Configuring your StorageWorks subsystem

HSJ50 Array Controllers

HSOF Version 5.0

StorageWorks

Part Number: EK-HAJ50-CG.A01
July 1996

While Digital Equipment Corporation believes the information included in this manual is correct as of the date of publication, it is subject to change without notice. Digital makes no representations that the interconnection of its products in the manner described in this document will not infringe existing or future patent rights, nor do the descriptions contained in this document imply the granting of licenses to make, use, or sell equipment or software in accordance with the description. No responsibility is assumed for the use or reliability of firmware on equipment not supplied by Digital or its affiliated companies. Possession, use, or copying of the software or firmware described in this documentation is authorized only pursuant to a valid written license from Digital, an authorized sublicensor, or the identified licensor.

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

© Digital Equipment Corporation, 1996.
Printed in U.S.A.
All rights reserved.

Alpha, CI, DCL, DECconnect, DECserver, Digital, DSSI, HSC, HSJ, HSD, HSZ, MSCP, OpenVMS, StorageWorks, TMSCP, VAX, VAXcluster, VAX 7000, VAX 10000, VMS, VMScluster, and the Digital logo are trademarks of Digital Equipment Corporation. All other trademarks and registered trademarks are the property of their respective holders.

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

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

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

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

Contents

1 Introduction, 1
Features of your controller, 2
Anatomy of your controller, 5
Key steps for configuring your subsystem, 6

2 Configuring your controller, 7
Key steps for configuring a controller, 8
Communicating with a controller, 9
Configuring a controller, 12
Connecting a controller to the host, 18

3 Planning your storagesets, 21
Creating a profile, 22
Defining your storage requirements, 23
Choosing a storageset, 24
Using stripesets to increase I/O performance, 25
Using mirrorsets to ensure availability, 28
Using RAIDsets to increase performance and availability, 30
Using striped mirrorsets for the highest performance and availability, 32
Planning your partitions, 33
Choosing switches for your storagesets and devices, 35
RAIDset switches, 36
Mirrorset switches, 38
Device switches, 40

Initialize switches, 41
Unit switches, 45
Assigning unit numbers, 48
Creating a storageset map, 49
The next step..., 52

4 Automatically configuring storagesets, 53
Introducing CFMENU, 54
Using CFMENU, 56
Considerations for using CFMENU, 57

5 Manually configuring storagesets, 59
Adding disk drives, 60
Configuring a stripeset, 61
Configuring a mirrorset, 63
Configuring a RAIDset, 65
Configuring a striped mirrorset, 67
Configuring a single-disk unit, 69
Configuring a tape drive, 71
Configuring a tape loader, 72
Partitioning a storageset or disk drive, 73
Adding a disk drive to the spareset, 76
Removing a disk drive from the spareset, 77
Deleting a storageset, 78
Changing switches for a storageset or device, 79

6 Working with storage units, 81
OpenVMS disk capacity limitations, 82
OpenVMS command interactions, 83
Considerations for host-based shadow sets, 84

7 Periodic procedures, 85
Automatically cloning data for backup, 86
Shutting down your subsystem, 89
Restarting your subsystem, 90

Appendix, 91

Glossary, 1097

Index, 109

Related documents

The following table lists some of the documents related to the use of this product.

Document title	Part number
DECevent Installation Guide	AA–Q73JA–TE
StorageWorks BA350–MA Controller Shelf User's Guide	EK–350MA–UG
StorageWorks Configuration Manager for DEC OSF/1 Installation Guide	AA–QC38A–TE
StorageWorks Configuration Manager for DEC OSF/1 System Manager's Guide for HSZterm	AA–QC39A–TE
StorageWorks Solutions Configuration Guide	EK–BA350–CG
StorageWorks Solutions Shelf and SBB User's Guide	EK–BA350–UG
StorageWorks Solutions SW300-Series RAID Enclosure Installation and User's Guide	EK–SW300–UG
StorageWorks SW500-Series Cabinet Installation and User's Guide	EK–SW500–IG
StorageWorks SW800-Series Data Center Cabinet Installation and User's Guide	EK–SW800–IG
StorageWorks SW500-Series Cabinet Cable Distribution Unit Installation Guide	EK–SW5CU–IG
StorageWorks SW800-Series Data Center Cabinet Cable Distribution Unit Installation Sheet	EK–SWCDU–IS
The RAIDBOOK—A Source for RAID Technology	RAID Advisory Board
VAXcluster Console System User's Guide	AA–GV45D–TE
VAXcluster Systems Guidelines for VAXcluster System Configurations	EK–VAXCS–CG

1 Introduction

Features of your controller

Anatomy of your controller

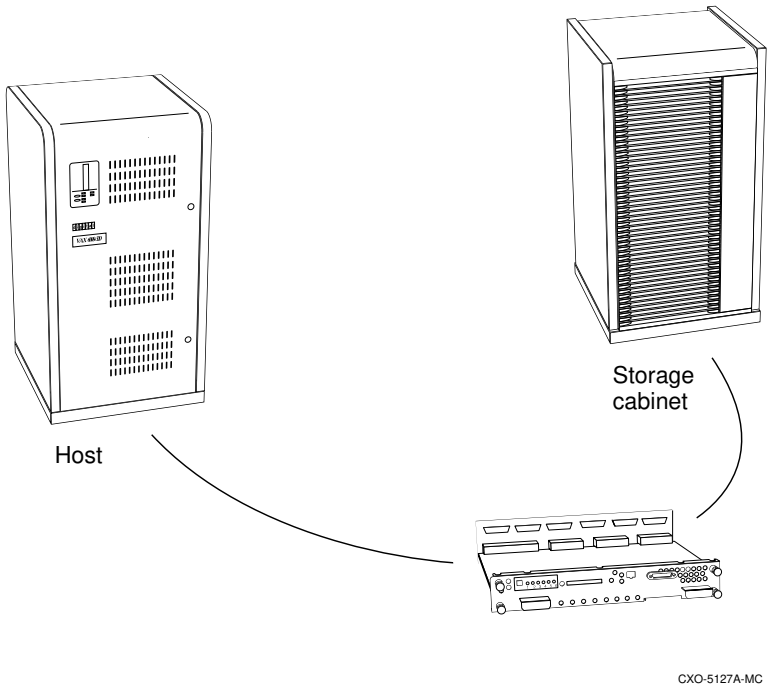
Key steps for configuring your subsystem

Features of your controller

Your HSJ50 controller is the intelligent bridge between your host and the devices in your subsystem.

From the host's perspective, the controller is simply another host that's connected to the cluster interface. Consequently, the host sends its packets to the controller just as it would to any other host on the CI.

Figure 1
Bridging the gap between the host and its storage subsystem



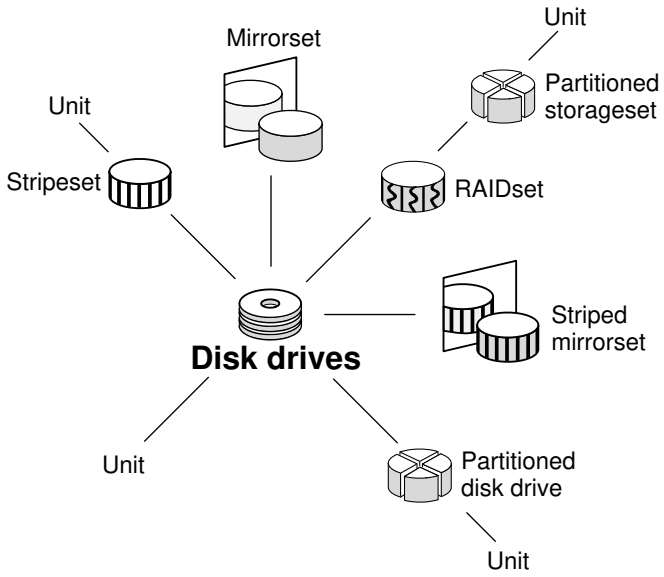
From the subsystem's perspective, the controller receives the I/O requests from the host and directs them to the devices in the subsystem. Since the controller processes all of the I/O requests, it eliminates the host-based processing that's typically associated with reading and writing data to multiple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

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

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

You create storage units by combining disk drives into storagesets, such as stripesets, RAIDsets, and mirrortsets, or by presenting them to the host as single-disk units as shown in Figure 2.

Figure 2
The host recognizes units created from storagesets, partitions, and disk drives



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

Of course, the controller also lets you add tape drives, loaders, and libraries to your subsytem to meet all of your storage requirements.

Table 1
Summary of features

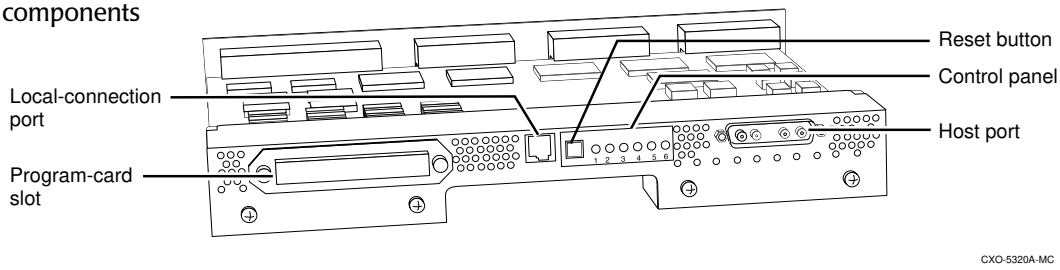
This table summarizes the features of your controller.

Feature	Supported
Host bus interconnect	CI
Host protocols	SCS, MSCP, TMSCP, DUP
Device protocol	SCSI-2
Number of SCSI device ports	6
Number of SCSI-2 devices in single configuration	42
Number of SCSI-2 devices in dual-redundant configuration	36
RAID levels supported	0, 1, 0+1, 3/5
Cache size	32, 64, or 128 MB
Preferred paths	✓
Disk warm swap	✓
Exercisers for testing devices	✓
Tape drives, loaders, and libraries	✓
Maximum number of configuration entities (devices + storagesets + partitions + units)	195
Maximum number of storagesets	30
Maximum number of partitions per storageset or disk drive	4
Maximum number of RAIDsets and mirrorsets running simultaneously	20
Maximum number of units presented to host	32
Maximum number of devices per unit	32
Largest device, storageset, or unit	120 GB

Anatomy of your controller

Your HSJ50 controller contains the key components shown in Figure 3.

Figure 3
Key components



Under normal circumstances, you won't need to remove the controller from its cabinet. For this reason, the components that you'll use most often are conveniently located on the front panel. For example, the local-connection port provides a convenient way to connect a terminal to your controller so that you can configure it for the first time.

After you've configured your controller, you should periodically check its control panel. The reset button flashes green about once every second to indicate that the controller is operating normally. If an error occurs, one or more of the amber LEDs on the control panel will flash in a pattern that will help you to diagnose the problem.

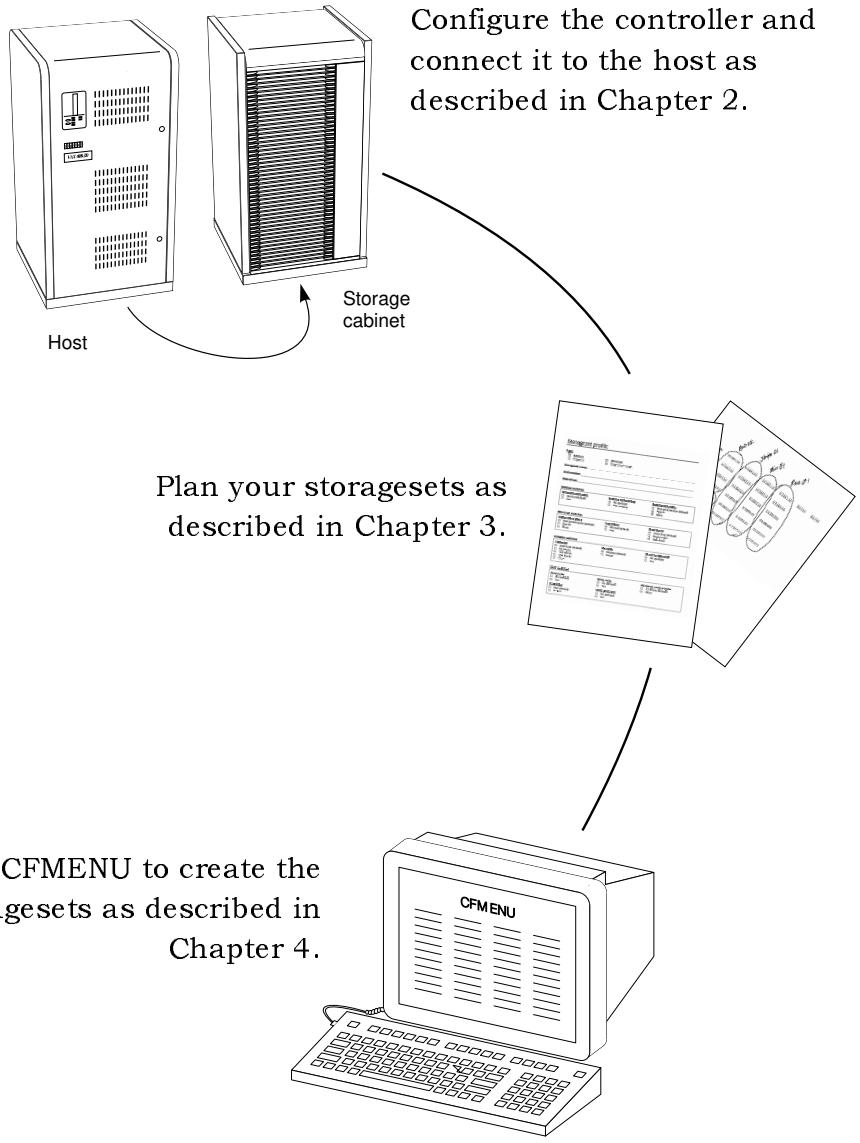
The host port and program-card slot are also located on the front panel, making it easy to update the HSOF software or to re-cable the controller.

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

Key steps for configuring your subsystem

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

Figure 4
Key steps for configuring your subsystem



2 Configuring your controller

Key steps for configuring a controller

Communicating with a controller

Configuring a controller

Connecting a controller to the host

Key steps for configuring a controller

Follow these key steps to configure a controller:

1. Establish communications with the controller. If you're configuring a controller for the first time, you MUST establish a local connection.
2. Configure the controller.
3. Connect the controller to the host.

After you've configured your controller, you can configure your subsystem by creating storagesets, single disk units, and other storage devices as described in Chapters 3, 4, and 5.

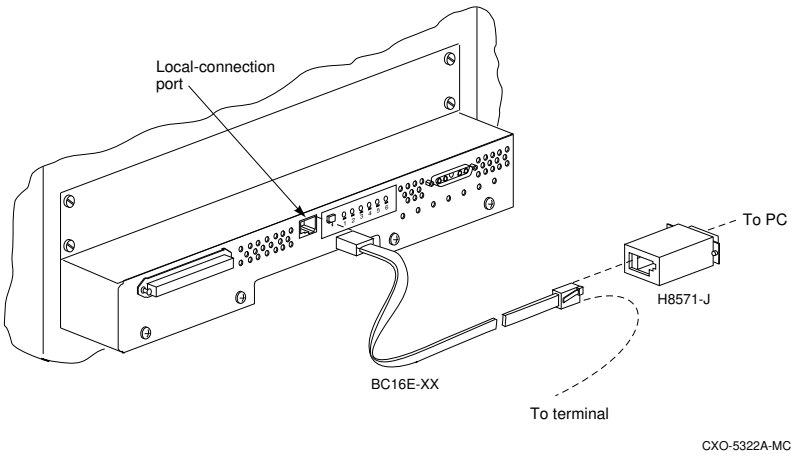
Communicating with a controller

You can communicate with a controller locally or remotely. Use a local connection to configure the controller for the first time. Use a remote connection for all subsequent configuration tasks.

Establishing a local connection

Use a terminal or a PC with terminal-emulation software to communicate with a controller locally.

Figure 5
Connect your terminal to the local-connection port to configure the controller for the first time



Caution

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

To establish a local connection for configuring the controller for the first time:

1. Turn off the terminal and connect it to the controller as shown in Figure 5. Plug one end of a DECconnect Office Cable (BC16E-XX) into the terminal; plug the other end into the local-connection port.

   If you're using a PC instead of a terminal, you'll need the serial-port adapter (H8571-J), also shown in Figure 5.

2. Turn on the terminal.

3. Configure the terminal for 9600 baud, 8 data bits, 1 stop bit, and no parity.

4. Press the Enter or Return key. A copyright notice and CLI prompt appear, indicating that you've established a local connection with the controller.

Establishing a remote connection

After configuring the controller for the first time, you can use a Diagnostics and Utility Protocol (DUP) terminal or a VAXcluster Console System (VCS) terminal to communicate with the controller remotely from a VMS™ host.

Installing the FYDRIVER

You'll need to install the FYDRIVER before starting a DUP terminal:

Note

Use SYSMAN rather than SYSGEN for AXP systems.

1. Access the SYSGEN utility:

   ```
   $ MCR SYSGEN
   ```

2. Specify an FY connection that does not use an adapter:

   ```
   SYSGEN> CONNECT FYA0 /NOADAPTER
   ```

3. Exit from SYSGEN:

   ```
   SYSGEN> EXIT
   ```

Starting a DUP terminal

1. Obtain DIAGNOSE privileges.
2. Start the DUP connection by entering the following command at the DCL prompt:

   ```
   $ SET HOST/DUP/SERVER=MSCP$DUP/TASK=CLI nodename
   ```

   A copyright notice and CLI prompt appear, indicating that you've established a remote connection to the controller

   Use the /LOG qualifier to create a log file of your sessions:

   ```
   $ SET HOST/DUP/SERVER=MSCP$DUP/TASK=CLI/LOG=LOG.INFO
   ```

If you want to run a local program such as DILX without going into the CLI, substitute the name of the program after the /TASK qualifier:

```
$ SET HOST/DUP/SERVER=MSCP$DUP/TASK=DILX/LOG=LOG.INFO
```

Starting a VCS terminal session

To communicate with the controller through a VCS terminal session, follow the instructions provided in the *VAXcluster Console System User's Guide*.

Configuring a controller

You can configure a controller to operate as a single, nonredundant controller or as one controller in a pair of dual-redundant controllers:

- Use a **single** configuration if you want to use only one controller to service the same group of storagesets, single-disk units, and other storage devices. Mount the controller in its own shelf and follow the steps for configuring a single controller that begin on the next page.
- If you have two HSJ50 controllers running HSOF Version 5.0, you can configure them as **dual-redundant** controllers. Use this configuration if you want to use two controllers to service the same group of storagesets, single-disk units, and other storage devices. Since they're redundant, either controller can continue to service all of the storage units if the other controller fails.

  Mount both controllers in the same shelf and follow the steps for configuring dual-redundant controllers that begin on page 15.

Configuring a single controller

To configure a single, nonredundant controller:

1. Establish a local connection to the controller.
2. Set the maximum number of nodes in the CI cluster:

   ```
   CLI> SET THIS_CONTROLLER MAX_NODES = n
   ```

   where *n* is 8, 16, or 32. Use the smallest number that's greater than the total number of nodes in the cluster.
3. Declare a CI node number for the controller:

   ```
   CLI> SET THIS_CONTROLLER ID = n
   ```

   where *n* is a number between 0 and (MAX_NODES–1) that's not already assigned to a node in the cluster.
4. Declare the maximum number of host adapters on the CI bus segment:

   ```
   CLI> SET THIS_CONTROLLER MAXIMUM_HOSTS = n
   ```

   where *n* is the number of host adapters. Keep in mind that one CPU can have more than one adapter. If you're going to add more adapters to the cluster in the future, you should declare the anticipated amount so you can add them without having to reconfigure the controller. This number must be smaller than the MAX_NODES number.
5. Declare an SCS node name for the controller:

   ```
   CLI> SET THIS_CONTROLLER SCS_NODENAME = "node name"
   ```

   where *node name* is an alphanumeric name that identifies the controller, such as "HSJ_1". The node name may contain up to six characters; it must begin with a letter and be enclosed in quotes.

6. Set the MSCP and TMSCP allocation classes:

```
CLI> SET THIS_CONTROLLER MSCP_ALLOCATION_CLASS = n
CLI> SET THIS_CONTROLLER TMSCP_ALLOCATION_CLASS = n
```

   where *n* is 0 through 255.

7. Optional: change the CLI prompt:

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

   where *new prompt* is a 16-character (or less) string that will appear as the prompt. For example, you could use the prompt to indicate the controller's name, such as "HSJ_1> ".

8. Optional: if one or more of your host adapters support 4K packets, enable 4K packets on the controller:

```
CLI> SET THIS_CONTROLLER CI_4K_PACKET_CAPABILITY
```

   The larger packet size can provide increased throughput to the host. The controller can use both 4K packets and smaller packets simutaneously, depending on the capabilities of each host and their adapters.

9. Disable the controller's CI ports:

```
CLI> SET THIS_CONTROLLER NOPATH=A
CLI> SET THIS_CONTROLLER NOPATH=B
```

10. Restart the controller:

```
CLI> RESTART THIS_CONTROLLER
```

11. When the CLI prompt reappears, verify the configuration:

```
CLI> SHOW THIS_CONTROLLER
```

12. Connect the controller to the host as described on page 18.

Configuring dual-redundant controllers

To configure a pair of dual-redundant controllers:

1. Establish a local connection to one of the controllers. (For the steps that follow, the controller to which you're connected is "this controller.")

2. Put "this controller" into dual-redundant (failover) mode:

   ```
   CLI> SET FAILOVER COPY = THIS_CONTROLLER
   ```

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

3. Set the maximum number of nodes in the CI cluster:

   ```
   CLI> SET THIS_CONTROLLER MAX_NODES = n
   CLI> SET OTHER_CONTROLLER MAX_NODES = n
   ```

   where *n* is 8, 16, or 32. Use the smallest number that's greater than the total number of nodes in the cluster.

4. Declare a CI node number for each controller:

   ```
   CLI> SET THIS_CONTROLLER ID = m
   CLI> SET OTHER_CONTROLLER ID = n
   ```

   where *m* and *n* are numbers between 0 and (MAX_NODES–1) that aren't already assigned to a node in the cluster.

5. Declare the maximum number of host adapters on the CI cluster:

   ```
   CLI> SET THIS_CONTROLLER MAXIMUM_HOSTS = n
   ```

   where *n* is the number of host adapters. Keep in mind that one CPU can have more than one adapter. If you're going to add more adapters to the cluster in the future, you should declare the anticipated amount so that you can add them without having to reconfigure the

controller. This number must be smaller than the MAX_NODES number.

6. Declare an SCS node name for each controller:

```
CLI> SET THIS_CONTROLLER SCS_NODENAME = "node name"
CLI> SET OTHER_CONTROLLER SCS_NODENAME = "node name"
```

   where *node name* is an alphanumeric name that identifies the controller, such as "HSJ_1" and "HSJ_2". The node name may contain up to six characters; it must begin with a letter and be enclosed in quotes.

7. Set the MSCP and TMSCP allocation classes for "this controller":

```
CLI> SET THIS_CONTROLLER MSCP_ALLOCATION_CLASS = n
CLI> SET THIS_CONTROLLER TMSCP_ALLOCATION_CLASS = n
```

   where *n* is 0 through 255. You only need to set these values for "this controller," since the "other controller" automatically inherits the same values.

8. Optional: change the CLI prompt for each controller:

```
CLI> SET THIS_CONTROLLER PROMPT = "new prompt"
CLI> SET OTHER_CONTROLLER PROMPT = "new prompt"
```

   where *new prompt* is a 16-character (or less) string that will appear as the controller's prompt. For example, you could use the prompt to indicate each controller's name, such as "HSJ_1> " and "HSJ_2> ".

9. Optional: if one or more of your host adapters support 4K packets, enable 4K packets on the controllers:

```
CLI> SET THIS_CONTROLLER CI_4K_PACKET_CAPABILITY
```

   The "other controller" inherits "this controller's" packet size. The larger packet size can provide increased throughput to the host. The controller can use both 4K packets and smaller packets simutaneously, depending on the capabilities of each host and their adapters.

10. Disable each controller's CI ports:

```
CLI> SET THIS_CONTROLLER NOPATH=A
CLI> SET THIS_CONTROLLER NOPATH=B
CLI> SET OTHER_CONTROLLER NOPATH=A
CLI> SET OTHER_CONTROLLER NOPATH=B
```

11. Restart the controllers:

```
CLI> RESTART OTHER_CONTROLLER
CLI> RESTART THIS_CONTROLLER
```

12. When the CLI prompt reappears, verify the configuration for each controller:

```
CLI> SHOW THIS_CONTROLLER FULL
CLI> SHOW OTHER_CONTROLLER FULL
```

13. Connect the controllers to the host as described on page 18.

Connecting a controller to the host

Note

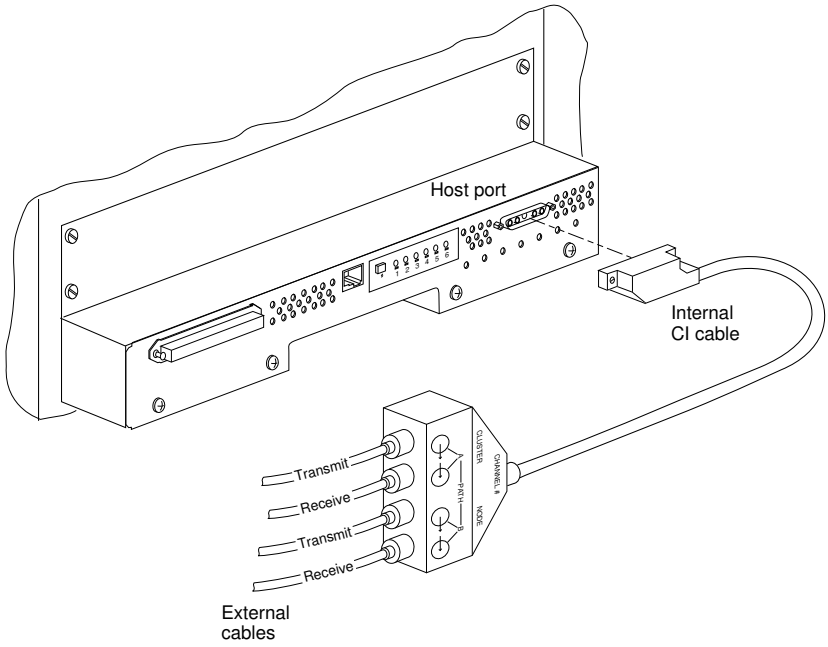
Whether you're installing a new controller or moving an old one to a new location, you must always configure the controller before connecting it to a host.

The controller's configuration determines how you'll connect it to a host. Use one of the following procedures described in this section:

- connecting a single controller to the host
- connecting dual-redundant controllers to the host

Connecting a single controller to the host

Figure 6
Connecting a single controller to the CI cluster



To connect a single, nonredundant controller to the CI cluster:

1. Connect the internal CI cable (17-03427-01) to the host port on the front of the controller as shown in Figure 6.
2. Connect the four, blue external cables to the internal cable as shown in Figure 6. Table 2 lists the maximum cable lengths.

Table 2
Maximum cable lengths

Cable	Meters	Feet
Internal CI cable	1.83	6
External CI cables (blue) to star coupler	45	147.6

3. Connect the external cables to the star coupler in the following order:

   Path B Receive
   Path B Transmit
   Path A Receive
   Path A Transmit

4. Route and tie the cables as desired.

5. Establish a local connection to the controller and enable its host port paths:

```
CLI> SET THIS_CONTROLLER PATH_A
CLI> SET THIS_CONTROLLER PATH_B
```

6. Verify the connection by logging onto the cluster and entering the following commands:

```
$ SHOW CLUSTER/CONT
  CLUSTER> ADD RPORT
  CLUSTER> ADD LPORT
```

   Make sure that the controller is listed with the correct node number (RPORT) and that its path is enabled.

Connecting dual-redundant controllers to the host

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

1. Connect an internal CI cable (17-03427-01) to the host port on the front of each controller as shown in Figure 6.
2. Connect the four, blue external cables to the internal cable for each controller as shown in Figure 6. Table 2 lists the maximum cable lengths.
3. Connect the external cables to the star coupler in the following order for each controller:

   Path B Receive
   Path B Transmit
   Path A Receive
   Path A Transmit
4. Route and tie the cables as desired.
5. Establish a local connection to one of the controllers and enable their host port paths:

```
CLI> SET THIS_CONTROLLER PATH_A
CLI> SET THIS_CONTROLLER PATH_B
CLI> SET OTHER_CONTROLLER PATH_A
CLI> SET OTHER_CONTROLLER PATH_B
```

6. Verify the connections by logging onto the cluster and entering the following commands:

```
$ SHOW CLUSTER/CONT
  CLUSTER> ADD RPORT
  CLUSTER> ADD LPORT
```

   Make sure that the controllers are listed with the correct node numbers (RPORT) and that the paths are enabled.

3 Planning your storagesets

Creating a profile

Defining your storage requirements

Choosing a storageset

Using stripesets to increase I/O perfromance

Using mirrorsets to ensure availability

Using RAIDsets to increase performance and availability

Using striped mirrorsets for the highest performance and availability

Planning your partitions

Choosing switches for your storagesets

Assigning unit numbers

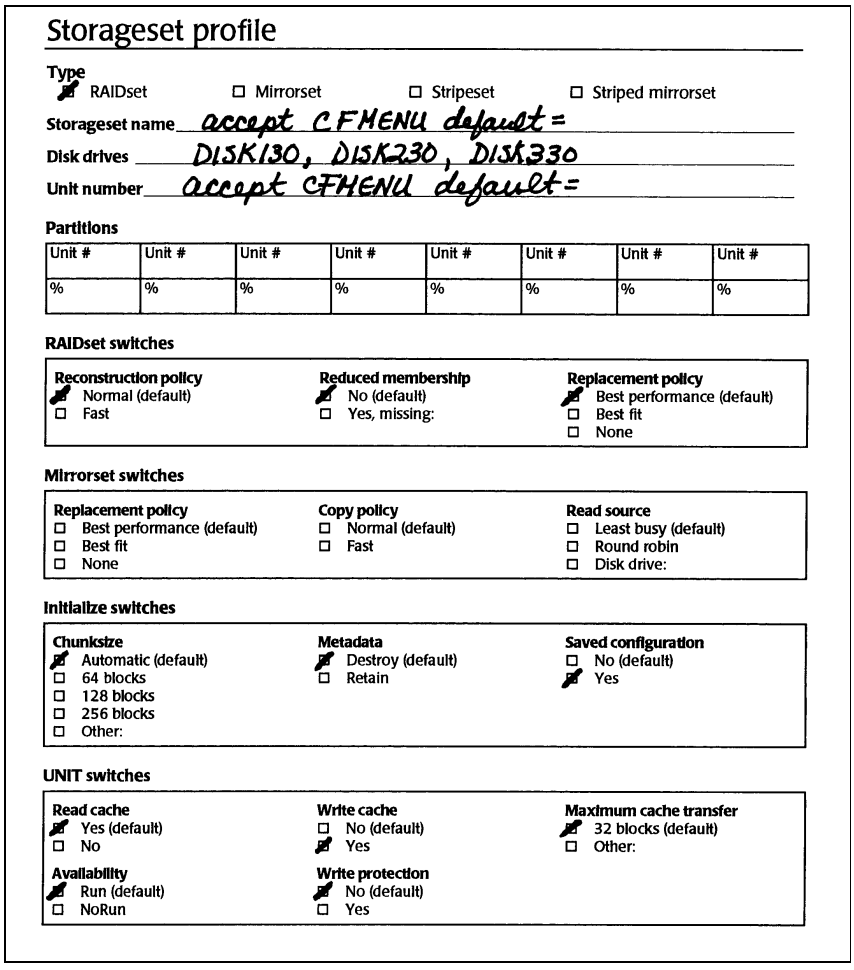
Creating a storageset map

Creating a profile

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

Take a few moments to familiarize yourself with the kinds of information contained in a storageset profile.

Figure 7
A typical storageset profile



Storageset profile

Type
☒ RAIDset ☐ Mirrorset ☐ Stripeset ☐ Striped mirrorset

Storageset name accept CFMENU default =

Disk drives DISK130, DISK230, DISK330

Unit number accept CFMENU default =

Partitions

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

RAIDset switches

Reconstruction policy	Reduced membership	Replacement policy
☒ Normal (default)	☒ No (default)	☒ Best performance (default)
☐ Fast	☐ Yes, missing:	☐ Best fit
		☐ None

Mirrorset switches

Replacement policy	Copy policy	Read source
☐ Best performance (default)	☐ Normal (default)	☐ Least busy (default)
☐ Best fit	☐ Fast	☐ Round robin
☐ None		☐ Disk drive:

Initialize switches

Chunksize	Metadata	Saved configuration
☒ Automatic (default)	☒ Destroy (default)	☐ No (default)
☐ 64 blocks	☐ Retain	☒ Yes
☐ 128 blocks		
☐ 256 blocks		
☐ Other:		

UNIT switches

Read cache	Write cache	Maximum cache transfer
☒ Yes (default)	☐ No (default)	☒ 32 blocks (default)
☐ No	☒ Yes	☐ Other:
Availability	**Write protection**	
☒ Run (default)	☒ No (default)	
☐ NoRun	☐ Yes	

The appendix contains blank profiles that you can copy and use to record the details for your storagesets.

Defining your storage requirements

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

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

Choosing a storageset

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

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

Table 3
A comparison of different kinds of storagesets

Compare the different kinds of storagesets to determine which ones satisfy your requirements.

Storageset	Relative availability	Request rate (read/write)	Transfer rate (read/write)	Applications
Array of disk drives (JBOD)	Proportionate to number of disk drives.	Identical to single disk drive.	Identical to single disk drive.	
Stripeset (RAID 0)	Proportionate to number of disk drives; worse than single disk drive.	Excellent if used with large chunk size.	Excellent if used with small chunk size.	Applications that require high performance for non-critical data.
Mirrorset (RAID 1)	Excellent	Good/Fair	Good/Fair	System drives; critical files.
RAIDset (RAID 3/5)	Excellent	Excellent/Fair	Good/Poor	High request rates, read-intensive, data lookup.
Striped Mirrorset (RAID 0+1)	Excellent	Excellent if used with large chunk size.	Excellent if used with small chunk size.	Any critical response-time application.

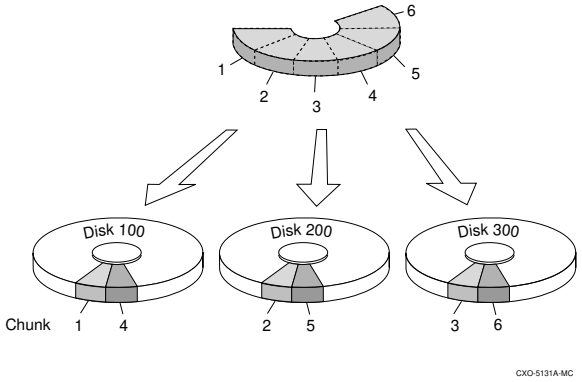
For a comprehensive discussion of RAID, refer to *The RAIDBOOK—A Source Book for Disk Array Technology,* pubished by the The RAID Advisory Board, St. Peter, MN.

Using stripesets to increase I/O performance

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

For example, in a three-member stripeset that contains disk drives 100, 200, and 300, the first chunk of an I/O request is written to 100, the second to 200, the third to 300, the fourth to 100, and so forth until all of the data has been saved.

Figure 8
Striping lets several disk drives participate in each I/O request



The relationship between the chunk size and the average request size determines if striping maximizes the request rate or the data-transfer rate. You can set the chunk size or let the controller set it automatically. See *Chunk size* on page 41 for information about setting the chunk size.

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

Considerations for planning a stripeset

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

- A storageset should only contain disk drives of the same capacity. The controller limits the capacity of each member to the capacity of the smallest member in the storageset. Thus, if you combine 2 GB disk drives with 1 GB disk drives in the same storageset, you'll waste 1 GB of capacity on each 2 GB member.

Note

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

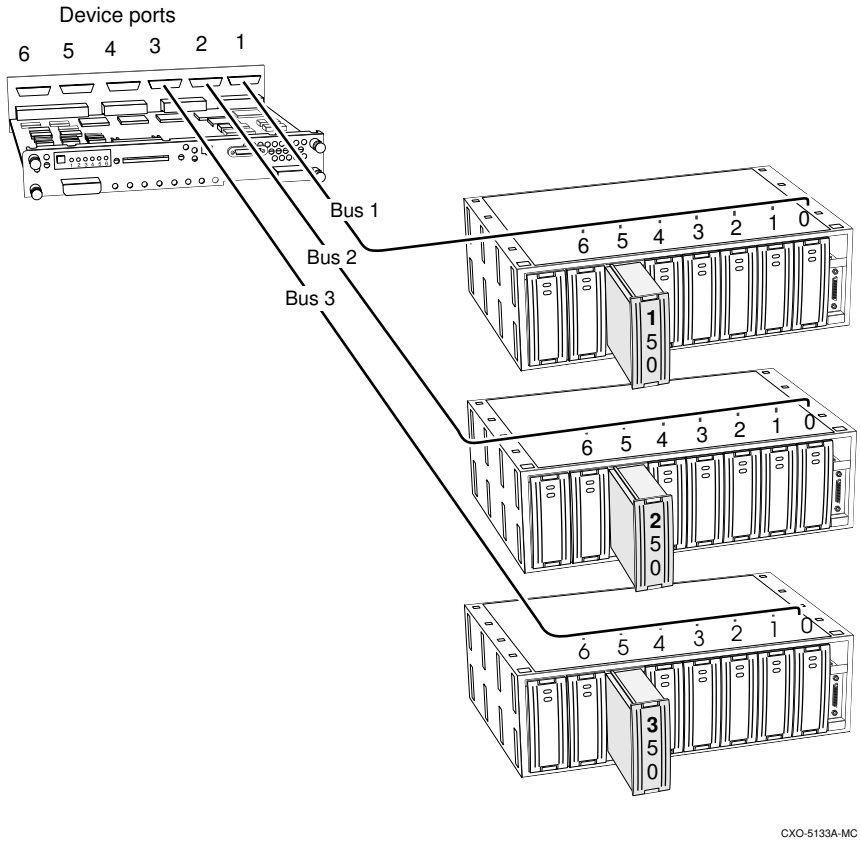
- Striping doesn't protect against data loss. In fact, because the failure of one member is equivalent to the failure of the entire stripeset, the likelihood of losing data is higher for a stripeset than for a single disk drive.

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

  For this reason, you should avoid using a stripeset to store critical data. Stripesets are more suitable for storing data that can be reproduced easily or whose loss doesn't 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 9.

Figure 9
Distribute members across ports

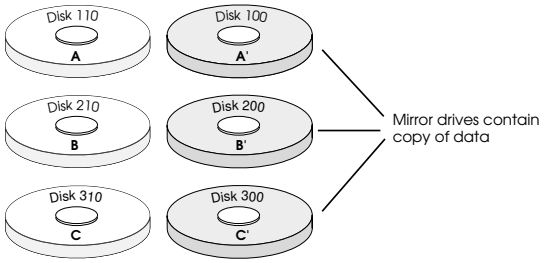


By spreading the traffic evenly across the buses, you'll ensure that no bus handles the majority of data to the storageset.

Using mirrorsets to ensure availability

Mirrorsets use redundancy to ensure availability. 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 10
Mirrorsets maintain two copies of the same data



Considerations for planning a mirrorset

Keep these points in mind as you plan your mirrorsets:

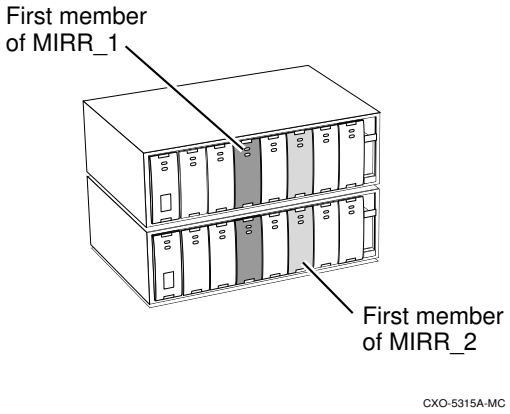
Note

If availability is your top priority, consider using redundant power supplies and dual-redundant controllers.

- Data availability with a mirrorset is excellent but costly—you'll need twice as many disk drives to satisfy a given capacity requirement.
- You can configure up to 20 mirrorsets per controller or pair of dual-redundant controllers. Each mirrorset may contain up to six members.
- A write-back cache module is required for mirrorsets.
- If you're using more than one mirrorset in your subsystem, you should put the first member of each mirrorset on different busses. (The first member of a mirrorset is the first disk drive you add with CFMENU or the ADD MIRRORSET command.)

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

Figure 11
Put first members on different busses



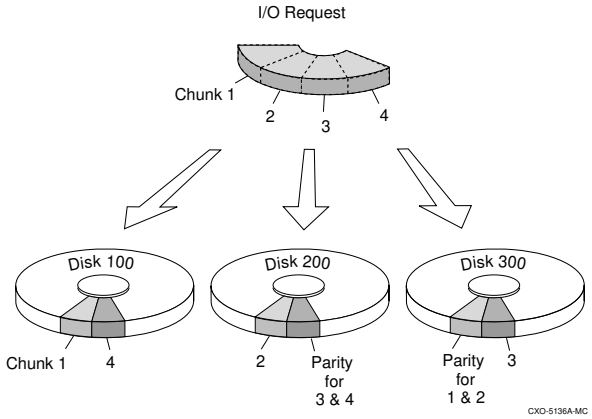
To avoid an I/O bottleneck on one bus, you can simply put the first members on different busses—in other words, put them in different shelves, as shown in Figure 11. Additionally, you can set the read-source switch to Round Robin. See *Read source* on page 39 for more information about this switch.

- A storageset should only contain disk drives of the same capacity. The controller limits the capacity of each member to the capacity of the smallest member in the storageset. Thus, if you combine 2 GB disk drives with 1 GB disk drives in the same storageset, you'll waste 1 GB of capacity on each 2 GB member.
- Evenly distribute the members across the device ports to balance load and provide multiple paths as shown in Figure 9.

Using RAIDsets to increase performance and availability

RAIDsets are enhanced stripesets—they use striping to increase I/O performance and distributed-parity data to ensure data availability.

Figure 12
Parity ensures availability; striping provides good performance



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

For example, in a three-member RAIDset that contains disk drives 100, 200, and 300, the first chunk of an I/O request is written to 100, the second to 200, then parity is calculated and written to 300; the third chunk is written to 300, the forth to 100, and so forth until all of the data is saved.

The relationship between the chunk size and the average request size determines if striping maximizes the request rate or the data-transfer rates. You can set the chunk size

or let the controller set it automatically. See *Chunk size* on page 41 for information about setting the chunk size.

Considerations for planning a RAIDset

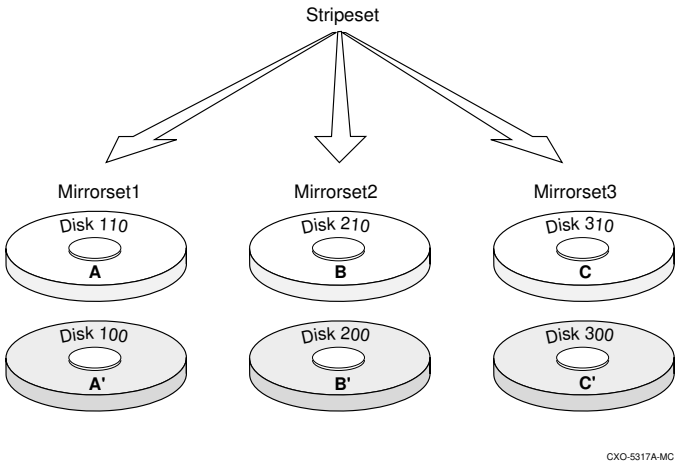
Keep these points in mind as you plan your RAIDsets:

- A write-back cache module is required for RAIDsets, but write-back cache needn't be enabled for the RAIDset to function properly.
- A RAIDset must include at least three disk drives, but no more than 14.
- Evenly distribute the members across the device ports to balance load and provide multiple paths as shown in Figure 9.
- A storageset should only contain disk drives of the same capacity. The controller limits the capacity of each member to the capacity of the smallest member in the storageset. Thus, if you combine 2 GB disk drives with 1 GB disk drives in the same storageset, you'll waste 1 GB of capacity on each 2 GB member.

Using striped mirrorsets for the highest performance and availability

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

Figure 13
Striping and mirroring in the same storageset



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

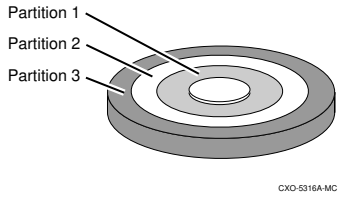
Considerations for planning a striped mirrorset

Plan the mirrorset members, then plan the stripeset that will contain them. Follow the considerations for stripesets and mirrorsets provided in this chapter.

Planning your partitions

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

Figure 14
Each partition can be presented to the host as a storage unit



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

Defining a partition

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

In any case, an unpartitioned storage unit has more capacity than a partition that uses the whole unit. That's because each partition requires 5 blocks of adminstrative

metadata. Thus, a single disk unit that contains one partition can store *n*-5 blocks of user or application data.

See *Partitioning a storageset or disk drive* in Chapter 5 for information on manually partitioninig a storageset or single-disk unit.

Guidelines for partitioning storagesets and disk drives

Keep these points in mind as you plan your partitions:

- You can create up to four partitions per storageset or disk drive.
- All of the partitions on the same storageset or disk drive must be addressed through the same controller. Thus, if you set a preferred controller for one partition in a storageset, all of the partitions in that storageset will inherit that preferred controller. This ensures a transparent failover of devices should one of the dual-redundant controllers fail.
- Partitions cannot be combined into storagesets. For example, you can't divide a disk drive into three partitions, then combine those partitions into a RAIDset.
- Once you've partitioned a container, you cannot unpartition it without reinitializing the container.
- Just as with storagesets, you don't have to assign unit numbers to partitions until you're ready to use them.

Choosing switches for your storagesets and devices

Depending upon the kind of storageset or device you're configuring, you can enable the following kinds of options or "switches:"

- RAIDset and Mirrorset switches
- Initialize switches
- Unit switches
- Device switches

Enabling switches

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

If you use CLI commands to configure the storageset or device manually, the procedures in Chapter 5 indicate when and how to enable each switch.

Changing switches

You can change the RAIDset, Mirrorset, Device, and Unit switches at any time. See *Changing switches for a storageset or device* in Chapter 5.

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

RAIDset switches

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

- Replacement policy
- Reconstruction policy
- Membership

Replacement policy

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

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

Reconstruction policy

Specify the speed with which the controller reconstructs the data from a failed disk drive then writes it to a replacement disk drive:

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

Membership

Indicate to the controller that the RAIDset you're adding is complete or "reduced," which means it's missing one of its members:

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

Mirrorset switches

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

- Replacement policy
- Copy speed
- Read source

Replacement policy

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

policy=best_performance (default) puts the failed disk drive in the failedset then tries to find a replacement (from the spareset) that is on a different device port than the remaining, operational disk drives. If more than one disk drive meets this criterion, this switch selects the drive that also provides the best fit.

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

Copy speed

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

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

Read source

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

- READ_SOURCE=ROUND ROBIN forces the controller to read data sequentially from all "normal" or operational members in a mirrorset. For example, in a four-member mirrorset (A, B, C, and D), the controller reads from A, then B, then C, then D, then A, then B, and so forth. No preference is given to any member.
- READ_SOURCE=LEAST BUSY (default) forces the controller to read data from the "normal" or operational member that has the least-busy work queue.
- READ_SOURCE=DISK*nnn* forces the controller to always read data from a particular "normal" or operational member. If the specified member fails, the controller reads from the least busy member.

Device switches

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

- Transportability
- Transfer rate

Transporatability

Note

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

Indicate whether a disk drive is transportable or not when you add it to your subsystem:

- NOTRANSPORTABLE disk drives (default) are marked with StorageWorks-exclusive metadata. This metadata supports the error-detection and recovery methods that the controller uses to ensure data availability. Disk drives that contain this metadata can't be used in non-StorageWorks subsystems.
- TRANSPORTABLE disk drives can be used in non-StorageWorks subsystems. Transportable disk drives can be used as single-disk units in StorageWorks subsystem as well as disk drives in other systems. They can't be combined into storagesets in a StorageWorks subsystem.

Transfer rate

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

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

Initialize switches

Note

After you've initialized the storageset or disk drive, you cannot change these switches without reinitializing the storageset or disk drive.

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

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

Chunk size

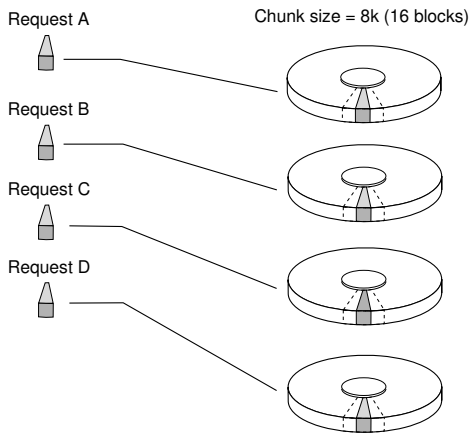
Specify a chunk size to control the stripesize used for RAIDsets and stripesets:

- CHUNKSIZE=DEFAULT lets the controller set the chunk size based on the number of disk drives (*d*) in a stripeset or RAIDset. If $d \leq 9$ then chunk size = 256. If $d > 9$ then chunk size = 128. However, if the cache size < 16MB then chunk size = 64 regardless of *d*.
- 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.

Figure 15
If chunk size is larger than the request size, then each disk drive in the storageset can respond to a separate I/O request.



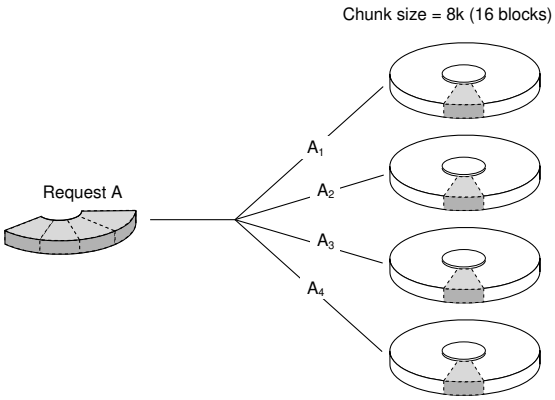
Applications such as interactive transaction processing, office automation, and file services for general timesharing tend to require high I/O request rates.

Large chunk sizes also tend to increase the performance of random reads and writes. Digital recommends a chunk size of 10 to 20 times the average request size, rounded up to the nearest multiple of 64. In general, a chunk size of 256 works well for UNIX systems; 128 works well for OpenVMS systems.

Increasing the data transfer rate

A small chunk size relative to the average request size increases the data transfer rate by allowing multiple disk drives to participate in one I/O request.

Figure 16
Chunk size is smaller than the request size, then more than one disk drive can respond to the same I/O request.



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

Increasing sequential write performance

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

Chunk size doesn't significantly affect sequential read performance.

Maximum chunk size for RAIDsets

Don't exceed the following chunk sizes for a RAIDset. (The maximum chunk size is derived by $2048/(d-1)$ where d is the number of disk drives in the RAIDset.)

Table 4
Maximum chunk sizes for a RAIDset

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

Save configuration

Indicate whether or not to save the subystem's configuration on the storage unit when you initialize it:

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

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

Overwrite

Specify whether to destroy or retain the user data and metadata when you're initializing a disk drive that has been previously used in a storageset or as a single-disk unit:

- DESTROY (default) overwrites the user data and forced-error metadata on a disk drive when it's 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 *CLI Reference Manual* for information on removing disk drives from a mirrorset.

Note

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

Unit switches

Table 5
Unit switches

You can enable the following Unit switches for the following storagesets and devices.

Unit switch	RAID	Stripe	Mirror	Disk Notrans	Disk Trans	Disk Optical	CD ROM	Tape	Pass through
Preferred path	✓	✓	✓	✓	✓	✓	✓	✓	✓
Read cache	✓	✓	✓	✓	✓	✓	✓		
Writeback cache	✓	✓	✓	✓		✓	✓		
Maximum cache transfer	✓	✓	✓	✓	✓	✓	✓		
Availability	✓	✓	✓	✓	✓	✓	✓		
Tape format								✓	
Write protection	✓	✓	✓	✓	✓	✓			✓

Preferred path dual-redundant configurations

Note

Preferred pathing doesn't apply to storage units that are serviced by a single controller.

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

- NOPREFERRED_PATH (default) allows either controller to access the storage unit.
- PREFERRED_PATH=THIS_CONTROLLER indicates that the controller to which you're connected handles all I/O activity to the storage unit. By establishing preferred paths, you can distribute the I/O load evenly between the two controllers by dividing the storage units into two equal groups—based on their I/O activities—and assigning each group to its own controller.
- PREFERRED_PATH=OTHER_CONTROLLER indicates that the other controller handles all I/O activity to the storage unit.

Read cache

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

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

Write-back cache

Note

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

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

- WRITEBACK_CACHE enables write-back caching.
- NOWRITEBACK_CACHE disables write-back caching.

Maximum cache transfer

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

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

Availability

Specify whether or not to make the storage unit available to the host:

- RUN (default) specifies that as soon as you provide a host-addressable unit number the storage unit will be made available to the host.

- NORUN specifies that the storage unit will not be made available to the host until you specify the RUN switch.

Tape format

Specify the tape format to be used unless it's overridden by the host. Not all tape drives support all formats:

- DEFAULT_FORMAT=DEVICE_DEFAULT lets the controller automatically determine and set the device default for the tape drive.
- DEFAULT_FORMAT=*n* lets you specify the tape format, such as TZ88 or HOST_SELECTED.

  To display the formats that are supported, enter the following command at the CLI prompt:

  ```
  CLI> SHOW tape-unit-number DEFAULT_FORMAT= ?
  ```

Write protection

Enable or disable write protection for the storage unit:

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

Assigning unit numbers

You'll need to assign a unique unit number to each storageset, single disk unit, or storage device that you want your host to know about in your subsystem. A unit number is an alpha-numeric tag that identifies each storage unit in your subsystem, such as D102 for a disk-based storage unit. The host uses these numbers to indicate the source or target for every I/O request it sends to a controller.

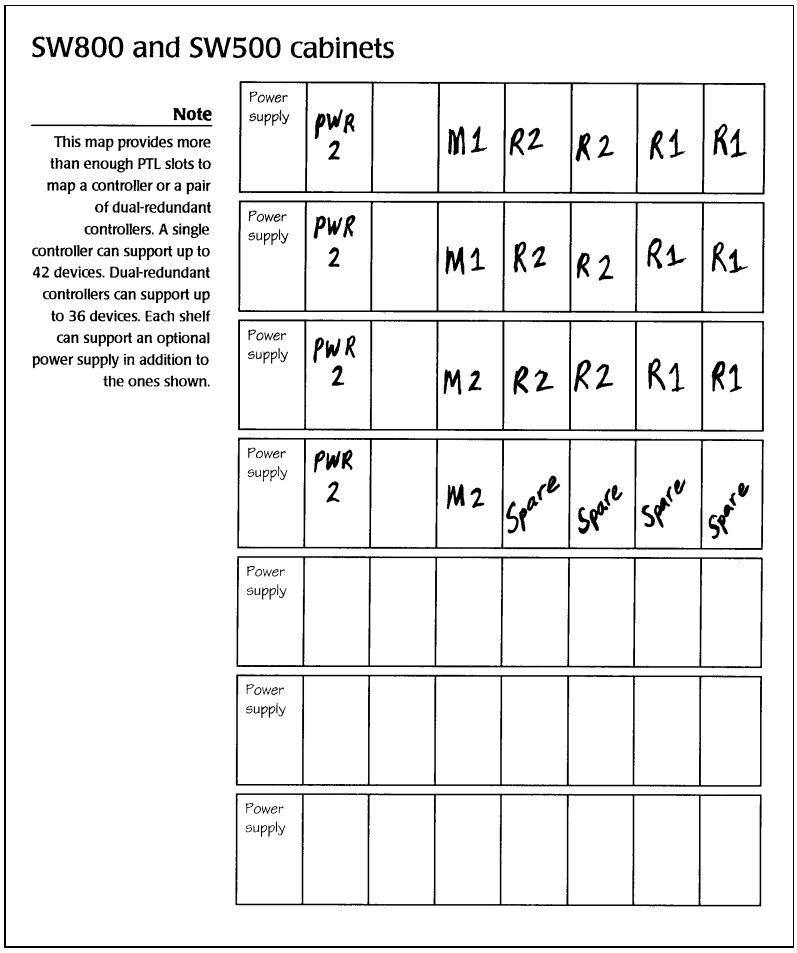
Each four-place unit number contains the following:

- A letter that indicates the kind of devices in the storage unit: use D for disk drives (including CD-ROMs and command disks for loaders and libraries), P for passthrough devices, and T for tape drives. (If you're using CFMENU to configure your storagesets and devices, it will automatically supply a device letter for you.)
- Any number from 0-4094 that uniquely identifies the storage unit.

Creating a storageset map

Configuring your subsytem will be easier if you know how the storagesets correspond to the disk drives in your subsystem. You can see this relationship by creating a storageset map like the one shown here.

Figure 17
A storageset map for a subsystem that contains two RAIDsets, two mirrorsets, and four disk drives in the spareset. Each shelf also has dual power supplies.



To create a storageset map:

1. Copy the appropriate cabinet template from the Appendix.
2. Establish a local or remote connection to one of the controllers in your subsystem.

3. Show the devices that are assigned to the controller:

   ```
   CLI> SHOW DEVICES
   ```

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

   ```
   CLI> LOCATE device_name
   ```

   The LOCATE command causes the device's LED to flash continuously. To turn off the LED:

   ```
   CLI> LOCATE CANCEL
   ```

   The controller names each device based on its Port-Target-LUN (PTL) location. See *PTL addressing convention* below).

5. Repeat steps 2 through 4 for each controller or dual-redundant pair of controllers.

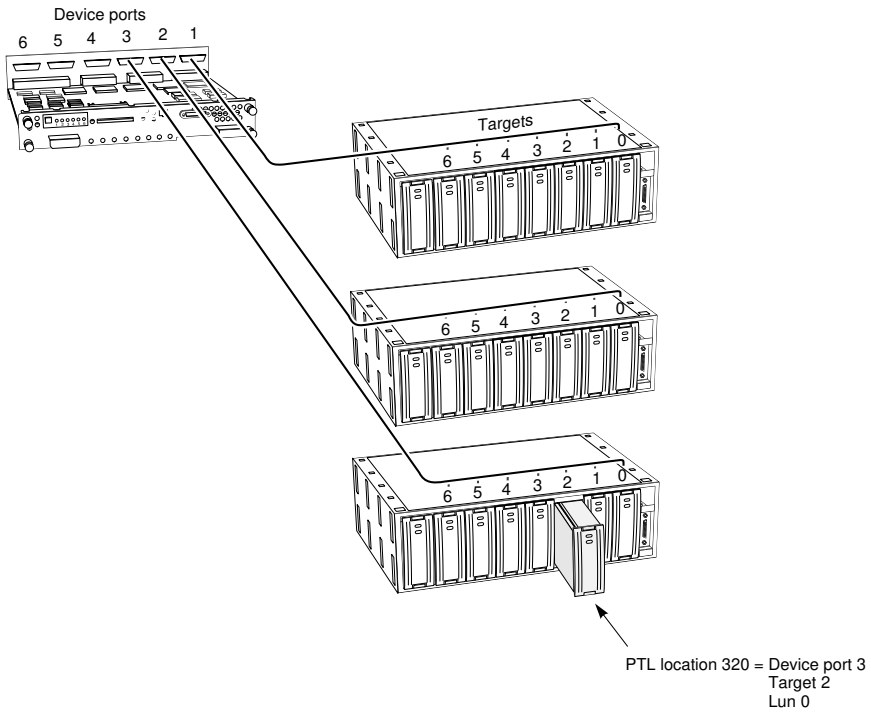
6. After you have mapped the devices to your cabinet template, create the storageset map by circling each group of disk drives that you want to combine into a storageset or put into the spareset. Label each group with its storageset name, for example: RAID1 for a RAIDset; Mirr1 for a mirrorset; and Stripe1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

PTL addressing convention

Your controller has six SCSI–2 device ports. Each device port connects to a shelf that supports up to seven devices or "targets." And every device uses LUN 0, except some tape loaders, which use LUN 1.

As shown in Figure 18, the controller addresses DISK320 through device port 3, target 2, LUN 0. Thus, the PTL location indicates the pathway that the controller uses to address a disk or tape drive. It also indicates the device name.

Figure 18
PTL addressing



The controller uses the PTL location to name each device that you add to your subsystem with the CONFIG utility or CFMENU. (Factory-installed devices are added with the CONFIG utility. Thus, their names derive from their PTL locations.) For example, if the controller finds a disk in PTL 320, it names it DISK320; if it finds a tape drive at PTL 320, it names it TAPE320.

When your controller receives an I/O request, it identifies the unit number for the request, then correlates the unit number to the storageset name. From the storageset name, the controller locates the appropriate devices for the I/O request. (For example, the RAIDset “R1” might contain DISK150, DISK250, and DISK350.) The controller generates the read or write request to the appropriate device using the PTL addressing convention.

The next step...

Turn to Chapter 4 if you want to configure your storage units automatically with CFMENU, a menu-based utility that simplifies the configuration process. This utility is especially helpful if you're configuring storagesets for the first time.

Turn to Chapter 5 if you want to configure your storage units manually by issuing CLI commands from a local or remote connection. Configuring your storage units manually give you more authority when it comes to naming the storage units, since CFMENU automatically names them for you. However, with authority comes responsibility, so you should complete a profile for each storage unit or device that you want configure in your subsytem before you begin the procedures provided in Chapter 5.

4 Automatically configuring storagesets

Introducing CFMENU

Using CFMENU

Considerations for using CFMENU

Introducing CFMENU

CFMENU is a modest but effective utility that simplifies the task of configuring storagesets and devices.

Note

See Chapter 5 if you want to modify an existing storageset or configure a tape drive or tape loader.

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

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

Figure 19
Main Menu

```
------------------------------------------- CFMENU  Configuration Menu Utility -------------------------------------------

MAIN MENU:               |Unconfig'd|     Config'd  Device       Product       Stor.set Stor.set Chnk  Trn  In- Re-       P W W
1. Add/delete devices    | Dev.PTLs |       PTLs     Name          ID            Name    Typ/Sz  Size  sp. it'd duc  Unit T P B
2. Add/delete mirrorsets |----------|      ------ --------- ----------------  --------- ------- ------ --- ---- --- ----- - - -
3. Add/delete stripesets |          |disks:  120    DISK120 RZ28B    (C) DEC                          N    N
4. Add/delete raidsets/  |          |        130    DISK130 RZ28D    (C) DEC                          N    N
   sparesets/failedsets  |          |        140    DISK140            ?                             N    ?        D7    N N
5. Partition Processing  |          |        150    DISK150 RZ28D    (C) DEC                          N    Y           P
6. Add/delete passthrough|          |        200    DISK200 RZ28D    (C) DEC                          Y    Y       D120   N N
7. Initialize devices    |          |        230    DISK230 RZ26     (C) DEC                          N    N
   and/or storagesets    |          |        250    DISK250 RZ26L    (C) DEC                          N    N
8. Add/delete units      |          |miror:  100    DISK100 RZ26     (C) DEC      M0    MIR/2                Y   N   D190   N Y
9. Setup terminal        |          |        300    DISK300 RZ28D    (C) DEC      "       "                  "   "    "     " "
10. Exit CFMENU          |          |        220    DISK220 RZ28D    (C) DEC      M2    MIR/2                Y   N
                         |          |        420    DISK420 RZ28D    (C) DEC      "       "                  "   "
                         |          |        310    DISK310 RZ28D    (C) DEC      M6    MIR/1                Y   N   D196   N Y
D=Scroll down U=Scroll up|          |        450    DISK450 RZ29B    (C) DEC      M5    MIR/2                Y   N   D195   N N
-------------------------------------------------------------------------------------------------------------------------

Enter menu choice (1,10) [10] ?
```

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

Table 6 on the next page lists the heading and contents for each column on the Main Menu.

Table 6
Interpreting CFMENU columns

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

Using CFMENU

Configuring a storageset with CFMENU is as easy as choosing menu options and responding to prompts. Just remember to move through the Main Menu item from top to bottom for each storageset you want to configure.

Note

Press "D" to scroll down CFMENU's columns. Press "U" to scroll up.

1. Start CFMENU:

   ```
   CLI> RUN CFMENU
   ```

2. Choose menu item 1 to add the disk drives you want to combine into a storageset.

3. (Skip this step if you're configuring a single-disk unit.) Choose menu items 2, 3, or 4 to add the storageset to the controller's list of available storagesets. CFMENU prompts you for a storageset number and the switches you can apply to the storageset.

4. Choose menu item 6 to initialize the storageset or disk drive. CFMENU prompts you for the Initialize switches you can apply to the storageset or single-disk unit.

5. Choose menu item 7 to assign a host-addressable unit-number to the storageset. CFMENU prompts you for the Unit switches you can apply to the storage unit.

6. Choose menu item 9 to exit CFMENU or repeat these steps to configure another storageset.

If you've configured storagesets with CLI commands, then you probably recognize these steps. That's because CFMENU assembles the appropriate CLI commands and issues them to the controller for you. These are the same commands you would issue if you followed the steps in Chapter 5 for configuring a storageset.

Considerations for using CFMENU

Keep the following points in mind for using CFMENU:

- Configure your storagesets manually if you want to use your own naming scheme. CFMENU names each storageset according to a simple naming convention: M*n* for mirrorsets, S*n* for stripesets, and R*n* for RAIDsets, where *n* is a sequentially indexed number. CFMENU also automatically provides unit-number prefixes; you specify the actual number.
- You can create and delete storagesets with CFMENU, however, you cannot modify them once they've been created. Follw the steps in *Changing switches for a storageset or device* in Chapter 5.
- If you're using dual-redundant controllers, you don't need to run CFMENU on both controllers simultaneously. The "other controller" automatically inherits the configuration you create with CFMENU.
- CFMENU can't configure tape loaders. See *Configuring a tape drive* and *Configuring a tape loader* in Chapter 5.
- CFMENU can't partition striped mirrorsets. Follow the steps for *Partitioning a storageset or disk drive* in Chapter 5 to partition a striped mirrorset.

5 Manually configuring storagesets

Adding disk drives

Configuring a stripeset

Configuring a mirrorset

Configuring a RAIDset

Configuring a striped mirrorset

Configuring a single disk unit

Configuring tape drives and tape loaders

Partitioning a storageset or disk drive

Adding a disk drive to the spareset

Removing a disk drive from the spareset

Deleting a storageset

Changing switches for a storageset or device

Adding disk drives

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

Adding one disk drive at a time

To add one new disk drive to your controller's list of eligible devices:

```
CLI> ADD DISK DISKnnn PTL-location
```

Adding several disk drives at a time

To add several new disk drives to your controller's list of eligible devices:

```
CLI> RUN CONFIG
```

Configuring a stripeset

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

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

   ```
   CLI> ADD STRIPESET stripeset-name DISKnnn DISKnnn
   ```

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

   ```
   CLI> INITIALIZE stripeset-name SWITCH_VALUE
   ```

Table 7
Initialize switches

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

3. Present the stripeset to the host by giving it a unit number the host can recognize:

   ```
   CLI> ADD UNIT unit-number stripeset-name
   ```

4. Optional: set the Unit switches or skip this step to accept the defaults(*). For each switch you want to set, enter the following command:

   ```
   CLI> SET unit-number SWITCH_VALUE
   ```

Table 8
Unit switches

Unit switch	Value and syntax
Maximum cached transfer	MAXIMUM_CACHED_TRANSFER=32* MAXIMUM_CACHED_TRANSFER=N
Read cache	READ_CACHE* NOREAD_CACHE

Unit switch	Value and syntax
Write-back cache	NOWRITEBACK_CACHE* WRITEBACK_CACHE
Availability	RUN* NORUN
Write protection	NOWRITE_PROTECT* WRITE_PROTECT

5. Verify the stripeset configuration and switches:

   CLI> **SHOW *stripeset-name***

6. Verify the unit configuration and switches:

   CLI> **SHOW *unit-number***

For example

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

```
CLI> ADD STRIPESET Stripe1 disk100 disk200 disk300
CLI> INITIALIZE Stripe1 CHUNKSIZE=128
CLI> ADD UNIT D100 Stripe1
CLI> SET D100 MAXIMUM_CACHED_TRANSFER=16
CLI> SET D100 WRITEBACK_CACHE
CLI> SHOW Stripe1
CLI> SHOW D100
```

Configuring a mirrorset

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

1. Create the mirrorset by adding its name to the controller's list of storagesets and specifying the disk drives it contains:

   ```
   CLI> ADD MIRRORSET mirrorset-name DISKnnn DISKnnn
   ```

2. Optional: set the Mirrorset switches or skip this step to accept the defaults(*). For each switch you want to set, enter the following command:

   ```
   CLI> SET mirrorset-name SWITCH_VALUE
   ```

Table 9
Mirrorset switches

Mirrorset switch	Value and syntax
Replacement policy	POLICY=BEST_PERFORMANCE* POLICY=BEST_FIT NOPOLICY
Copy speed	COPY=NORMAL* COPY=FAST
Read source	READ_SOURCE=LEAST_BUSY* READ_SOURCE=ROUND_ROBIN READ_SOURCE=DISKnnn

3. Initialize the mirrorset. If you want to set any Initialize switches, you must do so in this step:

   ```
   CLI> INITIALIZE mirrorset-name SWITCH_VALUE
   ```

Table 10
Initialize switches

Initialize switch	Value and syntax
Save configuration	NOSAVE_CONFIGURATION* SAVE_CONFIGURATION
Destroy	NODESTROY* DESTROY

4. Present the mirrorset to the host by giving it a unit number the host can recognize:

   ```
   CLI> ADD UNIT unit-number mirrorset-name
   ```

5. Optional: set the Unit switches or skip this step to accept the defaults(*). For each switch you want to set, enter the following command:

```
CLI> SET unit-number SWITCH_VALUE
```

Table 11
Unit switches

Unit switch	Value and syntax
Maximum cached transfer	MAXIMUM_CACHED_TRANSFER=32* MAXIMUM_CACHED_TRANSFER=N
Read cache	READ_CACHE* NOREAD_CACHE
Write-back cache	NOWRITEBACK_CACHE* WRITEBACK_CACHE
Availability	RUN* NORUN
Write protection	NOWRITE_PROTECT* WRITE_PROTECT

6. Verify the mirrorset configuration and switches:

```
CLI> SHOW mirrorset-name
```

7. Verify the unit configuration and switches:

```
CLI> SHOW unit-number
```

For example

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

```
CLI> ADD MIRRORSET Mirr1 disk100 disk200
CLI> INITIALIZE Mirr1
CLI> ADD UNIT D200 Mirr1
CLI> SET D200 WRITEBACK_CACHE
CLI> SHOW Mirr1
CLI> SHOW D200
```

Configuring a RAIDset

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

1. Create the RAIDset by adding its name to the controller's list of storagesets and specifying the disk drives it contains:

   CLI> **ADD RAIDSET *RAIDset-name DISKnnn DISKnnn DISKnnn***

2. Optional: set the RAIDset switches or skip this step to accept the defaults(*). For each switch you want to set, enter the following command:

   CLI> **SET *RAIDset-name SWITCH_VALUE***

Table 12
RAIDset switches

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

3. Initialize the RAIDset. Optional: if you want to set the Initialize switches, you must do so in this step:

   CLI> **INITIALIZE *RAIDset-name SWITCH_VALUE***

   Use the values in this table:

Table 13
Initialize switches

Initialize switch	Value and syntax
Chunk size	CHUNKSIZE=DEFAULT* CHUNKSIZE=*N*
Save configuration	NOSAVE_CONFIGURATION* SAVE_CONFIGURATION

4. Present the RAIDset to the host by giving it a unit number the host can recognize:

   CLI> **ADD UNIT *unit-number RAIDset-name***

5. Optional: set the Unit switches or skip this step to accept the defaults(*). For each switch you want to set, enter the following command:

```
CLI> SET unit-number SWITCH_VALUE
```

Table 14
nit switches

Unit switch	Value and syntax
Maximum cached transfer	MAXIMUM_CACHED_TRANSFER=32* MAXIMUM_CACHED_TRANSFER=N
Read cache	READ_CACHE* NOREAD_CACHE
Write-back cache	NOWRITEBACK_CACHE* WRITEBACK_CACHE
Availability	RUN* NORUN
Write protection	NOWRITE_PROTECT* WRITE_PROTECT

6. Verify the RAIDset configuration and switches:

```
CLI> SHOW RAIDset-name
```

7. Verify the unit configuration and switches:

```
CLI> SHOW unit-number
```

For example

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

```
CLI> ADD RAIDSET Raid1 disk100 disk200 disk300
CLI> INITIALIZE Raid1
CLI> ADD UNIT D300 Raid1
CLI> SET D300 WRITEBACK_CACHE
CLI> SHOW Raid1
CLI> SHOW D300
```

Configuring a striped mirrorset

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

1. Create—but don't initialize—at least two mirrorsets.
2. Create a stripeset and specify the mirrorsets it contains:

   CLI> **ADD STRIPESET *mirrorset_1 mirrorset_2***

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

   CLI> **INITIALIZE *stripeset-name SWITCH_VALUE***

Table 15
Initialize switches

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

4. Present the stripeset to the host by giving it a unit number the host can recognize:

   CLI> **ADD UNIT *unit-number stripeset-name***

5. Optional: set the Unit switches or skip this step to accept the defaults(*). For each switch you want to set, enter the following command:

   CLI> **SET *unit-number SWITCH_VALUE***

Table 16
Unit switches

Unit switch	Value and syntax
Maximum cached transfer	MAXIMUM_CACHED_TRANSFER=32* MAXIMUM_CACHED_TRANSFER=N
Read cache	READ_CACHE* NOREAD_CACHE
Write-back cache	NOWRITEBACK_CACHE* WRITEBACK_CACHE
Availability	RUN* NORUN
Write protection	NOWRITE_PROTECT* WRITE_PROTECT

6. Verify the striped mirrorset configuration and switches:

   CLI> **SHOW *stripeset-name***

7. Verify the unit configuration and switches:

   CLI> **SHOW *unit-number***

For 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.

```
CLI> ADD MIRRORSET Mirr1 disk100 disk200
CLI> ADD MIRRORSET Mirr2 disk300 disk400
CLI> ADD MIRRORSET Mirr3 disk500 disk600
CLI> ADD STRIPESET Stripe1 Mirr1 Mirr2 Mirr3
CLI> INITIALIZE Stripe1 CHUNKSIZE=default
CLI> ADD UNIT D101 Stripe1
CLI> SET D101 WRITEBACK_CACHE
CLI> SHOW Stripe1
CLI> SHOW D101
```

Configuring a single-disk unit

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

1. Add the disk drive by following the steps for *Adding disk drives* on page 60.
2. Optional: set the device switches for the disk drive or skip this step to accept the defaults(*). For each switch you want to set, enter the following command:

   ```
   CLI> SET DISKnnn SWITCH_VALUE
   ```

Table 17
Device switches

Device switch	Value and syntax
Transportability	NOTRANSPORTABLE* TRANSPORTABLE
Transfer rate	TRANSFER_RATE_REQUESTED=10MHZ* TRANSFER_RATE_REQUESTED=5MHZ TRANSFER_RATE_REQUESTED=ASYNCHRONOUS

3. Present the disk drive to the host by giving it a unit number the host can recognize:

   ```
   CLI> ADD UNIT unit-number DISKnnn
   ```

4. Optional: set the Unit switches or skip this step to accept the defaults(*). For each switch you want to set, enter the following command:

   ```
   CLI> SET unit-number SWITCH_VALUE
   ```

Table 18
Unit switches

Unit switch	Value and syntax
Maximum cached transfer	MAXIMUM_CACHED_TRANSFER=32* MAXIMUM_CACHED_TRANSFER=N
Read cache	READ_CACHE* NOREAD_CACHE
Write-back cache	NOWRITEBACK_CACHE* WRITEBACK_CACHE
Availability	RUN* NORUN
Write protection	NOWRITE_PROTECT* WRITE_PROTECT

6. Verify the configuration:

```
CLI> SHOW DEVICES
```

For example

The following example shows the commands you would use to configure DISK100 as a single-disk unit.

```
CLI> ADD DISK DISK100 1 0 0
CLI> ADD UNIT D101 disk100
CLI> SET D101 SAVE_CONFIGURATION
CLI> SHOW DEVICES
```

Configuring a tape drive

See Chapter 3 for information about creating a profile and understanding the Device switches you can set.

1. Add the tape drive to the controller's list of devices:

   ```
   CLI> ADD TAPE tapedrive-name PTL-location
   ```

2. Present the tape drive to the host by giving it a unit number the host can recognize:

   ```
   CLI> ADD UNIT unit-number tapedrive-name
   ```

3. Verify the configuration:

   ```
   CLI> SHOW DEVICES
   ```

For example

The following example shows the commands you would use to add Tape100 to your subsystem.

```
CLI> ADD TAPE Tape100 1 0 0
CLI> ADD UNIT T100 Tape100
CLI> SHOW DEVICES
```

Configuring a tape loader

The controller uses a passthrough device to transport the host's commands to and from a SCSI device such as a tape loader. For this reason, configuring a tape loader involves creating a passthrough device, then associating that device with the loader.

1. Install and configure the tape drive.
2. Install the tape loader.
3. Add the loader to the controller's list of devices:

   ```
   CLI> ADD LOADER loader-name PTL-location
   ```

4. Create a passthrough device at the same PTL location to logically represent the tape loader:

   ```
   CLI> ADD PASSTHROUGH passthrough-name PTL-location
   ```

5. Present the passthrough device to the host by giving it a unit number the host can recognize:

   ```
   CLI> ADD UNIT unit-number passthrough-name
   ```

6. Verify the configuration:

   ```
   CLI> SHOW DEVICES
   ```

7. Install and configure the host-based software that controls the loader. (This software is not provided with your HS array controller or its software.)

For example

The following example shows the commands you would use to add a loader to your subystem.

```
CLI> ADD LOADER Load130 1 3 0
CLI> ADD PASSTHROUGH Pass130 1 3 0
CLI> ADD UNIT T130 Pass130
CLI> SHOW PASSTHROUGH
```

Partitioning a storageset or disk drive

See *Planning your partitions* in Chapter 3 for information about partitioning a storage unit.

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

   ```
   CLI> ADD STORAGESET storageset-name DISKnnn DISKnnn
   ```

   or

   ```
   CLI> ADD DISK DISKnnn PTL-location
   ```

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

   ```
   CLI> INITIALIZE storageset-name
   ```

3. Create each partition in the storageset or disk drive by indicating the partition's size:

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

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

4. Verify the partitions:

   ```
   CLI> SHOW storageset-name
   ```

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

5. Present each partition to the host by giving it a unit number the host can recognize. You can skip this step until you're ready to put the partitions online:

   ```
   CLI> ADD UNIT unit-number storageset-name
        PARTITION=partition-number
   ```

6. Verify the unit numbers for the partitions:

   ```
   CLI> SHOW storageset-name
   ```

7. Optional: set the Unit switches for each partition, or skip this step to accept the defaults(*). For each switch you want to set:

```
CLI> SET unit-number VALUE
```

Table 19
Syntax for Unit switches

Unit switch	Value and syntax
Maximum cached transfer	MAXIMUM_CACHED_TRANSFER=32* MAXIMUM_CACHED_TRANSFER=N
Read cache	READ_CACHE* NOREAD_CACHE
Write-back cache	NOWRITEBACK_CACHE* WRITEBACK_CACHE
Availability	RUN* NORUN
Write protection	NOWRITE_PROTECT* WRITE_PROTECT

For example

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

```
CLI> ADD RAIDSET Raid1 disk100 disk200 disk300
CLI> INITIALIZE Raid1
CLI> CREATE_PARTITION Raid1 SIZE=25
CLI> CREATE_PARTITION Raid1 SIZE=25
CLI> CREATE_PARTITION Raid1 SIZE=25
CLI> CREATE_PARTITION Raid1 SIZE=LARGEST
CLI> SHOW Raid1
.
.
.
Partition number   Size           Starting Block   Used by
------------------------------------------------------------
1                  1915 (0.98 MB) 0
2                  1915 (0.98 MB) 1920
3                  1915 (0.98 MB) 3840
4                  2371 (1.21 MB) 5760
.
.
.
CLI> ADD UNIT D1 Raid1 PARTITION=1
CLI> ADD UNIT D2 Raid1 PARTITION=2
CLI> ADD UNIT D3 Raid1 PARTITION=3
CLI> ADD UNIT D4 Raid1 PARTITION=4
CLI> SHOW Raid1
.
.
.
Partition number   Size           Starting Block   Used by
------------------------------------------------------------
1                  1915 (0.98 MB) 0                D1
2                  1915 (0.98 MB) 1920             D2
3                  1915 (0.98 MB) 3840             D3
4                  2371 (1.21 MB) 5760             D4
.
.
.
CLI> SET D1 WRITEBACK_CACHE
CLI> SET D2 WRITEBACK_CACHE
CLI> SET D3 WRITEBACK_CACHE
CLI> SET D4 WRITEBACK_CACHE
```

Adding a disk drive to the spareset

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

Follow these steps to add a disk drive to the spareset:

1. Add the disk drive to the controller's spareset list:

   ```
   CLI> ADD SPARESET DISKnnn
   ```

2. Repeat step 1 for each disk drive you want to add to the spareset:

3. Verify the contents of the spareset:

   ```
   CLI> SHOW SPARESET
   ```

For example

The following example shows the commands you would use to add DISK600 and DISK610 to the spareset.

```
CLI> ADD SPARESET disk600
CLI> ADD SPARESET disk610
CLI> SHOW SPARESET
```

Removing a disk drive from the spareset

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

1. Show the contents of the spareset:

   ```
   CLI> SHOW SPARESET
   ```

2. Delete the desired disk drive:

   ```
   CLI> DELETE SPARESET DISKnnn
   ```

3. Verify the contents of the spareset:

   ```
   CLI> SHOW SPARESET
   ```

For example

The following example shows the commands you would use to remove DISK600 from the spareset.

```
CLI> SHOW SPARESET

Name       Storageset     Uses             Used by
-------------------------------------------------
SPARESET   spareset       disk600
                          disk610


CLI> DELETE SPARESET disk600
CLI> SHOW SPARESET

Name       Storageset      Uses             Used by
-------------------------------------------------
SPARESET   spareset        disk610
```

Deleting a storageset

Follow these steps to delete a storageset:

1. Show the configuration:

   ```
   CLI> SHOW STORAGESETS
   ```

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

   ```
   CLI> DELETE unit-number
   ```

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

   ```
   CLI> DELETE storageset-name
   ```

4. Verify the configuration:

   ```
   CLI> SHOW STORAGESETS
   ```

For example

The following example shows the commands you would use to delete Stripe1, a three-member stripeset that comprises DISK100, DISK200, and DISK300.

```
CLI> SHOW STORAGESETS

Name       Storageset     Uses       Used by
-------------------------------------------
STRIPE1    stripeset      DISK100    D100
                          DISK200
                          DISK300

CLI> DELETE D100
CLI> DELETE Stripe1
CLI> SHOW STORAGESETS
```

Changing switches for a storageset or device

You can optimize a storageset or device at any time by changing the switches that are associated with it. See *Choosing switches for your storagesets and devices* in Chapter 3 for an explanation of the switches.

Note

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:

```
CLI> SHOW storageset-name FULL
```

Changing RAIDset and Mirrorset switches

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

```
CLI> SET Raid1 POLICY=best_fit
```

Changing Device switches

Use the SET command to change the device switches. For example, the following command enables DISK100 to be used in a non-StorageWorks environment:

```
CLI> SET DISK100 TRANSPORTABLE
```

Changing Initialize switches

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

Changing Unit switches

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

```
CLI> SET D100 WRITE_PROTECT
```

6 Working with storage units

OpenVMS disk capacity limitations

OpenVMS command interactions

Considerations for host-based shadow sets

OpenVMS disk capacity limitations

OpenVMS VAX Versions 5.5-2 and earlier don't support disk capacities larger than 16,772,216 blocks (about 8.5 GB) as file-structured devices. You must keep this in mind when creating storage units, since stripesets and RAIDsets can easily exceed this limit.

The HSOF Version 3.0 software enforces a maximum byte count for ERASE commands of 4,194,303 blocks (about 2 GB). OpenVMS facilities that rely on ERASE commands automatically adjust to this behavior. It is only a concern for applications that issue ERASE commands directly.

OpenVMS command interactions

SHOW DEVICE

In OpenVMS Versions 5.5-2 and earlier, the SHOW DEVICE command doesn't recognize TRANSPORTABLE disk drives. Use the DCL lexical F$GETDVI to find the device type for a transportable disk drive.

INITIALIZE and BACKUP/IMAGE Commands

If you're using VAX VMS Version 5.5-1 or earlier to initialize a storage unit with the DCL INITIALIZE or BACKUP/IMAGE commands, you may receive the following message:

```
$ INITIALIZE device-name
  unfavorable cluster size
```

For the INITIALIZE command, you can avoid this message by specifying the cluster size:

```
$ INITIALIZE CLUSTER_SIZE = blocks device-name
```

For the BACKUP/IMAGE command, you can avoid the message by first initializing the target device as above, then using the NOINITIALIZE qualifier:

```
$ INITIALIZE CLUSTER_SIZE = blocks device-name
$ BACKUP/IMAGE/NOINITIALIZE...
```

Considerations for host-based shadow sets

Keep these points in mind when creating host-based shadow sets:

- In OpenVMS VAX versions earlier than Version 6.0, members may be dropped from the shadow set due to timed-out I/O requests. Change the value of the SYSGEN parameter SHADOW_MBR_TMO to at least 120 (seconds) to avoid dropping members.
- You cannot use transportable units as shadow set members.
- Host-based shadow sets may only use storage units that comprise the same device types and device capacities. For example, you can create a host-based shadow set from two stripesets of RZ26 disk drives, but you can't create a shadow set from a stripeset of RZ26 disk drives and a stripeset of RZ74 disk drives.

Note

The SAVE_CONFIGURATION switch affects the capacities of the disk drives in a storage unit. Thus, all or none of the storage units you want to combine into a host-based shadow set must be initialized with the SAVE_CONFIGURATION switch.

7 Periodic procedures

Automatically cloning data for backup

Shutting down your subsystem

Restarting your subsystem

Automatically cloning data for backup

Use CLONE to duplicate the data on any unpartitioned single-disk unit, stripeset, or mirrorset in preparation for backup. When the cloning operation is done, you can backup the clones rather than the storageset or single-disk unit, which can continue to service its I/O load.

Note

When you're cloning a mirrorset, CLONE doesn't need to create a temporary mirrorset. Instead, it adds a temporary member to the mirrorset and copies the data onto this new member.

CLONE 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're 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 CLONE 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.

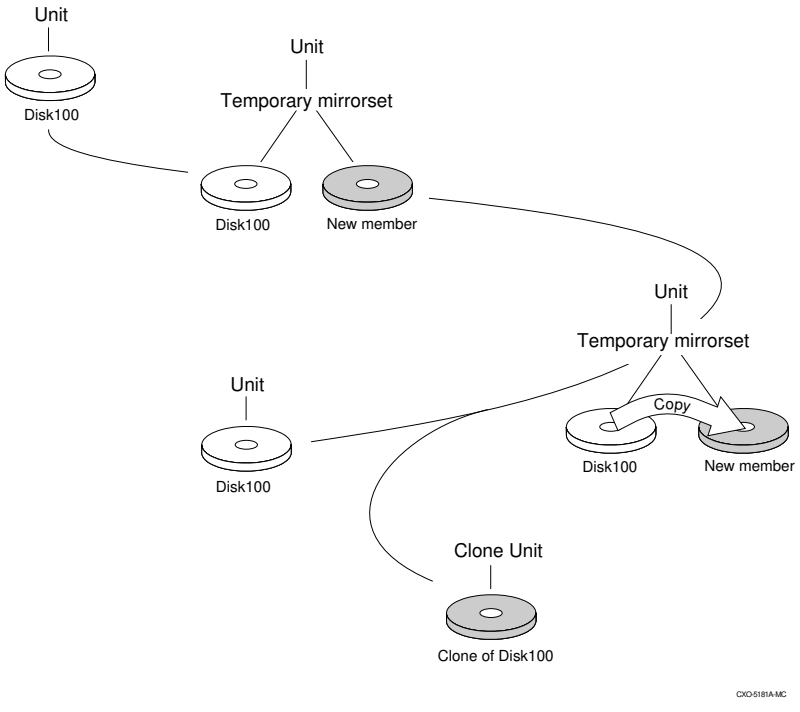


Figure 20 CLONE uses these key steps to duplicate each member of a unit.

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:

   ```
   CLI> 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. Backup the clone unit.

For example

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

```
CLI> RUN CLONE
Clone Local Program Invoked
Units available for cloning: 110
                             204

Enter unit to clone ? 204
Clone will create a new unit which is a copy of unit 204.

Enter the unit number which you want assigned to the new
unit ? 205
The new unit may be added using one of the following
methods:
1. Clone will pause after all members have been copied. The
   user must then press RETURN to cause the new unit to be
   added.
2. After all members have been copied, the unit will be
   added automatically.
```

```
Under which above method should the new unit be added[]?1
Devices available for clone targets:
DISK220 (size=832317)
DISK240 (size=832317)
DISK310 (size=832317)

Use available device DISK220(size=832317) for member
DISK130(size=832317) (y,n) [y] ? y
mirror DISK130 C_MA
set C_MA nopolicy
set C_MA members=2
set C_MA replace=DISK220

Devices available for clone targets:
DISK240 (size=832317)
DISK310 (size=832317)

Use available device DISK240(size=832317) for member
DISK200(size=832317) (y,n) [y] ? y
mirror DISK200 C_MB
set C_MB nopolicy
set C_MB members=2
set C_MB replace=DISK240

Copy in progress for each new member. Please be patient...
.
.
copy from DISK130 to DISK220 is 100% complete
copy from DISK200 to DISK240 is 100% complete

Press RETURN when you want the new unit to be created
reduce  DISK220 DISK240
unmirror DISK130
unmirror DISK200
add mirrorset C_MA       DISK220
add mirrorset C_MB       DISK240
add stripeset C_ST1  C_MA C_MB
init C_ST1     nodestroy chunk=128
add unit D205 C_ST1

D205 has been created. It is a clone of D204.

Clone - Normal Termination
```

Shutting down your subsystem

Follow these steps to shut down your «Brand» subsystem for any reason, such as a long holiday, a system move, or maintenance.

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

   ```
   CLI> SHUTDOWN OTHER_CONTROLLER
   CLI> SHUTDOWN THIS_CONTROLLER
   ```

4. Turn off the power to the subsystem.
5. Unplug the subsystem’s power cord.

Restarting your subsystem

Follow these steps to restart your subsystem:

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

   ```
   CLI> SHOW THIS_CONTROLLER
   ```

   If the battery is low, you won't be able to access your RAIDsets or mirrorsets until it recharges unless you change the cache policy to "B:"

   ```
   CLI> SET CACHE_POLICY=B
   ```

Note

Cache Policy "B" enables you to access RAIDsets and mirrorsets even if the backup battery is drained. However, you risk losing data if the power is lost before the batteries have recharged completely.

Appendix

Storageset profile

Device profile

SW800 and SW500 cabinet templates

SW300 cabinet template

Storageset profile

Type

☐ RAIDset ☐ Mirrorset ☐ Stripeset ☐ Striped mirrorset

Storageset name _____

Disk drives _____

Unit number _____

Partitions

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

RAIDset switches

Reconstruction policy
- ☐ Normal (default)
- ☐ Fast

Reduced membership
- ☐ No (default)
- ☐ Yes, missing:

Replacement policy
- ☐ Best performance (default)
- ☐ Best fit
- ☐ None

Mirrorset switches

Replacement policy
- ☐ Best performance (default)
- ☐ Best fit
- ☐ None

Copy policy
- ☐ Normal (default)
- ☐ Fast

Read source
- ☐ Least busy (default)
- ☐ Round robin
- ☐ Disk drive:

Initialize switches

Chunksize
- ☐ Automatic (default)
- ☐ 64 blocks
- ☐ 128 blocks
- ☐ 256 blocks
- ☐ Other:

Metadata
- ☐ Destroy (default)
- ☐ Retain

Saved configuration
- ☐ No (default)
- ☐ Yes

UNIT switches

Read cache
- ☐ Yes (default)
- ☐ No

Write cache
- ☐ No (default)
- ☐ Yes

Maximum cache transfer
- ☐ 32 blocks (default)
- ☐ Other:

Availability
- ☐ Run (default)
- ☐ NoRun

Write protection
- ☐ No (default)
- ☐ Yes

Device profile

Type

- ☐ Platter disk drive
- ☐ Tape drive
- ☐ Optical disk drive
- ☐ CD-ROM

Device name _____

Unit number _____

Device switches

Transportability
- ☐ No (default)
- ☐ Yes

Initialze switches

Chunksize
- ☐ Automatic (default)
- ☐ 64 blocks
- ☐ 128 blocks
- ☐ 256 blocks
- ☐ Other:

Saved configuration
- ☐ No (default)
- ☐ Yes

Metadata
- ☐ Destroy (default)
- ☐ Retain

UNIT switches

Read cache
- ☐ Yes (default)
- ☐ No

Write cache
- ☐ No (default)
- ☐ Yes

Maximum cache transfer
- ☐ 32 blocks (default)
- ☐ Other:

Availability
- ☐ Run (default)
- ☐ NoRun

Write protection
- ☐ No (default)
- ☐ Yes

SW800 and SW500 cabinets

Note

This map provides more than enough PTL slots to map a controller or a pair of dual-redundant controllers. A single controller can support up to 42 devices. Dual-redundant controllers can support up to 36 devices. Each shelf can support an optional power supply in addition to the ones shown.

Power supply							
Power supply							
Power supply							
Power supply							
Power supply							
Power supply							
Power supply							

SW300 cabinet

Note

This map provides more than enough PTL slots to map a controller or a pair of dual-redundant controllers. A single controller can support up to 42 devices. Dual-redundant controllers can support up to 36 devices. Each shelf can support an optional power supply in addition to the ones shown.

Power supply							
Power supply							
Power supply							
Power supply							

Glossary

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

Allocation class
A numerical value assigned to a controller to identify units across multiple, independent controllers. (Controllers in a dual-redundant configuration must have the same allocation class.)

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

BBR
Bad block replacement. The procedure used to locate a replacement block, mark the bad block as replaced, and move the data from the bad block to the replacement block.

BBU
Battery backup unit. A StorageWorks SBB option that extends power availability after the loss of primary ac power or a power supply to protect against the corruption or loss of data.

Block
The smallest data unit addressable on a disk. Also called a sector. In integrated storage elements, a block contains 512 bytes of data, EDC, ECC, flags, and the block's address header.

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

CI bus
Computer interconnect bus. Uses two serial paths, each with a transfer rate of 70 Mb/s (8.75 MB/s).

CLI
Command line interpreter. Operator command line interface for the HS family controller firmware.

Controller shelf
A StorageWorks shelf designed to contain controller and cache memory modules.

CRC
An 8-character cyclic redundancy check string used in conjunction with the customer identification string for turning on licensed features such as write-back caching.

Data center cabinet
A generic reference to the large cabinets, such as the SW800-series, in which StorageWorks components can be mounted.

DDL
Dual data link. The ability to operate on the CI bus using both paths simultaneously to the same remote node.

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

DILX
Disk inline exerciser. Diagnostic firmware used to test the data transfer capabilities of disk drives in a way that simulates a high level of user activity.

DSA
Digital storage architecture. A set of specifications and interfaces describing standards for designing mass storage products. DSA defines the functions performed by host computers, controllers, and disk drives. It also specifies how they interact to accomplish mass storage management.

DSSI
Digital storage system interconnect. A Digital-specific data bus with an 8-bit data transfer rate of 4 MB/s.

Dual-redundant configuration
Two controllers in one controller shelf providing the ability for one controller to take over the work of the other controller in the event of a failure of the other controller.

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

DUP
Diagnostic and utility protocol. Host application software that allows a host terminal to be connected to the controller's command line interpreter.

DWZZA
The StorageWorks compatible SCSI bus signal converter.

ECB
External cache battery.

ECC
Error correction code. One or more cyclic redundancy check (CRC) words that allow detection of a mismatch between transmitted and received data in a communications system, or between stored and retrieved data in a storage system. The ECC allows for location and correction of an error in the received/retrieved data. All ECCs have limited correction power.

EDC
Error detection code. One or more checksum words that allow detection of a mismatch between transmitted and received data in a communications system, or between stored and retrieved data in a storage system. The EDC has no data correction capability.

ESD
Electrostatic discharge. The discharge of a potentially harmful static electric voltage as a result of improper grounding.

Failedset
A group of disk drives that have been removed from RAIDsets due to a failure or a manual removal. Disk drives in the failedset should be considered defective and should be tested, repaired, and then placed into the spareset.

Failover
The process that takes place when one controller in a dual-redundant configuration assumes the workload of a failed controller.

Flush
The act of writing data from the cache module to the media.

FRU
Field replaceable unit. A hardware component that can be replaced.

FWD SCSI
Fast, wide, differential SCSI. The differential SCSI bus with a 16-bit parallel data path that yields a transfer rate of up to 20 MB/s.

Half-height device
A device that occupies half of a 5.25 inch SBB carrier. Two half-height devices can be mounted in a 5.25 inch SBB carrier. The first half-height device is normally mounted in the lower part of the carrier. The second device is normally mounted in the upper part of the carrier.

HBVS
Host-based volume shadowing. Also known as Phase 2 volume shadowing.

HSOF
Hierarchical storage operating firmware. Software contained on a program card that provides the logic for the HS array controllers.

HIS
Host interconnect services. The firmware in the HS array controller that communicates with the host.

Host
Any computer to which a storage subsystem can be attached.

Hot swap
A method of replacing a device whereby the system that contains the device remains online and active during replacement. The device being replaced is the only device that cannot perform operations during a hot swap.

Initiator
A SCSI device that requests an I/O process to be performed by another SCSI device (a target). This is always the controller.

Local terminal
A terminal plugged into the EIA-423 maintenance port on the front bezel of the HS array controller. Also called a maintenance terminal.

Logical unit
The physical device or storage unit seen by the host. Often these logical units are spread across more than one physical device, especially in RAID implementations. This is *not* a LUN.

Logical Unit Number
See LUN.

LRU
Least recently used. This is cache terminology for the block replacement policy for the read cache.

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

Maintenance terminal
Any EIA-423 compatible terminal to be plugged into the HS controller. This terminal is used to identify the controller, enable host paths, define the configuration, and check controller status. It is not required for normal operations. It is sometimes referred to as a local terminal.

Metadata
Data written on the physical disk that is not visible to the host/customer that allows the HS array controller to maintain a high integrity of customer data.

Mirrorset
Two or more physical disks configured to present one highly reliable virtual unit to the host.

MSCP
Mass storage control protocol. The protocol by which blocks of information are transferred between the host and the controller.

Non-redundant configuration
A single controller configuration. A controller configuration which does not include an second backup controller permitting failover in the event of a failure.

Normal member
A mirrorset member whose entire contents is guaranteed to be the same as all other NORMAL members. All NORMAL members are exactly equivalent.

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. Data that has never been written may differ among NORMALIZING members.

NV
Nonvolatile. A term used to describe memory that can retain data during a power loss to the controller.

Partition
A percentage of a storageset or single-disk unit that may be presented to the host as a storage unit.

PPort
The hardware and software used to connect a host controller to a communication bus, such as CI, DSSI, or SCSI bus. This term also is used to describe the connection between the controller and its SCSI storage devices.

PTL
Port-target-LUN. A method of device notation where **P** designates the controller's device port (1-6), **T** designates the target ID of the device (0-6), and **L** designates the LUN of the device (0-7).

Qualified device
A device that has been fully tested in an approved StorageWorks configuration, (that is, shelf, cabinet, power supply, cabling, and so forth) and is in complete compliance with country-specific standards (for example, FCC, TUV, and so forth) and with all Digital standards.

Quiesce
To make a bus inactive or dormant. The operator must quiesce SCSI bus operations, for example, during a device warm swap.

RAID
Redundant array of independent disks. The multiple storage access methods devised for performance (RAID 0, striping) and/or various cost levels of availability (RAID 1 through RAID 5).

RAIDset
Three or more physical disks that are configured to present an array of disks as a single virtual unit to the host.

Read cache
The cache used to accelerate read operations by retaining data which has been previously read, written, or erased, based on a prediction that it will be reread.

Replacement policy
The method by which a spare disk is selected to replace a disk that has failed in a RAIDset.

SBB
StorageWorks building block. A modular carrier plus the individual mechanical and electromechanical interface required

to mount it into a standard StorageWorks shelf. Any device conforming to shelf mechanical and electrical standards is considered an SBB.

SBB shelf
StorageWorks building block shelf. A StorageWorks shelf, such as the BA350--Sx, designed to house plug-in SBB modules.

SCS
System communication services. A delivery protocol for packets of information (commands or data) to or from the host.

SCSI
Small computer system interface. An ANSI interface defining the physical and electrical parameters of a parallel I/O bus used to connect initiators to a maximum of seven devices. The StorageWorks device interface is implemented according to SCSI--2 standard, allowing the synchronous transfer of 8-bit data at rates of up to 10 MB/s.

SCSI device
A host computer adapter, a peripheral controller, or a storage element that can be attached to the SCSI bus.

SCSI device ID
The bit-significant representation of the SCSI addressing that refers to one of the signal lines numbered 0 through 7. Also referred to as a *target* ID.

SCSI--A cable
A 50-conductor (25 twisted pair cable used for single-ended, SCSI bus connections.

SCSI--P cable
A 68-conductor (34 twisted pair cable used for differential bus connections.

Small Computer System Interface
See SCSI.

Spareset
A pool of disk drives used by the controller to replace failed members of a RAIDset.

SPD
Software product description. A document that contains the legal description of a product.

Storageset
Any collection of containers, such as stripesets, RAIDsets, the spareset, and the failedset, that make up a container.

Storage unit
The general term that refers to storagesets, single disk units, and all other storage devices that can be installed in your subsystem and accessed by a 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
Digital's family of modular data storage products that allows 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 enclosure to form storage subsystems.

StorageWorks building block
See SBB.

Stripeset
A virtual disk drive with its physical data spread across multiple physical disks. Stripeset configurations do not include a data recovery mechanism.

Striped mirrorset
Stripesets whose members have been mirrored.

Tagged command queuing
A SCSI feature that allows a device to have multiple I/O requests outstanding to it at one time.

Target
A SCSI device that performs an operation requested by an initiator. The target number is determined by the device's address on its SCSI bus.

TILX
Tape inline exerciser. Diagnostic firmware used to test the data transfer capabilities of tape drives in a way that simulates a high level of user activity.

TMSCP
Tape mass storage control protocol. The protocol by which blocks of information are transferred between the host and the controller.

Unit
The host's view of a container on an HS array controller. A unit may be made up of simply a physical disk or tape drive, or a more complex container such as a RAIDset.

Unwritten cached data
Data in the write-back cache which has not yet been written to the physical device, but the user has been notified that the data has been written.

VCS
VAXcluster console system.

Virtual terminal
A software path from an operator terminal on the host to the controller's CLI. The path can be established via the host port on the controller (using DUP) or via the maintenance port through on intermediary host (VCS). A virtual terminal is also sometimes called a host console.

Warm swap
A method for adding or replacing a device whereby the system remains online, but all activity on the device's bus must be halted for the duration of the swap.

Write-back caching
A caching strategy that writes data to the cache memory, then flushes the data to the intended device at some future time. From the user's perspective, the write operation is complete when the data is stored in the cache memory. This strategy avoids unnecessary access of the devices.

Write hole
Undetectable RAID level 1 or 5 data corruption. A write hole is caused by the successful writing of some, but not all, of the storageset members. 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 and writes can be retried.

Write-through cache

A cache write strategy in which the destination of the write data is the primary storage media. This operation may update, invalidate, or delete data from the cache memory accordingly, to ensure that the cache does not contain obsolete data. The user sees the operation as complete only after the backup storage device has been updated.

Index

A

Adapter, 98
Adding
  disk drives to configuration, 60
  disk drives to spareset, 76
  a reduced RAIDset, 37
Addressing, 50
Allocation class, 98
Array controller, 98
Automatic naming convention, 57
Automatically configuring
  mirrorsets, 56
  RAIDsets, 56
  striped mirrorsets, 56
  stripesets, 56

B

Backup, 86
BBR, 98
BBU, 98
Block, 98
Bus
  distributing members across, 27, 29, 31

C

Cache sizes supported, 4
CDU, 98
CFMENU, 54
  configuring storagesets, 56
  considerations for using, 57
  running on dual-redundant controllers, 57
Changing switches, 35
Chunk size
  choosing for RAIDsets and stripesets, 41
  displaying with CFMENU, 55
  maximum for RAIDsets, 43
  using to increase request rate, 41
  using to increase transfer rate, 42
CI bus, 98
CI cables
  maximum cable lengths, 18, 20
CLI, 98
  accessing from a VCS, 11
Cloning data for backup, 86
Command disk, 72
Comparison of storagesets, 24
Configuration
  dual-redundant controllers and CFMENU, 57
  key steps, 6
  saving, 43
Configured PTLs
  displaying with CFMENU, 55
Configuring
  disk drives, 60
  partitions, 73
  storagesets with CFMENU, 56
  tape drive, 71
  tape loader, 72
Control panel, 5
Controller
  components, 5
  features and general operation, 2
Controller shelf, 98
Copy speed
  setting for mirrorsets, 38
CRC, 99
Creating
  partitions, 73
  storageset profile, 22

D

Data center cabinet, 99
DDL, 99
Default device name, 51
Defining your storage requirements, 23
Deleting storagesets, 78
Device name
  displaying with CFMENU, 55
Device ports, 5
Device protocol, 4
Devices
  assigning unit numbers to, 48
  changing switches, 79
  switches, 40
  transfer rate, 40
Differential SCSI bus, 99
DILX, 99
Disk drive capacity limitations, 82

Disk drives
adding to spareset, 76
configuring, 60
configuring a single-disk drive unit, 69
removing from spareset, 77
Displaying switches, 79
Distributing
first members of multiple mirrorsets, 28
members of storageset, 27, 29, 31
DSA, 99
DSSI, 99
Dual-redundant configuration, 99
Dual-redundant controllers
using CFMENU, 57
DUART, 99
DUP, 99
DWZZA, 100

E

ECB, 100
ECC, 100
EDC, 100
Enabling switches, 35
Erasing metadata, 44
ESD, 100
Examples
configuring a spareset, 76
configuring a tape drive, 71
configuring a tape loader, 72
manually configuring loaders, 72
manually configuring mirrorsets, 64
manually configuring RAIDsets, 66
manually configuring single-disk unit, 70
manually configuring striped mirrorsets, 68
manually configuring stripesets, 62
manually partitioning a storage unit, 75

F

Failedset, 100
Failover, 100
Flush, 100
Format for tapes, 47
FRU, 100
FWD SCSI, 100

G

Guidelines for partitioning storagesets and disk drives, 34

H

Half-height device, 101
HBVS, 101
HIS, 101
Host, 101
Host port, 5
Host protocol, 4
Host-based shadow sets, 84
Hot swap, 101
HSOF, 101

I

I/O request routing, 51
Initialize switches, 41
Initiator, 101
Interconnect supported, 4

J

JBOD, 24

L

Largest device supported, 4
Loader
example of configuring manually, 72
Local connection jack, 5
Local terminal, 101
LOG qualifier, 10
Logical unit, 101
LRU, 101
LUN, 102

M

Maintenance terminal, 102
Manually configuring
mirrorsets, 63
RAIDset, 65
single-disk drive unit, 69
striped mirrorsets, 67
stripesets, 61

Mapping your storagesets, 49
Mean time between failures
  calculating, 26
Members
  distributing first members of mirrorsets, 28
  distributing on bus, 27, 29, 31
  maximum for RAIDset, 31
Metadata, 102
  erasing or retaining, 44
Mirrorset
  configuring automatically, 56
  configuring manually, 63
  considerations for planning, 28
  defined, 28
  distributing first members, 28
  example of configuring manually, 64
  setting copy speed, 38
  setting read source, 39
  switches, 38
Mirrorsets
  changing switches, 79
MSCP, 102
MTBF
  calculating, 26

N

Non-redundant configuration, 102
Normal member, 102
NV, 102

O

OpenVMS command interactions, 83
OpenVMS disk capacity limitations, 82
Options
  for devices, 40
  for mirrorsets, 38
  for RAIDsets, 36
  for storage units, 45
  initialize, 41
Options for storagesets and devices, 35

P

Partitioning a storageset or disk drive, 73
Partitions
  displaying with CFMENU, 55
  example of configuring manually, 75
  planning, 33
Path
  preferring for storage units, 45
Planning
  mirrorset, 28
  partitions, 33
  RAIDset, 30
  striped mirrorset, 32
  stripeset, 25
Port, 103
Preferred paths for storage units, 45
Product ID
  displaying with CFMENU, 55
Profile
  creating for storagesets and devices, 22
Program card slot, 5
Protocol
  devices, 4
  host, 4
PTL addressing convention, 50

Q

Qualified device, 103
Quiesce, 103

R

RAID levels supported, 4
RAIDset
  choosing a chunk size for, 41
  configuring automatically, 56
  considerations for planning, 31
  defined, 30
  example of configuring manually, 66
  manually configuring, 65
  maximum chunk size for, 43
  maximum membership, 31
  setting reconstruction policy, 37
  setting reduced membership, 37
  setting replacement policy, 36, 38
  switches, 36
RAIDsets
  changing switches, 79
Read cache, 46, 103

Read source
  setting for mirrorsets, 39
Reconstruction policy
  setting for RAIDsets, 37
Reduced membership
  setting for RAIDsets, 37
Removing
  disk drives from spareset, 77
Removing storagesets, 78
Replacement policy
  setting for RAIDsets, 36, 38
Request rate
  how chunk size affects, 41
Reset button, 5
Restarting your subsystem, 90

S

Saving your configuration, 43
SBB, 103
SCS, 104
SCSI, 104
Shadow sets, 84
Shutting down your subsystem, 89
Single-disk unit
  example of configuring manually, 70
Spareset
  example of configuring, 76
Sparesets, 76
  adding disk drives, 76
  removing disk drives, 77
SPD, 104
Starting your subsystem, 90
Storage requirements
  defining, 23
Storage unit, 105
Storageset
  changing switches, 79
  switches for, 35
Storageset map, 49
Storageset profile
  creating, 22
Storageset type
  displaying with CFMENU, 55
Storagesets
  assigning unit numbers, 48
  configuring with CFMENU, 56
  deleting, 78
  displaying switches for, 79
  overview, 24
  setting preferred path, 45
Striped mirrorset
  configuring automatically, 56
  considerations for planning, 32
  defined, 32
  example of configuring manually, 68
  manually configuring, 67
Stripeset
  choosing a chunk size for, 41
  configuring automatically, 56
  configuring manually, 61
  considerations for planning, 26
  defined, 25
  distributing members, 27, 29, 31
  example of configuring manually, 62
Switches
  changing, 35, 79
  changing device switches, 79
  changing Initialize switches, 79
  changing RAIDset and mirrorset switches, 79
  changing unit switches, 80
  enabling, 35
  for devices, 40
  for mirrorsets, 38
  for RAIDsets, 36
  for storage units, 45
  initialize, 41
Switches for storagesets and devices, 35

T

Tagged command queuing, 105
Tape drives
  configuring, 71
  example of configuring, 71
Tape formats, 47
Tape loaders
  configuring, 72
  example of configuring, 72
Target, 105
TILX, 105
TMSCP, 106
Transfer rate, 40
  how chunk size affects, 42

Transporatability, 40

U

Unconfigured Device PTLs
  displaying with CFMENU, 55
Unit, 106
Unit number
  displaying with CFMENU, 55
Unit numbers
  assigning to storagesets, 48
Unit switches, 45
  availability, 46
  maximum cache transfer, 46
  read cache, 46
  tape format, 47
  write protection, 47
  write-back cache, 46

V

VAXcluster console system. See VCS
VCS, 11, 106
Virtual terminal, 106

W

Warm swap, 106
Write hole, 106
Write protection switch
  displaying with CFMENU, 55
Write-back cache switch
  displaying with CFMENU, 55
Write-through cache, 107