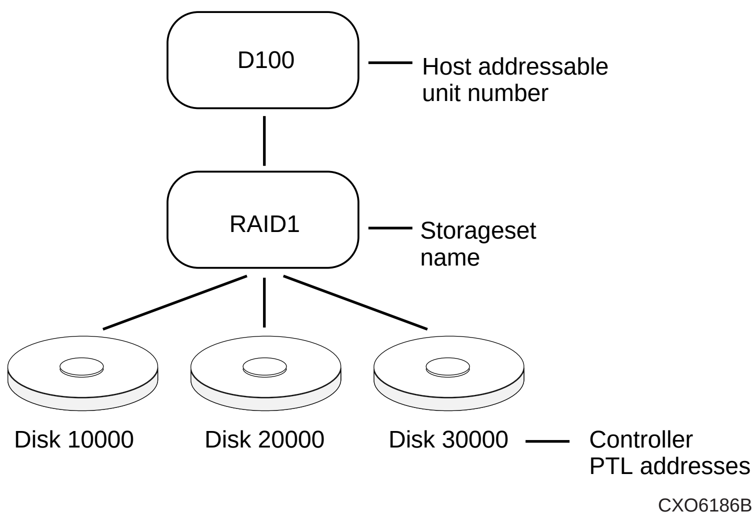


Figure 2–3. Mapping a unit to physical disk drives

Determining Storage Requirements

You cannot adequately plan your subsystem storage without first determining what your storage requirements are. Here are a few of the questions you should ask yourself of the subsystem usage:

- 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?

Choosing a Container Type

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

In choosing a container, you choose between independent disks (JBODs) or one of several storageset types. The independent disks and the selected storageset may also be partitioned.

The storagesets 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.

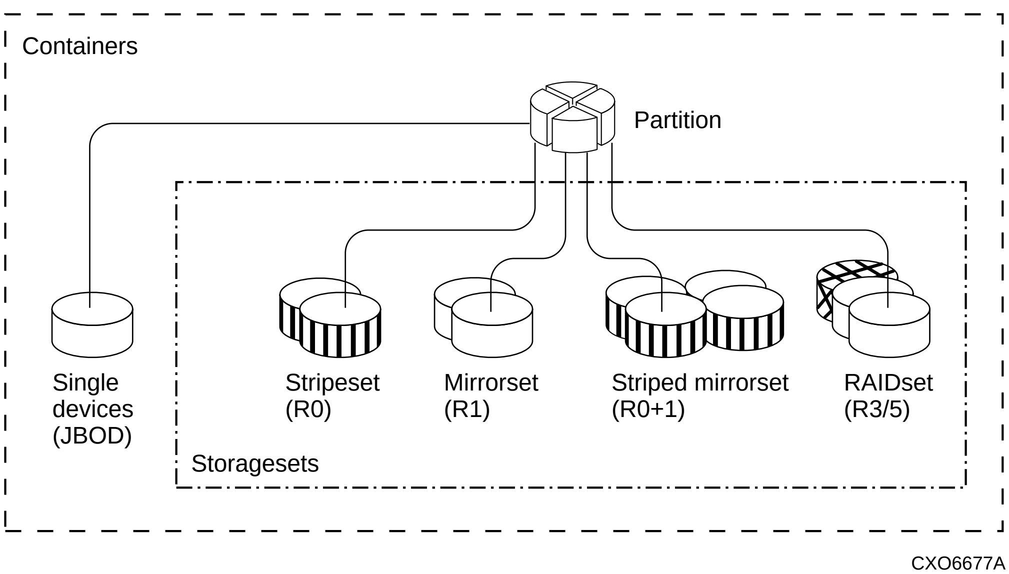


Figure 2–4. Container types

Table 2–1 compares the different kinds of containers to help you determine which ones satisfy your requirements.

Table 2–1 A Comparison of Container Types

Container Name	Relative Availability	Request Rate (Read/Write) I/O per second	Transfer Rate (Read/Write) MB per second	Applications
Independent disk drives (JBOD)	Equal to number of JBOD disk drives	Comparable to single disk drive	Comparable 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/good	Read: excellent (if used with small chunk sizes) Write: good (if used with small chunk sizes)	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.*

Creating a Storageset Profile

Creating a profile for your storagesets, partitions, and devices can simplify the configuration process. Filling out a storageset profile 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.

See the example storageset profile shown in Figure 2–5.

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

Type of Storageset:

_____ Mirrorset	__X_ RAIDset	_____ Stripeset	_____ Striped Mirrorset	____ JBOD

Storageset Name *R1*

Disk Drives *D10300, D20300, D30300, D40300, D50300, D60300*

Unit Number *D101*

Partitions:

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

RAIDset Switches:

Reconstruction Policy	Reduced Membership	Replacement Policy
_X_Normal (default)	_X_No (default)	_X_Best performance (default)
___Fast	___Yes, missing:	___Best fit
		___None

Mirrorset Switches:

Replacement Policy	Copy Policy	Read Source
___Best performance (default)	___Normal (default)	___Least busy (default)
___Best fit	___Fast	___Round robin
___None		___Disk drive:

Initialize Switches:

Chunk size	Save Configuration	Metadata
_X_ Automatic (default)	___No (default)	_X_Destroy (default)
___ 64 blocks	_X_Yes	___Retain
___ 128 blocks		
___ 256 blocks		
___ Other:		

Unit Switches:

Caching	Access by following hosts enabled
Read caching_______X__	_ALL______________________________
Read-ahead caching_____	__________________________________
Write-back caching___X__	__________________________________
Write-through caching____	__________________________________

Figure 2–5. An example storageset profile

Storageset Planning Considerations

This section contains the guidelines for choosing the storageset type needed for your subsystem:

- Stripeset Planning Considerations
- Mirrorset Planning Considerations
- Partition Planning Considerations
- RAIDset Planning Considerations
- Striped Mirrorset Planning Considerations

Stripeset Planning Considerations

Stripesets (RAID 0) 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 simultaneously “striped” across the disk drives in the storageset, thereby allowing several disk drives to participate in one I/O request.

For example, in a three-member stripeset that contains disk drives D10000, D20000, and D30000, the first chunk of an I/O request is written to D10000, the second to D20000, the third to D30000, the fourth to D10000, and so forth until all of the data has been written to the drives (Figure 2–6).

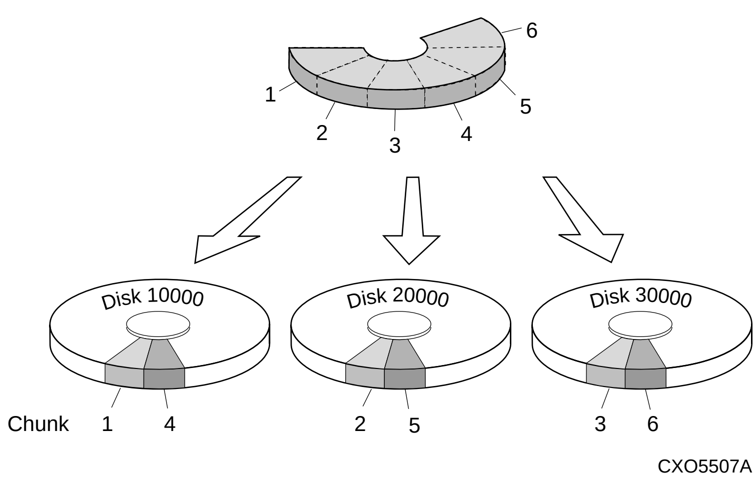


Figure 2–6. A 3-member RAID 0 stripeset (example 1)

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 use the default setting (see “Chunk Size,” page 2–24, for information about setting the chunk size). Figure 2–7 shows another example of a three-member RAID 0 Stripeset.

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

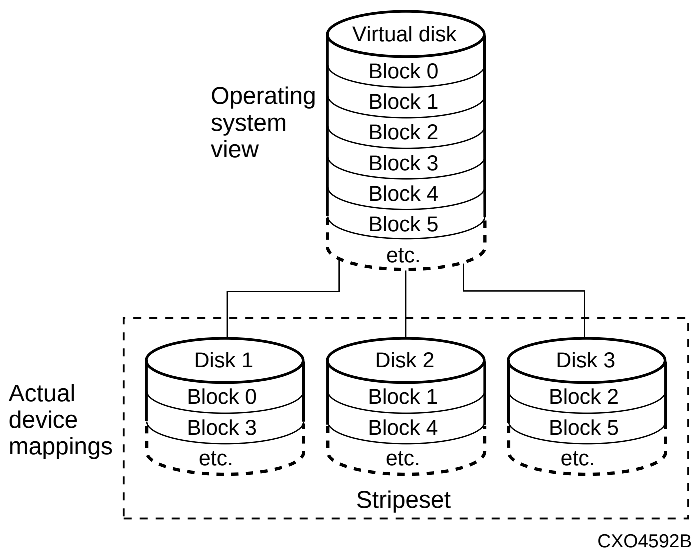


Figure 2–7. A 3-member RAID 0 stripeset (example 2)

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. The release notes that came with your platform kit contain details about these restrictions.
- A storageset should only contain disk drives of the same capacity. The controller limits the effective capacity of each member to the capacity of the smallest member in the storageset (base member size) when the storageset is initialized. Thus, if you combine 9 GB disk drives with 4 GB disk drives in the same storageset, you will waste 5 GB of capacity on each 9 GB member.

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

- Striping does not protect against data loss. In fact, because the failure of one member is equivalent to the failure of the entire stripeset, the likelihood of losing data is higher for a stripeset than for a single disk drive.

  For example, if the mean time between failures (MTBF) for a single disk is l hour, then the MTBF for a stripeset that comprises N such disks is l/N hours. As another example, if the MTBF of a a single disk 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 Figure 2–8.
- Stripesets may contain between two and 24 members.
- Stripesets are well-suited for the following applications:
  - Storing program image libraries or run-time libraries for rapid loading.
  - Storing large tables or other structures of read-only data for rapid application access.
  - Collecting data from external sources at very high data transfer rates.
- Stripesets are not well-suited for the following applications:
  - A storage solution for data that cannot be easily reproduced or for data that must be available for system operation.
  - Applications that make requests for small amounts of sequentially located data.
  - Applications that make synchronous random requests for small amounts of data.

By spreading the traffic evenly across the SCSI buses (I/O device ports), you ensure that no one bus (port) handles the majority of data to the storageset.

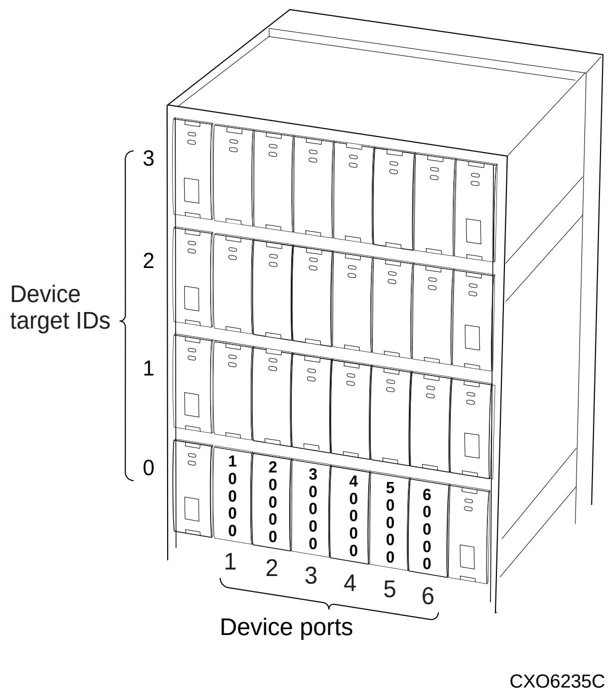


Figure 2–8. Distribute members across device ports

Mirrorset Planning Considerations

Mirrorsets (RAID 1) use redundancy to ensure availability, as illustrated in Figure 2–9. 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 2–10 shows a second example of a Mirrorset.

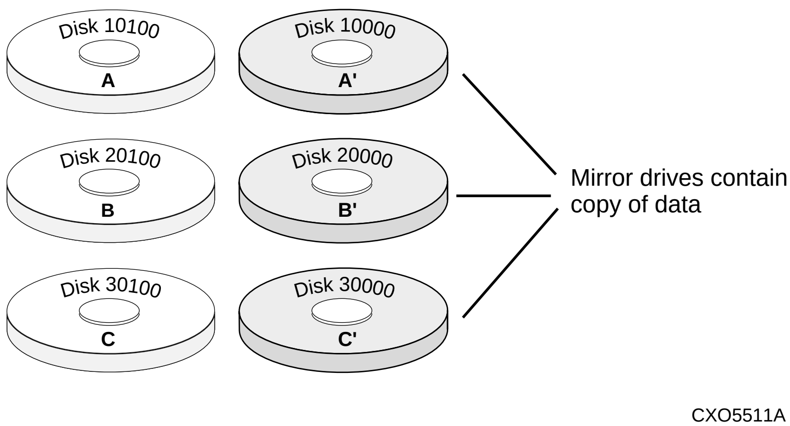


Figure 2–9. Mirrorsets maintain two copies of the same data

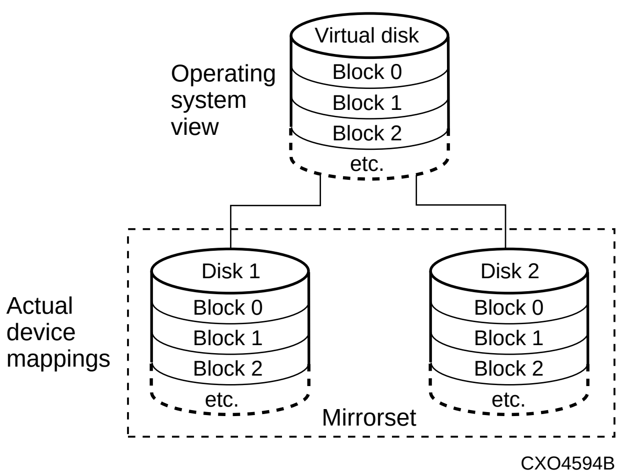
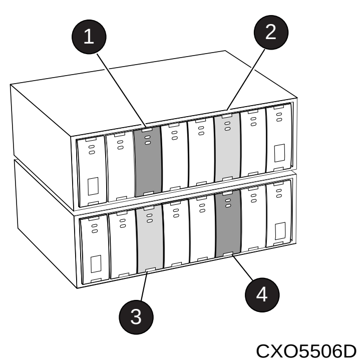


Figure 2–10. Mirrorset example 2

Keep these points in mind as you plan your mirrorsets:

- Data availability with a mirrorset is excellent but comes with a higher cost—you need twice as many disk drives to satisfy a given capacity requirement. If availability is your top priority, consider using dual-redundant controllers and redundant power supplies.
- You can configure up to 20 mirrorsets per controller or pair of dual-redundant controllers. Each mirrorset may contain up to 6 members.
- Both write-back cache modules must be the same size.
- If you are using more than one mirrorset in your subsystem, you should put the first member of each mirrorset on different buses as shown in Figure 2–11. (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.



❶ First member of first mirrorset

❷ First member of second mirrorset

❸ Second member of first mirrorset

❹ Second member of second mirrorset

Figure 2–11. First mirrorset members placed on different buses

- Place mirrorsets and RAIDsets on different I/O ports to minimize risk in the event of a single port failure.
- A storageset should only contain disk drives of the same capacity.
- Evenly distribute the members across the I/O ports to balance load and provide multiple paths as shown in Figure 2–8.
- 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 (a performance hit of 10% will occur)
- ❏ Applications for which cost is a primary issue

RAIDset Planning Considerations

RAIDsets (RAID 3/5) are enhanced stripesets—they use striping to increase I/O performance and distributed-parity data to ensure data availability. Figure 2–12 shows an example of a RAIDset that uses five members.

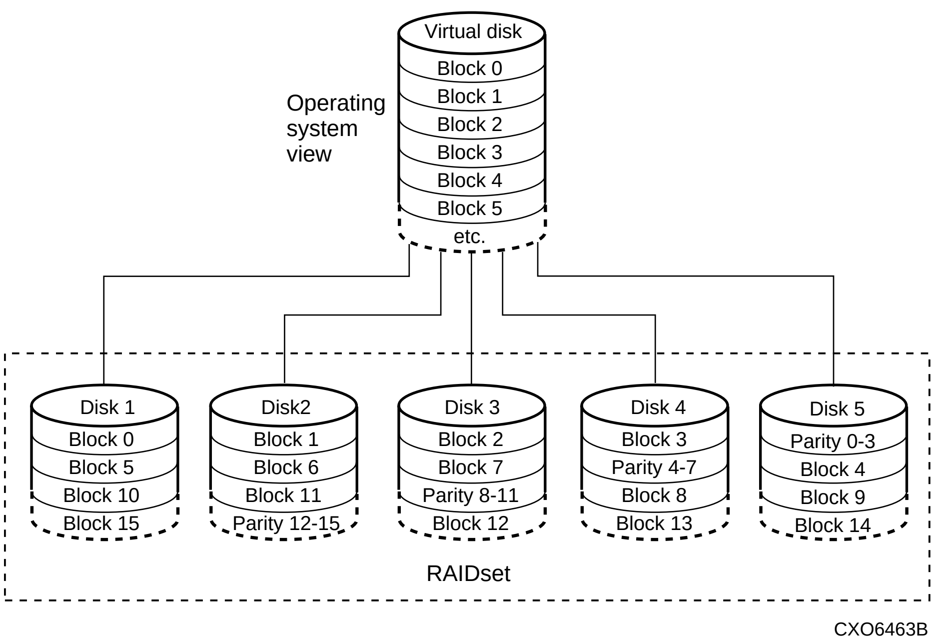


Figure 2–12. A 5-member RAIDset using parity

RAIDsets are similar to stripesets in that the I/O requests are broken into smaller “chunks” and striped across the disk drives. RAIDsets also create chunks of parity data and stripe them across all the members of the RAIDset. This parity data is derived mathematically from the I/O data and enables the controller to reconstruct the I/O data if a single disk drive fails. Thus, it becomes possible to lose a disk drive without losing access to the data it contained. Data could be lost, however, if a second disk drive fails before the controller replaces the first failed disk drive and reconstructs the data.

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 use the default setting. See “Chunk Size,” page 2–24, for information about setting the chunk size.

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 release notes that came with your platform kit for details about these restrictions.
- Both cache modules must be the same size.
- A RAIDset must include at least 3 disk drives, but no more than 14.
- Evenly distribute the members across the device ports to balance the I/O load and provide multiple paths as shown in Figure 2–8 on page 2–13.
- A storageset should only contain disk drives of the same capacity. The controller limits the capacity of each member to the capacity of the smallest member in the storageset. Thus, if you combine 9 GB disk drives with 4 GB disk drives in the same storageset, you will waste 5 GB of capacity on each 9 GB member.
- Place RAIDsets 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
  - Database applications in which fields are continually updated
  - Transaction processing

Striped Mirrorset Planning Considerations

Striped mirrorsets (RAID 0+1) are a configuration of stripesets whose members are also mirrorsets (Figure 2–13). 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 2–14 shows a second example of a striped mirrorset using six members.

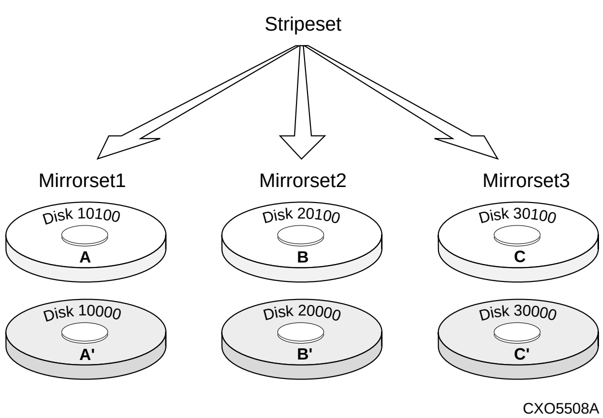


Figure 2–13. Striped mirrorset (example 1)

The failure of a single disk drive has no effect on the ability of the storageset to deliver data to the host. Under normal circumstances, it also 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.

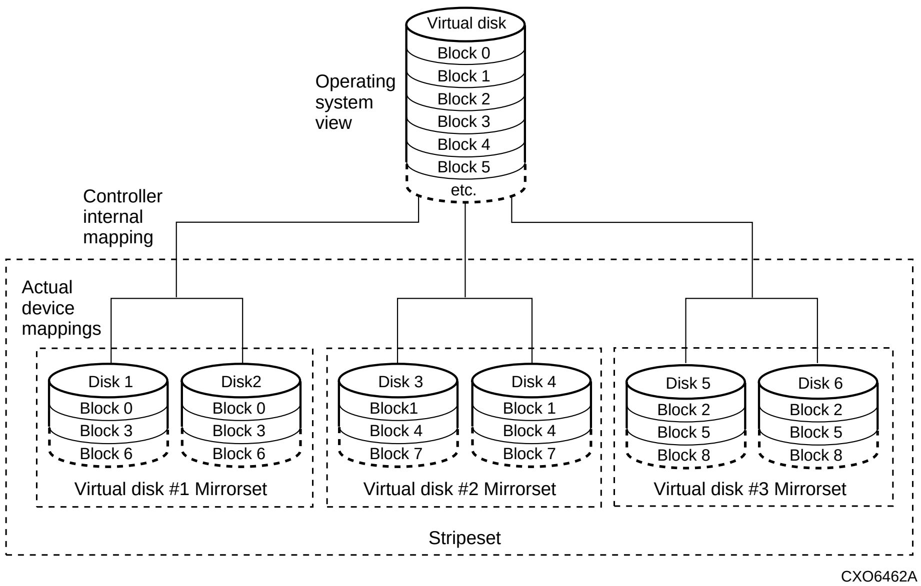


Figure 2–14. Striped mirrorset (example 2)

Plan the mirrorset members, then plan the stripeset that will contain them. Review the recommendations in "Storageset Planning Considerations," page 2–10, and "Mirrorset Planning Considerations," page 2–13.

Partition Planning Considerations

Use partitions to divide a container (storageset or individual disk drive) into smaller pieces, each of which can be presented to the host as its own storage unit. Figure 2–15 shows the conceptual effects of partitioning a single-disk container.

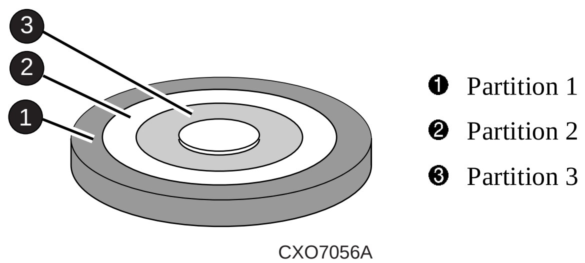


Figure 2–15. One example of a partitioned single-disk unit

You can create up to eight partitions per storageset (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. Partitions are separately addressable storage units, therefore, 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:

- 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.
- RAIDsets and stripesets—the controller allocates the largest whole number of stripes that are less than or equal to the percentage you specify.
  - ❑ Stripesets—the stripe size = chunk size × number of members.
  - ❑ RAIDsets, the stripe size = chunk size × (number of members minus 1)

An unpartitioned storage unit has more capacity than a partition that uses the whole unit because each partition requires a small amount of disk space for metadata.

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.

- In transparent failover mode, all the partitions of a particular container must be on the same host port. Partitions cannot be split across host ports.
- In multiple-bus failover mode, all the partitions of a particular container must be on the same controller. Partitions cannot be split across controllers.
- Partitions cannot be combined into storagesets. For example, you cannot divide a disk drive into three partitions, then combine those partitions into a RAIDset.
- Just as with storagesets, you do not have to assign unit numbers to partitions until you are ready to use them.
- The CLONE utility cannot be used with partitioned mirrorsets or partitioned stripesets.

Changing Characteristics through Switches

CLI command switches allow the user another level of command options. There are three type of switches that modify the storageset and unit characteristics:

- storageset switches
- initialization switches
- unit switches

The following sections:

- Describe how to enable/modify switches
- Contain a description of the major CLI command switches

Enabling Switches

If you use the StorageWorks Command Console (SWCC) to configure the device or storageset, you can set switches from the SWCC screens during the configuration process, and SWCC automatically applies them to the storageset or device. See the online SWCC help for information about using the SWCC.

If you use CLI commands to configure the storageset or device manually, the configuration procedure found in Chapter 4 of this guide indicates when and how to enable each switch. The *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide* contains the details of the CLI command and their switches.

Changing Switches

You can change the RAIDset, mirrorset, device, and unit switches at any time. 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.

CAUTION: Initializing a storageset is similar to formatting a disk drive; all of the data is destroyed during this procedure.

Storageset Switches

The characteristics of a particular storageset can be set by specifying switches when the storageset is added to the controllers' configuration. Once a storageset has been added, the switches can be changed by using a SET command. Switches can be set for the following types of storagesets:

- RAIDset
- Mirrorset
- Partition

Stripesets have no specific switches associated with their ADD and SET commands.

RAIDset Switches

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

- Replacement policy
- Reconstruction policy
- Remove/replace policy

For details on the use of these switches refer to SET RAIDSET and SET *RAIDset-name* commands in the *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide*.

Mirrorset Switches

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

- Replacement policy
- Copy speed
- Read source
- Membership

For details on the use of these switches refer to ADD MIRRORSET and SET *mirrorset-name* commands in the *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide*.

Partition Switches

When you create a partition, you can specify the following switches:

- Size
- Geometry

For details on the use of these switches refer to CREATE_PARTITION command in the *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide*.

Initialization Switches

Initialization switches set characteristics for established storagesets before they are made into units. You can enable the following kinds of switches to affect the format of a disk drive or storageset:

- Chunk Size (for stripesets and RAIDsets only)
- Save Configuration
- Destroy/Nodestroy
- Geometry

Each of these is described in the following sections.

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

Chunk Size

Specify the chunk size of the data to be stored to control the stripesize used in 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.

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 2–16.

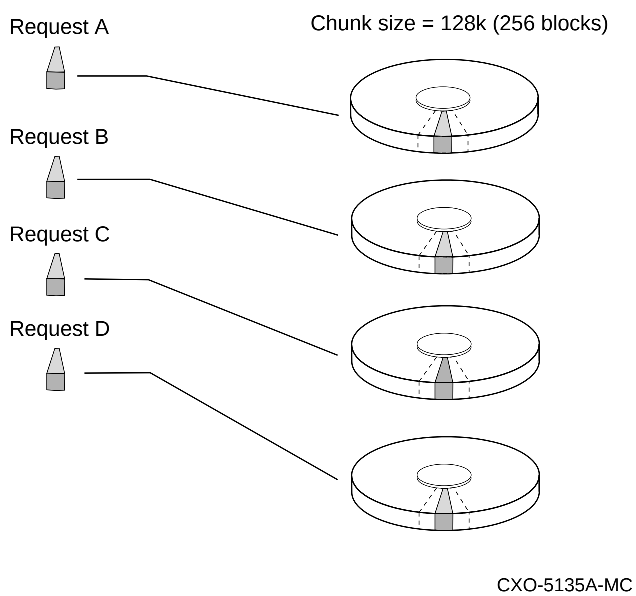


Figure 2–16. Chunk size larger than the request size

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 to the closest prime number. In general, 113 works well for OpenVMS™ systems with a transfer size of 8 sectors.

To calculate the chunk size that should be used for your subsystem, you first must analyze the types of requests that are being made to the subsystem:

- Lots of parallel I/O that use a small area of disk should use a chunk size of 10 times the average transfer request rate.
- Random I/Os that are scattered over all the areas of the disks should use a chunk size of 20 times the average transfer request rate.
- If you don't know, then you should use a chunk size of 15 times the average transfer request rate.
- If you have mostly sequential reads or writes (like those needed to work with large graphic files), then make the chunk size a small number (i.e. 17 sectors).

Table 2–2 shows a few examples of chunk size selection.

Table 2–2 Example Chunk Sizes

Transfer Size (KB)	Small Area of I/O Transfers	Unknown	Random Areas of I/O Transfers
2	41	59	79
4	79	113	163
8	157	239	317

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 2–17.

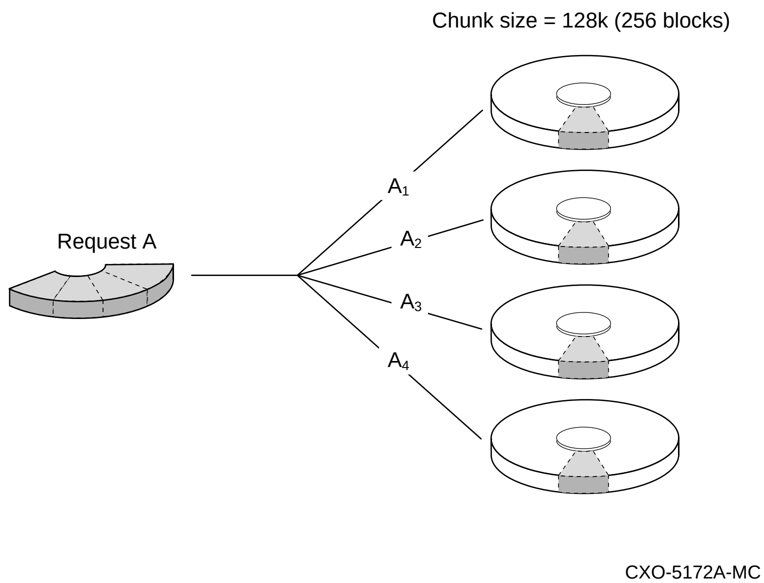


Figure 2–17. Chunk size smaller than the request size

Increasing Sequential Write Performance

Sequential write and read requests on stripesets (or striped mirrorsets), should use a small chunk size relative to the I/O size to increase the performance. A chunk size of 17 generally works well

Save Configuration

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

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

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

Destroy/Nodestroy

You must specify whether to destroy or retain the user data and metadata when a disk is initialized that has been previously used in a mirrorset or as a single-disk unit.

NOTE: The *DESTROY* and *NODESTROY* switches are only valid for mirrorsets and striped 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 *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide* for information on removing disk drives from a mirrorset.

  NODESTROY is ignored for members of a RAIDset.

Geometry

The geometry parameters of a storageset can be specified. The geometry switches are:

- *CAPACITY*—the number of logical blocks. The range is from 1 to the maximum container size.
- *CYLINDERS*—the number of cylinders used. The range is from 1 to 16777215.
- *HEADS*—the number of disk heads used. The range is from 1 to 255.
- *SECTORS_PER_TRACK*—the number of sectors per track us. The range is from 1 to 255.

Unit Switches

There are several switches that control the characteristics of units. The unit switches are described under the SET *unit-number* command in the *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide*.

One unit switch, *ENABLE/DISABLE_ACCESS_PATH*, determines which host connections can access the unit, and is part of the larger topic of matching units to specific hosts. This complex topic is covered in Chapter 1 under the following headings:

- “Assigning Unit Numbers,” page 1–16
- “Restricting Host Access (Selective Storage Presentation),” page 1–20

Storage Maps

Configuring your subsystem will be easier if you know how the storagesets, partitions, and JBODs correspond to the disk drives in your subsystem. You can more easily see this relationship by creating a hardcopy representation (a storage map). Figure is a representative blank storage map showing a simplified physical representation of the enclosure (each cell in the map represents a disk drive in the enclosure). The location of the drive determines the PTL location (Figure 2–19 and Figure 2–2).

Creating a Storage Map

If you want to make a storage map, fill out a blank storage map as you add storagesets, partitions, and JBOD disks to your configuration and assign them unit numbers. Appendix A contains blank templates you may use in the creation of your subsystem storage map. Label each disk drive in the map with the higher levels it is associated with, up to the unit level.

	Port 1	Port 2	Port 3	Port 4	Port 5	Port 6		Targets
Power Supply	D10300	D20300	D30300	D40300	D50300	D60300	Power Supply	3
Power Supply	D10200	D20200	D30200	D40200	D50200	D60200	Power Supply	2
Power Supply	D10100	D20100	D30100	D40100	D50100	D60100	Power Supply	1
Power Supply	D10000	D20000	D30000	D40000	D50000	D60000	Power Supply	0

Figure 2–18. Blank storage map

Example Storage Map

Figure 2–19 is an example of a completed storage map in a single-enclosure subsystem with the following configured storage:

- Unit D100 is a 6-member RAID 3/5 storageset named R1. R1 consists of Disk10000, Disk20000, Disk30000, Disk40000, Disk50000, and Disk60000.
- Unit D101 is a 6-member RAID 3/5 storageset named R2. R2 consists of Disk10100, Disk20100, Disk 30100, Disk40100, Disk50100, and Disk60100.
- Unit D102 is a 2-member striped mirrorset named S1. S1 consists of M1 and M2:
  - M1 is a 2-member striped mirrorset consisting of Disk10200 and Disk20200.
  - M2 is a 2-member mirrorset consisting of Disk30200 and Disk40200.
- Unit D103 is a 2-member mirrorset named M3. M3 consists of Disk50200 and Disk60200.
- Unit D104 is a 4-member stripeset named S2. S2 consists of Disk 10300, Disk20300, Disk30300, and Disk40300.
- Unit D105 is a single (JBOD) disk named Disk50300.
- Disk60300 is an spareset member.

	Port 1	Port 2	Port 3	Port 4	Port 5	Port 6		Targets
Power Supply	D104 S2 D10300	D104 S2 D20300	D104 S2 D30300	S2 D40300	D105 D50300	spareset member D60300	Power Supply	3
Power Supply	D102 S1 MI D10200	D102 S1 M1 D20200	D102 S1 M2 D30200	D102 S1 M2 D40200	D103 M3 D50200	D103 M3 D60200	Power Supply	2
Power Supply	D101 R2 D10100	D101 R2 D20100	D101 R2 D30100	D101 R2 D40100	D101 R2 D50100	D101 R2 D60100	Power Supply	1
Power Supply	D100 R1 D10000	D100 R1 D20000	D100 R1 D30000	D100 R1 D40000	D100 R1 D50000	D100 R1 D60000	Power Supply	0

Figure 2–19. Example storage map

Using the LOCATE Command to Find Devices

If you want to complete a storage map at a later time but do not remember where everything is, use the CLI command LOCATE. The LOCATE command flashes the (fault) LED on the drives associated with the specific storageset or unit. To turn off the flashing LEDs, enter the CLI command LOCATE *cancel*.

The following is an example of the commands needed to locate all the disk drives that make up unit D104:

```
LOCATE D104
```

The LEDs on the disk drives that make up unit D104 flash. After you have noted the position of all the drives contained within D104, enter the following to turn off the flashing LEDs:

```
LOCATE CANCEL
```

The following is an example of the commands needed to locate all the drives that make up RAIDset R1:

```
LOCATE R1
```

After you have noted the position of the drives contained within R1, enter the following to turn off the flashing LEDs:

```
LOCATE CANCEL
```

The Next Step...

Chapter 3 details a procedure for configuring a fabric subsystem. Chapter 4 details a procedure for configuring an arbitrated loop subsystem.

Chapter **3**

Configuration Procedures for Fabric Subsystems

This chapter describes a step-by-step procedure to configure a subsystem that uses Fibre Channel fabric topology. In fabric topology, the controller connects to its host(s) through switches.

The configuration procedure detailed in this chapter uses the command line interpreter (CLI), which is the low-level interface to the controller. There is also a graphic user interface called StorageWorks Command Console (SWCC) which can be used to configure the subsystem after a command console LUN is set up through the CLI. If you want to use SWCC for configuration, see the SWCC online help for assistance.

NOTE: The configuration process is complex because of the many options available; it is therefore highly recommended that you follow the flowchart, Figure 3–2, when configuring your subsystem.

The configuration procedures described in this chapter make the following assumptions:

- controllers and cache modules installed in a fully functional and populated enclosure
- PCMCIA cards installed

To install a controller or cache module, and for instructions in installing the PCMCIA card, see the *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 Maintenance and Service Guide*.

The controllers in a dual-redundant pair are referred to as “controller A” and “controller B.” CLI commands frequently refer to “this controller” and “other controller.” For a definition of these terms, see “Terminology,” page 1–1.

Establishing a Local Connection

A local connection is required to configure the controller until a command console LUN (CCL) is established using the CLI; after that, communication with the controller can be through either the CLI or through the SWCC graphic interface.

The maintenance port, as shown in Figure 3–1, provides a way to connect a maintenance terminal. The maintenance terminal can be either an EIA-423 compatible terminal or a computer running a terminal emulator program. This port accepts a standard RS-232 jack. The maintenance port cable shown in Figure 3–1 has a 9-pin connector molded onto the end for a PC connection. If you need a terminal connection or a 25-pin connection, optional cabling is available and can be ordered.

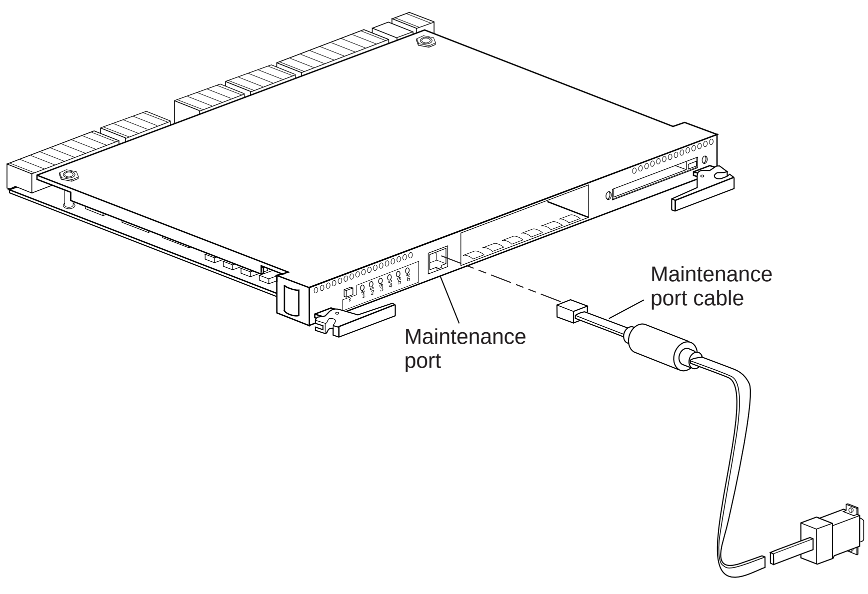


Figure 3–1. Maintenance port connection

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

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

1. Turn off the computer or terminal
2. Connect the computer or terminal to the controller as shown in Figure 3–1. The connection to the computer is through the COMM1 or COMM2 ports.
3. Turn on the computer or terminal.
4. Configure the computer or terminal as follows:
   - 9600 baud
   - 8 data bits
   - 1 stop bit
   - no parity
5. Press the Enter or Return key. A copyright notice and the CLI prompt appear, indicating that you established a local connection with the controller.

Configuration Procedure Flowchart

Before a subsystem is configured it must be planned. Planning a subsystem is detailed in Chapter 1, and planning storagesets is detailed in Chapter 2.

When you have decided on a configuration, you are ready to cable and configure the controller(s). Figure 3–2 shows a flow chart of the configuration process, keyed to the sections of this chapter.

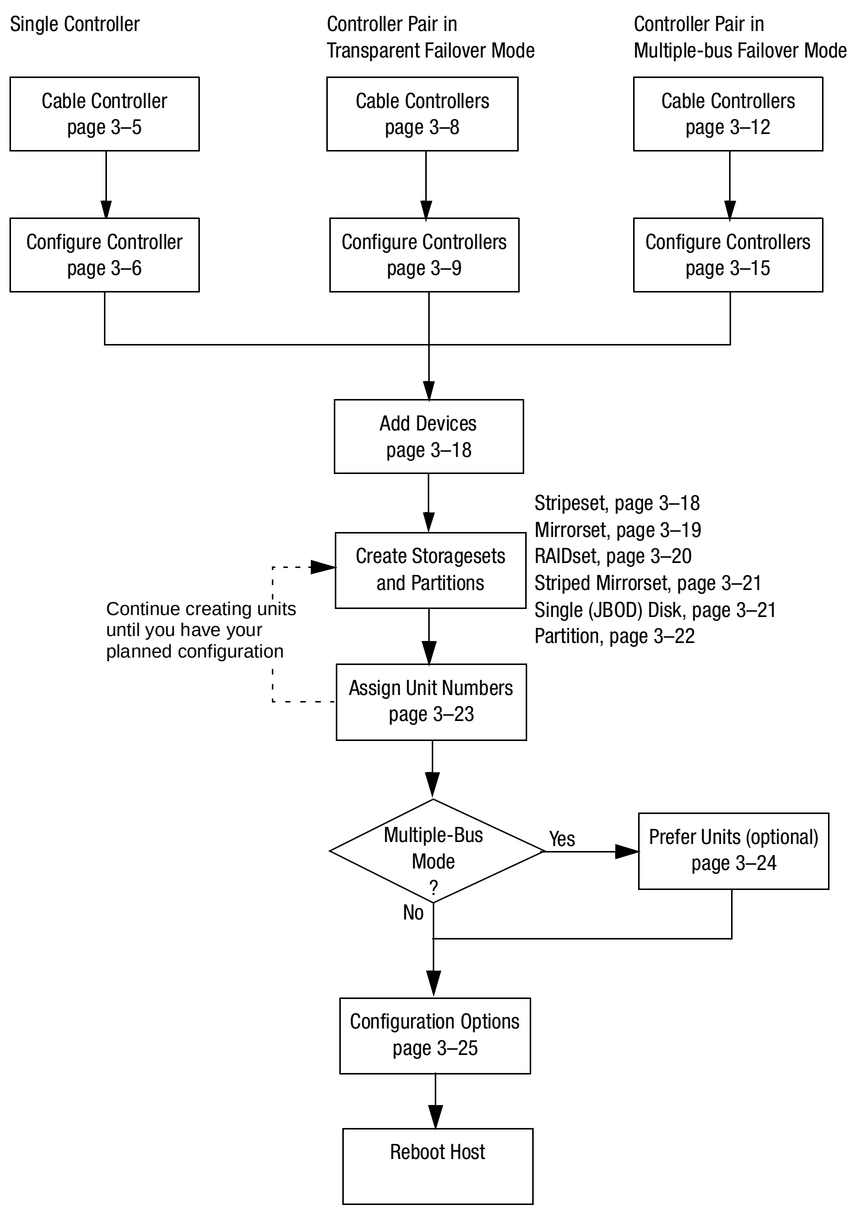


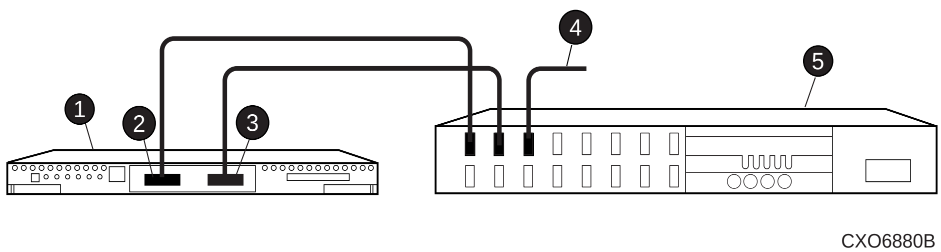
Figure 3–2. Configuration flowchart

Configuring a Single Controller

Cabling a Single controller

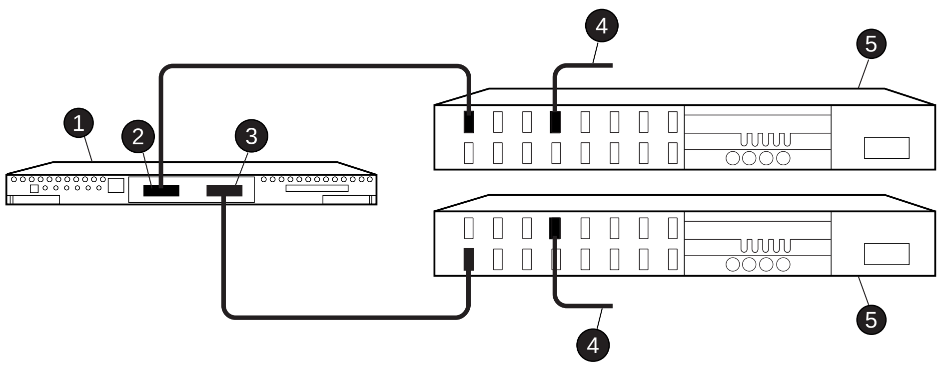
The cabling for a single controller with one switch is shown in Figure 3–3. The cabling for a single controller with two switches is shown in Figure 3–4.

NOTE: It is a good idea to plug only the controller cables into the switch. The host cables are plugged into the switch as part of the configuration procedure (“CLI Configuration Procedure for a Single Controller,” page 3–6).



❶ controller
❷ host port 1
❸ host port 2
❹ cable from the switch to the host Fibre Channel adapter
❺ switch

Figure 3–3. Single controller cabling with one switch



❶ controller

❷ host port 1

❸ host port 2

❹ cable from the switch to the host Fibre Channel adapter

❺ switch

Figure 3–4. Single controller cabling with two switches

CLI Configuration Procedure for a Single Controller

1. Apply power to the subsystem.

   The powerup sequence takes approximately 45 seconds. At the end of the powerup sequence, the audible alarm on the EMU will sound and the EMU error LED will be solidly lit. Turn off the alarm by pressing once and releasing the reset button on the EMU.

2. Attach the maintenance terminal to the controller, as shown in Figure 3–1.

3. If working with anything but a factory-new controller, enter the following command to take it out of any failover mode that may have been previously configured:

   ```
   SET NOFAILOVER
   ```

   If the controller did have a failover mode previously set, the CLI may report an error. Clear the error with this command:

   ```
   CLEAR_ERRORS CLI
   ```

4. Set the topology for the controller. If both ports are used, set topology for both ports:

   ```
   SET THIS PORT_1_TOPOLOGY=FABRIC
   SET THIS PORT_2_TOPOLOGY=FABRIC
   ```

NOTE: If the controller is not factory-new, it may have another topology set, in which case these commands will result in an error message. If this happens, take both ports offline first, then reset the topology:

```
SET THIS PORT_1_TOPOLOGY=OFFLINE
SET THIS PORT_2_TOPOLOGY=OFFLINE
SET THIS PORT_1_TOPOLOGY=FABRIC
SET THIS PORT_2_TOPOLOGY=FABRIC
```

5. Set the time on the controller using the following syntax:

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

6. Use the FRUTIL utility to set up the battery discharge timer. Enter the following command to start FRUTIL:

   ```
   RUN FRUTIL
   ```

   When FRUTIL asks if you intend to replace the battery, answer "y":

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

   FRUTIL will print out a procedure, but won’t give you a prompt. Ignore the procedure and just press enter.

7. Set up any additional optional controller settings, such as changing the CLI prompt. See the SET *this_controller/other controller* command in the *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide* for the format of optional settings.

8. Enter a SHOW *this_controller* command to verify that all changes have taken place:

   ```
   SHOW THIS_CONTROLLER
   ```

9. Plug in the Fibre Channel cable from the first adapter in the first host into the switch. Enter a SHOW *connections* command to view the connection table:

   ```
   SHOW CONNECTIONS
   ```

   The first connection will have one or more entries in the connection table. (How many there are depends on cabling configuration.) Each connection will have a default name of the form !NEWCONxx, where xx is a number representing the order in which the connection was added to the connection table.

10. Rename the connection(s) to something meaningful to the system and easy to remember. (For a recommended naming convention, see “Naming Connections,” page 1–11.) For example, to assign the name ANGEL1A1 to connection !NEWCON01, enter:

    ```
    RENAME !NEWCON01 ANGEL1A1
    ```

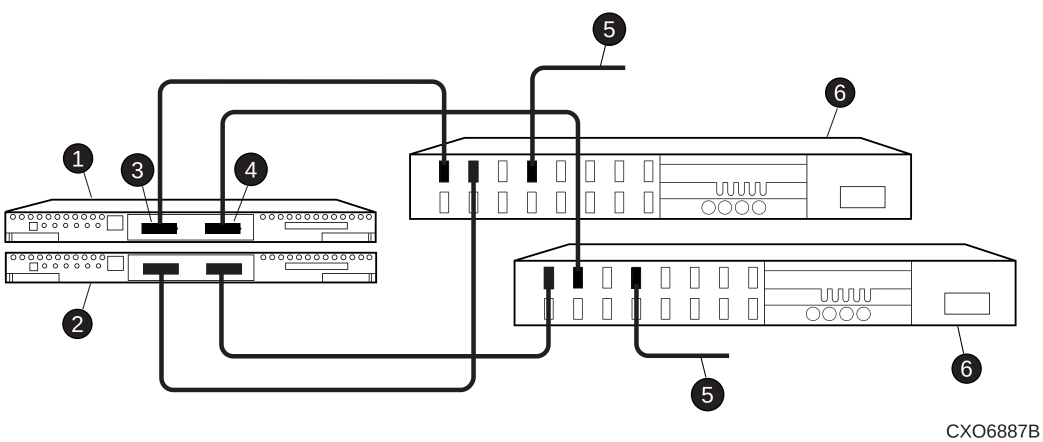
Configuration Procedure for Transparent Failover Mode

Cabling Controllers in Transparent Failover Mode

In transparent failover mode, port 1 of controller A is on the same fabric as port 1 of controller B. Being on the same fabric means that the ports connect to their hosts through the same switches. Likewise port 2 of controller A is on the fabric as port 2 of controller B. The port 1 fabric and the port 2 fabric can be either different fabrics or the same fabric. The use of port 2 is optional.

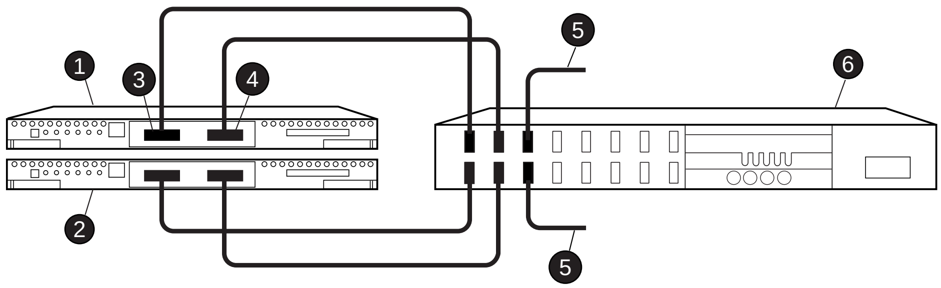
The cabling for a two-switch configuration, in which ports 1 and 2 are on separate fabrics, is shown in Figure 3–5. The cabling for a one-switch configuration, in which both ports 1 and 2 are on the same fabric, is shown in Figure 3–6.

NOTE: It is a good idea to plug only the controller cables into the switch. The host cables are plugged into the switch as part of the configuration procedure (“CLI Configuration Procedure for Transparent Failover Mode,” page 3–9).



❶ controller A
❷ controller B
❸ host port 1
❹ host port 2
❺ cable from the switch to the host FC adapter
❻ switch

Figure 3–5. Transparent failover cabling with two switches



❶ controller A
❷ controller B
❸ host port 1
❹ host port 2
❺ cable from the switch to the host FC adapter
❻ switch

Figure 3–6. Transparent failover cabling with one switch

CLI Configuration Procedure for Transparent Failover Mode

1. Apply power to the subsystem.

   The powerup sequence takes approximately 45 seconds. At the end of the powerup sequence, the audible alarm on the EMU will sound and the EMU error LED will be solidly lit. Turn off the alarm by pressing once and releasing the reset button on the EMU.

   The alarm will go off, but the LED will remain on until the controllers are bound into failover mode. The CLI will display a copyright notice and a last event report from the “other” controller.

2. Attach the maintenance terminal to the top controller, as shown in Figure 3–1.

3. If working with anything but factory-new controllers, enter the following command to remove any failover mode that may have been previously configured:

   SET NOFAILOVER

4. Enter the following command to stop the CLI from reporting a misconfiguration error resulting from having no failover mode specified:

   CLEAR_ERRORS CLI

5. Put the controller pair into transparent failover mode, using the following command:

   SET FAILOVER COPY=THIS

The copy qualifier specifies where the good copy of the array configuration is. Initially, the copy qualifier is meaningless because there is no array configuration yet, but it is part of the command syntax and must be specified.

> **NOTE:** If there is configuration information that you want to keep, which will probably be the case if the controllers are not factory-new, set COPY to the controller that contains the good configuration information

When the command is entered, the "other" controller (the one to which the maintenance terminal is not attached) will restart. The restart may set off the audible alarm of the EMU. To silence the alarm, press once and release the button on the EMU. The alarm will stop, but the LED will remain on until the controllers bind into transparent failover mode. The binding process takes about 15 seconds. The CLI will print out a last event report from the "other" controller; this just indicates that the "other" controller restarted. The CLI will continue reporting this condition until cleared with the following command:

```
CLEAR_ERRORS CLI
```

6. Optional: Set up mirrored cache for the controller pair using the following command:

   ```
   SET THIS_CONTROLLER MIRRORED_CACHE
   ```

   This command causes a restart, so the EMU audible alarm may sound.

7. Set the topology for both ports of both controllers:

   ```
   SET THIS PORT_1_TOPOLOGY=FABRIC
   SET THIS PORT_2_TOPOLOGY=FABRIC
   SET OTHER PORT_1_TOPOLOGY=FABRIC
   SET OTHER PORT_2_TOPOLOGY=FABRIC
   ```

   > **NOTE:** If the controllers are not factory-new, they may have another topology set, in which case these commands will result in an error message. If this happens, take all ports offline first, then reset the topology:

   ```
   SET THIS PORT_1_TOPOLOGY=OFFLINE
   SET THIS PORT_2_TOPOLOGY=OFFLINE
   SET OTHER PORT_1_TOPOLOGY=OFFLINE
   SET OTHER PORT_2_TOPOLOGY=OFFLINE
   SET THIS PORT_1_TOPOLOGY=FABRIC
   SET THIS PORT_2_TOPOLOGY=FABRIC
   SET OTHER PORT_1_TOPOLOGY=FABRIC
   SET OTHER PORT_2_TOPOLOGY=FABRIC
   ```

8. Set the time on “this” controller with the following syntax:

   SET THIS TIME=DD-MMM-YYYY:HH:MM:SS

9. Use the FRUTIL utility to set up the battery discharge timer in “this” controller. Enter the following command to start FRUTIL:

   RUN FRUTIL

   When FRUTIL asks if you intend to replace the battery, answer "y":

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

   FRUTIL will print out a procedure, but won't give you a prompt. Ignore the procedure and just press return.

10. Move the maintenance cable to the lower controller and repeat step 9.

11. Set up any additional optional controller settings, such as changing the CLI prompt. See the SET *this_controller/other controller* command in the *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide* for the format of optional settings.

12. Enter a SHOW *this_controller* command and a SHOW *other_controller* command to verify that all changes have taken place:

    SHOW THIS_CONTROLLER

    SHOW OTHER_CONTROLLER

13. Plug in the Fibre Channel cable from the first adapter in the first host into the switch. Enter a SHOW *connections* command to view the connection table:

    SHOW CONNECTIONS

    The first connection will have one or more entries in the connection table. (How many there are depends on cabling configuration.) Each connection will have a default name of the form !NEWCONxx, where xx is a number representing the order in which the connection was added to the connection table.

14. Rename the connection(s) to something meaningful to the system and easy to remember. (For a recommended naming convention, see “Naming Connections,” page 1–11.) For example, to assign the name ANGEL1A1 to connection !NEWCON01, enter:

    RENAME !NEWCON01 ANGEL1A1

Configuration Procedure for Multiple-Bus Failover Mode

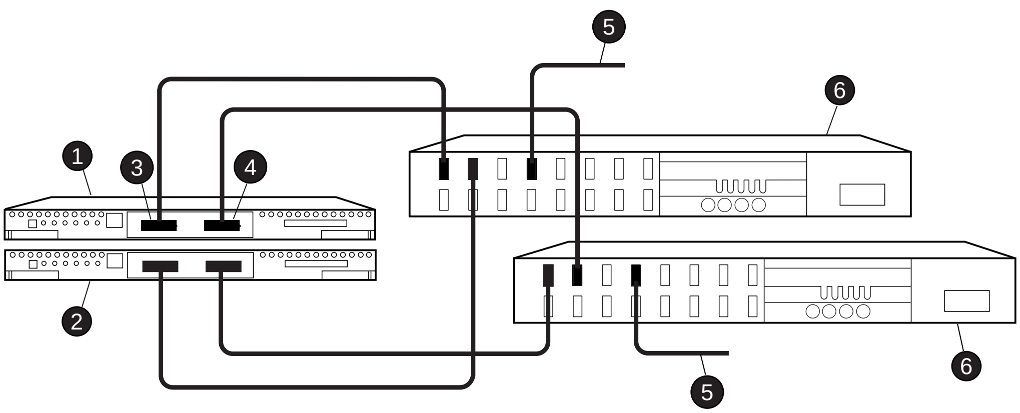
In multiple-bus failover mode, there are two or more paths between each unit and its host(s). There are three ways to configure a controller pair in multiple-bus mode. The following sections show the three cabling options. For simplicity, the drawings show only the two host connections required for one host, but additional host connections are possible.

Cabling Controllers in Multiple-Bus Failover Mode

The most common cabling option is shown in Figure 3–7; This option is relatively simple: one switch connects to port 1 of both controllers, and the second switch connects to port 2 of both controllers. This option provides path redundancy between the switches and the controllers (there is a path from each switch to each controller).

NOTE: It is a good idea to plug only the controller cables into the switch. The host cables are plugged into the switch as part of the configuration procedure (“CLI Configuration Procedure for Multiple-Bus Failover Mode,” page 3–15).

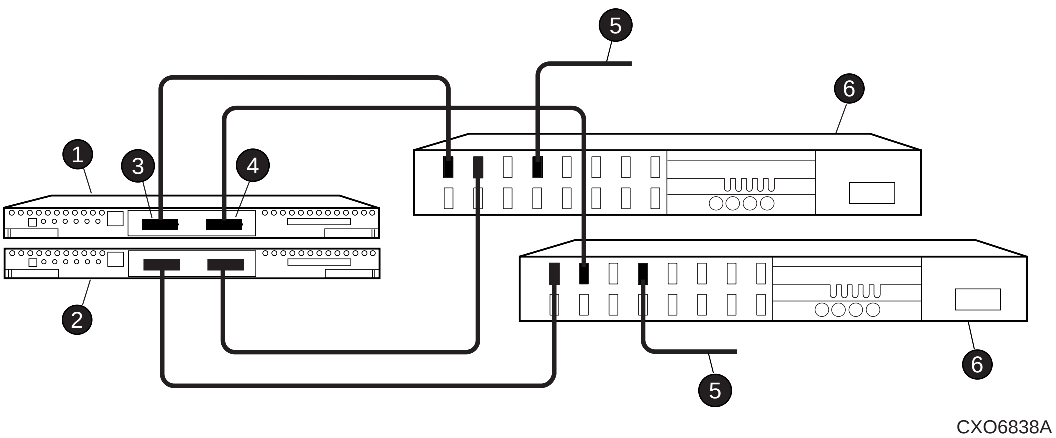
CAUTION: The configuration shown in Figure 3–7 is not recommended for Data Replication Manager applications. See the *Data Replication Manager HSG80 ACS Version 8.5P Operations Guide* for configurations supported in Data Replication Manager applications.



❶	controller A	❹	host port 2
❷	controller B	❺	cable from the switch to the host FC adapter
❸	host port 1	❻	switch

Figure 3–7. Multiple-bus failover cabling, option 1

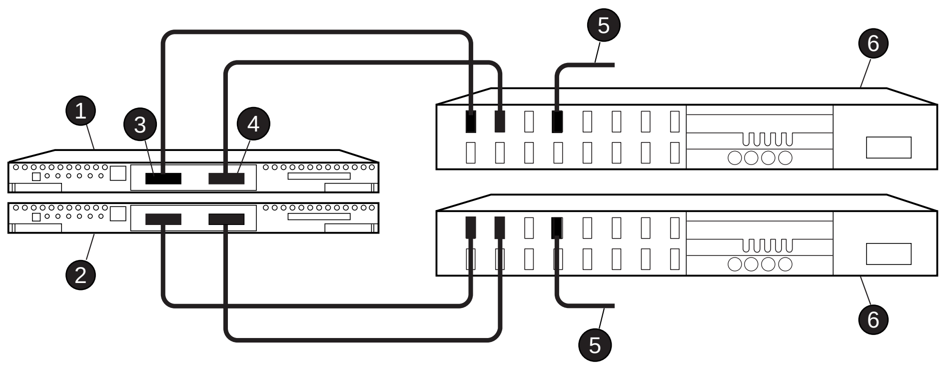
Another cabling option is shown in Figure 3–8. This option is more complex: each switch connects to port 1 of one controller and port 2 of the other controller. This option provides path redundancy between the switches and the controllers (there is a path from each switch to each controller).



❶ controller A
❷ controller B
❸ host port 1
❹ host port 2
❺ cable from the switch to the host FC adapter
❻ switch

Figure 3–8. Multiple-bus failover cabling, option 2

The third cabling option is shown in Figure 3–9. This option is simple: each switch connects to both ports of one controller. This option does not provide path redundancy between the switches and the controllers; if a switch fails, the controller to which it is attached becomes inaccessible.



❶	controller A	❹	host port 2
❷	controller B	❺	cable from the switch to the host FC adapter
❸	host port 1	❻	switch

Figure 3–9. Multiple-bus failover cabling, option 3 (limited path redundancy)

CLI Configuration Procedure for Multiple-Bus Failover Mode

1. Apply power to the subsystem.

   The powerup sequence takes approximately 45 seconds. At the end of the powerup sequence, the audible alarm on the EMU will sound and the EMU error LED will be solidly lit. Turn off the alarm by pressing once and releasing the reset button on the EMU.

   The alarm will go off, but the LED will remain on until the controllers are bound into failover mode. The CLI will display a copyright notice and a last event report from the "other" controller.

2. Attach the maintenance terminal to the top controller, as shown in Figure 3–1.

3. If working with anything but factory-new controllers, enter the following command to remove any failover mode that may have been previously configured:

   ```
   SET NOFAILOVER
   ```

4. Enter the following command to stop the CLI from reporting a misconfiguration error resulting from having no failover mode specified:

   CLEAR CLI

5. Put the controller pair into multiple-bus failover mode, using the following command:

   SET MULTIBUS COPY=THIS

   The copy qualifier specifies where the good copy of the array configuration is. Initially, the copy qualifier is meaningless because there is no array configuration yet, but it is part of the command syntax and must be specified.

   NOTE: If there is configuration information that you want to keep, which will probably be the case if the controllers are not factory-new, set COPY to the controller that contains the good configuration information

   When the command is entered, the “other” controller (the one to which the serial line is not attached) will restart. The restart may set off the audible alarm of the EMU. To silence the alarm, press once and release the button on the EMU. The alarm will stop, but the LED will remain on until the controllers bind into transparent failover mode. The binding process takes about 15 seconds. The CLI will print out a last event report from the “other” controller. This just indicates that the “other” controller restarted. The CLI will continue reporting this condition until cleared with the following command:

   CLEAR CLI

6. Set up mirrored cache, if desired, for the controller pair using the following command:

   SET THIS MIRRORED_CACHE

   This command causes a restart, so the EMU audible alarm may sound.

7. Set the topology for both ports of both controllers:

   SET THIS PORT_1_TOPOLOGY=FABRIC

   SET THIS PORT_2_TOPOLOGY=FABRIC

   SET OTHER PORT_1_TOPOLOGY=FABRIC

   SET OTHER PORT_2_TOPOLOGY=FABRIC

NOTE: If the controllers are not factory-new, they may have another topology set, in which case these commands will result in an error message. If this happens, take all ports offline first, then reset the topology:

```
SET THIS PORT_1_TOPOLOGY=OFFLINE
SET THIS PORT_2_TOPOLOGY=OFFLINE
SET OTHER PORT_1_TOPOLOGY=OFFLINE
SET OTHER PORT_2_TOPOLOGY=OFFLINE
SET THIS PORT_1_TOPOLOGY=FABRIC
SET THIS PORT_2_TOPOLOGY=FABRIC
SET OTHER PORT_1_TOPOLOGY=FABRIC
SET OTHER PORT_2_TOPOLOGY=FABRIC
```

8. Set the time on “this” controller using the following syntax:

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

9. Use the FRUTIL utility to set up the battery discharge timer in “this” controller. Enter the following command to start FRUTIL:

   ```
   RUN FRUTIL
   ```

   When FRUTIL asks if you intend to replace the battery, answer "y":

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

   FRUTIL will print out a procedure, but won't give you a prompt. Ignore the procedure and just press return.

10. Move the serial cable to the lower controller, and repeat step 9.

11. Set up any additional optional controller settings, such as changing the CLI prompt. See the SET *this_controller/other_controller* command in the *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide* for the format of optional settings.

12. Enter a SHOW *this_controller* command and a SHOW *other_controller* command to verify that all changes have taken place:

    ```
    SHOW THIS_CONTROLLER
    SHOW OTHER_CONTROLLER
    ```

13. Plug in the Fibre Channel cable from the first adapter in the first host into the switch. Enter a SHOW *connections* command to view the connection table:

    ```
    SHOW CONNECTIONS
    ```

The first connection will have one or more entries in the connection table. (How many there are depends on cabling configuration.) Each connection will have a default name of the form !NEWCONxx, where xx is a number representing the order in which the connection was added to the connection table.

14. Rename the connection(s) to something meaningful to the system and easy to remember. (For a recommended naming convention, see “Naming Connections,” page 1–11.) For example, to assign the name ANGEL1A1 to connection !NEWCON01, enter:

    RENAME !NEWCON01 ANGEL1A1

Configuring Devices

The devices on the device bus can be configured either manually or by the CONFIG utility. The CONFIG utility is easier. Invoke CONFIG with the following command:

RUN CONFIG

CONFIG takes about 2 minutes to discover and map the configuration of a completely populated pedestal. The message that CONFIG is running looks like the following:

```
Config Local Program Invoked
Config is building its tables and determining what devices exist
on the subsystem. Please be patient.
add disk DISK10000   1 0 0
add disk DISK10100   1 1 0
add disk DISK20000   2 0 0
add disk DISK20100   2 1 0
...
Config - Normal Termination
```

Configuring a Stripeset

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 DISKNNNNN DISKNNNNN

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

   INITIALIZE STRIPESET-NAME SWITCH

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

   SHOW STRIPESET-NAME

4. Assign the stripeset a unit number to make it accessible by the host(s). See “Assigning Unit Numbers and Unit Qualifiers” on page 3–23.

Example

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

```
ADD STRIPESET STRIPE1 DISK10000 DISK20000 DISK30000
INITIALIZE STRIPE1 CHUNKSIZE=128
SHOW STRIPE1
```

See Chapter 2 for more information on stripeset switches and values.

Configuring a Mirrorset

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 DISKNNNNN DISKNNNNN SWITCHES

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

   INITIALIZE MIRRORSET-NAME SWITCHES

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

   SHOW MIRRORSET-NAME

4. Assign the mirrorset a unit number to make it accessible by the host(s). See “Assigning Unit Numbers and Unit Qualifiers” on page 3–23.

Example

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

```
ADD MIRRORSET MIRR1 DISK10000 DISK20000
INITIALIZE MIRR1
SHOW MIRR1
```

See Chapter 2 for more information on stripeset switches and values.

Configuring a RAIDset

To configure a RAIDset:

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

   Use the following syntax to create a RAIDset:

   ```
   ADD RAIDSET RAIDSET-NAME DISKNNNNN DISKNNNNN DISKNNNNN SWITCH
   ```

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

   ```
   INITIALIZE RAIDSET-NAME SWITCH
   ```

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

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

   ```
   SHOW RAIDSET-NAME
   ```

4. Assign the RAIDset a unit number to make it accessible by the host(s). See "Assigning Unit Numbers and Unit Qualifiers" on page 3–23.

Example

The following example shows the commands you would use to create RAID1, a three-member RAIDset:

```
ADD RAIDSET RAID1 DISK10000 DISK20000 DISK30000
INITIALIZE RAID1
SHOW RAID1
```

See Chapter 2 for more information on RAIDset switches and values.

Configuring a Striped Mirrorset

To configure a striped mirrorset:

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

   ADD STRIPESET STRIPESET-NAME MIRRORSET-1 MIRRORSET-2....MIRRORSET-N

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

   INITIALIZE STRIPESET-NAME SWITCH

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

   SHOW STRIPESET-NAME

5. Assign the stripeset mirrorset a unit number to make it accessible by the host(s). See “Assigning Unit Numbers and Unit Qualifiers” on page 3–23.

Example

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

```
ADD MIRRORSET MIRR1 DISK10000 DISK20000
ADD MIRRORSET MIRR2 DISK30000 DISK40000
ADD MIRRORSET MIRR3 DISK50000 DISK60000
ADD STRIPESET STRIPE1 MIRR1 MIRR2 MIRR3
INITIALIZE STRIPE1 CHUNKSIZE=DEFAULT
SHOW STRIPE1
```

See Chapter 2 for more information on stripeset and mirrorset switches and values.

Configuring a Single-Disk Unit

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

1. Initialize the disk drive using the following syntax:

   INITIALIZE DISKNNN SWITCH

2. Assign the disk a unit number to make it accessible by the host(s). See “Assigning Unit Numbers and Unit Qualifiers” on page 3–23.
3. Verify the configuration using the following command:

   SHOW DEVICES

Configuring a Partition

To partition a storageset or disk drive:

1. Initialize the storageset or disk drive. If you want to set any initialization switches, you must do so in this step. Use which syntax is appropriate:

   INITIALIZE STORAGESET-NAME SWITCHES

   or

   INITIALIZE DISK-NAME SWITCHES

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

   or

   CREATE_PARTITION DISK-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

   or

   SHOW DISK-NAME

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

4. Assign the partition a unit number to make it accessible by the host(s). See “Assigning Unit Numbers and Unit Qualifiers” on page 3–23.

Example

The following example shows the commands you would use to create RAID1, a three-member RAIDset, then partition it into two storage units:

```
ADD RAIDSET RAID1 DISK10000 DISK20000 DISK30000
INITIALIZE RAID1
CREATE_PARTITION RAID1 SIZE=25
CREATE_PARTITION RAID1 SIZE=LARGEST
SHOW RAID1
```

See Chapter 2 for more information on partition switches and values.

Assigning Unit Numbers and Unit Qualifiers

Each storageset, partition, or single (JBOD) disk must be assigned a unit number for the host to access. As the units are added, their properties can be specified through use of command qualifiers, which are discussed in detail under the ADD UNIT command in the *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide*.

Each unit can be reserved for the exclusive use of a host or group of hosts. See "Restricting Host Access in Transparent Failover Mode," page 1–20 and "Restricting Host Access in Multiple-Bus Failover Mode," page 1–24.

Assigning a Unit Number to a Storageset

To assign a unit number to a storageset, use the following syntax:

```
ADD UNIT UNIT-NUMBER STORAGESET-NAME
```

Example:

To assign unit D102 to RAIDset R1 use the following command:

```
ADD UNIT D102 R1
```

Assigning a Unit Number to a Single (JBOD) Disk

To assign a unit number to a single (JBOD) disk, use the following syntax:

```
ADD UNIT UNIT-NUMBER DISK-NAME
```

Example:

To assign unit D4 to disk20300, use the following command:

```
ADD UNIT D4 DISK20300
```

Assigning a Unit Number to a Partition

To assign a unit number to a partition, use the following syntax:

```
ADD UNIT UNIT-NUMBER STORAGESET-NAME PARTITION=PARTITION-NUMBER
```

Example:

To assign unit D100 to partition 3 of mirrorset mirr1, use the following command:

```
ADD UNIT D100 MIRROR1 PARTITION=3
```

Preferring Units in Multiple-Bus Failover Mode

In multiple-bus failover mode, individual units can be preferred to a specific controller. To prefer, for example, unit D102 to “this controller,” use the following command:

```
SET D102 PREFERRED_PATH=THIS
```

RESTART commands must be issued to both controllers for this command to take effect:

```
RESTART THIS_CONTROLLER
RESTART OTHER_CONTROLLER
```

NOTE: The controllers need to restart together for the preferred settings to take effect. The RESTART *other_controller* command must be entered immediately after the RESTART *this_controller* command.

Configuration Options

There are many options to choose from when configuring a subsystem. This section shows how to set-up some of the more common ones.

Changing the CLI Prompt

To change the CLI prompt, enter a 1- to 16- character string as the new prompt, as follows:

```
SET THIS_CONTROLLER PROMPT = "NEW PROMPT"
```

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

```
SET OTHER_CONTROLLER PROMPT = "NEW PROMPT"
```

It is suggested that the prompts reflect something about the controllers. For example, if the subsystem is the third one in a lab, give the top controller a prompt like LAB3A and the bottom controller, LAB3B.

Adding Disk Drives

If you add new disk drives to the subsystem, the disk drives must be added to the controllers’ list of known devices:

- To add one new disk drive to the list of known devices, enter the following command:

  ```
  ADD DISK DISKNNN PTL-LOCATION
  ```

- To add several new disk drives to the list of known devices, enter the following command:

  ```
  RUN CONFIG
  ```

Adding a Disk Drive to the Spareset

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

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

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

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

   ```
   ADD SPARESET DISKNNNNN
   ```

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

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

   ```
   SHOW SPARESET
   ```

Example

The following example shows the commands for adding DISK60000 and DISK60100 to the spareset.

```
ADD SPARESET DISK60000
ADD SPARESET DISK60100
SHOW SPARESET
```

Removing a Disk Drive from the Spareset

You can delete disks in the spareset if you need to use them elsewhere in your subsystem. To remove a disk drive from the spareset:

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

   ```
   SHOW SPARESET
   ```

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

   ```
   DELETE SPARESET DISKNNNNN
   ```

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

   ```
   SHOW SPARESET
   ```

Enabling Autospare

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

To enable autospare use the following command:

```
SET FAILEDSET AUTOSPARE
```

To disable autospare use the following command:

```
SET FAILEDSET NOAUTOSPARE
```

During initialization, AUTOSPARE checks to see if the new disk drive contains metadata. Metadata is information the controller writes on the disk drive when the disk drive is configured into a storageset. Therefore, the presence of metadata indicates the disk drive belongs to, or has been used by, a storageset. If the disk drive contains metadata, initialization stops. (A new disk drive will not contain metadata but a repaired or re-used disk drive might. To erase metadata from a disk drive, add it to the controller's list of devices, then set it to be transportable and initialize it.)

Deleting a Storageset

NOTE: If the storageset you are deleting is partitioned, you must delete each partitioned unit before you can delete the storageset.

Use the following steps to delete a storageset:

1. Show the configuration using the following command:

   ```
   SHOW STORAGESETS
   ```

2. Delete the unit number that uses the storageset. Use the following command:

   ```
   DELETE UNIT-NUMBER
   ```

3. Delete the storageset. Use the following command:

   ```
   DELETE STORAGESET-NAME
   ```

4. Verify the configuration using the following command:

   ```
   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. Remember to update the storageset's profile when you change its switches.

Displaying the Current Switches

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

```
SHOW STORAGESET-NAME OR DEVICE-NAME FULL
```

Changing RAIDset and Mirrorset Switches

Use the SET storageset-name command to change the RAIDset and Mirrorset switches associated with an existing storageset. For example, the following command changes the replacement policy for RAIDset RAID1 to BEST_FIT:

```
SET RAID1 POLICY=BEST_FIT
```

Changing Device Switches

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

```
SET DISK10000 TRANSPORTABLE
```

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

Changing Initialize Switches

The initialization 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 the characteristics of a unit. For example, the following command enables write protection for unit D100:

```
SET D100 WRITE_PROTECT
```

Chapter **4**

Configuration Procedures for Loop Subsystems

This chapter describes a step-by-step procedure to configure a subsystem that uses Fibre Channel arbitrated loop topology. In loop topology, the controller connects to its host(s) through hubs.

The configuration procedure detailed in this chapter uses the command line interpreter (CLI), which is the low-level interface to the controller. There is also a graphic user interface called StorageWorks Command Console (SWCC) which can be used to configure the subsystem after a command console LUN is set up through the CLI. If you want to use SWCC for configuration, refer to the SWCC online help for assistance.

> **NOTE:** The configuration process is complex because of the many options available; it is therefore highly recommended that you follow the flowchart, Figure 4–2, when configuring your subsystem.

The configuration procedures described in this chapter make the following assumptions:

- controllers and cache modules installed in a fully functional and populated enclosure
- PCMCIA cards installed

To install a controller or cache module, and for instructions in installing the PCMCIA card, see the *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 Maintenance and Service Guide*.

The controllers in a dual-redundant pair are referred to as “controller A” and “controller B.” CLI commands frequently refer to “this controller” and “other controller.” For a definition of these terms, see “Terminology,” page 1–1.

Establishing a Local Connection

A local connection is required to configure the controller until a command console LUN (CCL) is established using the CLI; after that, communication with the controller can be through either the CLI or through the SWCC graphic interface.

The maintenance port, as shown in Figure 4–1, provides a way to connect a maintenance terminal. The maintenance terminal can be either an EIA-423 compatible terminal or a computer running a terminal emulator program. This port accepts a standard RS-232 jack. The maintenance port cable shown in Figure 4–1 has a 9-pin connector molded onto the end for a PC connection. If you need a terminal connection or a 25-pin connection, optional cabling is available and can be ordered.

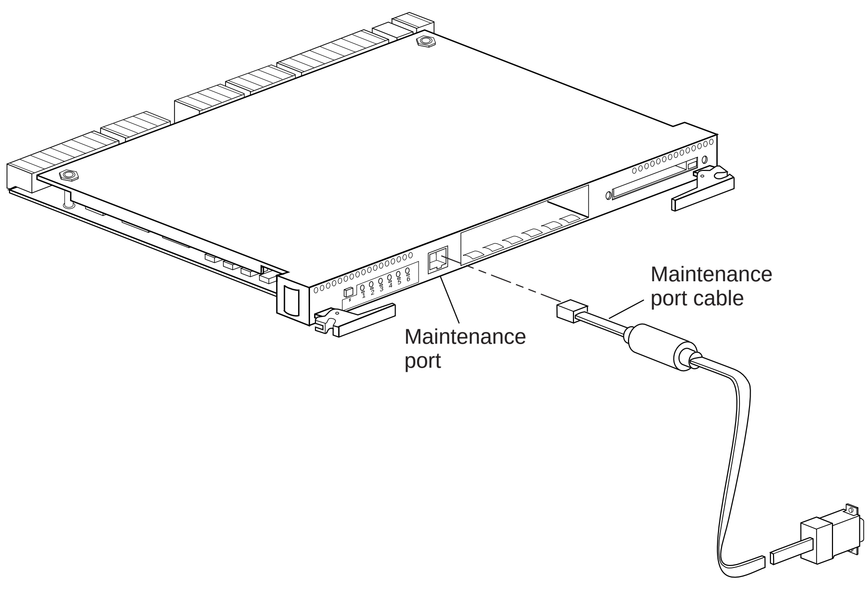


Figure 4–1. Maintenance port connection

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

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

1. Turn off the computer or terminal
2. Connect the computer or terminal to the controller as shown in Figure 4–1. The connection to the computer is through the COMM1 or COMM2 ports.
3. Turn on the computer or terminal.
4. Configure the computer or terminal as follows:
   - 9600 baud
   - 8 data bits
   - 1 stop bit
   - no parity
5. Press the Enter or Return key. A copyright notice and the CLI prompt appear, indicating that you established a local connection with the controller.

Configuration Procedure Flowchart

Before a subsystem is configured it must be planned. Planning a subsystem is detailed in Chapter 1, and planning storagesets is detailed in Chapter 2.

When you have decided on a configuration, you are ready to cable and configure the controller(s). Figure 4–2 shows a flow chart of the configuration process, keyed to the sections of this chapter.

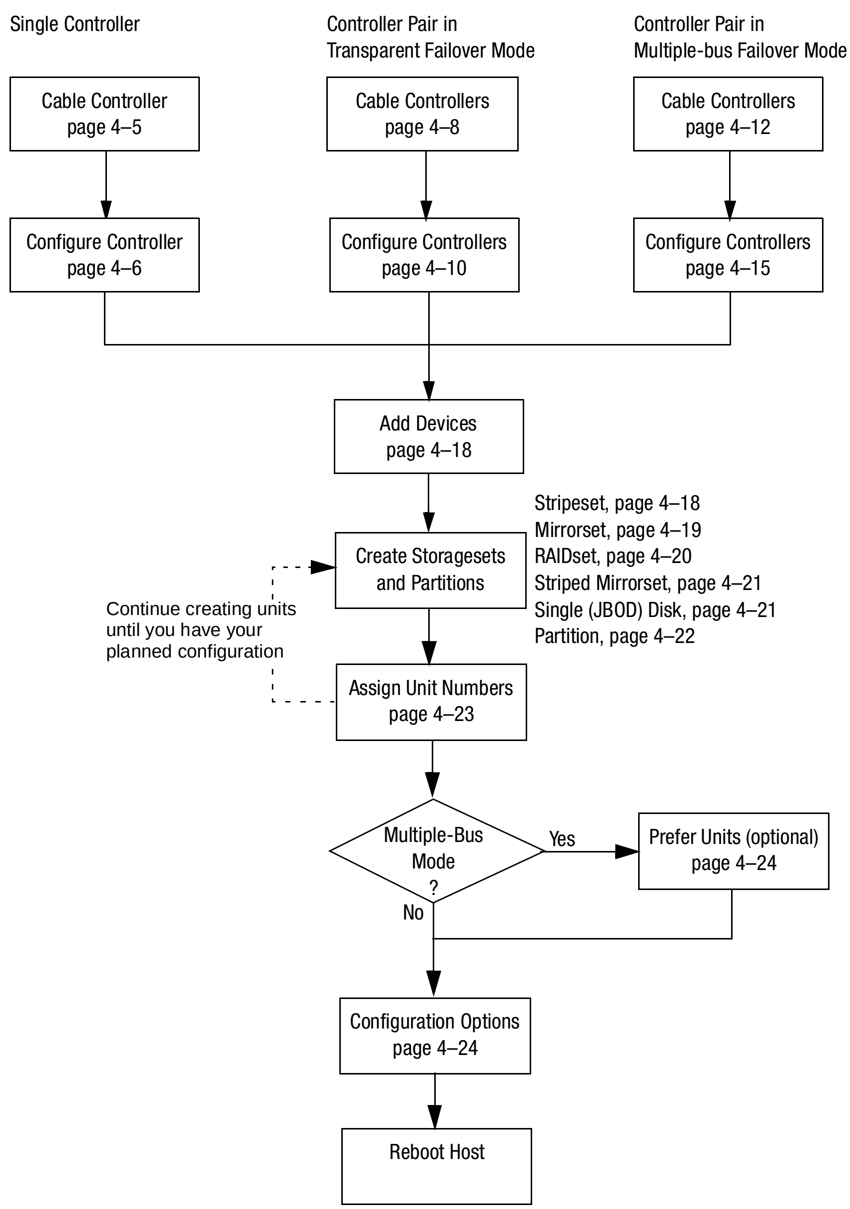


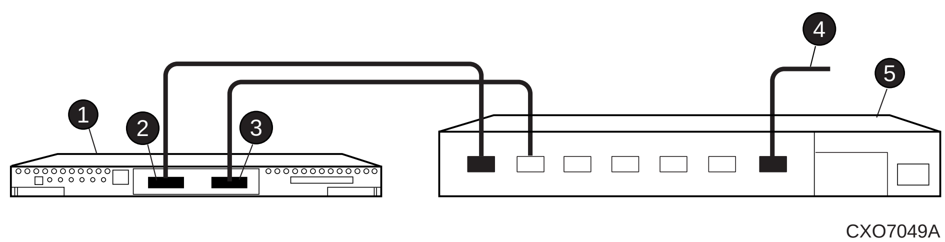
Figure 4–2. Configuration flowchart

Configuring a Single Controller

Cabling a Single controller

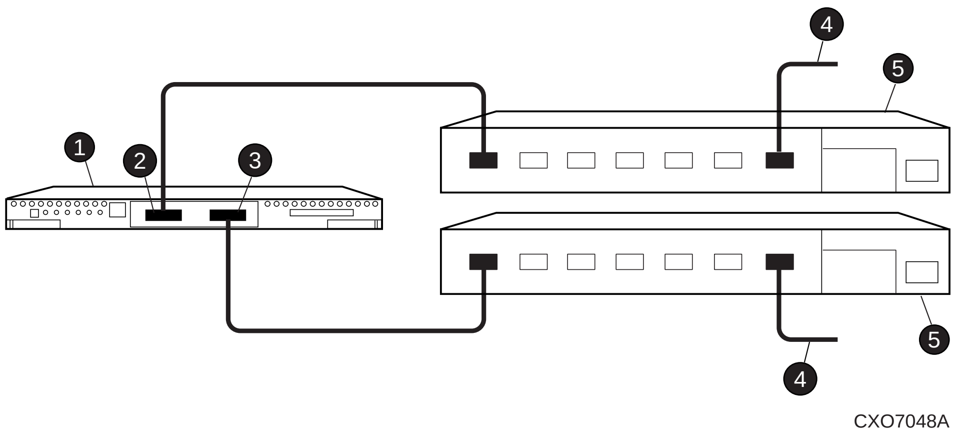
The cabling for a single controller with one hub is shown in Figure 4–3. The cabling for a single controller with two hubs is shown in Figure 4–4.

NOTE: It is a good idea to plug only the controller cables into the hub. The host cables are plugged into the hub as part of the configuration procedure (“CLI Configuration Procedure for a Single Controller,” page 4–6).



❶ controller
❷ host port 1
❸ host port 2
❹ cable from the hub to the host Fibre Channel adapter
❺ hub

Figure 4–3. Single controller cabling with one hub



❶ controller
❷ host port 1
❸ host port 2
❹ cable from the hub to the host bus adapter
❺ hub

Figure 4–4. Single controller cabling with two hubs

CLI Configuration Procedure for a Single Controller

1. Apply power to the subsystem.

   The powerup sequence takes approximately 45 seconds. At the end of the powerup sequence, the audible alarm on the EMU will sound and the EMU error LED will be solidly lit. Turn off the alarm by pressing once and releasing the reset button on the EMU.

2. Attach the maintenance terminal to the controller, as shown in Figure 4–1.

3. If working with anything but a factory-new controller, enter the following command to take it out of any failover mode that may have been previously configured:

   ```
   SET NOFAILOVER
   ```

   If the controller did have a failover mode previously set, the CLI may report an error. Clear the error with this command:

   ```
   CLEAR_ERRORS CLI
   ```

4. Set the topology for the controller. If both ports are used, set topology for both ports:

   NOTE: Some operating systems require topology to be set to *LOOP_HARD*.

   SET THIS PORT_1_TOPOLOGY=LOOP_SOFT

   SET THIS PORT_2_TOPOLOGY=LOOP_SOFT

   NOTE: If the controller is not factory-new, it may have another topology set, in which case these commands will result in an error message. If this happens, take both ports offline first, then reset the topology:

   SET THIS PORT_1_TOPOLOGY=OFFLINE

   SET THIS PORT_2_TOPOLOGY=OFFLINE

   SET THIS PORT_1_TOPOLOGY=LOOP_SOFT

   SET THIS PORT_2_TOPOLOGY=LOOP_SOFT

5. Set the time on the controller using the following syntax:

   SET THIS TIME=DD-MMM-YYYY:HH:MM:SS

6. Use the FRUTIL utility to set up the battery discharge timer. Enter the following command to start FRUTIL:

   RUN FRUTIL

   When FRUTIL asks if you intend to replace the battery, answer "y":

   ```
   Do you intend to replace this controller's cache battery? Y/N [N] Y
   ```

   FRUTIL will print out a procedure, but won't give you a prompt. Ignore the procedure and just press enter.

7. Set up any additional optional controller settings, such as changing the CLI prompt. See the SET *this_controller/other controller* command in the *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide* for the format of optional settings.

8. Enter a SHOW *this_controller* command to verify that all changes have taken place:

   SHOW THIS_CONTROLLER

9. Plug in the Fibre Channel cable from the first adapter in the first host into the hub. Enter a SHOW *connections* command to view the connection table:

   SHOW CONNECTIONS

   The first connection will have one or more entries in the connection table. (How many there are depends on cabling configuration.) Each connection will have a default name of the form !NEWCONxx, where xx is a number representing the order in which the connection was added to the connection table.

10. Rename the connection(s) to something meaningful to the system and easy to remember. (For a recommended naming convention, see “Naming Connections,” page 1–11.) For example, to assign the name ANGEL1A1 to connection !NEWCON01, enter:

```
RENAME !NEWCON01 ANGEL1A1
```

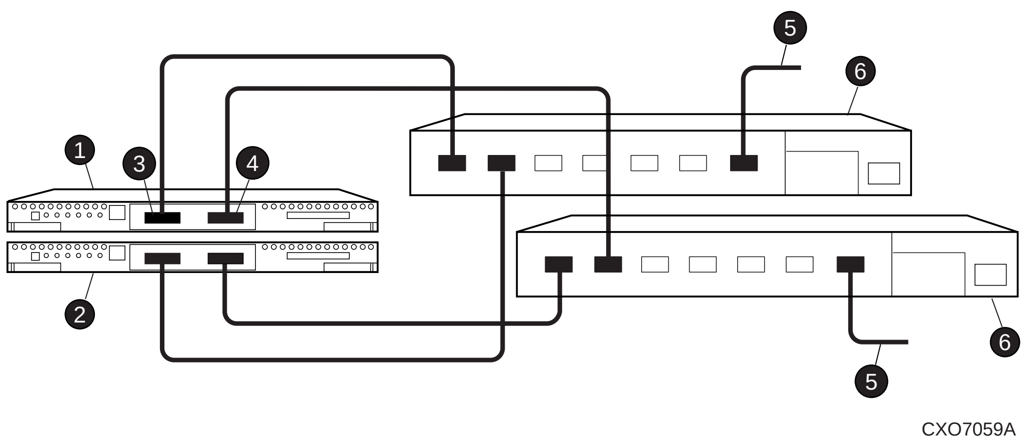
Configuration Procedure for Transparent Failover Mode

Cabling Controllers in Transparent Failover Mode

In transparent failover mode, port 1 of controller A is on the same loop as port 1 of controller B. Being on the same loop means that the ports connect to their hosts through the same hub. Likewise port 2 of controller A is on the loop as port 2 of controller B. The port 1 loop and the port 2 loop can be either different loops or the same loop. The use of port 2 is optional.

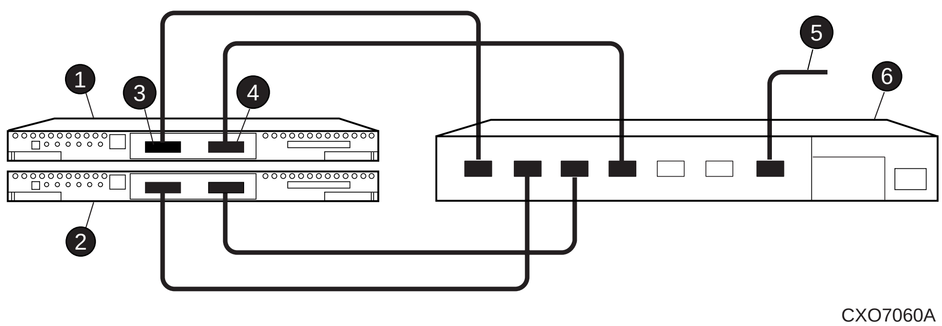
The cabling for a two-hub configuration, in which ports 1 and 2 are on separate loops, is shown in Figure 4–5. The cabling for a one-hub configuration, in which both ports 1 and 2 are on the same loop, is shown in Figure 4–6.

NOTE: It is a good idea to plug only the controller cables into the hub. The host cables are plugged into the hub as part of the configuration procedure (“CLI Configuration Procedure for Transparent Failover Mode,” page 4–10).



❶ controller A
❷ controller B
❸ host port 1
❹ host port 2
❺ cable from the hub to the host FC adapter
❻ hub

Figure 4–5. Transparent failover cabling with two hubs



❶ controller A
❷ controller B
❸ host port 1
❹ host port 2
❺ cable from the hub to the host FC adapter
❻ hub

Figure 4–6. Transparent failover cabling with one hub

CLI Configuration Procedure for Transparent Failover Mode

1. Apply power to the subsystem.

   The powerup sequence takes approximately 45 seconds. At the end of the powerup sequence, the audible alarm on the EMU will sound and the EMU error LED will be solidly lit. Turn off the alarm by pressing once and releasing the reset button on the EMU.

   The alarm will go off, but the LED will remain on until the controllers are bound into failover mode. The CLI will display a copyright notice and a last event report from the "other" controller.

2. Attach the maintenance terminal to the top controller, as shown in Figure 4–1.
3. If working with anything but factory-new controllers, enter the following command to remove any failover mode that may have been previously configured:

   SET NOFAILOVER

4. Enter the following command to stop the CLI from reporting a misconfiguration error resulting from having no failover mode specified:

   CLEAR_ERRORS CLI

5. Put the controller pair into transparent failover mode, using the following command:

   SET FAILOVER COPY=THIS

   The copy qualifier specifies where the good copy of the array configuration is. Initially, the copy qualifier is meaningless because there is no array configuration yet, but it is part of the command syntax and must be specified.

   NOTE: If there is configuration information that you want to keep, which will probably be the case if the controllers are not factory-new, set COPY to the controller that contains the good configuration information

   When the command is entered, the "other" controller (the one to which the maintenance terminal is not attached) will restart. The restart may set off the audible alarm of the EMU. To silence the alarm, press once and release the button on the EMU. The alarm will stop, but the LED will remain on until the controllers bind into transparent failover mode. The binding process takes about 15 seconds. The CLI will print out a last event report from the "other" controller; this just indicates that the "other" controller restarted. The CLI will continue reporting this condition until cleared with the following command:

   CLEAR_ERRORS CLI

6. Optional: Set up mirrored cache for the controller pair using the following command:

   SET THIS_CONTROLLER MIRRORED_CACHE

This command causes a restart, so the EMU audible alarm may sound.

7. Set the topology for both ports of both controllers:

NOTE: Some operating systems require topology to be set to *LOOP_HARD*.

SET THIS PORT_1_TOPOLOGY=LOOP_SOFT

SET THIS PORT_2_TOPOLOGY=LOOP_SOFT

SET OTHER PORT_1_TOPOLOGY=LOOP_SOFT

SET OTHER PORT_2_TOPOLOGY=LOOP_SOFT

NOTE: If the controllers are not factory-new, they may have another topology set, in which case these commands will result in an error message. If this happens, take all ports offline first, then reset the topology:

SET THIS PORT_1_TOPOLOGY=OFFLINE

SET THIS PORT_2_TOPOLOGY=OFFLINE

SET OTHER PORT_1_TOPOLOGY=OFFLINE

SET OTHER PORT_2_TOPOLOGY=OFFLINE

SET THIS PORT_1_TOPOLOGY=LOOP_SOFT

SET THIS PORT_2_TOPOLOGY=LOOP_SOFT

SET OTHER PORT_1_TOPOLOGY=LOOP_SOFT

SET OTHER PORT_2_TOPOLOGY=LOOP_SOFT

8. Set the time on “this” controller with the following syntax:

SET THIS TIME=DD-MMM-YYYY:HH:MM:SS

9. Use the FRUTIL utility to set up the battery discharge timer in “this” controller. Enter the following command to start FRUTIL:

RUN FRUTIL

When FRUTIL asks if you intend to replace the battery, answer "y":

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

FRUTIL will print out a procedure, but won't give you a prompt. Ignore the procedure and just press return.

10. Move the maintenance cable to the lower controller and repeat step 9.

11. Set up any additional optional controller settings, such as changing the CLI prompt. See the SET *this_controller/other controller* command in the *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide* for the format of optional settings.

12. Enter a SHOW *this_controller* command and a SHOW *other_controller* command to verify that all changes have taken place:

    SHOW THIS_CONTROLLER

    SHOW OTHER_CONTROLLER

13. Plug in the Fibre Channel cable from the first adapter in the first host into the hub. Enter a SHOW *connections* command to view the connection table:

    SHOW CONNECTIONS

    The first connection will have one or more entries in the connection table. (How many there are depends on cabling configuration.) Each connection will have a default name of the form !NEWCONxx, where xx is a number representing the order in which the connection was added to the connection table.

14. Rename the connection(s) to something meaningful to the system and easy to remember. (For a recommended naming convention, see "Naming Connections," page 1–11.) For example, to assign the name ANGEL1A1 to connection !NEWCON01, enter:

    RENAME !NEWCON01 ANGEL1A1

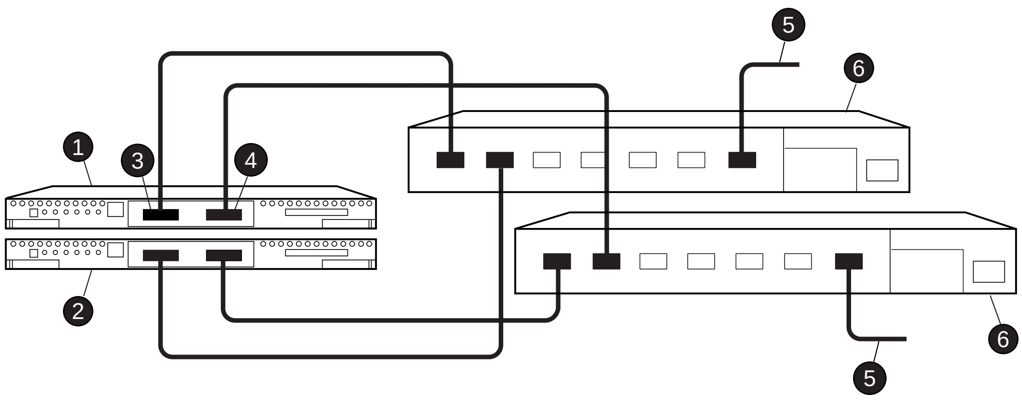
Configuration Procedure for Multiple-Bus Failover Mode

In multiple-bus failover mode, there are two or more paths between each unit and its host(s). There are three ways to configure a controller pair in multiple-bus mode. The following sections show the three cabling options. For simplicity, the drawings show only the two host connections required for one host, but additional host connections are possible.

Cabling Controllers in Multiple-Bus Failover Mode

The most common cabling option is shown in Figure 4–7. This option is relatively simple: one hub connects to port 1 of both controllers, and the second hub connects to port 2 of both controllers. This option provides path redundancy between the hubs and the controllers (there is a path from each hub to each controller).

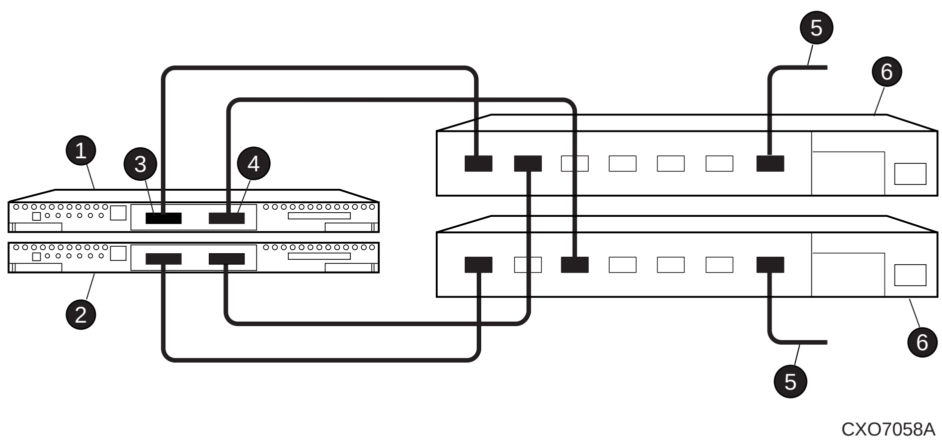
> **NOTE:** It is a good idea to plug only the controller cables into the hub. The host cables are plugged into the hub as part of the configuration procedure ("CLI Configuration Procedure for Multiple-Bus Failover Mode," page 4–15).



❶ controller A
❷ controller B
❸ host port 1
❹ host port 2
❺ cable from the hub to the host FC adapter
❻ hub

Figure 4–7. Multiple-bus failover cabling, option 1

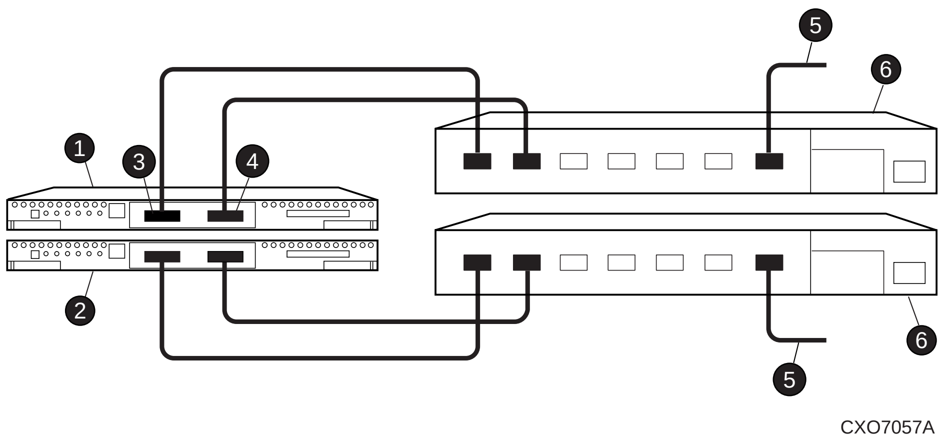
Another cabling option is shown in Figure 4–8. This option is more complex: each hub connects to port 1 of one controller and port 2 of the other controller. This option provides path redundancy between the hubs and the controllers (there is a path from each hub to each controller).



❶ controller A
❷ controller B
❸ host port 1
❹ host port 2
❺ cable from the hub to the host FC adapter
❻ hub

Figure 4–8. Multiple-bus failover cabling, option 2

The third cabling option is shown in Figure 4–9. This option is simple: each hub connects to both ports of one controller. This option does not provide path redundancy between the hubs and the controllers; if a hub fails, the controller to which it is attached becomes inaccessible.



❶ controller A
❷ controller B
❸ host port 1
❹ host port 2
❺ cable from the hub to the host FC adapter
❻ hub

Figure 4–9. Multiple-bus failover cabling, option 3 (limited path redundancy)

CLI Configuration Procedure for Multiple-Bus Failover Mode

1. Apply power to the subsystem.

   The powerup sequence takes approximately 45 seconds. At the end of the powerup sequence, the audible alarm on the EMU will sound and the EMU error LED will be solidly lit. Turn off the alarm by pressing once and releasing the reset button on the EMU.

   The alarm will go off, but the LED will remain on until the controllers are bound into failover mode. The CLI will display a copyright notice and a last event report from the “other” controller.

2. Attach the maintenance terminal to the top controller, as shown in Figure 4–1.

3. If working with anything but factory-new controllers, enter the following command to remove any failover mode that may have been previously configured:

   SET NOFAILOVER

4. Enter the following command to stop the CLI from reporting a misconfiguration error resulting from having no failover mode specified:

   CLEAR CLI

5. Put the controller pair into multiple-bus failover mode, using the following command:

   SET MULTIBUS COPY=THIS

   The copy qualifier specifies where the good copy of the array configuration is. Initially, the copy qualifier is meaningless because there is no array configuration yet, but it is part of the command syntax and must be specified.

   NOTE: If there is configuration information that you want to keep, which will probably be the case if the controllers are not factory-new, set COPY to the controller that contains the good configuration information

   When the command is entered, the "other" controller (the one to which the serial line is not attached) will restart. The restart may set off the audible alarm of the EMU. To silence the alarm, press once and release the button on the EMU. The alarm will stop, but the LED will remain on until the controllers bind into transparent failover mode. The binding process takes about 15 seconds. The CLI will print out a last event report from the "other" controller. This just indicates that the "other" controller restarted. The CLI will continue reporting this condition until cleared with the following command:

   CLEAR CLI

6. Set up mirrored cache, if desired, for the controller pair using the following command:

   SET THIS MIRRORED_CACHE

   This command causes a restart, so the EMU audible alarm may sound.

7. Set the topology for both ports of both controllers:

   NOTE: Some operating systems require topology to be set to *LOOP_HARD*.

   SET THIS PORT_1_TOPOLOGY=LOOP_SOFT

   SET THIS PORT_2_TOPOLOGY=LOOP_SOFT

   SET OTHER PORT_1_TOPOLOGY=LOOP_SOFT

   SET OTHER PORT_2_TOPOLOGY=LOOP_SOFT

NOTE: If the controllers are not factory-new, they may have another topology set, in which case these commands will result in an error message. If this happens, take all ports offline first, then reset the topology:

SET THIS PORT_1_TOPOLOGY=OFFLINE

SET THIS PORT_2_TOPOLOGY=OFFLINE

SET OTHER PORT_1_TOPOLOGY=OFFLINE

SET OTHER PORT_2_TOPOLOGY=OFFLINE

SET THIS PORT_1_TOPOLOGY=LOOP_SOFT

SET THIS PORT_2_TOPOLOGY=LOOP_SOFT

SET OTHER PORT_1_TOPOLOGY=LOOP_SOFT

SET OTHER PORT_2_TOPOLOGY=LOOP_SOFT

8. Set the time on “this” controller using the following syntax:

   SET THIS TIME=DD-MMM-YYYY:HH:MM:SS

9. Use the FRUTIL utility to set up the battery discharge timer in “this” controller. Enter the following command to start FRUTIL:

   RUN FRUTIL

   When FRUTIL asks if you intend to replace the battery, answer "y":

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

   FRUTIL will print out a procedure, but won't give you a prompt. Ignore the procedure and just press return.

10. Move the serial cable to the lower controller, and repeat step 9.

11. Set up any additional optional controller settings, such as changing the CLI prompt. See the SET *this_controller/other_controller* command in the *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide* for the format of optional settings.

12. Enter a SHOW *this_controller* command and a SHOW *other_controller* command to verify that all changes have taken place:

    SHOW THIS_CONTROLLER

    SHOW OTHER_CONTROLLER

13. Plug in the Fibre Channel cable from the first adapter in the first host into the hub. Enter a SHOW *connections* command to view the connection table:

    SHOW CONNECTIONS

The first connection will have one or more entries in the connection table. (How many there are depends on cabling configuration.) Each connection will have a default name of the form !NEWCONxx, where xx is a number representing the order in which the connection was added to the connection table.

14. Rename the connection(s) to something meaningful to the system and easy to remember. (For a recommended naming convention, see “Naming Connections,” page 1–11.) For example, to assign the name ANGEL1A1 to connection !NEWCON01, enter:

    RENAME !NEWCON01 ANGEL1A1

Configuring Devices

The devices on the device bus can be configured either manually or by the CONFIG utility. The CONFIG utility is easier.

Invoke CONFIG with the following command:

RUN CONFIG

CONFIG takes about 2 minutes to discover and map the configuration of a completely populated pedestal. The message that CONFIG is running looks like the following:

```
Config Local Program Invoked
Config is building its tables and determining what devices exist
on the subsystem. Please be patient.
add disk DISK10000   1 0 0
add disk DISK10100   1 1 0
add disk DISK20000   2 0 0
add disk DISK20100   2 1 0
...
Config - Normal Termination
```

Configuring a Stripeset

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 DISKNNNNN DISKNNNNN

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

   INITIALIZE STRIPESET-NAME SWITCH

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

   SHOW STRIPESET-NAME

4. Assign the stripeset a unit number to make it accessible by the host(s). See “Assigning Unit Numbers and Unit Qualifiers” on page 4–23.

Example

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

```
ADD STRIPESET STRIPE1 DISK10000 DISK20000 DISK30000
INITIALIZE STRIPE1 CHUNKSIZE=128
SHOW STRIPE1
```

See Chapter 2 for more information on stripeset switches and values.

Configuring a Mirrorset

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 DISKNNNNN DISKNNNNN SWITCHES

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

   INITIALIZE MIRRORSET-NAME SWITCHES

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

   SHOW MIRRORSET-NAME

4. Assign the mirrorset a unit number to make it accessible by the host(s). See “Assigning Unit Numbers and Unit Qualifiers” on page 4–23.

Example

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

```
ADD MIRRORSET MIRR1 DISK10000 DISK20000
INITIALIZE MIRR1
SHOW MIRR1
```

See Chapter 2 for more information on stripeset switches and values.

Configuring a RAIDset

To configure a RAIDset:

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

   Use the following syntax to create a RAIDset:

   ```
   ADD RAIDSET RAIDSET-NAME DISKNNNNN DISKNNNNN DISKNNNNN SWITCH
   ```

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

   ```
   INITIALIZE RAIDSET-NAME SWITCH
   ```

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

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

   ```
   SHOW RAIDSET-NAME
   ```

4. Assign the RAIDset a unit number to make it accessible by the host(s). See "Assigning Unit Numbers and Unit Qualifiers" on page 4–23.

Example

The following example shows the commands you would use to create RAID1, a three-member RAIDset:

```
ADD RAIDSET RAID1 DISK10000 DISK20000 DISK30000
INITIALIZE RAID1
SHOW RAID1
```

See Chapter 2 for more information on RAIDset switches and values.

Configuring a Striped Mirrorset

To configure a striped mirrorset:

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

   ADD STRIPESET STRIPESET-NAME MIRRORSET-1 MIRRORSET-2....MIRRORSET-N
3. Initialize the stripeset. If you want to set any Initialize switches, you must do so in this step. Use the following command:

   INITIALIZE STRIPESET-NAME SWITCH
4. Verify the striped mirrorset configuration and switches. Use the following command:

   SHOW STRIPESET-NAME
5. Assign the stripeset mirrorset a unit number to make it accessible by the host(s). See “Assigning Unit Numbers and Unit Qualifiers” on page 4–23.

Example

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

```
ADD MIRRORSET MIRR1 DISK10000 DISK20000
ADD MIRRORSET MIRR2 DISK30000 DISK40000
ADD MIRRORSET MIRR3 DISK50000 DISK60000
ADD STRIPESET STRIPE1 MIRR1 MIRR2 MIRR3
INITIALIZE STRIPE1 CHUNKSIZE=DEFAULT
SHOW STRIPE1
```

See Chapter 2 for more information on stripeset and mirrorset switches and values.

Configuring a Single-Disk Unit

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

1. Initialize the disk drive using the following syntax:

   INITIALIZE DISKNNN SWITCH

2. Assign the disk a unit number to make it accessible by the host(s). See “Assigning Unit Numbers and Unit Qualifiers” on page 4–23.
3. Verify the configuration using the following command:

   SHOW DEVICES

Configuring a Partition

To partition a storageset or disk drive:

1. Initialize the storageset or disk drive. If you want to set any initialization switches, you must do so in this step. Use which syntax is appropriate:

   INITIALIZE STORAGESET-NAME SWITCHES

   or

   INITIALIZE DISK-NAME SWITCHES

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

   or

   CREATE_PARTITION DISK-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

   or

   SHOW DISK-NAME

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

4. Assign the partition a unit number to make it accessible by the host(s). See “Assigning Unit Numbers and Unit Qualifiers” on page 4–23.

Example

The following example shows the commands you would use to create RAID1, a three-member RAIDset, then partition it into two storage units:

```
ADD RAIDSET RAID1 DISK10000 DISK20000 DISK30000
INITIALIZE RAID1
CREATE_PARTITION RAID1 SIZE=25
CREATE_PARTITION RAID1 SIZE=LARGEST
SHOW RAID1
```

See Chapter 2 for more information on partition switches and values.

Assigning Unit Numbers and Unit Qualifiers

Each storageset, partition, or single (JBOD) disk must be assigned a unit number for the host to access. As the units are added, their properties can be specified through use of command qualifiers, which are discussed in detail under the ADD UNIT command in the *Compaq StorageWorks HSG80 Array Controller ACS Version 8.5 CLI Reference Guide*.

Each unit can be reserved for the exclusive use of a host or group of hosts. See “Restricting Host Access in Transparent Failover Mode,” page 1–20 and “Restricting Host Access in Multiple-Bus Failover Mode,” page 1–24.

Assigning a Unit Number to a Storageset

To assign a unit number to a storageset, use the following syntax:

```
ADD UNIT UNIT-NUMBER STORAGESET-NAME
```

Example:

To assign unit D102 to RAIDset R1 use the following command:

```
ADD UNIT D102 R1
```

Assigning a Unit Number to a Single (JBOD) Disk

To assign a unit number to a single (JBOD) disk, use the following syntax:

```
ADD UNIT UNIT-NUMBER DISK-NAME
```

Example:

To assign unit D4 to disk20300, use the following command:

```
ADD UNIT D4 DISK20300
```

Assigning a Unit Number to a Partition

To assign a unit number to a partition, use the following syntax:

```
ADD UNIT UNIT-NUMBER STORAGESET-NAME PARTITION=PARTITION-NUMBER
```

Example:

To assign unit D100 to partition 3 of mirrorset mirr1, use the following command:

```
ADD UNIT D100 MIRROR1 PARTITION=3
```

Preferring Units in Multiple-Bus Failover Mode

In multiple-bus failover mode, individual units can be preferred to a specific controller. To prefer, for example, unit D102 to "this controller," use the following command:

```
SET D102 PREFERRED_PATH=THIS
```

RESTART commands must be issued to both controllers for this command to take effect:

```
RESTART THIS_CONTROLLER
RESTART OTHER_CONTROLLER
```

NOTE: The controllers need to restart together for the preferred settings to take effect. The RESTART *other_controller* command must be entered immediately after the RESTART *this_controller* command.

Configuration Options

There are many options to choose from when configuring a subsystem. This section shows how to set up some of the more common ones.

Changing the CLI Prompt

To change the CLI prompt, enter a 1- to 16- character string as the new prompt, as follows:

```
SET THIS_CONTROLLER PROMPT = "NEW PROMPT"
```

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

```
SET OTHER_CONTROLLER PROMPT = "NEW PROMPT"
```

It is suggested that the prompts reflect something about the controllers. For example, if the subsystem is the third one in a lab, give the top controller a prompt like LAB3A and the bottom controller, LAB3B.

Adding Disk Drives

If you add new disk drives to the subsystem, the disk drives must be added to the controllers' list of known devices:

- To add one new disk drive to the list of known devices, enter the following command:

  ```
  ADD DISK DISKNNN PTL-LOCATION
  ```

- To add several new disk drives to the list of known devices, enter the following command:

  ```
  RUN CONFIG
  ```

Adding a Disk Drive to the Spareset

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

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

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

1. To add the disk drive to the controller's spareset list. Use the following command:

   ```
   ADD SPARESET DISKNNNNN
   ```

   Repeat this step for each disk drive you want to add to the spareset:
2. Verify the contents of the spareset using the following command:

   ```
   SHOW SPARESET
   ```

Example

The following example shows the commands for adding DISK60000 and DISK60100 to the spareset.

```
ADD SPARESET DISK60000
ADD SPARESET DISK60100
SHOW SPARESET
```

Removing a Disk Drive from the Spareset

You can delete disks in the spareset if you need to use them elsewhere in your subsystem. To remove a disk drive from the spareset:

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

   ```
   SHOW SPARESET
   ```

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

   ```
   DELETE SPARESET DISKNNNNN
   ```

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

   ```
   SHOW SPARESET
   ```

Enabling Autospare

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

To enable autospare use the following command:

```
SET FAILEDSET AUTOSPARE
```

To disable autospare use the following command:

```
SET FAILEDSET NOAUTOSPARE
```

During initialization, AUTOSPARE checks to see if the new disk drive contains metadata. Metadata is information the controller writes on the disk when the disk is configured into a storageset. Therefore, the presence of metadata indicates the disk drive belongs to, or has been used by, a storageset. If the disk drive contains metadata, initialization stops. (A new disk drive will not contain metadata but a repaired or re-used disk drive might. To erase metadata from a disk drive, add it to the controller's list of devices, then set it to be transportable and initialize it.)

Deleting a Storageset

NOTE: If the storageset you are deleting is partitioned, you must delete each partitioned unit before you can delete the storageset.

Use the following steps to delete a storageset:

1. Show the configuration using the following command:

   SHOW STORAGESETS

2. Delete the unit number that uses the storageset. Use the following command:

   DELETE UNIT-NUMBER

3. Delete the storageset. Use the following command:

   DELETE STORAGESET-NAME

4. Verify the configuration using the following command:

   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. Remember to update the storageset's profile when you change its switches.

Displaying the Current Switches

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

SHOW STORAGESET-NAME OR DEVICE-NAME FULL

Changing RAIDset and Mirrorset Switches

Use the SET storageset-name command to change the RAIDset and Mirrorset switches associated with an existing storageset. For example, the following command changes the replacement policy for RAIDset RAID1 to BEST_FIT:

```
SET RAID1 POLICY=BEST_FIT
```

Changing Device Switches

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

```
SET DISK10000 TRANSPORTABLE
```

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

Changing Initialize Switches

The initialization 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 the characteristics of a unit. For example, the following command enables write protection for unit D100:

```
SET D100 WRITE_PROTECT
```

Chapter **5**

Other Procedures

This chapter describes some common procedures that are not part of configuration.

Backing Up the 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 use the SHOW *this_controller FULL* command to find out if the save configuration feature is active and which devices are being used to store the configuration. The display includes a line that indicates status and how many devices have copies of the configuration, as shown in the following example.

```
SHOW THIS_CONTROLLER FULL
```

The last line of the status the CLI returns shows how many devices the configuration is backed up on.

Also, the SHOW *devices FULL* command shows which disk drive are set up to back up the configuration. Enter this command:

```
SHOW DEVICES FULL
```

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.

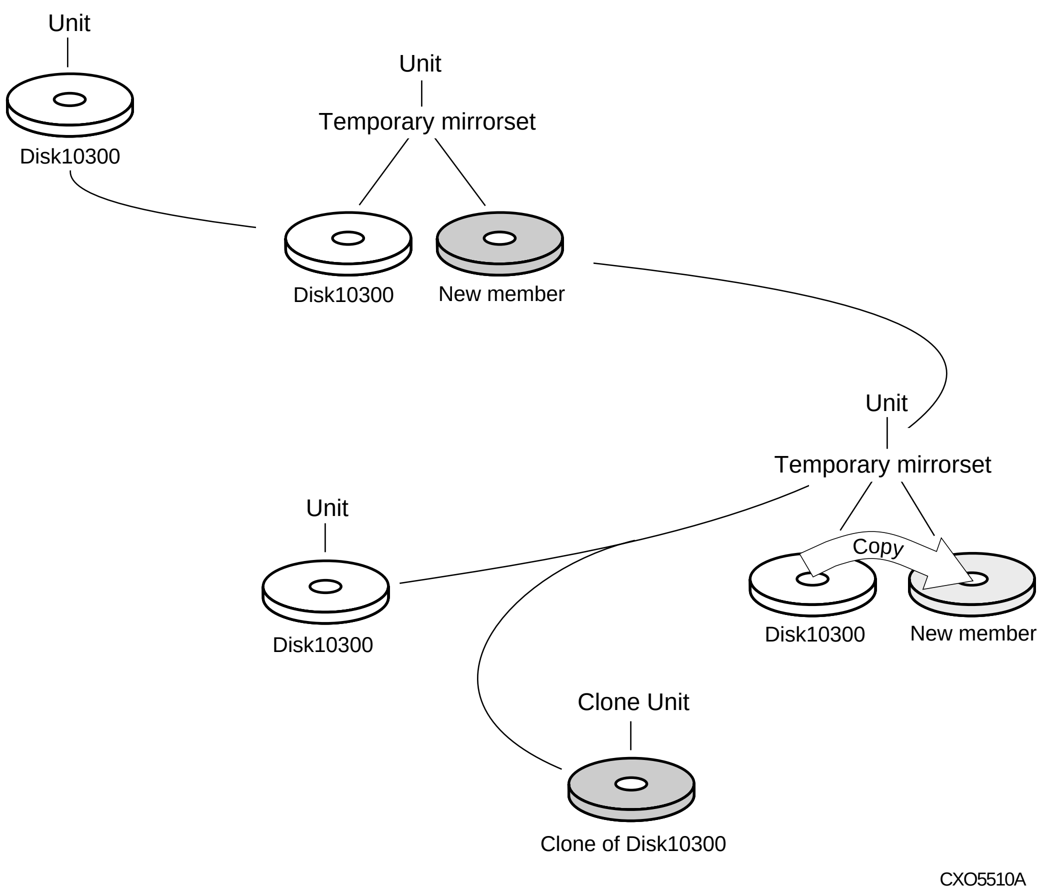


Figure 5–1. Steps the CLONE utility follows for duplicating unit members

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

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

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

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

Example

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

```
RUN CLONE

CLONE LOCAL PROGRAM INVOKED

UNITS AVAILABLE FOR CLONING:

        98

ENTER UNIT TO CLONE ? 98

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

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

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

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

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

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

DEVICES AVAILABLE FOR CLONE TARGETS:

DISK20200 (SIZE=832317)

DISK20300 (SIZE=832317)

DISK30100 (SIZE=832317)

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

MIRROR DISK10300 C_MA

SET C_MA NOPOLICY

SET C_MA MEMBERS=2

SET C_MA REPLACE=DISK20200

DEVICES AVAILABLE FOR CLONE TARGETS:

DISK20300 (SIZE=832317)

DISK30100 (SIZE=832317)

USE AVAILABLE DEVICE DISK10400(SIZE=832317) FOR MEMBER DISK10000(SIZE=832317)
```

```
(Y,N) [Y] ? Y
MIRROR DISK10000 C_MB
SET C_MB NOPOLICY
SET C_MB MEMBERS=2
SET C_MB REPLACE=DISK10400
COPY IN PROGRESS FOR EACH NEW MEMBER. PLEASE BE PATIENT...
.
.
COPY FROM DISK10300 TO DISK20200 IS 100% COMPLETE
COPY FROM DISK10000 TO DISK10400 IS 100% COMPLETE

PRESS RETURN WHEN YOU WANT THE NEW UNIT TO BE CREATED
REDUCE  DISK20200 DISK10400
UNMIRROR DISK10300
UNMIRROR DISK10000
ADD MIRRORSET C_MA      DISK20200
ADD MIRRORSET C_MB      DISK10400
ADD STRIPESET C_ST1  C_MA C_MB
INIT C_ST1     NODESTROY
ADD UNIT D99 C_ST1
D99 HAS BEEN CREATED. IT IS A CLONE OF D98.
CLONE - NORMAL TERMINATION
```

Moving Storagesets

You can move a storageset from one subsystem to another without destroying its data, as shown in Figure 5–1. You also can follow the steps in this section to move a storageset to a new location within the same subsystem.

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

You can use the procedure in this section to migrate wide SCSI devices from an HSZ70 controller in a BA370 rack-mountable enclosure to an HSG80 environment. See the release notes for the version of your controller's software for information on which drives can be supported.

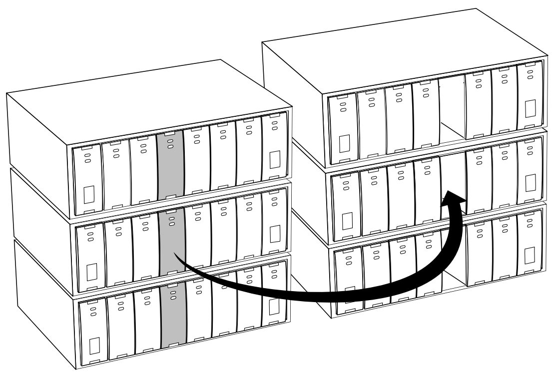


Figure 5–2. Moving a storageset from one subsystem to another

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

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

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

   SHOW STORAGESET-NAME

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

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

   LOCATE DISK-NAME

   To cancel the locate command, enter the following:

   LOCATE CANCEL

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

   DELETE UNIT-NUMBER

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

   DELETE STORAGESET-NAME

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

   DELETE DISK-NAME

   DELETE DISK-NAME

   DELETE DISK-NAME

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

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

   ADD DISK DISK-NAME PTL-LOCATION

   ADD DISK DISK-NAME PTL-LOCATION

   ADD DISK DISK-NAME PTL-LOCATION

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

   ADD STORAGESET-NAME DISK-NAME DISK-NAME

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

   ADD UNIT UNIT-NUMBER STORAGESET-NAME

Example

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

DELETE D100

DELETE RAID99

DELETE DISK10000

DELETE DISK20000

DELETE DISK30000

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

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

Appendix A

Subsystem Profile Templates

This appendix contains storageset profiles you can copy and 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:

Storageset Profile

Type of Storageset:

_____ Mirrorset __X_ RAIDset _____ Stripeset _____ Striped Mirrorset ____ JBOD

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:

Caching	Access by following hosts enabled
Read caching__________	____
Read-ahead caching_____	____
Write-back caching______	____
Write-through caching____	____

Storage Map Template 1

Use this template for:

- single-enclosure subsystems
- first enclosure of multiple-enclosure subsystems

Port

	1	2	3	4	5	6		Targets
Power Supply	D10300	D20300	D30300	D40300	D50300	D60300	Power Supply	3
Power Supply	D10200	D20200	D30200	D40200	D50200	D60200	Power Supply	2
Power Supply	D10100	D20100	D30100	D40100	D50100	D60100	Power Supply	1
Power Supply	D10000	D20000	D30000	D40000	D50000	D60000	Power Supply	0

Storage Map Template 2

Use this template for the second enclosure of multiple-enclosure subsystems.

Port

	1	2	3	4	5	6		Targets
Power Supply	D11100	D21100	D31100	D41100	D51100	D61100	Power Supply	11
Power Supply	D11000	D21000	D31000	D41000	D51000	D61000	Power Supply	10
Power Supply	D10900	D20900	D30900	D40900	D50900	D60900	Power Supply	9
Power Supply	D10800	D20800	D30800	D40800	D50800	D60800	Power Supply	8

Storage Map Template 3

Use this template for the third enclosure of multiple-enclosure subsystems.

	Port 1	Port 2	Port 3	Port 4	Port 5	Port 6		Targets
Power Supply	D11500	D21500	D31500	D41500	D51500	D61500	Power Supply	15
Power Supply	D11400	D21400	D31400	D41400	D51400	D61400	Power Supply	14
Power Supply	D11300	D21300	D31300	D41300	D51300	D611300	Power Supply	13
Power Supply	D11200	D21200	D31200	D41200	D51200	D61200	Power Supply	12

Glossary

This glossary defines terms pertaining to the HSG80 Fibre Channel array controller. It is not a comprehensive glossary of computer terms.

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

ACS *See* array controller software.

AL_PA *See* arbitrated loop physical address.

ANSI Pronounced “ann-see.” Acronym for the American National Standards Institute. An organization who develops standards used voluntarily by many manufacturers within the USA. ANSI is not a government agency.

arbitrated loop physical address Abbreviated AL_PA. A one-byte value used to identify a port in an Arbitrated Loop topology.

array controller *See* controller.

array controller software Abbreviated ACS. Software contained on a removable ROM program card that provides the operating system for the array controller.

asynchronous Pertaining to events that are scheduled as the result of a signal asking for the event; pertaining to that which is without any specified time relation. *See also* synchronous.

autospare A controller feature that automatically replaces a failed disk drive. To aid the controller in automatically replacing failed disk drives, you can enable the AUTOSPARE switch for the failedset causing physically replaced disk drives to be automatically placed into the spareset. Also called “autonewspare.”

bad block A data block that contains a physical defect.

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

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

BBR *See* bad block replacement.

BIST *See* built-in self-test.

bit A single binary digit having a value of either 0 or 1. A bit is the smallest unit of data a computer can process.

block Also called a sector. The smallest collection of consecutive bytes addressable on a disk drive. In integrated storage elements, a block contains 512 bytes of data, error codes, flags, and the block’s address header.

bootstrapping A method used to bring a system or device into a defined state by means of its own action. For example, a machine routine whose first few instructions are enough to bring the rest of the routine into the computer from an input device.

built-in self-test A diagnostic test performed by the array controller software on the controller’s policy processor.

byte A binary character string made up of 8 bits operated on as a unit.

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

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

channel An interface which allows high speed transfer of large amounts of data. Another term for a SCSI bus. *See also* SCSI.

chunk A block of data written by the host.

chunk size The number of data blocks, assigned by a system administrator, written to the primary RAIDset or stripeset member before the remaining data blocks are written to the next RAIDset or stripeset member.

CLCP An abbreviation for code-load code-patch utility.

CLI *See* command line interpreter.

coax A two-conductor wire in which one conductor completely wraps the other with the two separated by insulation.

coaxial cable *See* coaxial cable.

cold swap A method of device replacement that requires the entire subsystem to be turned off before the device can be replaced. *See also* hot swap and warm swap.

command line interpreter The configuration interface to operate the controller software.

configuration file A file that contains a representation of a storage subsystem's configuration.

container 1) Any entity that is capable of storing data, whether it is a physical device or a group of physical devices. (2) A virtual, internal controller structure representing either a single disk or a group of disk drives linked as a storageset. Stripesets and mirrorsets are examples of storageset containers the controller uses to create units.

controller A hardware device that, with proprietary software, facilitates communications between a host and one or more devices organized in an array. HS family controllers are examples of array controllers.

copying A state in which data to be copied to the mirrorset is inconsistent with other members of the mirrorset. *See also* normalizing.

copying member Any member that joins the mirrorset after the mirrorset is created is regarded as a copying member. Once all the data from the normal member (or members) is copied to a normalizing or copying member, the copying member then becomes a normal member. *See also* normalizing member.

CSR An acronym for control and status register.

DAEMON Pronounced "demon." A program usually associated with a UNIX systems that performs a utility (housekeeping or maintenance) function without being requested or even known of by the user. A daemon is a diagnostic and execution monitor.

data center cabinet A generic reference to large DIGITAL subsystem cabinets, such as the SW600-series and 800-series cabinets in which StorageWorks components can be mounted.

data striping The process of segmenting logically sequential data, such as a single file, so that segments can be written to multiple physical devices (usually disk drives) in a round-robin fashion. This technique is useful if the processor is capable of reading or writing data faster than a single disk can supply or accept the data. While data is being transferred from the first disk, the second disk can locate the next segment.

device *See* node and peripheral device.

differential I/O module A 16-bit I/O module with SCSI bus converter circuitry for extending a differential SCSI bus. *See also* I/O module.

differential SCSI bus A bus in which a signal's level is determined by the potential difference between two wires. A differential bus is more robust and less subject to electrical noise than is a single-ended bus.

DIMM Dual inline Memory Module.

dirty data The write-back cached data that has not been written to storage media, even though the host operation processing the data has completed.

DMA Direct Memory Access.

DOC DWZZA-On-a-Chip. An NCR53C120 SCSI bus extender chip used to connect a SCSI bus in an expansion cabinet to the corresponding SCSI bus in another cabinet.

driver A hardware device or a program that controls or regulates another device. For example, a device driver is a driver developed for a specific device that allows a computer to operate with the device, such as a printer or a disk drive.

dual-redundant configuration A controller configuration consisting of two active controllers operating as a single controller. If one controller fails, the other controller assumes control of the failing controller's devices.

dual-simplex A communications protocol that allows simultaneous transmission in both directions in a link, usually with no flow control.

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

ECB External cache battery. The unit that supplies backup power to the cache module in the event the primary power source fails or is interrupted.

ECC Error checking and correction.

EDC Error detection code.

EIA	The abbreviation for Electronic Industries Association. EIA is a standards organization specializing in the electrical and functional characteristics of interface equipment. Same as Electronic Industries Association.
EMU	Environmental monitoring unit. A unit that provides increased protection against catastrophic failures. Some subsystem enclosures include an EMU which works with the controller to detect conditions such as failed power supplies, failed blowers, elevated temperatures, and external air sense faults. The EMU also controls certain cabinet hardware including DOC chips, alarms, and fan speeds.
ESD	Electrostatic discharge. The discharge of potentially harmful static electrical voltage as a result of improper grounding.
extended subsystem	A subsystem in which two cabinets are connected to the primary cabinet.
external cache battery	*See* ECB.
F_Port	A port in a fabric where an N_Port or NL_Port may attach.
fabric	A group of interconnections between ports that includes a fabric element.
failedset	A group of failed mirrorset or RAIDset devices automatically created by the controller.
failover	The process that takes place when one controller in a dual-redundant configuration assumes the workload of a failed companion controller. Failover continues until the failed controller is repaired or replaced.
FC–AL	The Fibre Channel Arbitrated Loop standard.
FC–ATM	ATM AAL5 over Fibre Channel
FC–FG	Fibre Channel Fabric Generic Requirements
FG–FP	Fibre Channel Framing Protocol (HIPPI on FC)
FC-GS-1	Fibre Channel Generic Services-1
FC–GS-2	Fibre Channel Generic Services-2
FC–IG	Fibre Channel Implementation Guide
FC–LE	Fibre Channel Link Encapsulation (ISO 8802.2)
FC–PH	The Fibre Channel Physical and Signaling standard.

FC–SB	Fibre Channel Single Byte Command Code Set
FC–SW	Fibre Channel Switched Topology and Switch Controls
FCC	Federal Communications Commission. The federal agency responsible for establishing standards and approving electronic devices within the United States.
FCC Class A	This certification label appears on electronic devices that can only be used in a commercial environment within the United States.
FCC Class B	This certification label appears on electronic devices that can be used in either a home or a commercial environment within the United States.
FCP	The mapping of SCSI-3 operations to Fibre Channel.
FDDI	Fiber Distributed Data Interface. An ANSI standard for 100 megabaud transmission over fiber optic cable.
FD SCSI	The fast, narrow, differential SCSI bus with an 8-bit data transfer rate of 10 MB/s. *See also* FWD SCSI and SCSI.
fiber	A fiber or optical strand. Spelled ***fibre*** in Fibre Channel.
fiber optic cable	A transmission medium designed to transmit digital signals in the form of pulses of light. Fiber optic cable is noted for its properties of electrical isolation and resistance to electrostatic contamination.
FL_Port	A port in a fabric where N_Port or an NL_Port may be connected.
flush	The act of writing dirty data from cache to a storage media.
FMU	Fault management utility.
forced errors	A data bit indicating a corresponding logical data block contains unrecoverable data.
frame	An invisible unit used to transfer information in Fibre Channel.
FRU	Field replaceable unit. A hardware component that can be replaced at the customer's location by DIGITAL service personnel or qualified customer service personnel.
full duplex (n)	A communications system in which there is a capability for 2-way transmission and acceptance between two sites at the same time.
full duplex (adj)	Pertaining to a communications method in which data can be transmitted and received at the same time.

FWD SCSI A fast, wide, differential SCSI bus with a maximum 16-bit data transfer rate of 20 MB/s. *See also* SCSI and FD SCSI.

GBIC Gigabit Interface Converter. The devices that are inserted into the ports of the Fibre Channel switch and that hold the Fibre Channel cables.

GLM Gigabit link module

giga A prefix indicating a billion (10^9) units, as in gigabaud or gigabyte.

gigabaud An encoded bit transmission rate of one billion (10^9) bits per second.

gigabyte A value normally associated with a disk drives storage capacity, meaning a billion (10^9) bytes. The decimal value 1024 is usually used for one thousand.

half-duplex (adj) Pertaining to a communications system in which data can be either transmitted or received but only in one direction at one time.

hard address The AL_PA which an NL_Port attempts to acquire during loop initialization.

heterogeneous host support Also called *noncooperating host support*.

HIPPI–FC Fibre Channel over HIPPI

host The primary or controlling computer to which a storage subsystem is attached.

host adapter A device that connects a host system to a SCSI bus. The host adapter usually performs the lowest layers of the SCSI protocol. This function may be logically and physically integrated into the host system.

hot disks A disk containing multiple hot spots. Hot disks occur when the workload is poorly distributed across storage devices which prevents optimum subsystem performance. *See also* hot spots.

hot spots A portion of a disk drive frequently accessed by the host. Because the data being accessed is concentrated in one area, rather than spread across an array of disks providing parallel access, I/O performance is significantly reduced. *See 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.

IBR	Initial Boot Record.
ILF	Illegal function.
INIT	Initialize input and output.
initiator	A SCSI device that requests an I/O process to be performed by another SCSI device, namely, the SCSI target. The controller is the initiator on the device bus. The host is the initiator on the host bus.
instance code	A four-byte value displayed in most text error messages and issued by the controller when a subsystem error occurs. The instance code indicates when during software processing the error was detected.
interface	A set of protocols used between components, such as cables, connectors, and signal levels.
I/O	Refers to input and output functions.
I/O driver	The set of code in the kernel that handles the physical I/O to a device. This is implemented as a fork process. Same as driver.
I/O interface	*See* interface.
I/O module	A 16-bit SBB shelf device that integrates the SBB shelf with either an 8-bit single ended, 16-bit single-ended, or 16-bit differential SCSI bus.
I/O operation	The process of requesting a transfer of data from a peripheral device to memory (or visa versa), the actual transfer of the data, and the processing and overlaying activity to make both of those happen.
IPI	Intelligent Peripheral Interface. An ANSI standard for controlling peripheral devices by a host computer.
IPI-3 Disk	Intelligent Peripheral Interface Level 3 for Disk
IPI-3 Tape	Intelligent Peripheral Interface Level 3 for Tape
JBOD	Just a bunch of disks. A term used to describe a group of single-device logical units.
kernel	The most privileged processor access mode.
LBN	Logical Block Number.
L_port	A node or fabric port capable of performing arbitrated loop functions and protocols. NL_Ports and FL_Ports are loop-capable ports.

LED Light Emitting Diode.

link A connection between two Fibre Channel ports consisting of a transmit fibre and a receive fibre.

logical block number *See* LBN.

local connection A connection to the subsystem using either its serial maintenance port or the host's SCSI bus. A local connection enables you to connect to one subsystem controller within the physical range of the serial or host SCSI cable.

local terminal A terminal plugged into the EIA-423 maintenance port located on the front bezel of the controller. *See also* maintenance terminal.

logical bus A single-ended bus connected to a differential bus by a SCSI bus signal converter.

logical unit A physical or virtual device addressable through a target ID number. LUNs use their target's bus connection to communicate on the SCSI bus.

logical unit number A value that identifies a specific logical unit belonging to a SCSI target ID number. A number associated with a physical device unit during a task's I/O operations. Each task in the system must establish its own correspondence between logical unit numbers and physical devices.

logon Also called login. A procedure whereby a participant, either a person or network connection, is identified as being an authorized network participant.

loop *See* arbitrated loop.

loop_ID A seven-bit value numbered contiguously from zero to 126-decimal and represent the 127 legal AL_PA values on a loop (not all of the 256 hex values are allowed as AL_PA values per FC-AL.

loop tenancy The period of time between the following events: when a port wins loop arbitration and when the port returns to a monitoring state.

L_Port A node or fabric port capable of performing Arbitrated Loop functions and protocols. NL_Ports and FL_Ports are loop-capable ports.

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

Mbps Approximately one million (10^6) bits per second—that is, megabits per second.

MBps Approximately one million (10^6) bytes per second—that is, megabytes per second.

maintenance terminal An EIA-423-compatible terminal used with the controller. This terminal is used to identify the controller, enable host paths, enter configuration information, and check the controller's status. The maintenance terminal is not required for normal operations.
See also local terminal.

member A container that is a storage element in a RAID array.

metadata The data written to a disk for the purposes of controller administration. Metadata improves error detection and media defect management for the disk drive. It is also used to support storageset configuration and partitioning. Nontransportable disks also contain metadata to indicate they are uniquely configured for StorageWorks environments. Metadata can be thought of as "data about data."

mirroring The act of creating an exact copy or image of data.

mirrorset *See* RAID level 1.

MIST Module Integrity Self-Test.

N_port A port attached to a node for use with point-to-point topology or fabric topology.

NL_port A port attached to a node for use in all three topologies.

network A data communication, a configuration in which two or more terminals or devices are connected to enable information transfer.

node In data communications, the point at which one or more functional units connect transmission lines.

Non-L_Port A Node of Fabric port that is not capable of performing the Arbitrated Loop functions and protocols. N_Ports and F_Ports loop-capable ports.

non-participating mode A mode within an L_Port that inhibits the port from participating in loop activities. L_Ports in this mode continue to retransmit received transmission words but are not permitted to arbitrate or originate frames. An L_Port in non-participating mode may or may not have an AL_PA. *See also* participating mode.

nominal membership The desired number of mirrorset members when the mirrorset is fully populated with active devices. If a member is removed from a mirrorset, the actual number of members may fall below the "nominal" membership.

node	In data communications, the point at which one or more functional units connect transmission lines. In Fibre Channel, a device that has at least one N_Port or NL_Port.
nonredundant controller configuration	(1) A single controller configuration. (2) A controller configuration that does not include a second controller.
normal member	A mirrorset member that, block-for-block, contains the same data as other normal members within the mirrorset. Read requests from the host are always satisfied by normal members.
normalizing	Normalizing is a state in which, block-for-block, data written by the host to a mirrorset member is consistent with the data on other normal and normalizing members. The normalizing state exists only after a mirrorset is initialized. Therefore, no customer data is on the mirrorset.
normalizing member	A mirrorset member whose contents is the same as all other normal and normalizing members for data that has been written since the mirrorset was created or lost cache data was cleared. A normalizing member is created by a normal member when either all of the normal members fail or all of the normal members are removed from the mirrorset. *See also* copying member.
NVM	Non-Volatile Memory. A type of memory where the contents survive power loss. Also sometimes referred to as NVMEM.
OCP	Operator control panel. The control or indicator panel associated with a device. The OCP is usually mounted on the device and is accessible to the operator.
other controller	The controller in a dual-redundant pair that is connected to the controller serving your current CLI session. *See also* this controller.
outbound fiber	One fiber in a link that carries information away from a port.
parallel data transmission	A data communication technique in which more than one code element (for example, bit) of each byte is sent or received simultaneously.
parity	A method of checking if binary numbers or characters are correct by counting the ONE bits. In odd parity, the total number of ONE bits must be odd; in even parity, the total number of ONE bits must be even.
parity bit	A binary digit added to a group of bits that checks to see if errors exist in the transmission.

parity check A method of detecting errors when data is sent over a communications line. With even parity, the number of ones in a set of binary data should be even. With odd parity, the number of ones should be odd.

participating mode A mode within an L_Port that allows the port to participate in loop activities. A port must have a valid AL_PA to be in participating mode.

PCM Polycenter Console Manager.

PCMCIA Personal Computer Memory Card Industry Association. An international association formed to promote a common standard for PC card-based peripherals to be plugged into notebook computers. The card commonly known as a PCMCIA card is about the size of a credit card.

parity A method of checking if binary numbers or characters are correct by counting the ONE bits. In odd parity, the total number of ONE bits must be odd; in even parity, the total number of ONE bits must be even. Parity information can be used to correct corrupted data. RAIDsets use parity to improve the availability of data.

parity bit A binary digit added to a group of bits that checks to see if there are errors in the transmission.

parity RAID *See* RAIDset.

partition A logical division of a container, represented to the host as a logical unit.

peripheral device Any unit, distinct from the CPU and physical memory, that can provide the system with input or accept any output from it. Terminals, printers, tape drives, and disks are peripheral devices.

point-to-point connection A network configuration in which a connection is established between two, and only two, terminal installations. The connection may include switching facilities.

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

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

port_name A 64-bit unique identifier assigned to each Fibre Channel port. The Port_Name is communicated during the logon and port discovery process.

preferred address The AL_PA which an NL_Port attempts to acquire first during initialization.

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

private NL_Port An NL_Port which does not attempt login with the fabric and only communicates with NL_Ports on the same loop.

public NL_Port An NL_Port that attempts login with the fabric and can observe the rules of either public or private loop behavior. A public NL_Port may communicate with both private and public NL_Ports.

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

protocol The conventions or rules for the format and timing of messages sent and received.

PTL Port-Target-LUN. The controller's method of locating a device on the controller's device bus.

PVA module Power Verification and Addressing module.

quiesce The act of rendering bus activity inactive or dormant. For example, "quiesce the SCSI bus operations during a device warm-swap."

RAID Redundant Array of Independent Disks. Represents multiple levels of storage access developed to improve performance or availability or both.

RAID level 0 A RAID storageset that stripes data across an array of disk drives. A single logical disk spans multiple physical disks, allowing parallel data processing for increased I/O performance. While the performance characteristics of RAID level 0 is excellent, this RAID level is the only one that does not provide redundancy. Raid level 0 storagesets are sometimes referred to as stripesets.

RAID level 0+1 A RAID storageset that stripes data across an array of disks (RAID level 0) and mirrors the striped data (RAID level 1) to provide high I/O performance and high availability. This RAID level is alternatively called a striped mirrorset. Raid level 0+1 storagesets are sometimes referred to as striped mirrorsets.

RAID level 1 A RAID storageset of two or more physical disks that maintains a complete and independent copy of the entire virtual disk's data. This type of storageset has the advantage of being highly reliable and extremely tolerant of device failure. Raid level 1 storagesets are sometimes referred to as mirrorsets.

RAID level 3 A RAID storageset that transfers data parallel across the array's disk drives a byte at a time, causing individual blocks of data to be spread over several disks serving as one enormous virtual disk. A separate redundant check disk for the entire array stores parity on a dedicated disk drive within the storageset. *See also* RAID level 5.

RAID level 5 A RAID storageset that, unlike RAID level 3, stores the parity information across all of the disk drives within the storageset. *See also* RAID level 3.

RAID level 3/5 A DIGITAL-developed RAID storageset that stripes data and parity across three or more members in a disk array. A RAIDset combines the best characteristics of RAID level 3 and RAID level 5. A RAIDset is the best choice for most applications with small to medium I/O requests, unless the application is write intensive. A RAIDset is sometimes called parity RAID. Raid level 3/5 storagesets are sometimes referred to as RAIDsets.

RAIDset *See* RAID level 3/5.

RAM Random access memory.

read ahead caching A caching technique for improving performance of synchronous sequential reads by prefetching data from disk.

read caching A cache management method used to decrease the subsystem's response time to a read request by allowing the controller to satisfy the request from the cache memory rather than from the disk drives.

reconstruction The process of regenerating the contents of a failed member's data. The reconstruct process writes the data to a spareset disk and then incorporates the spareset disk into the mirrorset, striped mirrorset, or RAIDset from which the failed member came. *See also* regeneration.

reduced Indicates that a mirrorset or RAIDset is missing one member because the member has failed or has been physically removed.

redundancy The provision of multiple interchangeable components to perform a single function in order to cope with failures and errors. A RAIDset is considered to be redundant when user data is recorded directly to one member and all of the other members include associated parity information.

regeneration (1) The process of calculating missing data from redundant data. (2) The process of recreating a portion of the data from a failing or failed drive using the data and parity information from the other members within the storageset. The regeneration of an entire RAIDset member is called reconstruction. *See also* reconstruction.

request rate The rate at which requests are arriving at a servicing entity.

RFI Radio frequency interference. The disturbance of a signal by an unwanted radio signal or frequency.

replacement policy	The policy specified by a switch with the SET FAILEDSET command indicating whether a failed disk from a mirrorset or RAIDset is to be automatically replaced with a disk from the spareset. The two switch choices are AUTOSPARE and NOAUTOSPARE.
SBB	StorageWorks building block. (1) A modular carrier plus the interface required to mount the carrier into a standard StorageWorks shelf. (2) any device conforming to shelf mechanical and electrical standards installed in a 3.5-inch or 5.25-inch carrier, whether it is a storage device or power supply.
SCSI	Small computer system interface. (1) An ANSI interface standard defining the physical and electrical parameters of a parallel I/O bus used to connect initiators to devices. (2) a processor-independent standard protocol for system-level interfacing between a computer and intelligent devices including hard drives, floppy disks, CD-ROMs, printers, scanners, and others.
SCSI-A cable	A 50-conductor (25 twisted-pair) cable generally used for single-ended, SCSI-bus connections.
SCSI bus signal converter	Sometimes referred to as an adapter. (1) A device used to interface between the subsystem and a peripheral device unable to be mounted directly into the SBB shelf of the subsystem. (2) a device used to connect a differential SCSI bus to a single-ended SCSI bus. (3) A device used to extend the length of a differential or single-ended SCSI bus. *See also* I/O module.
SCSI device	(1) A host computer adapter, a peripheral controller, or an intelligent peripheral that can be attached to the SCSI bus. (2) Any physical unit that can communicate on a SCSI bus.
SCSI device ID number	A bit-significant representation of the SCSI address referring to one of the signal lines, numbered 0 through 7 for an 8-bit bus, or 0 through 15 for a 16-bit bus. *See also* target ID number.
SCSI ID number	The representation of the SCSI address that refers to one of the signal lines numbered 0 through 15.
SCSI-P cable	A 68-conductor (34 twisted-pair) cable generally used for differential bus connections.
SCSI port	(1) Software: The channel controlling communications to and from a specific SCSI bus in the system. (2) Hardware: The name of the logical socket at the back of the system unit to which a SCSI device is connected.

serial transmission A method transmission in which each bit of information is sent sequentially on a single channel rather than simultaneously as in parallel transmission.

service rate The rate at which an entity is able to service requests For example, the rate at which an Arbitrated Loop is able to service arbitrated requests.

signal converter *See* SCSI bus signal converter.

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

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

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

storage array An integrated set of storage devices.

storage array subsystem *See* storage subsystem.

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

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

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

StorageWorks A family of DIGITAL modular data storage products that allow customers to design and configure their own storage subsystems. Components include power, packaging, cabling, devices, controllers, and software. Customers can integrate devices and array controllers in StorageWorks enclosures to form storage subsystems.

StorageWorks systems include integrated SBBs and array controllers to form storage subsystems. System-level enclosures to house the shelves and standard mounting devices for SBBs are also included.

stripe The data divided into blocks and written across two or more member disks in an array.

striped mirrorset *See* RAID level 0+1.

stripeset *See* RAID level 0.

stripe size The stripe capacity as determined by n–1 times the chunksize, where n is the number of RAIDset members.

striping The technique used to divide data into segments, also called chunks. The segments are striped, or distributed, across members of the stripeset. This technique helps to distribute hot spots across the array of physical devices to prevent hot spots and hot disks.

Each stripeset member receives an equal share of the I/O request load, improving performance.

surviving controller The controller in a dual-redundant configuration pair that serves its companion's devices when the companion controller fails.

switch A method that controls the flow of functions and operations in software.

synchronous Pertaining to a method of data transmission which allows each event to operate in relation to a timing signal. *See also* asynchronous.

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

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

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

topology An interconnection scheme that allows multiple Fibre Channel ports to communicate with each other. For example, point-to-point, Arbitrated Loop, and switched fabric are all Fibre Channel topologies.

transfer data rate The speed at which data may be exchanged with the central processor, expressed in thousands of bytes per second.

ULP Upper Layer Protocol.

ULP process A function executing within a Fibre Channel node which conforms to the Upper Layer Protocol (ULP) requirements when interacting with other ULP processes.

Ultra SCSI A Fast-20 SCSI bus. *See also* Wide Ultra SCSI.

unit A container made accessible to a host. A unit may be created from a single disk drive or tape drive. A unit may also be created from a more complex container such as a RAIDset. The controller supports a maximum of eight units on each target. *See also* target and target ID number.

unwritten cached data Sometimes called unflushed data. *See* dirty data.

UPS Uninterruptible power supply. A battery-powered power supply guaranteed to provide power to an electrical device in the event of an unexpected interruption to the primary power supply. Uninterruptible power supplies are usually rated by the amount of voltage supplied and the length of time the voltage is supplied.

VHDCI Very high-density-cable interface. A 68-pin interface. Required for Ultra-SCSI connections.

virtual terminal A software path from an operator terminal on the host to the controller's CLI interface, sometimes called a host console. The path can be established via the host port on the controller (using HSZterm) or via the maintenance port through an intermediary host.

VTDPY An abbreviation for Virtual Terminal Display Utility.

warm swap A device replacement method that allows the complete system remains online during device removal or insertion. The system bus may be halted, or quiesced, for a brief period of time during the warm-swap procedure.

Wide Ultra SCSI Fast/20 on a Wide SCSI bus.

Worldwide name A unique 64-bit number assigned to a subsystem by the Institute of Electrical and Electronics Engineers (IEEE) and set by DIGITAL manufacturing prior to shipping. This name is referred to as the node ID within the CLI.

write-back caching A cache management method used to decrease the subsystem's response time to write requests by allowing the controller to declare the write operation "complete" as soon as the data reaches its cache memory. The controller performs the slower operation of writing the data to the disk drives at a later time.

write-through caching A cache management method used to decrease the subsystem's response time to a read. This method allows the controller to satisfy the request from the cache memory rather than from the disk drives.

write hole The period of time in a RAID level 1 or RAID level 5 write operation when an opportunity emerges for undetectable RAIDset data corruption. Write holes occur under conditions such as power outages, where the writing of multiple members can be abruptly interrupted. A battery backed-up cache design eliminates the write hole because data is preserved in cache and unsuccessful write operations can be retried.

write-through cache A cache management technique for retaining host write requests in read cache. When the host requests a write operation, the controller writes data directly to the storage device. This technique allows the controller to complete some read requests from the cache, greatly improving the response time to retrieve data. The operation is complete only after the data to be written is received by the target storage device.

This cache management method may update, invalidate, or delete data from the cache memory accordingly, to ensure that the cache contains the most current data.

Index

S