

DIGITAL StorageWorks  
HSD50 Array Controller

---

H5OF 5.1  
Configuration Manual

Part Number: EK-HSD50-CG. B01

**March 1997**

**Digital Equipment Corporation  
Maynard, Massachusetts**

---

**March 1997**

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, 1997.

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

CI, DIGITAL, HSC, HSJ, HSD, HSZ, OpenVMS, StorageWorks, VAX, VAXcluster, 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.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company LTD.

SUN is a registered trademark of Sun Microsystems Corp.

IBM is a registered trademark of International Business Machines Corp.

Windows NT is a registered trademark of Microsoft Corporation.

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.

---

# Table of Contents

<b>1 Introduction .....</b>	<b>1-1</b>
Features of your controller.....	1-2
Anatomy of your controller .....	1-5
Key steps for configuring your subsystem.....	1-6
<b>2 Configuring your controller.....</b>	<b>2-1</b>
Key steps for configuring a controller.....	2-2
Communicating with a controller.....	2-2
Establishing a local connection.....	2-3
Establishing a remote connection.....	2-4
Installing the FYDRIVER .....	2-4
Starting a DUP terminal .....	2-4
Starting a VCS terminal session .....	2-5
Configuring a controller .....	2-5
Configuring a single controller .....	2-6
Configuring dual-redundant controllers.....	2-7
Connecting a controller to the host .....	2-9
Connecting a single controller to the host .....	2-9
Connecting dual-redundant controllers to the host.....	2-11
<b>3 Planning your storage sets .....</b>	<b>3-1</b>
Creating a profile.....	3-2
Defining your storage requirements .....	3-3
Choosing a storage set .....	3-3
Using stripesets to increase I/O performance .....	3-4
Considerations for planning a stripeset.....	3-6

Using mirrorsets to ensure availability.....	3-8
Considerations for planning a mirrorset .....	3-8
Using RAIDsets to increase performance and availability.....	3-10
Considerations for planning a RAIDset.....	3-11
Using striped mirrorsets for the highest performance and availability.....	3-11
Considerations for planning a striped mirrorset.....	3-12
Planning your partitions.....	3-12
Defining a partition .....	3-12
Guidelines for partitioning storagesets and disk drives.....	3-13
Choosing switches for your storagesets and devices.....	3-13
Enabling switches.....	3-14
Changing switches.....	3-14
RAIDset switches .....	3-14
Replacement policy .....	3-14
Reconstruction policy .....	3-15
Membership.....	3-15
Mirrorset switches .....	3-15
Replacement policy .....	3-15
Copy speed.....	3-16
Read source.....	3-16
Device switches.....	3-17
Transportability .....	3-17
Transfer rate .....	3-17
Initialize switches.....	3-18
Chunk size.....	3-18
Increasing the request rate .....	3-18
Increasing the data transfer rate .....	3-19
Increasing sequential write performance .....	3-20
Maximum chunk size for RAIDsets .....	3-20
Save configuration.....	3-21
Overwrite .....	3-22
Unit switches.....	3-23
Preferred path dual-redundant configurations.....	3-23
Read cache .....	3-24
Write-back cache.....	3-24
Maximum cache transfer .....	3-24
Availability .....	3-24
Tape format.....	3-25
Write protection .....	3-25
Assigning unit numbers .....	3-25
Creating a storageset map.....	3-26
PTL addressing convention.....	3-27
The next step.....	3-29

## 4 Automatically configuring storagesets .....4-1

Introducing CFMENU .....	4-2
Considerations for using CFMENU .....	4-4
Adding devices with CFMENU .....	4-4
Deleting Devices with CFMENU.....	4-5
Creating a Storageset with CFMENU .....	4-5
Deleting a Storageset with CFMENU .....	4-7
Adding a Disk Drive to a Spareset with CFMENU .....	4-7
Initializing Containers with CFMENU.....	4-8
Adding Units with CFMENU.....	4-8
Deleting a Disk Drive from a Spareset with CFMENU .....	4-9
Partitioning a Container with CFMENU .....	4-9
Adding a Passthrough with CFMENU .....	4-10
Deleting a Passthrough with CFMENU.....	4-10
Adding a Partitioned Unit with CFMENU .....	4-11
Deleting a Partitioned Unit with CFMENU.....	4-11

## 5 Manually configuring storagesets .....5-1

Adding disk drives.....	5-2
Adding one disk drive at a time .....	5-2
Adding several disk drives at a time.....	5-2
Adding CD-ROM drives .....	5-2
Adding one CD-ROM drive at a time.....	5-2
Adding several CD-ROM drives at a time .....	5-2
Configuring a stripeset.....	5-3
Configuring a mirrorset .....	5-5
Configuring a RAIDset.....	5-7
Configuring a striped mirrorset.....	5-9
Configuring a single-disk unit.....	5-11
Configuring a tape drive .....	5-12
Configuring a tape loader .....	5-13
Partitioning a storageset or disk drive .....	5-14
Adding a disk drive to the spareset .....	5-16
Removing a disk drive from the spareset .....	5-17
Deleting a storageset.....	5-17
Changing switches for a storageset or device .....	5-18
Displaying the current switches .....	5-18
Changing RAIDset and Mirrorset switches .....	5-19
Changing Device switches .....	5-19
Changing Initialize switches .....	5-19
Changing Unit switches .....	5-19

<b>6 Working with storage units.....</b>	<b>6-1</b>
OpenVMS disk capacity limitations .....	6-2
OpenVMS command interactions .....	6-3
SHOW DEVICE.....	6-3
INITIALIZE and BACKUP/IMAGE Commands .....	6-3
Considerations for host-based shadow sets.....	6-4
<b>7 Periodic Procedures .....</b>	<b>7-1</b>
Automatically cloning data for backup .....	7-2
For example.....	7-3
Shutting down your subsystem .....	7-4
Restarting your subsystem.....	7-5

## Appendix A Profiles and Templates

### Glossary

### Index

### Figures

Figure 1-1 Bridging the gap between the host and its storage subsystem .....	1-2
Figure 1-2 The host recognizes units created from storagesets, partitions, and disk drives .....	1-3
Figure 1-3 Key Components .....	1-5
Figure 1-4 Key steps for configuring your subsystem.....	1-6
Figure 2-1 Connect your terminal to the local-connection port to configure the controller for the first time .....	2-3
Figure 2-2 Connecting a single controller to its host .....	2-9
Figure 2-3 Connecting dual-redundant controllers to the host .....	2-11
Figure 3-1 A typical storageset profile .....	3-2
Figure 3-2 Striping lets several disk drives participate in each I/O request .....	3-5
Figure 3-3 Distribute members across ports .....	3-7
Figure 3-4 Mirrorsets maintain two copies of the same data.....	3-8
Figure 3-5 Put first members on different busses.....	3-9
Figure 3-6 Parity ensures availability; striping provides good performance.....	3-10
Figure 3-7 Striping and mirroring in the same storageset .....	3-11
Figure 3-8 Each partition can be presented to the host as a storage unit .....	3-12

Figure 3-9	If chunk size is larger than the request size, then each disk drive in the storageset can respond to a separate I/O request. ....	3-19
Figure 3-10	If chunk size is smaller than the request size, then more than one disk drive can respond to the same I/O request. ....	3-20
Figure 3-11	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. ....	3-26
Figure 3-12	PTL addressing .....	3-28
Figure 7-1	CLONE uses these key steps to duplicate each member of a unit. ....	7-2

## Tables

Table 1-1	Summary of features .....	1-4
Table 2-1	Maximum DSSI cable lengths.....	2-10
Table 3-1	A comparison of different kinds of storagesets.....	3-4
Table 3-2	Maximum chunk sizes for a RAIDset .....	3-20
Table 3-3	Unit switches .....	3-23
Table 4-1	Interpreting CFMENU Columns.....	4-3
Table 5-1	Initialize switches .....	5-3
Table 5-2	Unit switches .....	5-4
Table 5-3	Mirrorset switches .....	5-5
Table 5-4	Initialize switches .....	5-5
Table 5-5	Unit switches .....	5-6
Table 5-6	RAIDset switches .....	5-7
Table 5-7	Initialize switches .....	5-7
Table 5-8	Unit switches .....	5-8
Table 5-9	Initialize switches .....	5-9
Table 5-10	Unit switches .....	5-10
Table 5-11	Device switches .....	5-11
Table 5-12	Unit switches .....	5-12
Table 5-13	Syntax for Unit switches .....	5-15

## Related documents

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

<b>Document title</b>	<b>Part number</b>
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 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-UG
StorageWorks SW800-Series Data Center Cabinet Installation and User's Guide	EK-SW800-UG
The RAIDBOOK—A Source for RAID Technology	RAID Advisory Board
Polycenter Console Manager User's Guide	Computer Associates
VAXcluster Systems Guidelines for VAXcluster System Configurations	EK-VAXCS-CG
16-Bit SBB User's Guide	EK-SBB16-UG
7-Bit SBB Shelf (BA356 Series) User's Guide	EK-BA356-UG
SBB User's Guide	EK-SBB35-UG

# 1

## Introduction

Features of your controller

Anatomy of your controller

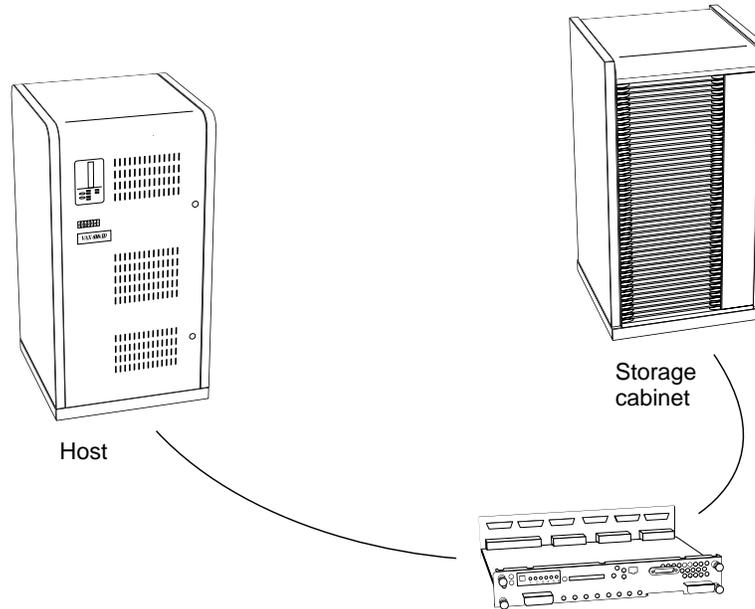
Key steps for configuring your subsystem

## Features of your controller

Your HSD50 controller is the intelligent bridge between your host and the devices in your subsystem.

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

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



CXO-5127A-MC

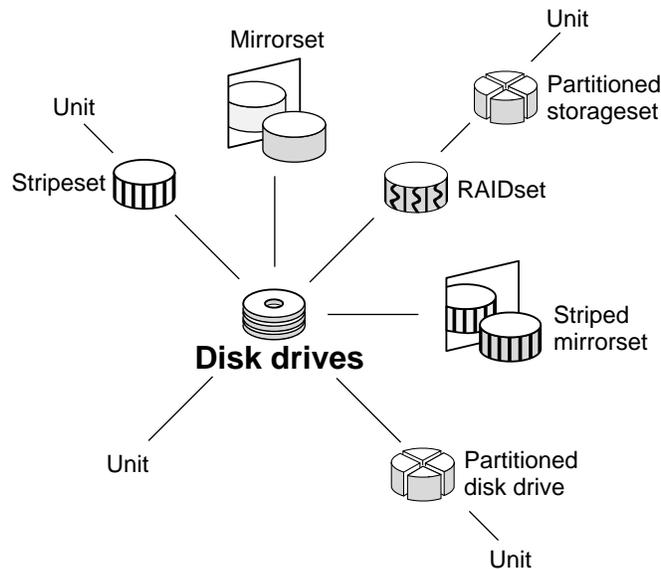
From the subsystem's perspective, the controller receives the I/O requests from the host and directs them to the devices in the subsystem. Since the controller processes all of the I/O requests, it eliminates the host-based processing that's typically associated with reading and writing data to multiple 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 mirrorsets, or by presenting them to the host as single-disk units as shown in Figure 1-2.

**Figure 1-2 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, CD-ROM drives, loaders, and libraries to your subsystem to meet all of your storage requirements.

**Table 1-1 Summary of features**

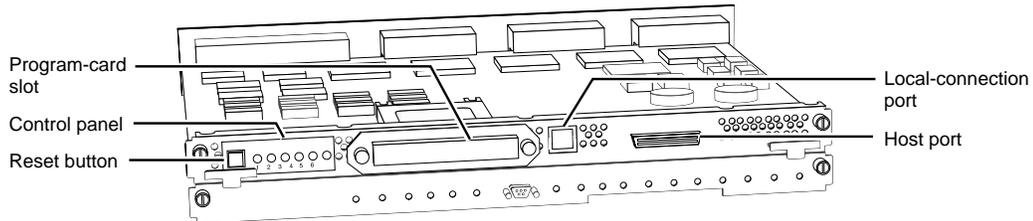
This table summarizes the features of your controller.

<b>Feature</b>	<b>Supported</b>
Host bus interconnect	DSSI
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	✓
Device warm swap	✓
Error correction code	✓
Exercisers for testing devices	✓
Tape drives, loaders, and libraries	✓
Maximum number of configuration entities (devices + storage sets + partitions + units)	195
Maximum number of storage sets	30
Maximum number of partitions per storage set or disk drive	4
Number of RAID sets and mirror sets running simultaneously	20
Maximum number of units presented to host	32
Maximum number of devices per unit	32
Largest storage set	120 GB
Remote Disk Copy Data	✓

## Anatomy of your controller

Your HSD50 controller contains the key components shown in Figure 1-3.

**Figure 1-3 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.

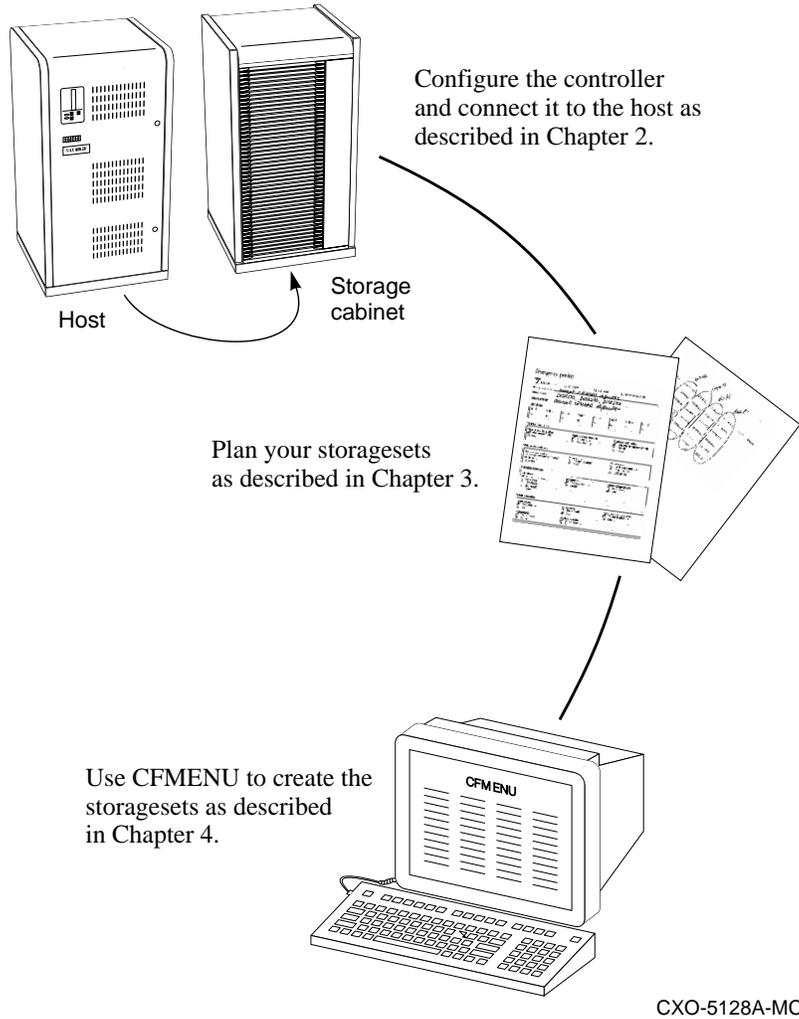
The host port and program-card slot are also located on the front panel, making it easy to update the HSOE software or to connect the controller to a different host.

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

## Key steps for configuring your subsystem

Figure 1-4 shows the key steps you'll follow to set up and configure your subsystem and its controller. Each of these key steps is explained later in this book.

**Figure 1-4 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.

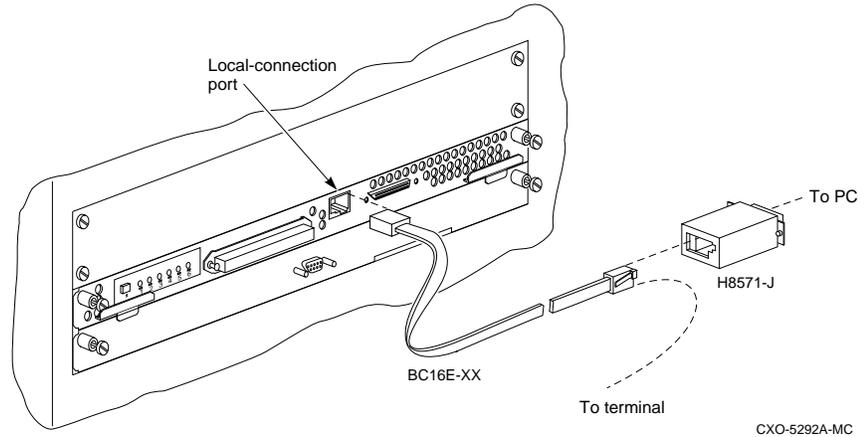
## Communicating with a controller

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 2-1 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 2-1. 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 2-1.

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, 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:

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** controller only if you want to use 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 HSD50 controllers running the same HSOF Version, 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 both controllers service the same storage units, either controller can continue to service all of the 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 2-7.

## Configuring a single controller

To configure a single, nonredundant controller:

1. Establish a local connection to the controller.
2. Declare a DSSI node number for the controller:

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

where *n* is a node number (0-7) that's unique among all devices on the DSSI interconnect.

3. Declare the maximum number of host adapters on the DSSI bus segment:

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

where *n* is the number of host adapters (1-4). 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.

4. 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 "HSD\_7". The node name may contain up to six characters; it must begin with a letter and be enclosed in quotes.

5. 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 a number from one to 255.

6. 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 "HSD>".

7. Disable the controller's DSSI port:

```
CLI> SET THIS_CONTROLLER NOPATH
```

8. Restart the controller:  

```
CLI> RESTART THIS_CONTROLLER
```
9. When the CLI prompt reappears, verify the configuration:  

```
CLI> SHOW THIS_CONTROLLER
```
10. Connect the controller to the host as described on page 2-9.

## Configuring dual-redundant controllers

---

### Note

---

This procedure assumes that one controller has already been setup and installed, as described in the previous section.

---

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. Declare a DSSI node number for each controller:

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

where *n* and *m* are numbers between one and seven that are unique among all devices on the DSSI interconnect.

4. Declare the maximum number of host adapters on the DSSI bus segment:

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

where *n* is the number of host adapters (1-4). 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.

5. Declare an SCS node name for each controller:

```
CLI> SET THIS_CONTROLLER SCS_NODENAME = "x"  
CLI> SET OTHER_CONTROLLER SCS_NODENAME = "y"
```

where “x” and “y” are alphanumeric names that identifies each controller, such as “HSD\_1>” and “HSD\_2>”. 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 a number from one to 255. You only need to set these values for one controller—the other controller automatically inherits the same values.

7. 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 “HSD\_1> ” and “HSD\_2> ”.

8. Disable each controller’s DSSI port:

```
CLI> SET THIS_CONTROLLER NOPATH  
CLI> SET OTHER_CONTROLLER NOPATH
```

9. Restart the controllers:

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

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

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

11. Connect the controller to the host as described on page 2-9.

## 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 configure the controller before connecting it to a host.

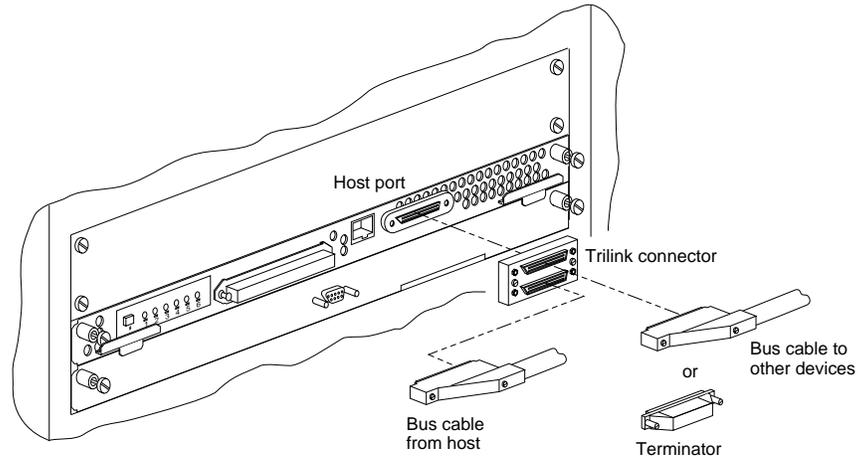
---

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 2-2 Connecting a single controller to its host**



To connect a single, nonredundant controller to the host:

1. Stop all I/O from the host to its devices on the bus to which you're connecting the controller.
2. Remove the trilink connector (12-39921-02) from the controller. This connector is a 68-pin Y-adapter that maintains bus continuity even when it's disconnected from the controller.

3. Connect the DSSI bus cable from the previous device to one of the connectors on the front of the trilink connector as shown in Figure 2-2. Table 2-1 lists the maximum cable lengths.

**Table 2-1 Maximum DSSI cable lengths**

<b>Cable</b>	<b>Meters</b>	<b>Feet</b>
From host to controller	9.14	30
Total length of all cables	18.28	60

4. If the controller is at the end of the host bus, connect a terminator to the other connector on the front of the trilink connector. Otherwise, connect a cable that continues to the next device on the bus. (Be sure to install a terminator at the end of the bus.)
5. Reconnect the trilink connector to the host port on the controller. Don't disconnect the host cables from the trilink connector.
6. Route and tie the cables as desired.
7. Restart the I/O from the host. Some operating systems may require you to reboot the host to see the devices attached to the new controller.
8. Establish a local connection to the controller and enable its host port path:

```
CLI> SET THIS_CONTROLLER_PATH
```

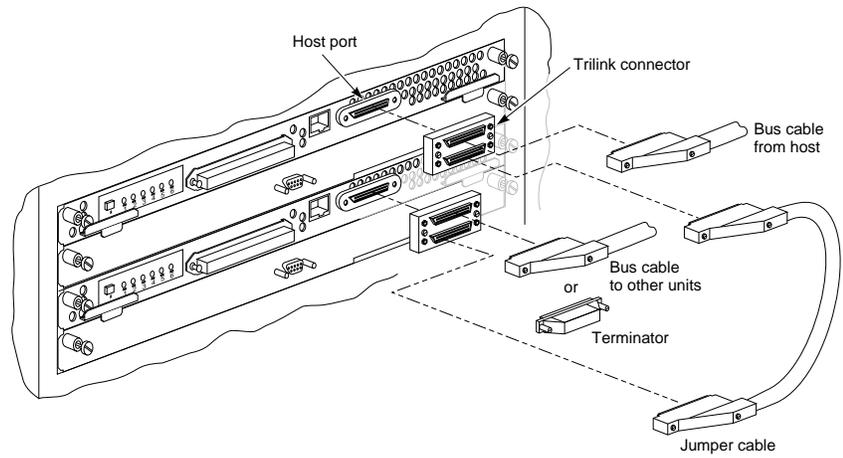
9. Verify the connection by logging onto the cluster and entering the following commands:

```
$ 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

**Figure 2-3 Connecting dual-redundant controllers to the host**



CXO-5301A-MC

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

1. Stop all I/O from the host to its devices on the bus to which you're connecting the controllers.
2. Remove the trilink connectors (12-39921-02) from both controllers. These connectors are 68-pin Y-adapters that maintain bus continuity even when they're disconnected from their controller.
3. Connect the DSSI bus cable from the previous device to one of the connectors on the front of one of the trilink connectors as shown in Figure 2-3. Table 2-1 lists the maximum cable lengths.
4. Connect the two trilink connectors with a jumper cable.
5. If the controllers are at the end of the bus, connect a terminator to the open connector on the front of the trilink connector. Otherwise, connect a cable that continues to the next device on the bus. (Be sure to install a terminator at the end of the bus.)
6. Reconnect the trilink connectors to the host ports on the controllers. Don't disconnect the host cables from the trilink connector.
7. Route and tie the cables as desired.

8. Restart the I/O from the host. Some operating systems may require you to reboot the host to see the devices attached to the new controller.
9. Establish a local connection to one of the controllers and enable the host port paths:

```
CLI> SET THIS_CONTROLLER PATH  
CLI> SET OTHER_CONTROLLER PATH
```

10. 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 storage sets

Creating a profile

Defining your storage requirements

Choosing a storage set

Using stripesets to increase I/O performance

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 storage sets

Assigning unit numbers

Creating a storage set map

## Creating a profile

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

Take a few moments to familiarize yourself with the kinds of information contained in a typical storage set profile, as shown in Figure 3-1.

Figure 3-1 A typical storage set profile

### Storage set profile

---

**Type**  
 RAIDset     Mirrorset     Stripeset     Striped mirrorset

**Storage set name** accept CFMENU default =

**Disk drives** DISK130, DISK230, DISK330

**Unit number** accept CFMENU default =

**Partitions**

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

**RAIDset switches**

<b>Reconstruction policy</b> <input checked="" type="checkbox"/> Normal (default) <input type="checkbox"/> Fast	<b>Reduced membership</b> <input checked="" type="checkbox"/> No (default) <input type="checkbox"/> Yes, missing:	<b>Replacement policy</b> <input checked="" type="checkbox"/> Best performance (default) <input type="checkbox"/> Best fit <input type="checkbox"/> None
---	---	---

**Mirrorset switches**

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

**Initialize switches**

<b>Chunksize</b> <input checked="" type="checkbox"/> Automatic (default) <input type="checkbox"/> 64 blocks <input type="checkbox"/> 128 blocks <input type="checkbox"/> 256 blocks <input type="checkbox"/> Other:	<b>Metadata</b> <input checked="" type="checkbox"/> Destroy (default) <input type="checkbox"/> Retain	<b>Saved configuration</b> <input type="checkbox"/> No (default) <input checked="" type="checkbox"/> Yes
--	---	--

**UNIT switches**

<b>Read cache</b> <input checked="" type="checkbox"/> Yes (default) <input type="checkbox"/> No	<b>Write cache</b> <input type="checkbox"/> No (default) <input checked="" type="checkbox"/> Yes	<b>Maximum cache transfer</b> <input checked="" type="checkbox"/> 32 blocks (default) <input type="checkbox"/> Other:
<b>Availability</b> <input checked="" type="checkbox"/> Run (default) <input type="checkbox"/> NoRun	<b>Write protection</b> <input checked="" type="checkbox"/> No (default) <input type="checkbox"/> Yes	

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

## 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 storage set

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

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

Compare the different kinds of storage sets in Table 3-1 to determine which ones satisfy your requirements.

**Table 3-1 A comparison of different kinds of storage sets**

Storage set	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)	Inversely proportionate to number of disk drives; worse than single disk drive. (More drives = less reliability.)	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—better than RAID 0	Good/Fair	Good/Fair	System drives; critical files.
RAIDset (RAID 3/5)	Excellent—better than RAID 1.	Excellent/Fair	Good/Poor	High request rates, read-intensive, data lookup.
Striped Mirrorset (RAID 0+1)	Excellent—better than RAID 3/5	Excellent if used with large chunk size. Otherwise, good.	Excellent if used with small chunk size.	Any critical response-time application.

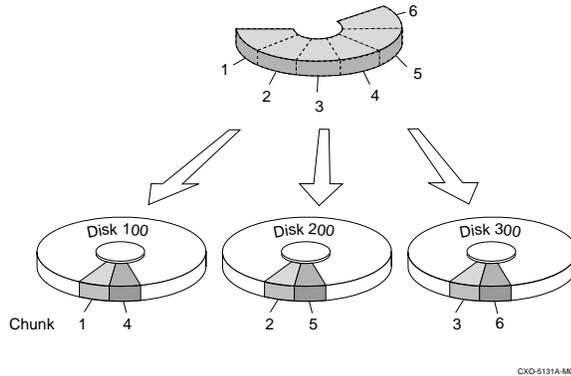
For a comprehensive discussion of RAID, refer to *The RAIDBOOK—A Source Book for Disk Array Technology*, published by the RAID Advisory Board, St. Peter, MN.

## 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 storage set, 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 3-2 Striping lets several disk drives participate in each I/O request**



The relationship between the chunk size and the average request size determines if striping maximizes the request rate or the data-transfer rate. You can set the chunk size or let the controller set it automatically. See *Chunk size* on page 3-19 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 storage set 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 storage set. Thus, if you combine 2 GB disk drives with 1 GB disk drives in the same storage set, you'll waste 1 GB of capacity on each 2 GB member.

---

### Note

---

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

---

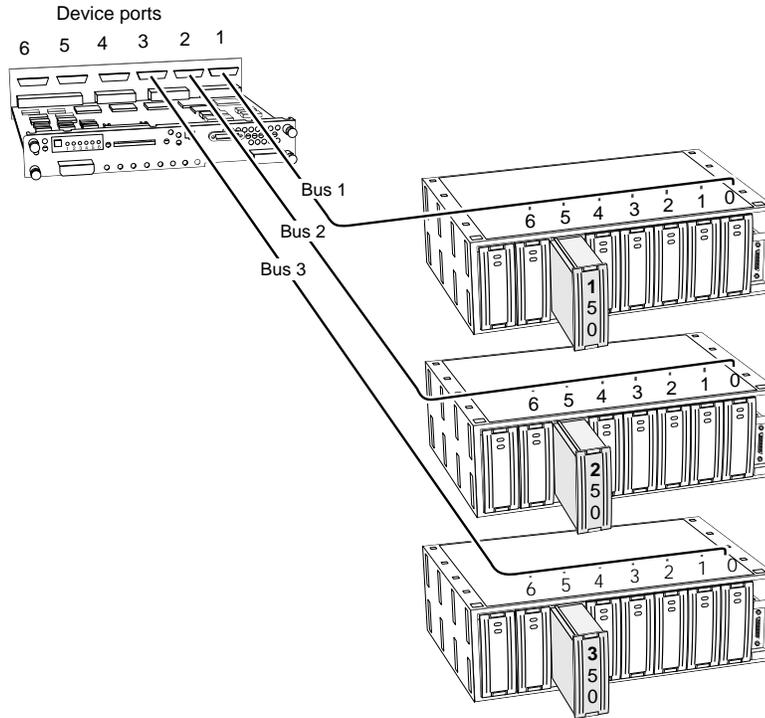
- Striping doesn't protect against data loss. In fact, because the failure of one member is equivalent to the failure of the entire stripeset, the likelihood of losing data is higher for a stripeset than for a single disk drive.

For example, if the mean time between failures (MTBF) for a single disk is  $\lambda$  hours, then the MTBF for a stripeset that comprises  $N$  such disks is  $\lambda/N$  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 3-3.

**Figure 3-3 Distribute members across ports**



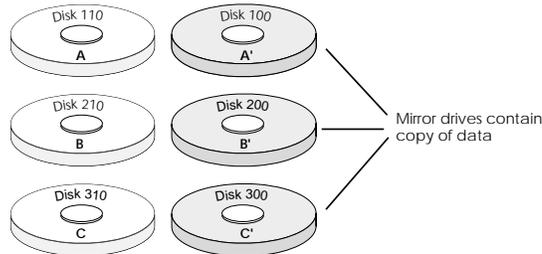
CXO-5133A-MC

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 3-4 Mirrorsets maintain two copies of the same data**



### Considerations for planning a mirrorset

Keep these points in mind as you plan your mirrorsets:

---

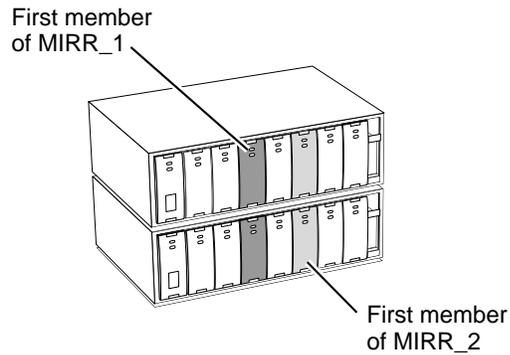
#### Note

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 3-5 Put first members on different busses**

CXO-5315A-MC

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 3-5. Additionally, you can set the read-source switch to Round Robin. See *Read source* on page 3-16 for more information about this switch.

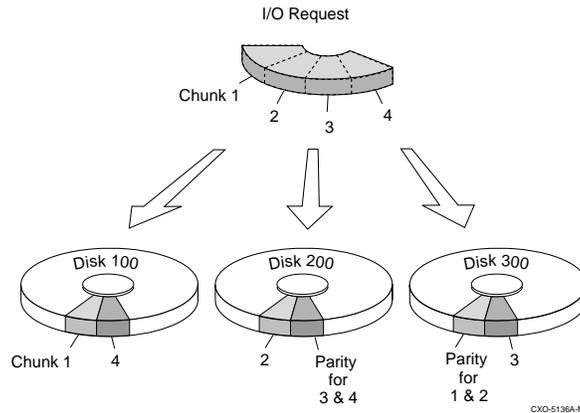
- A storageset should only contain disk drives of the same capacity. The controller limits the capacity of each member to the capacity of the smallest member in the storageset. Thus, if you combine 2 GB disk drives with 1 GB disk drives in the same storageset, you'll waste 1 GB of capacity on each 2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

Evenly distribute the members across the device ports to balance load and provide multiple paths as shown in Figure 3-3.

## Using RAIDsets to increase performance and availability

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

**Figure 3-6 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 fourth 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 3-18 for information about setting the chunk size.

### Considerations for planning a RAIDset

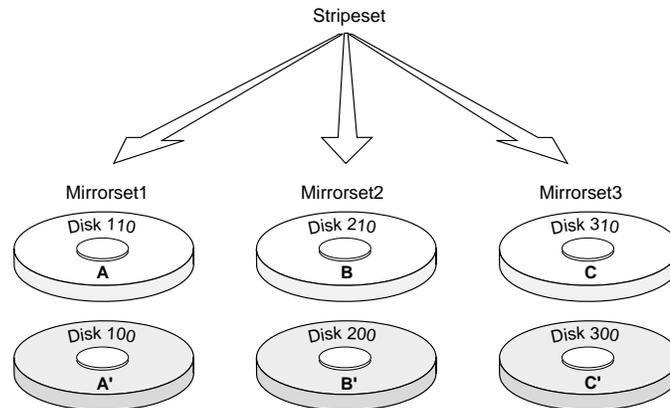
Keep these points in mind as you plan your RAIDsets:

- 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 3-3.
- 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 3-7 Striping and mirroring in the same storageset**



CXO-5317A-MC

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

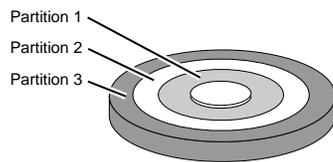
### Considerations for planning a striped mirror set

Plan the mirror set members, then plan the stripe set that will contain them. Follow the considerations for stripe sets and mirror sets provided in this chapter.

## Planning your partitions

Use partitions to divide a storage set or disk drive into smaller pieces, each of which can be presented to the host as its own storage unit. Figure 3-8 shows the conceptual effects of partitioning a single-disk unit.

**Figure 3-8** Each partition can be presented to the host as a storage unit



CXO-5316A-MC

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

### Defining a partition

Partitions are expressed as a percentage of the storage set or single disk unit that contains them. For mirror sets and single disk units, the controller allocates the largest whole number of blocks that are equal to or less than the percentage you specify. For RAID sets and stripe sets, the controller allocates the largest whole number of stripes that are less than or equal to the percentage you specify. For stripe sets, the stripe size = chunk size \* number of members. For RAID sets, 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 administrative metadata. Thus, a single disk unit that contains one partition can store  $n-5$  blocks of user or application data.

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

### **Guidelines for partitioning storagesets and disk drives**

Keep these points in mind as you plan your partitions:

- You can create up to 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 subsystem by using this switch.

## Mirrorset switches

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

- Replacement policy
- Copy speed
- Read source

### Replacement policy

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

`policy=best_performance` (default) puts the failed disk drive in the failedset then tries to find a replacement (from the spareset) that is on a different device port than the remaining, operational disk drives. If more than one

disk drive meets this criterion, this switch selects the drive that also provides the best fit.

- `POLICY=BEST_FIT` puts the failed disk drive in the failedset then tries to find a replacement (from the spareset) that most closely matches the size of the remaining, operational disk drives. If more than one disk drive meets this criterion, this switch selects the one that also provides the best performance.
- `nopolicy` puts the failed disk drive in the failedset and 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=DISKnnn` forces the controller to always read data from a particular "normal" or operational member. If the specified member fails, the controller reads from the least busy member.

## Device switches

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

- Transportability
- Transfer rate

### Transportability

---

**Note**

---

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

---

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

- NOTTRANSPORTABLE disk drives (default) are marked with StorageWorks-exclusive metadata. This metadata supports the error-detection and recovery methods that the controller uses to ensure data availability. Disk drives that contain this metadata 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

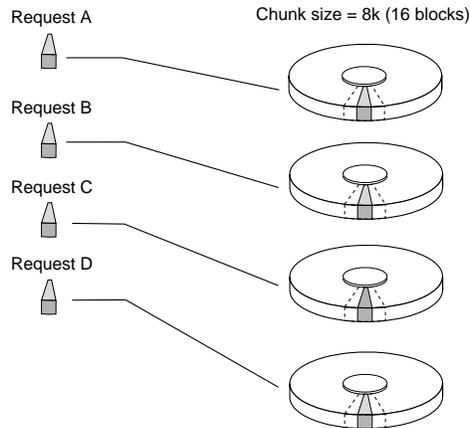
Specify a chunk size to control the stripesize used for RAIDsets and stripesets:

- `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 3-9** If chunk size is larger than the request size, then each disk drive in the storageset can respond to a separate I/O request.



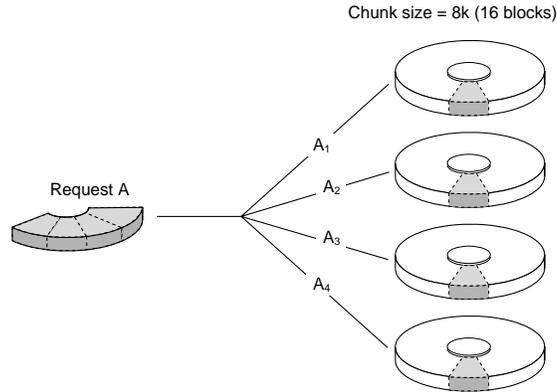
Applications such as interactive transaction processing, office automation, and file services for general timesharing tend to require high I/O request rates.

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 3-10** If 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 3-2** 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 subsystem's configuration on the storage unit when you initialize it:

- **NOSAVE\_CONFIGURATION** (default) means that the controller stores the subsystem's configuration in its nonvolatile memory. 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** only requires one disk be initialized with this option, but more may be used, if desired. **DIGITAL** *does not* recommend initializing *all* disks with the **SAVE\_CONFIGURATION** switch, because every update to non-volatile memory causes writes to all disks and can affect performance adversely.
- Specify **SAVE\_CONFIGURATION** when initializing any disk device or container on which you want to store a copy of the controller configuration. If you use the switch for a multi-device storageset, such as a stripeset, the complete information is stored on each device in the storageset.
- **DIGITAL** recommends that the **SAVE\_CONFIGURATION** switch only be used for single controller configurations. (Use the **SET FAILOVER COPY=** command to save configuration information for dual-redundant configurations).
- **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 when you change the configuration 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.
- Specify **NOSAVE\_CONFIGURATION** for devices and storagesets on which you do not want to store a copy of the controller configuration.
- **SAVE\_CONFIGURATION** is not available for upgrades of firmware or hardware, and will not perform inter-platform conversions.

## 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.

---

### Note

---

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

---

- NODESTROY preserves the user data and forced-error metadata when a disk drive is initialized. Use NODESTROY to create a single-disk unit from any disk drive that has been used as a member of a mirrorset. See the *Reduced* command in the *CLI Reference Manual* for information on removing disk drives from a mirrorset.

## Unit switches

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

**Table 3-3 Unit switches**

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. Use the `SHOW unit-name` command to check whether the cache is flushed

---

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 issue a `SET unit-number RUN` command.

## 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.

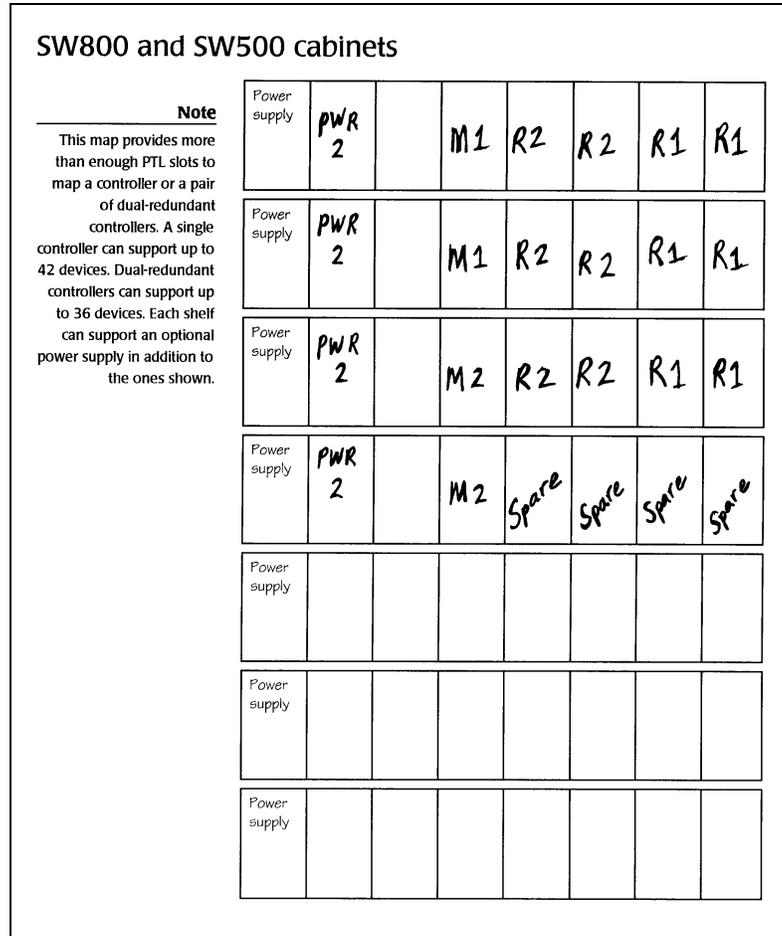
Each four-place unit number contains the following:

- 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 storage set map

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

**Figure 3-11 A storage set 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).

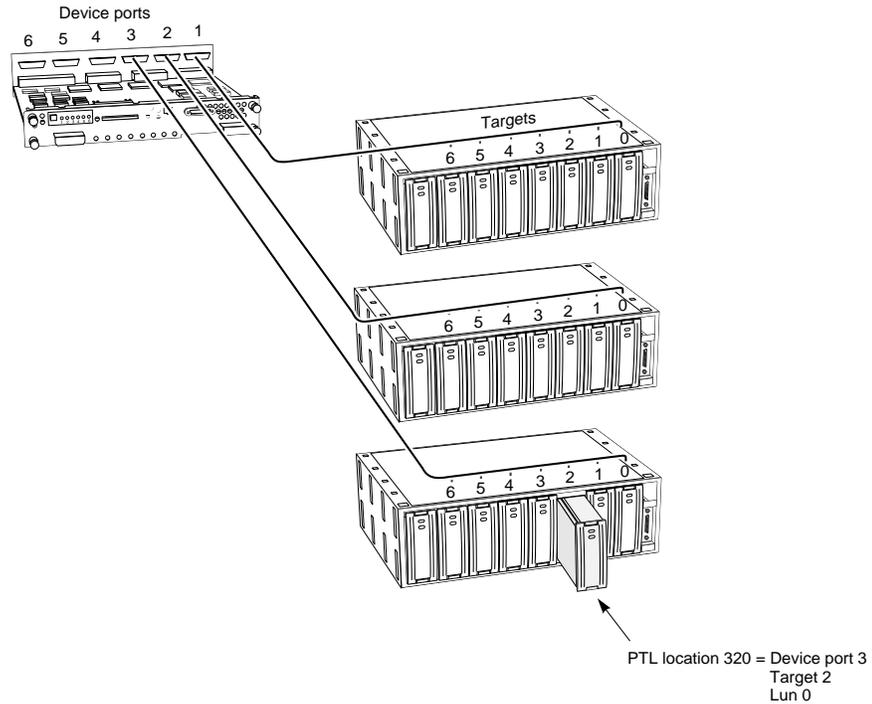
5. Repeat steps 2 through 4 for each controller or dual-redundant pair of controllers.
6. After you have mapped the devices to your cabinet template, create the storageset map by circling each group of disk drives that you want to combine into a storageset or put into the spareset. Label each group with 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 3-12, 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 3-12 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 storage set name. From the storage set 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 subsystem before you begin the procedures provided in Chapter 5.



# 4

## Automatically configuring storagesets

- Introducing CFMENU
- Considerations for using CFMENU
  - Adding devices with CFMENU
  - Deleting devices with CFMENU
- Creating a storageset with CFMENU
- Deleting a storageset with CFMENU
- Adding a disk drive to a spareset with CFMENU
  - Initializing containers with CFMENU
    - Adding a unit with CFMENU
  - Deleting a disk drive from a spareset with CFMENU
- Partitioning a storageset with CFMENU
  - Adding a passthrough with CFMENU
  - Deleting a passthrough with CFMENU
  - Adding a partitioned unit with CFMENU
  - Deleting a Partitioned unit with 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 from a menu of configuration tasks, such as adding a storageset or assigning a unit number.

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

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 4-1 lists the heading and contents for each column on the Main Menu.

**Table 4-1 Interpreting CFMENU Columns**

Column heading	Information Displayed
Main menu	Shows the tasks you can accomplish with CFMENU.
Unconfig'd Dev.PTLs	Shows the PTL locations of devices that have not yet been added to the controller's configuration. Use these devices to create single-disk units and storagesets, such as stripesets and RAIDsets.
Config'd PTLs	Shows the PTL locations of all devices that are used in—or are eligible to be used in—a storageset or a single-disk unit.
Device Name	Shows the names of all devices that are used in—or are eligible to be used in—a storageset or as a single-disk unit.
Product ID	Shows the model numbers of all devices that are used in—or are eligible to be used in—a storageset or as a single-disk unit.
Stor.set Name	Shows the name of all storagesets in the controller's list of configured storagesets: by convention, Sn for stripesets, Mn for mirrorsets, and Rn for RAIDsets.

<b>Column heading</b>	<b>Information Displayed</b>
Stor.set Typ/Sz	Shows the types of storage sets 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 storage set.
Trnsp.	Displays "Y" if you enabled the Transportable switch.
Init'd	Displays "Y" if you initialized the storage set.
Reduc	Displays "Y" if you initialized the storage set with the Reduced switch or if the storage set is in a reduced state due to the failure of one of its members.
Unit	Shows the unit numbers for all storage sets or devices.
PT	Displays "P" if the unit is partitioned.
WP	Displays "Y" if the unit is write protected.
WB	Displays "Y" if you enabled write-back cache.

## Considerations for using CFMENU

Keep the following points in mind for using CFMENU:

- Configure your storagesets manually if you want to use your own naming scheme. CFMENU names each storageset according to a simple naming convention: *Mn* for mirrorsets, *Sn* for stripesets, and *Rn* for RAIDsets, where *n* is a sequentially indexed number. CFMENU also automatically provides unit-number prefixes; you specify the actual number.
- You can create and delete storagesets with CFMENU, however, you cannot modify them once they've been created. Follow the steps in *Changing switches for a storageset or device* in Chapter 5.
- If you are 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 cannot configure tape loaders. See *Configuring a tape drive* and *Configuring a tape loader* in Chapter 5.

## Adding devices with CFMENU

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

1. Install the new disk drives in your storage cabinet.
2. Start CFMENU:  

```
CLI> RUN CFMENU
```
3. From the Main Menu, choose task 1 to go to the Device Menu.
4. From the Device Menu, choose task 1 to add disk drives.
5. CFMENU presents—one at a time—the disk drives or devices that you may add to the subsystem. Type Y to add the disk drive, N to skip to the next one.
6. Set the disk drive NOTTRANSPORTABLE. See *Initialize switches* in Chapter 3 for more information about this switch.
7. Return to the Main Menu and exit CFMENU.

## Deleting Devices with CFMENU

To delete a disk drive or other storage device from your subsystem with CFMENU:

1. Start CFMENU:  
`CLI> RUN CFMENU`
2. From the Main Menu, choose task 1 to go to the Device Menu.
3. From the Device Menu, choose task 2 to delete devices.
4. CFMENU presents—one at a time—the disk drives or devices that you may delete from the subsystem. Type Y to delete the disk drive, N to skip to the next one.
5. Return to the Main Menu and exit CFMENU.

## Creating a Storageset with CFMENU

Creating a storageset or single-disk unit with CFMENU is as easy as choosing menu options and responding to prompts. Just remember to move through the Main Menu items from top to bottom for each storageset or single-disk unit you want to create.

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

---

**Note**

---

Press “D” to scroll down CFMENU’s columns.  
Press “U” to scroll up.

---

1. Start CFMENU:  
`CLI> RUN CFMENU`
2. Go to step 9 if you’re configuring a single-disk unit, otherwise choose task 2, 3, or 4 depending on the kind of storageset you want to create.  
  
CFMENU displays a storageset menu that corresponds to your choice: Mirrorset Menu, Stripeset Menu, or RAIDset/sparesets/failedsets Menu.
3. From the storageset menu, choose task 1 to add a storageset to the controller’s list of available storagesets.
4. Enter the number of disk drives or members that you want to include in the storageset.

5. CFMENU presents—one at a time—the disk drives or members that you may include in the storageset. Type Y to include a member, N to skip to the next one.
6. When you reach number of members specified in step 5, CFMENU prompts you for the switches you can apply to the storageset. Indicate your choice or press Return to accept the default value.
7. CFMENU displays a message that indicates the storageset's type and name, as well as the names of all its members. Press Return to create the storageset.
8. Return to the Main Menu and repeat steps 2 through 8 for each storageset or single-disk unit you want to create.
9. From the Main Menu, choose task 6 to go to the Initialization Menu.
10. From the Initialization Menu, choose task 1 to initialize the storageset (or the disk drive if you're creating a single-disk unit).
11. CFMENU prompts you for the Initialize switches you can apply to the storageset or single-disk unit. Indicate your choice or press Return to accept the default value. See *Initialize switches* in Chapter 3 for more information about these switches.
12. Repeat steps 10 and 11 for each storageset or single-disk unit you want to initialize.
13. Return to the Main Menu.
14. From the Main Menu, choose task 7 to go to the Unit Menu.
15. From the Unit Menu, choose task 1 to assign a host-addressable unit number to the storageset or single-disk unit. See *Assigning unit numbers* in Chapter 3 for more information about choosing unit numbers.
16. CFMENU prompts you for the Unit switches you can apply to the unit. Indicate your choice or press Return to accept the default value. See *Unit switches* in Chapter 3 for more information about these switches.
17. Repeat steps 15 and 16 for each storageset or single-disk unit to which you want to assign a unit number.
18. Return to the Main Menu and exit CFMENU.

## Deleting a Storageset with CFMENU

To delete a storageset with CFMENU:

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

## Adding a Disk Drive to a Spareset with CFMENU

To add a disk drive to a spareset:

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

## Initializing Containers with CFMENU

To initialize devices or storagesets:

1. Start CFMENU:  

```
CLI> RUN CFMENU
```
2. From the Main Menu, choose task 6 to go to the Initialization Menu.
3. From the Initialization Menu, choose task 1.
4. CFMENU presents—one at a time—the devices or storagesets that you may initialize. Type Y to add the disk drive, N to skip to the next one.
5. In addition, CFMENU will prompt you to decide on other operating qualifiers, depending on whether the container is a device, mirrorset, stripeset, or RAIDset. (Refer to the ADD or SET commands in the *HSZ40 Array Controllers CLI Reference Manual* if you need help understanding the qualifiers.)
6. Return to the Main Menu and exit.

## Adding Units with CFMENU

To add units that are not partitioned:

1. Start CFMENU:  

```
CLI> RUN CFMENU
```
2. From the Main Menu, choose task 7 to go to the Unit Menu.
3. From the Unit menu, choose task 1 to add a unit.
4. CFMENU presents—one at a time—for the containers from which the new units will be created. Type Y to add the disk drive, N to skip to the next one.
5. CFMENU also prompts you to assign a unit number. (The program automatically assigns a “D” or “T” to the unit number when listing the unit.)
6. In addition, CFMENU prompts you to decide on other unit qualifiers. See the description of the ADD *unit* or SET *unit* commands in the *HSZ40 Array Controllers CLI Reference Manual*.
7. Return to the Main Menu and exit.

## Deleting a Disk Drive from a Spareset with CFMENU

To delete a disk drive from a the spareset:

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

---

### Note

---

CFMENU only allows you to delete one disk from a spareset at a time.

---

7. Return to the Main Menu and exit.

## Partitioning a Container with CFMENU

To partition a storageset or single-disk container:

1. Start CFMENU:
2. **CLI> RUN CFMENU**
3. From the Main Menu, choose task 5 and answer yes to processing the partition if there is a partitioned container to go to the Partition Processing menu.
4. CFMENU presents—one at a time—the storageset or single disk containers that you may partition. Type Y to select the container, N to skip to the next one.
5. From the Partition Menu, choose task 1 to partition the selected container.
6. Indicate the percentage of the container that you want to dedicate to the first partition.

7. Repeat step 6 for each partition you want to create on the container.
8. Return to the Main Menu and exit.

## Adding a Passthrough with CFMENU

You can use this CFMENU selection to add any generic SCSI device, but you will normally use it to add devices requiring the use of MSCP read and write operations as a transport for SCSI commands (such as “jukebox” style loaders).

To add a passthrough device to your subsystem with CFMENU:

1. Install the new devices in your storage cabinet.
2. Start CFMENU:  

```
CLI> RUN CFMENU
```
3. From the Main Menu, choose task 6 to go to the Passthrough Menu.
4. From the Passthrough Menu, choose task 1 to add passthrough devices.
5. CFMENU presents—one at a time—the devices that you may add to the subsystem. Type Y to add the device, N to skip to the next one.
6. Return to the Main Menu and exit CFMENU.

## Deleting a Passthrough with CFMENU

To delete a SCSI device from your subsystem with CFMENU:

1. Start CFMENU:  

```
CLI> RUN CFMENU
```
2. From the Main Menu, choose task 6 to go to the Passthrough Menu.
3. From the Passthrough Menu, choose task 2 to delete passthrough devices.
4. CFMENU presents—one at a time—the devices that you may delete from the subsystem. Type Y to delete the device, N to skip to the next one.  
  
Return to the Main Menu and exit CFMENU.

## Adding a Partitioned Unit with CFMENU

To add a partitioned unit to your subsystem with CFMENU:

1. Start CFMENU:  
`CLI> RUN CFMENU`
2. From the Main Menu, choose task 5 to go to the Partition Menu.
3. From the Partition Menu, choose task 2 to add partitioned units.
4. CFMENU presents—one at a time—the devices that you may add to the subsystem. Type Y to add the device, N to skip to the next one.
5. Return to the Main Menu and exit CFMENU.

## Deleting a Partitioned Unit with CFMENU

To delete a partitioned unit from your subsystem with CFMENU:

1. Start CFMENU:  
`CLI> RUN CFMENU`
2. From the Main Menu, choose task 5 to go to the Partition Menu.
3. From the Partition Menu, choose task 3 to delete the partitioned unit.
4. CFMENU presents—one at a time—the devices that you may delete from the subsystem. Type Y to add the device, N to skip to the next one.
5. Return to the Main Menu and exit CFMENU.



# 5

## Manually configuring storagesets

- Adding disk drives
- Adding CD-ROM 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 subsystem 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
```

## Adding CD-ROM drives

To add new CD-ROM drives to your subsystem, issue the following CLI commands:

### Adding one CD-ROM drive at a time

To add one new CD-ROM drive to your controller's list of eligible devices:

```
CLI> ADD CDROM CDROM-name SCSI-location
```

The *SCSI-location* parameter assigns a PTL address to the CD-ROM that is used by the controller. Refer to the CLI Manual for an explanation of the PTL numbering system.

### Adding several CD-ROM drives at a time

To add several new CD-ROM 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 5-1 Initialize switches**

Initialize switch	Value and syntax
Chunk size	CHUNKSIZE=DEFAULT* CHUNKSIZE= <i>n</i>
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 5-2 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

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 5-3 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 5-4 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
```

- 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 5-5 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

- Verify the mirrorset configuration and switches:

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

- 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 5-6 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 5-7 Initialize switches**

Initialize switch	Value and syntax
Chunk size	CHUNKSIZE=DEFAULT* CHUNKSIZE= <i>n</i>
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
```

- 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 5-8 Unit switches**

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

- Verify the RAIDset configuration and switches:

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

- 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 5-9 Initialize switches**

Initialize switch	Value and syntax
Chunk size	CHUNKSIZE=DEFAULT* CHUNKSIZE= <i>n</i>
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 5-10 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 5-1.
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 5-11 Device switches**

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

---

**Note**

---

Refer to Table 2-1 to determine the correct transfer rate for the device you are adding.

---

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 5-12 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 subsystem.

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

## Partitioning a storage set or disk drive

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

1. Add the storage set or disk drive to the controller's list of storage sets and specify the disk drives it contains:

```
CLI> ADD STORAGESET storage-set-name DISKnnn DISKnnn
```

or

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

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

```
CLI> INITIALIZE storage-set-name
```

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

```
CLI> CREATE_PARTITION storage-set-name SIZE=n
```

where *n* is the percentage of the disk drive or storage set 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 storage-set-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 storage-set-name  
PARTITION=partition-number
```

6. Verify the unit numbers for the partitions:

```
CLI> SHOW storage-set-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 5-13 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 DISK200 DISK300	D100

```
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.

---

### 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.

---

- 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.

# 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 storage set 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 7-1 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. Wait until the green light stops blinking and then 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. The system should reboot. If it does not automatically reboot, press and hold the reset button on the controller for three seconds, then release it.
3. 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
```

---

### 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.

---

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
```



# **A**

## **Profiles and Templates**

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

### 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	

## Device profile

---

### Type

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

**Device name** \_\_\_\_\_

**Unit number** \_\_\_\_\_

### Device switches

#### Transportability

- No (default)  
 Yes

### Initialize switches

#### 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 an 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, G-2
- Adding
  - disk drives to spareset, 5-16
- Adding a reduced RAIDset, 3-15
- Adding CD-ROM drives to configuration, 5-2
- Adding disk drives to configuration, 5-2
- Adding disk drives with CFMENU, 4-5
- Adding Passthroughs with CFMENU, 4-11
- Addressing
  - PTL convention, 3-27
- Allocation class, G-2
- Allocation classes, 2-6, 2-8
- Array controller, G-2
- Automatic naming convention, 4-4

## B

---

- Backup, 7-2
- BBR, G-2
- BBU, G-2
- Block, G-2
- Bus
  - distributing members across, 3-7, 3-9, 3-11

## C

---

- Cache sizes supported, 1-4

- CDU, G-2
- CFMENU, 4-2
  - adding devices, 4-5, 4-11
  - configuring storagesets, 4-6
  - considerations for using, 4-3
  - Deleting devices, 4-6, 4-11
  - initializing a storageset, 4-7
  - partitioning, 4-10
  - running on dual-redundant controllers, 4-5
- Changing switches, 3-14
- Chunk size
  - choosing for RAIDsets and stripesets, 3-18
  - displaying with CFMENU, 4-3
  - maximum for RAIDsets, 3-20
  - using to increase request rate, 3-18
  - using to increase transfer rate, 3-19
- CI bus, G-2
- CI cables
  - maximum cable lengths, 2-10, 2-11
- CLI, G-3
  - accessing from a VCS, 2-5
- Cloning data for backup, 7-2
- Command disk, 5-13
- Comparison of storagesets, 3-3
- Configuration
  - dual-redundant controllers and CFMENU, 4-5
  - key steps, 1-5

- saving, 3–21
- Configured PTLs
  - displaying with CFMENU, 4–3
- Configuring
  - partitions, 5–14
- Configuring a tape drive, 5–12
- Configuring a tape loader, 5–13
- Configuring CD-ROM drives, 5–2
- Configuring disk drives, 5–2
- Configuring storagesets with CFMENU, 4–6
- Control panel, 1–5
- Controller
  - components, 1–5
  - features and general operation, 1–2
- Controller shelf, G–3
- Copy speed
  - setting for mirrorsets, 3–16
- CRC, G–3
- Creating
  - partitions, 5–14
  - storageset profile, 3–2

## D

---

- Data center cabinet, G–3
- DDL, G–3
- Default device name, 3–28
- Defining your storage requirements, 3–3
- Deleting disk drives with CFMENU, 4–6
- Deleting Passthroughs with CFMENU, 4–11
- Deleting storagesets, 5–17
- Device name
  - displaying with CFMENU, 4–3
- Device ports, 1–5
- Device protocol, 1–4
- Devices
  - adding with CFMENU, 4–5, 4–11
  - assigning unit numbers to, 3–25
  - changing switches, 5–18
  - Deleting with CFMENU, 4–6, 4–11
  - switches, 3–17
  - switches for, 3–13
  - transfer rate, 3–17
- Differential SCSI bus, G–3
- DILX, G–3
- Disk drive capacity limitations, 6–2
- Disk drives

- adding to spareset, 5–16
- adding with CFMENU, 4–5
- configuring, 5–2
- Deleting with CFMENU, 4–6
- manually configuring a single-disk drive unit, 5–11
- removing from spareset, 5–17
- Displaying switches, 5–18
- Distributing
  - first members of multiple mirrorsets, 3–8
  - members of storageset, 3–7, 3–9, 3–11
- DSA, G–3
- DSSI, G–4
- Dual-redundant configuration, G–4
- Dual-redundant controllers
  - using CFMENU, 4–5
- DUART, G–4
- DUP, G–4
- DWZZA, G–4

## E

---

- ECB, G–4
- ECC, G–4
- EDC, G–4
- Enabling switches, 3–13
- Erasing metadata, 3–21
- ESD, G–4
- Examples
  - configuring a spareset, 5–16
  - configuring a tape drive, 5–13
  - configuring a tape loader, 5–13
  - manually configuring loaders, 5–13
  - manually configuring mirrorsets, 5–6
  - manually configuring RAIDsets, 5–8
  - manually configuring single-disk unit, 5–12
  - manually configuring striped mirrorsets, 5–10
  - manually configuring stripesets, 5–4
  - manually partitioning a storage unit, 5–15

## F

---

- Failedset, G–5
- Failover, G–5
- Flush, G–5
- Format for tapes, 3–25
- FRU, G–5

FWD SCSI, G-5

## G

Guidelines for partitioning storage sets and disk drives, 3-13

## H

Half-height device, G-5  
 HBVS, G-5  
 HIS, G-5  
 Host, G-6  
 Host port, 1-5  
 Host protocol, 1-4  
 Host-based shadow sets, 6-4  
 Hot swap, G-6  
 HSO, G-5

## I

I/O request routing, 3-28  
 Initialize switches, 3-18  
 Initialized devices  
   displaying with CFMENU, 4-3  
 Initializing a storage set, 4-7  
 Initiator, G-6  
 Interconnect supported, 1-4

## J

JBOD, 3-4

## L

Largest device supported, 1-4  
 Loader  
   example of configuring manually, 5-13  
 Local connection jack, 1-5  
 Local terminal, G-6  
 LOG qualifier, 2-4  
 Logical unit, G-6  
 LRU, G-6  
 LUN, G-6

## M

Maintenance terminal, G-6  
 Manually configuring  
   mirror sets, 5-5

RAID set, 5-7  
 single-disk drive unit, 5-11  
 striped mirror sets, 5-9  
 stripe sets, 5-3

Mapping your storage sets, 3-26

Mean time between failures  
   calculating, 3-6

Members

  distributing first members of mirror sets, 3-8  
   distributing on bus, 3-7, 3-9, 3-11  
   maximum for RAID set, 3-11

Metadata, G-7

  erasing or retaining, 3-21

Mirror set

  configuring manually, 5-5  
   configuring with CFMENU, 4-6  
   considerations for planning, 3-8  
   defined, 3-8  
   distributing first members, 3-8  
   example of configuring manually, 5-6  
   setting copy speed, 3-16  
   setting read source, 3-16  
   switches, 3-15

Mirror sets

  changing switches, 5-19

MSCP, 2-6, 2-8, G-7

MTBF

  calculating, 3-6

## N

Non-redundant configuration, G-7

Normal member, G-7

NV, G-7

## O

OpenVMS command interactions, 6-3

OpenVMS disk capacity limitations, 6-2

Options

  for devices, 3-17  
   for mirror sets, 3-15  
   for RAID sets, 3-14  
   for storage units, 3-23  
   initialize, 3-18

Options for storage sets and devices, 3-13

**P**

- 
- Partitioning a storageset or disk drive, 5–14
  - Partitioning a storageset with CFMENU, 4–10
  - Partitions
    - displaying with CFMENU, 4–3
    - example of configuring manually, 5–15
    - planning, 3–12
  - Passthroughs
    - adding with CFMENU, 4–11
    - deleting with CFMENU, 4–11
  - Path
    - preferring for storage units, 3–23
  - Planning
    - mirrorset, 3–8
    - partitions, 3–12
    - RAIDset, 3–10
    - striped mirrorset, 3–11
    - stripeset, 3–4
  - Port, G–7
  - Preferred paths for storage units, 3–23
  - Product ID
    - displaying with CFMENU, 4–3
  - Profile
    - creating for storagesets and devices, 3–2
  - Program card slot, 1–5
  - Protocol
    - devices, 1–4
    - host, 1–4
  - PTL addressing convention, 3–27

**Q**

- 
- Qualified device, G–8
  - Quiesce, G–8

**R**

- 
- RAID levels supported, 1–4
  - RAIDset
    - choosing a chunk size for, 3–18
    - configuring with CFMENU, 4–6
    - considerations for planning, 3–11
    - defined, 3–10
    - example of configuring manually, 5–8
    - manually configuring, 5–7
    - maximum chunk size for, 3–20
    - maximum membership, 3–11

- setting reconstruction policy, 3–15
- setting reduced membership, 3–15
- setting replacement policy, 3–14, 3–15
- switches, 3–14

**RAIDsets**

- changing switches, 5–19

**Read cache, 3–24, G–8****Read source**

- setting for mirrorsets, 3–16

**Reconstruction policy**

- setting for RAIDsets, 3–15

**Reduced membership**

- displaying with CFMENU, 4–3
- setting for RAIDsets, 3–15

**Removing**

- disk drives from spareset, 5–17

**Removing storagesets, 5–17****Replacement policy**

- setting for RAIDsets, 3–14, 3–15

**Request rate**

- how chunk size affects, 3–18

**Reset button, 1–5****Restarting your subsystem, 7–5****S****Saving your configuration, 3–21****SBB, G–8****SCS, G–9****SCS node names, 2–8****SCSI, G–9****Shadow sets, 6–4****Shutting down your subsystem, 7–4****Single-disk unit**

- example of configuring manually, 5–12

**Spareset**

- example of configuring, 5–16

**Sparesets, 5–16**

- adding disk drives, 5–16

- removing disk drives, 5–17

**SPD, G–10****Starting your subsystem, 7–5****Storage requirements**

- defining, 3–3

**Storage unit, G–10****Storageset**

- changing switches, 5–18

- initializing with CFMENU, 4–7
- partitioning with CFMENU, 4–10
- switches for, 3–13
- Storageset map, 3–26
- Storageset name
  - displaying with CFMENU, 4–3
- Storageset profile
  - creating, 3–2
- Storageset type
  - displaying with CFMENU, 4–3
- Storagesets
  - assigning unit numbers, 3–25
  - configuring with CFMENU, 4–6
  - deleting, 5–17
  - displaying switches for, 5–18
  - overview, 3–3
  - setting preferred path, 3–23
- Striped mirrorset
  - configuring with CFMENU, 4–6
  - considerations for planning, 3–12
  - defined, 3–11
  - example of configuring manually, 5–10
  - manually configuring, 5–9
- Stripeset
  - choosing a chunk size for, 3–18
  - configuring manually, 5–3
  - configuring with CFMENU, 4–6
  - considerations for planning, 3–6
  - defined, 3–4
  - distributing members, 3–7, 3–9, 3–11
  - example of configuring manually, 5–4
- Switches
  - changing, 3–14, 5–18
  - changing device switches, 5–19
  - changing Initialize switches, 5–19
  - changing RAIDset and mirrorset switches, 5–19
  - changing unit switches, 5–19
  - enabling, 3–13
  - for devices, 3–17
  - for mirrorsets, 3–15
  - for RAIDsets, 3–14
  - for storage units, 3–23
  - initialize, 3–18
- Switches for storagesets and devices, 3–13

---

**T**


---

- Tagged command queuing, G–10
- Tape drives
  - configuring, 5–12
  - example of configuring, 5–13
- Tape formats, 3–25
- Tape loaders
  - configuring, 5–13
  - example of configuring, 5–13
- Target, G–10
- TILX, G–11
- TMSCP, 2–6, 2–8, G–11
- Transfer rate, 3–17
  - how chunk size affects, 3–19
- Transportability, 3–17
- Transportability
  - displaying with CFMENU, 4–3

---

**U**


---

- Unconfigured Device PTLs
  - displaying with CFMENU, 4–3
- Unit, G–11
- Unit number
  - displaying with CFMENU, 4–3
- Unit numbers
  - assigning to storagesets, 3–25
- Unit switches, 3–23
  - availability, 3–24
  - maximum cache transfer, 3–24
  - read cache, 3–24
  - tape format, 3–25
  - write protection, 3–25
  - write-back cache, 3–24

---

**V**


---

- VAXcluster console system. See VCS*
- VCS, 2–5, G–11
- Virtual terminal, G–11

---

**W**


---

- Warm swap, G–11
- Write hole, G–12
- Write protection switch
  - displaying with CFMENU, 4–3

Write-back cache switch  
displaying with CFMENU, 4-3

Write-through cache, G-12