SCF Reference Manual for the Storage Subsystem

HP Part Number: 529937-022
Published: February 2014
Edition: J06.03 and subsequent J-series RVUs, H06.05 and subsequent H-series RVUs, G06.28 and subsequent G-series RVUs.

Contents

Tables

About This Document

This manual describes how to use the Subsystem Control Facility (SCF) to configure, control, and inquire about storage subsystem objects on HP Integrity NonStop™ NS-series servers and HP NonStop S-series servers. This manual is for anyone who configures new systems, changes or adds to existing system configurations, plans changes to systems, monitors the status of the storage subsystem, or operates a network of distributed systems. These devices are supported:

- Physical disks and virtual disks
- Tape drives
- Storage Management Foundation (SMF) objects
- Open SCSI and Open SCSI Fibre Channel objects
- ServerNet adapters
- CLuster I/O modules (CLIMs)

Supported Release Version Updates (RVUs)

This publication supports J06.03, H06.05, and G06.28.

Intended Audience

Anyone who is responsible for:

- Configuring new systems
- Changing or adding to existing system configurations
- Planning changes to systems
- Monitoring the status of the storage subsystem
- Operating a network of distributed systems

New and Changed Information

New and Changed Information in the February 2014 Edition

Chapter	Change
Chapter 14: "Storage Subsystem Commands" (page 190)	Updated syntax for "STATS DISK Command" (page 277)
Appendix A: "Storage Subsystem Error Messages" (page 298)	Added the error messages 00145, 00146, and 00147 in "Storage Subsystem Error Messages" (page 298)
Chapter 6: "Configuring Disks" (page 70)	Added a new consideration to "Considerations for WRITECACHE" (page 94).
Chapter 6: "Configuring Disks" (page 70)	Updated information under Section : "Configuring Disk Cache" (page 87)
Chapter 14: "Storage Subsystem Commands" (page 190)	Added a note to AUDITTRAILBUFFER description under "Attribute Descriptions for Disk Commands" (page 198).

New and Changed Information in the August 2013 Edition

Table 1

Chapter	Change
Chapter 5: “Displaying Information About Disk Drives” (page 48)	• Added statement that the FastBulkWrite attribute affects all 512-byte-sector disks to “Displaying Disk Configuration Information” (page 50) and “Displaying Profile Attributes” (page 54) • Updated sample display and explanation of fields in “Displaying Disk Configuration Information” (page 50), “Displaying Disk Cache Configuration Information” (page 59), and “Displaying Disk Cache Statistics” (page 60).
Chapter 6: “Configuring Disks” (page 70)	Under WRITECACHE, deleted unneeded phrase, “Make sure this chain of mirrored disk partitions involves only a few disks.” Added a note to “Types of Disks” (page 70) stating that the terms “disk” and “physical disk” include both magnetic and solid state disks. Added a consideration to “Considerations for WRITECACHE” (page 94) indicating that altering WRITECACHE on a busy system might cause disk aborts on some systems. Updated sample display in “Configuring Disk Cache” (page 87).
Chapter 12: “Configuring and Managing Open SCSI Devices” (page 170)	Changed “Examples of Detailed INFO SCSI Reports” (page 171) to read RECSIZE in “bytes” instead of “number of words.”
Chapter 14: “Storage Subsystem Commands” (page 190)	Under descriptions of WRITECACHE, deleted unneeded phrase, “Make sure this chain of mirrored disk partitions involves only a few disks.” Changed “ADD SCSI Attributes” (page 223) to read RECSIZE in “bytes” instead of “number of words.” Changed “ADD TAPE Attributes” (page 226) to read RECSIZE in “bytes” instead of “number of words.” Changed “ALTER SCSI Attributes” (page 238) to read RECSIZE in “bytes” instead of “number of words.” Changed “ALTER TAPE Attributes” (page 242) to read RECSIZE in “bytes” instead of “number of words.” Changed “Displaying Configuration Information” (page 180) to read “in bytes” instead of “in number of bytes.” Changed “Examples of INFO TAPE Detailed Reports” (page 181) to read “in bytes” instead of “in number of bytes.” Changed description of FASTBULKWRITE field under “Attribute Descriptions for Disk Commands” (page 198).
Throughout the manual	For references to magnetic disks, deleted the term “magnetic” or changed the term “magnetic” to “physical.”

New and Changed Information in the August 2012 Edition

Chapter	Change
Chapter 6: "Configuring Disks" (page 70)	• Added Caution about partitioning SAS drives under "Partitioning HDDs and SSDs" (page 92).
Chapter 13: "Configuring and Managing Tape Drives" (page 179)	• Added the LTO-5 tape object under "Managing Encrypted Tape Drives" (page 189).
Chapter 14: "Storage Subsystem Commands" (page 190)	• Changed AUDITTRAILBUFFER description under "Attribute Descriptions for Disk Commands" (page 198). • Under "ADD SCSI Attributes" (page 223), "ADD TAPE Attributes" (page 226), and "ALTER TAPE Attributes" (page 242), changed the RECSIZE field descriptions to state that larger block sizes give better performance. • Under the FASTBULKWRITE field of "Attribute Descriptions for Disk Commands" (page 198), added a note to describe restrictions on partitioned unstructured files on an XP storage array and on H-series, J-series, and S-series internal disks.

New and Changed Information in the February 2012 Edition

Chapter	Change
Chapter 13: "Configuring and Managing Tape Drives" (page 179)	Removed attribute BUFFERED from this chapter, including the ADD TAPE and ALTER TAPE commands.
General	Changed the RECSIZE value in all examples in the manual to 57344, which is the maximum.

New and Changed Information in the August 2011 Edition

The following information was added or changed in this edition of the manual:

Chapter	Change
Chapter 5: "Displaying Information About Disk Drives" (page 48)	• Added paragraph to the beginning of this chapter: Effective with the H06.23/J06.12 RVU, Solid State Drives (SSDs) are supported in addition to Hard Disk Drives (HDDs). SSDs are configured and displayed in the same manner as HDDs. • Added new section, "Displaying Information about Partitioned Disks" (page 56).
Chapter 6: "Configuring Disks" (page 70)	• Added an entry for CLIM-Attached Disks, such as SAS disks which include HDDs and SSDs, to "Types of Disks" (page 70). • Updated information under "Considerations for M8*xxx* Fibre Channel Disks" (page 71). • Added a new section, "Considerations for CLIM-Attached Disks" (page 72). • Added a new section, "Mirroring With CLIM-Attached Disks" (page 77). • Under "Considerations for ALTER DISK, MIRRORLOCATION and Online Disk Remirroring" (page 80), stated that you can use the NOSTART attribute with the ALTER DISK command to specify that the configured mirror disk should not be started. • Added notes to the steps under "Example of Online Disk Remirroring" (page 81). • Under "ADD DISK Examples for Different Disk Types" (page 83), added an example for the set of commands for CLIM-attached disks. • Under "Adding a Disk" (page 82), changed subtitle to "Steps to Add a Disk" (page 82). • Under "Considerations for ALTER DISK and Disks" (page 85), added new attributes that require the disk to be stopped. • Added new section, "Partitioning HDDs and SSDs" (page 92). • Made changes to "Write Caching" (page 94) and "Considerations for WRITECACHE" (page 94) to clarify and stress importance of having an HP rack mount UPS to protect data during power interruptions and to indicate writecache considerations for partitioned disks.
Chapter 7: "Managing Disks" (page 96)	• Added information about how to handle SSDs in event of power failures under "Handling Power Failures" (page 117), "Hard Disk Drives (HDDs) and Solid State Drives (SSDs)" (page 117). • Added a new section, "Storage CLIMs" (page 117) that describes how storage CLIMs can be configured to handle power failures. • Changed the wording of encryption operation to encryption key rotation under "Changing the Speed of Encryption Key Rotation" (page 117).
Chapter 13: "Configuring and Managing Tape Drives" (page 179)	• Added an example of the INFO DETAIL report with the BUFFERING attribute to "Examples of INFO TAPE Detailed Reports" (page 181).
Chapter 14: "Storage Subsystem Commands" (page 190)	• Added these disk partitioning commands for SAS disks: "ADD PARTITION Command" (page 216) "DELETE PARTITION Command" (page 250) "INFO PARTITION Command" (page 256) • Modified these commands to include information about disk partitions on SAS disks. Changed examples, where appropriate: "ADD DISK Command" (page 194) "ALTER DISK Command" (page 228)

Chapter	Change

	"INFO DISK Command" (page 254)
	"INFO CLIM Command" (page 253)
	"STATUS CLIM Command" (page 280)
	
	• Under the ADD DISK command, changed heading from Disk Attribute Descriptions to "Attribute Descriptions for Disk Commands" (page 198), and added statements under disk commands referring to this section.
	• Added information to the WRITECACHE attribute under "Attribute Descriptions for Disk Commands" (page 198).
	• Added example of how to create an obey file from which attributes can be replicated from one disk to another under "ADD DISK Examples for Physical Disks" (page 196). This example includes disk partitioning and write caching.
	• Added PRIMARYPARTITION, MIRRORPARTITION, SENDTO STORAGE, PRIMARYCLIM, MIRRORCLIM, BACKUPCLIM, and MBACKUPCLIM attributes to "Disk Attributes for the ADD Command" (page 195).
	• Added PRIMARYPARTITION, MIRRORPARTITION, NOSTART, and SENDTO STORAGE to "Disk Attributes for the ALTER DISK Command" (page 230).
	• Added description for the ALTER DISK attribute, NOSTART, to "Attribute Descriptions for Disk Commands" (page 198) and the "ALTER DISK Command" (page 228).
	• Added descriptions for the ALTER DISK attributes and the ADD DISK attributes, PRIMARYPARTITION, MIRRORPARTITION, and SENDTO STORAGE, to "Attribute Descriptions for Disk Commands" (page 198), "ALTER DISK Command" (page 228), and "ADD DISK Command" (page 194).
	• Changed the "Attribute Descriptions for Disk Commands" (page 198) to indicate that if you do not configure a value for AUDITTRAILBUFFER, or if the value you specify is less than 2, DP2 uses a value of 128.
	• Changed the "Attribute Descriptions for Disk Commands" (page 198) to indicate that the SERIALWRITES attribute cannot be enabled for BladeSystems' disk subsystems attached with CLIMs because DP2 uses parallel writes for better performance. Serial writes are not needed because the CLIM is fully buffered.
	• Added a detailed example for the INFO CLIM Command under "INFO CLIM Examples" (page 254).
	• Changed the example about how to display encrypted devices that use a CLIM under "STATUS CLIM Examples" (page 281).
	• Changed the example about how to display consistency information about a disk volume under "STATUS DISK Examples For Physical Disks" (page 283).
	• Added the BUFFERING attribute to the "ADD TAPE Attributes" (page 226) and "ALTER TAPE Attributes" (page 242). Added a buffering example for the ALTER TAPE command under "ALTER TAPE Examples" (page 244).
	• Added notes about security officer involvement for the CLEARENCRYPTKEY, NEWENCRYPTKEY, and KEYGENPOLICY attributes in these sections:
	  ◦ Section : "Attribute Descriptions for Disk Commands" (page 198)
	  ◦ Section : "ALTER TAPE Attributes" (page 242)
	  ◦ Section : "INITIALIZE DISK Command" (page 260)
	  ◦ Section : "Managing Encrypted Disk Drives" (page 116)

Appendix B: "Upgrade and Replacement Procedures" (page 362)	

Chapter	Change
	• Under Section : "Upgrading a Mirrored Volume Online" (page 363), added this note: You can only use a smaller replacement disk if it is able to accommodate the largest used sector address. Please note that because of fragmentation, the largest used sector address can be much larger than the actual disk space used.
Appendix, SCF Command Summary	• Deleted this appendix from the manual. The syntax information that it contained can be found at the beginning of each command in Chapter 14: "Storage Subsystem Commands" (page 190).
Appendix, Converting D-Series PUP and COUP Commands to SCF Commands	• Deleted this appendix from the manual. The information can be found in previous versions of this manual.
General	• In chapter headings and subheadings that referred to disks, deleted the term "magnetic" because these chapters also apply to solid-state disks. • Changed all occurrences of Global Customer Support Center (GCSC) to Global NonStop Support Center (GNSC).

New and Changed Information in the March 2010 edition

The following information was added or changed in this edition of the manual:

Chapter	Change
"Storage Subsystem Commands" (page 190)	Updated "Attribute Descriptions for Disk Commands" (page 198) to update the attribute REVIVERATE.

New and Changed Information in the November 2009 edition

The following information was added or changed in this edition of the manual:

Chapter	Change
"Storage Subsystem Objects" (page 30)	Updated "The CLIM Object" (page 32) with information about encrypted devices.
"Using the Storage Subsystem Manager" (page 40)	Updated "Displaying Current Status Information" (page 40) with information about encrypted devices.
"Displaying Information About Disk Drives" (page 48)	Added section "Displaying Information about Encrypted Disks" (page 58).
"Configuring Disks" (page 70)	Updated "Considerations for ESS Disks" (page 72) and "Write Caching" (page 94).
"Managing Disks" (page 96)	Added section "Managing Encrypted Disk Drives" (page 116).
"Configuring and Managing Tape Drives" (page 179)	Added these sections: • "Example of a STATUS TAPE, ENCRYPTION Report" (page 184) • "Managing Encrypted Tape Drives" (page 189)

"Storage Subsystem Commands" (page 190)	Updated "Attribute Descriptions for Disk Commands" (page 198) to update the attribute WRITECACHE.
	Updated or added these entries:
	• "Attribute Descriptions for Disk Commands" (page 198) to add these attributes:
	  ◦ CLEARENCRYPTKEY
	  ◦ ENCRYPTPRIORITY

Chapter	Change
	- ENCRYPTRATE - KEYALGORITHM - KEYSIZE - NEWENCRYPTKEY • "ALTER DISK Command" (page 228) to add these attributes: - CLEARENCRYPTKEY - ENCRYPTPRIORITY - ENCRYPTRATE • "ALTER TAPE Command" (page 241) to add these attributes: - KEYGENPOLICY - NEWENCRYPTKEY • "INITIALIZE DISK Command" (page 260) to add these attributes: - CLEARENCRYPTKEY - KEYALGORITHM - KEYSIZE - NEWENCRYPTKEY • "STATUS CLIM Command" (page 280) • "STATUS DISK Command" (page 282) • "STATUS TAPE Command" (page 289)

New and Changed Information in the May 2009 edition

The following information was added or changed in this release of the manual:

Chapter	Change
"Storage Subsystem Commands" (page 190)	• Updated these entries: "Attribute Descriptions for Disk Commands" (page 198) with: – Updated description for AUDITTRAILBUFFER – New limits for MAXLOCKSPEROCB and MAXLOCKSPERTCB – Changes to how DP2 handles FULLCHECKPOINTS, PROTECTDIRECTORY, and SERIALWRITES for 512-byte/sector disks - Updated the BULKIO option of the ALTER SUBSYS command with a change for H-series systems

New and Changed Information in the August 2008 edition

The following information was added or changed in this release of the manual:

Chapter	Change
"Storage Subsystem Objects" (page 30)	Added "The CLIM Object" (page 32)
"Displaying Information About Disk Drives" (page 48)	• Updated: ◦ "Displaying Disk Configuration Information" (page 50) to add WRITECACHE ◦ "Example of an INFO PROFILE Report" (page 54) to add WRITECACHE
"Configuring Disks" (page 70)	• Added "Write Caching" (page 94) • Updated "Considerations for ALTER DISK and Disks" (page 85) to add WRITECACHE
"Managing Disks" (page 96)	Updated "Considerations for STOP DISK" (page 101) with information aboutKernel-Managed Swap Facility (KMSF) swap files.
"Storage Subsystem Commands" (page 190)	• Updated these entries: ◦ "Disk Attributes for the ADD Command" (page 195) to add WRITECACHE and RECOVERYTIMEOUT ◦ "Attribute Descriptions for Disk Commands" (page 198) with the AUDITTRAILBUFFER and WRITECACHE attributes ◦ "Attribute Descriptions for Disk Commands" (page 198) to add CLIM configuration attributes BACKUPCLIM, MBACKUPCLIM, MIRRORCLIM, and PRIMARYCLIM. ◦ "INITIALIZE Command" (page 260) with information about deleted data ◦ "PROFILE Attributes" (page 222) ◦ "ALTER SUBSYS Command" (page 240) with BULKIO change for H-series systems • Added these commands to support the CLIM device: ◦ "INFO CLIM Command" (page 253) ◦ "STATUS CLIM Command" (page 280) ◦ "SWITCH CLIM Command" (page 294)

Document Organization

This document is organized as follows:

Chapter 1: Introduction to the Storage Subsystem

Describes how the Subsystem Control Facility (SCF)'s storage subsystem allows you to configure and manage storage devices online. It describes how to configure the Storage Subsystem Manager ($ZZSTO), how to make changes to the configuration, and how to display information about it. It also describes fault tolerance.

Chapter 2: Storage Subsystem Objects

Describes how storage subsystem components are organized. It describes object names, states, attributes, and type.

Chapter 3: Using the Storage Subsystem Manager

Describes the storage subsystem manager ($ZZSTO) in detail. It describes how to display information about it and make changes to its configuration.

Chapter 4: Configuring and Managing the SMF Master Process

Describes the SMF master process ($ZSMS) in detail. It describes how to display information about the process, and how to add, alter, and delete a master process. It also describes how to start, stop, reset, and switch a master process.

Chapter 5: Displaying Information About Disk Drives

Describes how to display information about disks: configuration, disk cache information, disk cache statistics, and error and status information.

Chapter 6: Configuring Disks

Describes how to configure disks. It describes the types of disk configurations, how to automate disk configuration, configuring disks to start automatically, and configuring disk profiles. It also describes how to add, name, relabel, and initialize disks, and how to change their attributes. Finally, it describes how to delete disks.

Chapter 7: Managing Disks

Describes detailed procedures for managing disks: starting and reviving disks, stopping and resetting disks and groups of disks, sparing sectors, correcting doubly allocated file extents, replacing the bootstrap program, enabling and disabling file extents, swapping processors, and changing the active path for a disk.

Chapter 8: Disk Load Balancing

Describes ways to distribute the disk work load across all fabrics and processors in the system and how to maintain the load balance. It contains guidelines for current and past RVUs.

Chapter 9: Configuring and Managing Storage Pools for Disks

Describes how to configure and manage a storage pool, a collection of disks that are managed by the Storage Management Foundation (SMF). It describes how to display information about storage pools, how to configure them, and how to manage a storage pool by starting, stopping, and resetting the pool. It also contains information about adding disk drives.

Chapter 10: Configuring and Managing Virtual Disks

Describes how to configure and manage virtual disks, which are created by configuring one or more disks in a storage pool. It describes how to display information about virtual disks, how to configure them, and how to manage them by stopping and starting virtual disks.

Chapter 11: Configuring and Managing ServerNet Storage Adapters

Describes how to configure and manage ServerNet storage adapters. It describes how to display information about them and configure them, and how to manage them by troubleshooting installations, downloading new firmware, and displaying and testing connections.

Chapter 12: Configuring and Managing Open SCSI Devices

Describes how to configure and manage devices that obey the ANSI standard protocol for the small computer system interface (SCSI) to communicate with the system through the Open SCSI I/O process. It describes how to display information about them and configure them, and how to manage them by starting, altering, resetting, and stopping them, as well as changing their active paths, swapping their processors, and deleting them.

Chapter 13: Configuring and Managing Tape Drives

Describes how to configure and manage tape drives. It describes how to display information about them and configure them, and how to manage them by enabling or displaying processing, and resetting, starting, and stopping them.

Appendix A: Storage Subsystem Error Messages

Lists error messages with their cause, effect, and recovery.

Appendix B: Upgrade and Replacement Procedures
: Describes upgrade and replacement procedures for tape drives, Open SCSI devices, and upgrading a mirrored drive.

Glossary : Glossary
: Defines terms used in this manual.

Notation Conventions

General Syntax Notation

This list summarizes the notation conventions for syntax presentation in this manual.

UPPERCASE LETTERS

Uppercase letters indicate keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

MAXATTACH

Italic Letters

Italic letters, regardless of font, indicate variable items that you supply. Items not enclosed in brackets are required. For example:

file-name

`Computer Type`

Computer type letters indicate:

- C and Open System Services (OSS) keywords, commands, and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

  Use the `cextdecs.h` header file.

- Text displayed by the computer. For example:

  `Last Logon: 14 May 2006, 08:02:23`

- A listing of computer code. For example

```
if (listen(sock, 1) < 0)
{
perror("Listen Error");
exit(-1);
}
```

Bold Text

Bold text in an example indicates user input typed at the terminal. For example:

```
ENTER RUN CODE

?123
CODE RECEIVED:      123.00
```

The user must press the Return key after typing the input.

[ ] Brackets

Brackets enclose optional syntax items. For example:

```
TERM [\system-name.]$terminal-name

INT[ERRUPTS]
```

A group of items enclosed in brackets is a list from which you can choose one item or none. The items in the list can be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
FC [ num  ]
   [ -num]
   [ text]
```

```
K [ X | D ] address
```

{ } Braces

A group of items enclosed in braces is a list from which you are required to choose one item. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
LISTOPENS PROCESS { $appl-mgr-name }
                  { $process-name  }
```

```
ALLOWSU { ON | OFF }
```

| Vertical Line

A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
INSPECT { OFF | ON | SAVEABEND }
```

… Ellipsis

An ellipsis immediately following a pair of brackets or braces indicates that you can repeat the enclosed sequence of syntax items any number of times. For example:

```
M address [ , new-value ]…
```

```
 - ] {0|1|2|3|4|5|6|7|8|9}…
```

An ellipsis immediately following a single syntax item indicates that you can repeat that syntax item any number of times. For example:

```
"s-char…"
```

Punctuation

Parentheses, commas, semicolons, and other symbols not previously described must be typed as shown. For example:

```
error := NEXTFILENAME ( file-name ) ;
```

```
LISTOPENS SU $process-name.#su-name
```

Quotation marks around a symbol such as a bracket or brace indicate the symbol is a required character that you must type as shown. For example:

```
"[" repetition-constant-list "]"
```

Item Spacing

Spaces shown between items are required unless one of the items is a punctuation symbol such as a parenthesis or a comma. For example:

```
CALL STEPMOM ( process-id ) ;
```

If there is no space between two items, spaces are not permitted. In this example, no spaces are permitted between the period and any other items:

```
$process-name.#su-name
```

Line Spacing

If the syntax of a command is too long to fit on a single line, each continuation line is indented three spaces and is separated from the preceding line by a blank line. This spacing distinguishes items in a continuation line from items in a vertical list of selections. For example:

```
ALTER [ / OUT file-spec / ] LINE

   [ , attribute-spec ]…
```

Notation for Messages

This list summarizes the notation conventions for the presentation of displayed messages in this manual.

Bold Text

Bold text in an example indicates user input typed at the terminal. For example:

```
ENTER RUN CODE

?123
CODE RECEIVED:      123.00
```

The user must press the Return key after typing the input.

Nonitalic Text

Nonitalic letters, numbers, and punctuation indicate text that is displayed or returned exactly as shown. For example:

```
Backup Up.
```

Italic Text

Italic text indicates variable items whose values are displayed or returned. For example:

p-register

process-name

[ ] Brackets

Brackets enclose items that are sometimes, but not always, displayed. For example:

Event number = *number* [Subject = *first-subject-value*]

A group of items enclosed in brackets is a list of all possible items that can be displayed, of which one or none might actually be displayed. The items in the list can be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

proc-name trapped [in SQL | in SQL file system]

{ } Braces

A group of items enclosed in braces is a list of all possible items that can be displayed, of which one is actually displayed. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

obj-type obj-name state changed to *state*, caused by
{ Object | Operator | Service }

process-name State changed from *old-objstate* to *objstate*
{ Operator Request. }
{ Unknown.   }

| Vertical Line

A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
Transfer status: { OK | Failed }
```

% Percent Sign

A percent sign precedes a number that is not in decimal notation. The % notation precedes an octal number. The %B notation precedes a binary number. The %H notation precedes a hexadecimal number. For example:

```
%005400

%B101111

%H2F
```

```
P=%p-register E=%e-register
```

Related Information

For a listing of manuals describing NonStop NS-series and NonStop S-series servers, see the *NonStop NS-Series Planning Guide* , the *NonStop S-Series Planning and Configuration Guide*, and the *NonStop BladeSystem Planning Guide*.

Publishing History

Part Number	Product Version	Publication Date
529937-017	Storage Config MGR H01 and G06	March 2010
529937-018	Storage Config MGR H01 and G06	August 2011
529937-019	Storage Config MGR H01 and G06	February 2012
529937-020	Storage Config MGR H01 and G06	August 2012
529937-021	Storage Config MGR H01 and G06	August 2013
529937-022	Storage Config MGR H01 and G06	February 2014

HP Encourages Your Comments

HP encourages your comments concerning this document. We are committed to providing documentation that meets your needs. Send any errors found, suggestions for improvement, or compliments to:

docsfeedback@hp.com

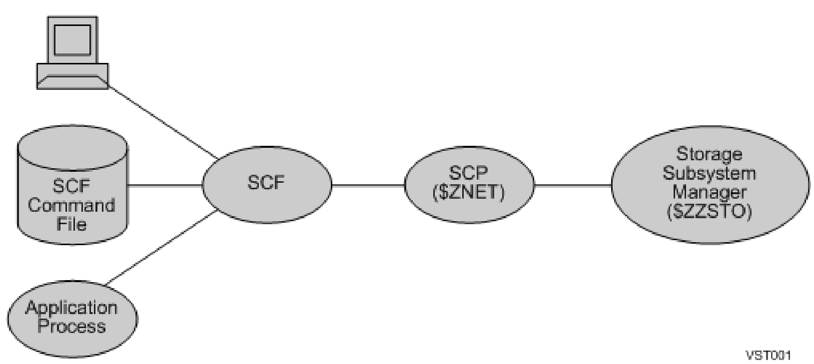
Include the document title, part number, and any comment, error found, or suggestion for improvement you have concerning this document.

1 Introduction to the Storage Subsystem

The storage subsystem allows you to use SCF to configure and manage storage devices online. This chapter contains these sections:

SCF Interface to the Storage Subsystem

The storage subsystem configures and manages all storage objects for G-series and H-series RVUs. You use SCF to configure, control, and inquire about these objects.



The interfaces between SCF, the Subsystem Control Point (SCP), and the $ZZSTO subsystem manager are described in the *NonStop S-Series Planning and Configuration Guide.*

When you enter an SCF command, SCF first determines which subsystem can process the command. SCF then directs the command to the appropriate subsystem through the Subsystem Control Point (SCP).

Manager Processes

Storage Subsystem Manager ($ZZSTO)

The $ZZSTO storage subsystem manager process is a generic process that uses SCF to perform persistent configuration and management. That is, when you use SCF to change the system configuration database, the changes survive a system load. After you configure storage subsystem objects online, $ZZSTO maintains this configuration information in the system configuration database ($SYSTEM.ZSYSCONF.CONFIG).

You can use SCF to configure any storage device except the system disk while the system continues to do other processing. Configuration changes made online using SCF take effect as soon as the object is restarted and persist through processor and system loads (unless you load the system with a new configuration file).

The storage subsystem includes features to manage both SCSI and Fibre Channel devices. The Fibre Channel features are collectively called the FCS Manager. FCS Manager assigns Fibre Channel loops for Fibre Channel disk modules (FCDMs) to the FCS monitors. FCS Manager also passes OSM requests to the FCS Monitors.

The FCS Monitor Process

The FCS monitor process (FCSMON) is a persistent process that runs in every processor. FCSMON:

- Monitors the Fibre Channel loops that connect to the Fibre Channel disk modules (FCDMs).
- Accepts requests from FCS Manager and notifies FCS Manager about FCDM configuration information
- Sends hardware change events and environmental status change events to $ZLOG
- Polls FCDMs for drive insertions and removals
- Polls the Environmental Management Units (EMUs) in the FCDMs
- Issues commands to the EMUs

Storage Management Foundation (SMF)

The Storage Management Foundation (SMF) uses SCF to help you manage data storage on multiple disks. Refer to the *Storage Management Foundation User's Guide* for information about:

- SMF configuration, management, and troubleshooting
- File relocation
- File protection and recovery
- Other SMF utilities

SMF uses these object types in the storage subsystem:

Object Type	Description	Chapter
ADAPTER	A ServerNet adapter, which controls communication between the system and attached devices	"Configuring and Managing ServerNet Storage Adapters " (page 153)
CLIM	A CLuster I/O Module (CLIM), which controls communication between the system and attached devices	"The CLIM Object" (page 32)
DISK	Virtual disk process	"Configuring and Managing Virtual Disks" (page 143)
MON	SMF monitor process $ZSMS	"Configuring and Managing the SMF Master Process" (page 42)
POOL	Storage pool process created from multiple disks	"Configuring and Managing Storage Pools for Disks" (page 134)
PROFILE	Preconfigured disk attributes	"The PROFILE Object" (page 36) and "Configuring Custom Profiles" (page 75)
SAC	ServerNet addressable controller on a storage adapter	"The SAC Object" (page 37)
SCSI	Open SCSI device	"Configuring and Managing Open SCSI Devices" (page 170)

Object Type	Description	Chapter
SUBSYS	The $ZZSTO storage subsystem manager process	"Using the Storage Subsystem Manager" (page 40)
TAPE	Tape drive	"Configuring and Managing Tape Drives" (page 179)

Configuration Overview

To Make Configuration Changes Safely

1. Create a command file containing the SCF commands.
2. Save the current $SYSTEM.ZSYSCONF.CONFIG file using the SCF SAVE command described in the *SCF Reference Manual for G-Series RVUs*.
3. Run SCF using the command file you created in Step 1.
4. If the new configuration creates a problem, you can return to the earlier configuration by loading the system from the saved configuration file.

Using OBEYFORM to Create an SCF Command File

The OBEYFORM attribute of the INFO command can help you automate the task of configuring objects by displaying configurable attributes in a form accepted by the ADD command.

You can use OBEYFORM on these objects:

- DISK, magnetic
- DISK, virtual
- MON
- POOL
- SCSI
- TAPE

Comparing OBEYFORM to LIKE

The LIKE attribute of the ADD command performs a function similar to the OBEYFORM attribute of the INFO command:

- Use the LIKE attribute to quickly add one or more similar objects to the current configuration file.
- Use the OBEYFORM attribute to see and change the attributes of new objects before adding them to the configuration file.
- You can log configuration information for existing objects and use that information later to add similar objects or to reconfigure objects that have been stopped or deleted.

By issuing successive INFO commands for each object, you can build a system-load command file that reflects modifications you have made to your configuration.

You can copy this command file to another system to add the same configuration records to a different configuration file.

Displaying Information About the Storage Subsystem

To display information about storage subsystem devices and processes, see:

- The SCF "INFO Command" (page 253)
- "LISTDEV Information for the Storage Subsystem" (page 28)

LISTDEV Information for the Storage Subsystem

You can use the SCF LISTDEV command to display configured information about processes that have a device type and that are known to SCF.

For these storage subsystem devices, you can specify an existing ldev number in any storage subsystem command (unless the command has a path specification):

- The $ZZSTO storage subsystem manager
- Disks (physical and virtual)
- Tape drives
- Open SCSI devices
- Storage pools
- The $ZSMS SMF manager
- The $ZSLM or $ZSLM2 SCSI lock manager

These devices are assigned an LDEV number in the order in which they were configured and as recorded in the destination control table.

Example

```
-> LISTDEV STORAGE

LDev   Name      PPID     BPID      Type     RSize Pri Program
   69  $ZZSTO     0,275    1,324    (65,0 )   4096 180 \COMM.$SYSTEM.SYS01.TZSTO
    6  $SYSTEM    0,257    1,257    ( 3,42)   4096 220 \COMM.$SYSTEM.SYS01.OSIMAGE
  114  $ZIMBU     9,46     8,67     ( 3,36)   4096 219 \COMM.$SYSTEM.SYS01.OVDP
  116  $ZERO      8,47     9,75     ( 3,36)   4096 219 \COMM.$SYSTEM.SYS01.OVDP
  123  $WORK2     5,264    4,276    ( 3,43)   4096 220 \COMM.$SYSTEM.SYS01.TSYSDP2
…
  138  $TAPE0     2,263    3,282    ( 4,9 )   2048 200 \COMM.$SYSTEM.SYS01.OTPPROCP
…
  316  $L700C16   1,282    0,286    ( 8,0 )   4096 220 \COMM.$SYSTEM.SYS01.TDSCSI
…
  201  $POOL6     9,19     8,22     (25,0 )   4096 180 \COMM.$SYSTEM.SYS01.OPP
…
   79  $ZSMS      5,30     4,39     (52,0 )   4096 180 \COMM.$SYSTEM.SYS01.OMP
  433  $ZSLM      0,281    1,266    (67,0 )   1024 221 \COMM.$SYSTEM.SYS01.TZSLM
```

Fault Tolerance

An important concept, and a possible source of confusion for users, is the three ways that the word primary can be used when configuring and managing devices.

Primary Processor

An IOP runs in two processors: primary and backup. The PRIMARYCPU and BACKUPCPU attributes of the ADD command are set when the device is configured.

To change which processor runs the primary process, specify that processor number in a PRIMARY command. For example, see “Swapping Processors for a Disk” (page 109). However, the PRIMARY command does not change the configured value of the PRIMARYCPU or BACKUPCPU attributes.

Primary Path

A nonmirrored device can be accessed through two paths: a primary path and a backup path. To set the paths use these attributes of the ADD command when the disk is configured:

PRIMARYLOCATION	PRIMARYSAC
BACKUPLOCATION	BACKUPSAC

The primary and backup paths go to the primary device.

A mirrored device, such as a mirrored disk, can be accessed through an additional two paths: a mirror path and a mirror-backup path. The paths are set by these attributes of the ADD command when the device is configured:

MIRRORLOCATION	MBACKUPLOCATION
MIRRORSAC	MBACKUPSAC

The primary and backup paths go to the primary device. The mirror and mirror-backup paths go the mirror device.

You can make two paths to a mirror volume active at the same time: one path to the primary device (either -P or -B) and one path to the mirror device (either -M or -MB). To make a path active, use the SWITCH command. For an example, see "Changing the Active Path for a Disk" (page 112).

Primary Disk

A mirrored disk volume has two disks: a primary disk and a mirror disk. This logical combination of two disks is set by the PRIMARYLOCATION and MIRRORLOCATION attributes of the ADD command when the disk is configured.

2 Storage Subsystem Objects

This chapter describes how storage subsystem components are organized for management through the SCF interface. It contains these sections:

Object Names

Object names are the device or process names you use to specify a particular object in an SCF command. Each object type has its own object-name format.

Object Type	Object-Name Format	Example Object Name
ADAPTER	$ZZSTO.#*type*.GRP-*g*.MOD-*m*.SLOT-*s*	$ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50
CLIM	`$ZZCIP.`*clim-name* (CIP subsystem perspective) `$ZZSTO.`*clim-name* (storage subsystem perspective)	$ZZCIP.N1002541 $ZZSTO.S1002543
DISK	*$disk$disk-path*	$DISK $DISK-P
MON	*$mon*	$ZSMS
POOL	*$pool*	$POOL
PROFILE	$ZZSTO.*internal-disk*[-*groupnum*]	$ZZSTO.INTERNAL-DISK-1
SAC	$ZZSTO.#*type*.SAC-*n*.GRP-*g*.MOD-*m*.SLOT-*s*	$ZZSTO.#FCSA.SAC-1.GRP-112.MOD-3.SLOT-5
SCSI	*$SCSI-device* *$SCSI-device-path*	$DEV $DEV0-P
SUBSYS	*$storage-manager*	$ZZSTO
TAPE	*$tape*	$TAPE

SCF Object Name Template

Most SCF commands accept more than one object name when placed in parentheses and separated by commas, for example:

```
-> SWITCH DISK ($DATA01-P, $DATA01-M)
```

Using Wild-Card Characters in an Object Name

When you issue an SCF command, you can use wild-card characters in the object name to indicate that the command affects multiple objects of the given object type. Specify multiple objects by using either:

- A single wild-card character
- Text with one or more wild-card characters

The storage subsystem supports these wild-card characters:

*	Represents a character string of undefined length. For example, to display the object names of all disks managed by the storage subsystem: `-> NAMES DISK $*`
?	Represents a single unknown character in a specific position. For example, to start disks named $DATA00, $DATA01, and so on: `-> START DISK $DATA0?`

You can use wild-card characters in any combination. For example, this command starts all disks whose name has D as the second character and the number 0 as the sixth character:

```
-> START DISK $D???0*
```

Object States and Substates

The state and substate of an object provide current status information. Knowing the object state and substate of an object can be critical. Some SCF commands have no effect on an object when the object is in a particular object state and substate but can affect the object when it is in another object state and substate. The substate further defines why the object is in a particular object state. For example, when a disk is being started, it might also need to be revived, which means it is in the STARTING state, substate REVIVE. Not all objects have substates.

Descriptions of states for...	Can be found in...
CLIM object	“CLIM Object States” (page 33)
DISK object (physical)	“Object States and Substates of Disks” (page 34)
DISK object (virtual)	“Virtual Disk Object States” (page 35)
MON object	“MON Object States” (page 36)
POOL object	“POOL Object States” (page 36)
SCSI object	“SCSI Object States” (page 38)
TAPE object	“TAPE Object States” (page 39)

Object Attributes

The attributes for each object are described under the “ADD Command” (page 194).

Object Types for the Storage Subsystem

Object Type	Definition
“The ADAPTER Object” (page 32)	A ServerNet adapter, which controls communication between the system and attached devices
“The CLIM Object” (page 32)	A CLuster I/O module (CLIM)
“The DISK Object” (page 33)	A disk volume, either physical or virtual
“The MON Object” (page 35)	The Storage Management Foundation (SMF) master process
“The POOL Object” (page 36)	A storage pool
“The PROFILE Object” (page 36)	A method used to preconfigure disks
“The SAC Object” (page 37)	A controller on a storage adapter. Most commands issued to a SAC are targeted at devices connected to the SAC.
“The SCSI Object” (page 38)	An Open SCSI device

Object Type	Definition
"The SUBSYS Object" (page 38)	The $ZZSTO storage subsystem manager process
"The TAPE Object" (page 39)	A tape drive

When issuing SCF commands, you need not specify the object type of the device if either of these conditions exists:

- You have fully specified the object name (no wild-card characters), and the device exists.
- You have already issued an SCF ASSUME command for the object type.

The ADAPTER Object

The ADAPTER object represents a specific ServerNet storage adapter that communicates with the system and attached peripheral devices. For more information, see "Configuring and Managing ServerNet Storage Adapters " (page 153).

The ADAPTER object has this format:

`$ZZSTO.#`*`type`*`.GRP-`*`g`*`.MOD-`*`m`*`.SLOT-`*`s`*

$*ZZSTO*

is the storage subsystem manager.

#*type*

is the adapter type:

PMF	PMF and PMF-2
IOMF	IOMF and IOMF-2
SNDA	6760 ServerNet device adapter

GRP-*g*.MOD-*m*.SLOT-*s*

is the physical location (group, module, and slot) of the adapter. *g* cannot have a leading zero.

Wild-card characters are supported.

Examples

```
$ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50
$ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-55
$ZZSTO.#SNDA.SAC-1.GRP-1.MOD-1.SLOT-53
```

The CLIM Object

From the perspective of the storage subsystem, the CLIM object is similar to the ADAPTER object because it provides communication connectivity between the system and attached, peripheral storage devices. You can obtain information about the CLIM from the storage subsystem by using the INFO CLIM $ZZSTO.*clim-name* or STATUS CLIM $ZZSTO.*clim-name* command. You also can obtain status and info about the CLIM by using the INFO CLIM $ZZCIP.*clim-name* (or STATUS CLIM $ZZCIP.*clim-name*).

The CLIM object exists in both the storage and CIP subsystem and certain management tasks like adding a CLIM are only accessible through the CIP subsystem. One task, SWITCH CLIM, can be done through either the CIP or storage subsystem. For information about adding or deleting a CLIM, see the *Cluster I/O Protocols (CIP) Configuration and Management Manual*.

CLIMs can be configured to support encryption so that disk or tape devices connected to them can be encrypted. For details about encryption, see the *NonStop Volume Level Encryption Guide*.

The CLIM object in the storage subsystem has this format:

`$ZZSTO.`*`clim-name`*

$ZZSTO

is the storage subsystem manager.

`clim-name`

is the user-defined CLIM name.

The CLIM object in the CIP subsystem has this format:

`$ZZCIP.`*`clim-name`*

$ZZCIP

is the cluster I/O protocol subsystem manager.

`clim-name`

is the user-defined CLIM name.

Wild-card characters are supported.

Examples

```
$ZZSTO.S1002541
```

CLIM Object States

The object state of a CLIM provides current status information about the disk or a path to that disk.

Table 2 CLIM Object States

SCF State	Description
STARTED	Accessible
STARTING	In transition to the STARTED state
STOPPED	Inaccessible

The DISK Object

The DISK object represents a disk storage device which can be a logical disk or a virtual disk:

- A disk is a disk device or device pair that stores and accesses data on disk media. For more information, see "Configuring Disks" (page 70). Depending on the command, the DISK object can refer to one of these aspects of a disk:
  - A single disk—a disk device.
  - A disk volume—a logical disk, which can be one or two disk devices. If mirrored, the two disks are treated as one.
  - A single path to a disk—the primary, backup, mirror, or mirror backup path.
- A virtual disk is a logical representation of disk space. This space can be part of a disk, an entire disk, or multiple disks. For more information, see "Configuring and Managing Virtual Disks" (page 143).

The DISK object has this format:

`$`*`disk`*` [ -P | -B | -M | -MB ]`

$*disk* [-P | -B | -M | -MB]

is the disk process controlling the disk and, optionally, the path (primary, backup, mirror, or mirror backup). Paths apply to disks only.

The dollar sign ($) is followed by a maximum of seven alphanumeric characters, the first of which must be alphabetic. Seven-character disk names are network-accessible on systems running H-series, G-series, and D-series RVUs.

Wild-card characters are supported.

Examples

```
$DATA00-P
$DATA00-M
$DATA00-B
$DATA00-MB
$DATA03
```

Object States and Substates of Disks

The object state and substate of a disk provide current status information about the disk or a path to that disk.

Table 3 Object States and Substates of Disks

State	Substate	Caused By	Description
SERVICING	--	--	The disk volume or disk path is in transition and is unavailable due to a pending service operation. (To take the disk volume or disk path out of the SERVICING state, issue a RESET DISK command.)
	SPECIAL	TSM, OSM; START command or ALTER command	The disk volume is being used by OSM, TSM, the START command, ALTER DISK, LABEL command, or some other privileged process.
	TEST	TSM	The disk volume or disk path is being tested by OSM or TSM.
STARTED	--	TSM or START command	The disk volume or disk path is accessible for user processes.
STARTING	--	START command	The disk volume or disk path is in transition from a STOPPED state to a STARTED state.
	REVIVE	START command	A revive operation is in progress. (When the revive operation is complete, the state of the disk and path changes to STARTED.)
STOPPED	CONFIG-ERROR	Device error	The disk volume is configured improperly.
	DOWN	TSM, OSM; STOP or RESET command	The disk volume or disk path is no longer logically accessible to user processes.
	HARDDOWN	ABORT DISK command; hardware errors; no disk in slot	The disk volume or disk path was put into the HARDDOWN substate by the ABORT DISK command or is physically inaccessible due to hardware errors.
	INACCESSIBLE	Processor halts	All paths to the processor are down.
	PREMATURE-TAKEOVER	Improper sequence of commands	The backup disk process was asked to take over for the primary disk process before the backup disk process had enough information.
	RESOURCE-UNAVAILABLE	Resource is down or in an inaccessible state	The disk process could not obtain a necessary resource.
	UNKNOWN-REASON	Communication error	The disk process is unavailable for an unknown reason.

Recovering From Various SCF Object States and Substates

This table describes how to prepare a disk for restarting. For all situations except CONFIG-ERROR, the RESET command puts a disk into the STOPPED state, substate DOWN, ready for a START command.

SCF State	SCF Substate	Description	Recovery
SERVICING	SPECIAL	The disk is reserved for service.	Issue the RESET command.
SERVICING	TEST	The disk is reserved for service.	Issue the RESET command.
STOPPED	CONFIG-ERROR	The disk is configured improperly.	Reconfigure the disk properly.
STOPPED	HARDDOWN	Caused by the ABORT command or a hardware error.	Issue the RESET command.
STOPPED	PREMATURE-TAKEOVER	The backup IOP was asked to take over for the primary IOP before it had the proper information.	Issue the RESET, FORCED command.If this fails, reload the processors.
STOPPED	RESOURCE-UNAVAILABLE	The IOP could not get a necessary resource.	Issue the RESET, FORCED command.
STOPPED	UNKNOWN-REASON	The IOP is down for an unknown reason.	Issue the RESET, FORCED command.

Virtual Disk Object States

Table 4 Virtual Disk Object States

SCF State	Description
STARTED	Accessible
STARTING	In transition to the STARTED state
STOPPED	Inaccessible

The MON Object

The MON object represents the $ZSMS Storage Management Foundation (SMF) master process, which maintains information about storage pools and virtual disks on the system. The SMF master process coordinates modification services for disks and virtual disks (such as defining, deleting, and redefining) and information services for storage pools. You use the MON object to query and configure the SMF master process. For more information, see "Configuring and Managing the SMF Master Process" (page 42).

The MON object has this format:

$*ZSMS*

$*ZSMS*

is the SMF master process.

Wild-card characters are supported.

Example

```
$ZSMS
```

MON Object States

The object state of a MON object provides current status information about the SMF master process

Table 5 MON Object States

State	Substate	Description
STARTED		The SMF master process is available.
STOPPED		The SMF master process is not available.
SERVICING	SPECIAL	The SMF master process is reserved by OSM, TSM, SCF, or some other privileged process.

The POOL Object

The POOL object represents a storage pool, which is a collection of disk volumes, all on the same system. A POOL object represents a resource of spaces where data files can be placed. Virtual disk processes can create files across volumes of a storage pool. A storage pool can have up to 144 disk volumes. The number of storage pools on a system is limited only by the current size of the destination control table (DCT). For more information, see "Configuring Custom Profiles" (page 75).

The POOL object has this format:

`$pool-name`

$*pool-name*

is the name of a storage pool. The format is a dollar sign ($) followed by a maximum of seven alphanumeric characters, the first of which must be alphabetic.

Wild-card characters are supported.

Example

```
$POOL01
$P00L
```

POOL Object States

The object state of a POOL object provides current status information about a storage pool.

Table 6 POOL Object States

State	Substate	Description
STARTED		The storage pool is available.
STOPPED		The storage pool is not available.
SERVICING	SPECIAL	The storage pool is reserved by SCF or some other privileged process.

The PROFILE Object

Use the PROFILE object to preconfigure an internal disk when it is inserted into the system. You can modify the default profile, or you can create a unique profile for each system enclosure.

The standard default profile, $ZZSTO.INTERNAL-DISK, contains all the configurable attributes of an ADD DISK command except PRIMARYLOCATION and MIRRORLOCATION. These attributes must be unique for each mirrored disk. For more information, see"Configuring Custom Profiles" (page 75).

The PROFILE object has this format:

`$ZZSTO.internal-disk[-groupnum ]`

$ZZSTO

is the storage subsystem manager.

internal-disk

is the type of disk for which a storage subsystem profile can be configured.

-groupnum

is the enclosure number to which a user-configured profile applies. *groupnum* cannot have a leading zero.

The default standard profile has no group number and can be used to preconfigure any internal disk inserted in a slot in any enclosure.

To tailor internal disk profiles based on which enclosure houses the disk, create additional profiles by specifying the group number of the enclosure as shown in this list. The enclosure number you specify must already exist. For details about how to create a profile, see "Creating a Custom Profile" (page 75). Valid group numbers are:

groupnum	**Enclosures**
1-8	Processor enclosures
11-19	I/O enclosures attached to processor group 1
21-29	I/O enclosures attached to processor group 2
31-39	I/O enclosures attached to processor group 3
41-49	I/O enclosures attached to processor group 4
51-59	I/O enclosures attached to processor group 5
61-69	I/O enclosures attached to processor group 6
71-79	I/O enclosures attached to processor group 7
81-89	I/O enclosures attached to processor group 8

Wild-card characters are supported.

Examples

- The PROFILE object for any disk slot in any enclosure is:

  `$ZZSTO.INTERNAL-DISK`

- The PROFILE object for disk slots in group 01 is:

  `$ZZSTO.INTERNAL-DISK-1`

The SAC Object

The SAC object represents a ServerNet addressable controller on a storage adapter. Commands issued for a SAC are often targeted at devices connected to the SAC. For example, you can use the SAC object to test or display information about:

- The connection to an arbitrated loop of a Fibre Channel disk module (FCDM)
- The ports in a Fibre Channel link
- The SAC itself

For more information, see "PROBE SAC Command" (page 268), "STATS SAC Command" (page 278), and "STATUS SAC Command" (page 285).

The SAC object has this format:

`$ZZSTO.#`*type*`.SAC-`*n*`.GRP-`*g*`.MOD-`*m*`.SLOT-`*s*

$ZZSTO

is the storage subsystem manager.

#type
: is the adapter type, for example FCSA.

#SAC-*n*
: is the number of the SAC on the adapter.

GRP-*g*.MOD-*m*.SLOT-*s*
: is the physical location (group, module, and slot) of the adapter. *g* cannot have a leading zero.

Wild-card characters are supported.

Example

```
$ZZSTO.#FCSA.SAC-2.GRP-111.MOD-2.SLOT-5
```

The SCSI Object

The SCSI object represents an Open SCSI device, which is a storage device that uses the small computer system interface (SCSI) to connect the device to the system.

The SCSI object can also represent an Open SCSI Fibre Channel device, which is a Fibre Channel device that uses the SCSI protocol to communicate with the system through the Open SCSI I/O process.

Unless stated otherwise, references to Open SCSI devices also refer to Open SCSI Fibre Channel devices.

For more information, see "Configuring and Managing Open SCSI Devices" (page 170).

The SCSI object has this format:

$SCSI-device-name [-P | -B]

$*SCSI-device-name*
: is the name of the Open SCSI process controlling the Open SCSI device. The format is a dollar sign ($) followed by a maximum of seven alphanumeric characters, the first of which must be alphabetic.

-P | -B
: indicates a path to the primary or backup Open SCSI device.

Wild-card characters are supported.

Examples

```
$DEV0
$DEV1-P
$SCSI1
```

SCSI Object States

The state of a SCSI object provides current status information about the SCSI device or a path to that SCSI device

Table 7 SCSI Object States

State	Description
STARTED	The device is available.
STOPPED	The device is not available.

The SUBSYS Object

The SUBSYS object represents the $ZZSTO storage subsystem manager process. Use the SUBSYS object to query and configure the storage subsystem manager, which configures and manages all storage objects in the storage subsystem. For more information, see "Using the Storage Subsystem Manager" (page 40).

The SUBSYS object has this format:

$ZZSTO

$*ZZSTO*

is the storage subsystem manager process.

Wild-card characters are supported.

Example

```
$ZZSTO
```

The TAPE Object

The TAPE object represents a specific tape drive, which is a storage device for archiving data. For more information, see"Configuring and Managing Tape Drives" (page 179) .

The TAPE object has this format:

$tape-name

$*tape-name*

is the name of the tape process controlling the tape device.

Wild-card characters are supported.

Examples

```
$TAPE0
$TAPE1
```

TAPE Object States

The object state of a TAPE object provides current status information about a tape drive

Table 8 TAPE Object States

State	Description
STARTED	The tape drive is available.
STOPPED	The tape drive is not available.

3 Using the Storage Subsystem Manager

The storage subsystem manager ($ZZSTO) is a generic process configured in the kernel subsystem. $ZZSTO monitors and controls all storage subsystem objects. $ZZSTO has the object type of SUBSYS. For information about this object, see "The SUBSYS Object" (page 38). This chapter contains these sections:

Displaying Information

To display information about the storage subsystem manager:

Displaying Configuration Information

The "INFO SUBSYS Command" (page 259) displays configured information about the storage subsystem manager:

```
-> INFO SUBSYS $ZZSTO

STORAGE - Info SUBSYS $ZZSTO
 AutoConfigure  AutoRevive  Autostart  BulkIO  LabelTape  UPS
 OFF            OFF         ON         ON      ON         OFF
```

Explanation of Fields — INFO SUBSYS Report

AutoConfigure	Shows whether automatic configuration of disk devices is permitted when an internal SCSI or M8*xxx* disk is inserted into a slot.
AutoRevive	Shows whether an automatic revive is permitted during a system load or when an internal SCSI or M8*xxx* disk is inserted into a slot.
Autostart	Shows whether the disk process is allowed to start automatically when an internal SCSI or M8*xxx* disk is inserted into a slot.
BulkIO	Shows whether direct bulk I/O is on or off.
LabelTape	Shows whether labeled-tape operation is on or off.
UPS	Shows whether the uninterruptible power supply option is on or off.

NOTE: M8*xxx* disks are described in "Types of Disks" (page 70).

Displaying Current Status Information

The "STATUS SUBSYS Command" (page 288) displays current information about the storage subsystem manager:

```
1-> STATUS SUBSYS $ZZSTO
STORAGE - Status SUBSYS $ZZSTO
 BulkIO  EncryptionLicense  LabelTape  UPS
 OFF     VALID              ON         ON
```

Explanation of Fields — STATUS SUBSYS Report

BulkIO	Shows whether direct bulk I/O is on or off.
LabelTape	Shows whether labeled-tape operation is on or off.

EncryptionLicense	Shows whether encryption license is valid or invalid.
UPS	Shows whether the uninterruptible power supply option is on or off.

Using the PRIMARY SUBSYS Command

The PRIMARY SUBSYS command replaces a series of PRIMARY commands for individual IOPs in the topology branch. The PRIMARY SUBSYS command does not start any processes.

At the completion of the command, the PRIMARY SUBSYS command returns a list of the IOPs that did not move the primary IOP. You must handle any exceptions individually.

On G06.10 and earlier RVUs, a primary processor change by one disk IOP might cause other disk IOPs to lose controller ownership and thus cause those other disk IOPs to perform primary changes. See “Guidelines for G06.10 and Earlier RVUs” (page 123).

4 Configuring and Managing the SMF Master Process

The SMF master process manages a centrally located database of information required at startup time for all other SMF processes, and it monitors and controls all storage pools and virtual disks in the system. The SMF master process, $ZSMS, has the object type of MON. For a description of this object, see “The MON Object” (page 35) and “MON Object States” (page 36). This chapter contains:

Displaying Information

To display information about the SMF master process:

Displaying Configuration Information

The “INFO MON Command” (page 256) displays configured information about the SMF master process.

Example of an INFO MON Report

```
-> INFO MON $ZSMS

STORAGE - Info MON \COMM.$ZSMS
*BackupCPU  CatalogLocation               *PrimaryCPU  *Program
 4          $VIRCFG.ZSMSCAT0               5             $SYSTEM.SYSTEM.OMP
```

Explanation of Fields — INFO MON Report

*	An attribute whose value you can change by using an ALTER MON command.
BackupCPU	The processor number of the backup SMF master process.
CatalogLocation	The location of the catalog for the SMF master process.

PrimaryCPU	The processor number of the primary SMF master process.
Program	The object file name of the SMF master process.

Example of a Detailed INFO MON Report

```
-> INFO MON $ZSMS, DETAIL

STORAGE - Detailed Info MON \COMM.$ZSMS
*BackupCPU............................. 4
 CatalogLocation....................... $VIRCFG.ZSMSCAT0
*HighPin............................... OFF
*Mode.................................. QUIET
*PrimaryCPU............................ 5
*Program............................... $SYSTEM.SYSTEM.OMP
```

Explanation of Fields — Detailed INFO MON Report	
*	An attribute whose value you can change by using an ALTER MON command.
BackupCPU	The processor number of the backup SMF master process.
CatalogLocation	The location of the catalog for the SMF master process.
Highpin	Shows whether the IOP can run at a high PIN (ON) or a low PIN (OFF). The PIN is the process identification number.
Mode	Shows whether the SMF master process generates additional EMS messages.
PrimaryCPU	The processor number of the primary SMF master process.
Program	The object file name of the SMF master process.

Displaying Current Status Information

The "STATUS MON Command" (page 284) displays status information about the SMF master process.

Example of a STATUS MON Command

```
-> STATUS MON $ZSMS

STORAGE - Status MON \COMM.$ZSMS
LDev    State        Primary    Backup    Type   Subtype
                     PID        PID
  79    STARTED       5,30       4,39     52     0
```

Example of a Detailed STATUS MON Report

```
-> STATUS MON $ZSMS, DETAIL

STORAGE - Detailed Status MON \COMM.$ZSMS

Mon Process Information:
 LDev    State        Primary    Backup    Type   Subtype
                      PID        PID
   79    STARTED       5,30       4,39     52     0

Mon I/O Process Information:
  Library File.......
  Program File....... $SYSTEM.SYS01.OMP
```

Explanation of Fields — Detailed STATUS MON Report	
LDev	The logical device number for the SMF master process.

State	The object state of the SMF master process.
Primary PID Backup PID	The processor number and PIN of the current primary and backup SMF master processes.
Type	The device type. The SMF master process is always type 52.
Subtype	The device subtype. The SMF master process is always subtype 0.
Mon I/O Process Information:	
Library File	Shows the library file name of the SMF master process.
Program File	Shows the program file name of the SMF master process.

Configuring the SMF Master Process

Configuration tasks for the SMF master process include:

- "Adding the SMF Master Process" (page 44)
- "Altering the Values of the SMF Master Process Attributes" (page 45)
- "Deleting the SMF Master Process" (page 45)

Adding the SMF Master Process

The "ADD MON Command" (page 215) adds the SMF master process to the system configuration database.

Considerations for ADD MON

Before adding an SMF master process to the system configuration, verify that:

- The HP NonStop Transaction Management Facility (TMF) product is running.
- The volume you plan to specify as the CATALOGLOCATION is in the STARTED state and enabled in TMF.
- The system disk is enabled in TMF if you do not specify a CATALOGLOCATION.
- Example

1. Add the process to the system, specifying its catalog location:

```
-> ADD MON $ZSMS, SENDTO STORAGE, PRIMARYCPU 5, &
-> BACKUPCPU 4, CATALOGLOCATION $VIRCFG
```

2. Verify the SMF master process has been properly configured:

```
-> INFO MON $ZSMS
```

Example of Adding the SMF Master Process Using the OBEYFORM Attribute

To configure the SMF master process for another system, create a command file by using the OBEYFORM attribute of the INFO DISK command.

Capture the existing default profile:

```
-> INFO / OUT LOG / $ZSMS, OBEYFORM

== STORAGE - Detailed Info MON in obeyform: \COMM.$ZSMS
ADD MON $ZSMS , &
    SENDTO STORAGE , &
    BACKUPCPU 4 , &
    CATALOGLOCATION $VIRCFG.ZSMSCAT0, &
    HIGHPIN OFF, &
    MODE QUIET, &
    PRIMARYCPU 5 , &
    PROGRAM $SYSTEM.SYSTEM.OMP
```

- Enter the log file contents either by copying and pasting into an SCF command line or by using the log file as a command file.

Altering the Values of the SMF Master Process Attributes

The "ALTER MON Command" (page 232) changes attributes of the SMF master process in the system configuration database.

Considerations for ALTER MON

CAUTION: Using wild-card characters can affect the MON objects of other subsystems (QIO, for example).

- The process must be in the STOPPED state or not running before you can change its configuration.
- Changes take effect when you restart the process.
- Example

1. Stop the SMF master process:

```
-> STOP $ZSMS
```

2. Change one or more "ALTER MON Attributes" (page 233). For example, to change the mode to generate all the EMS messages:

```
-> ALTER $ZSMS, MODE NOISY
```

3. Verify the change has been made:

```
-> INFO $ZSMS, DETAIL

STORAGE - Detailed Info MON \COMM.$ZSMS
*BackupCPU............................ 4
 CatalogLocation...................... $VIRCFG.ZSMSCAT0
*HighPin.............................. OFF
*Mode................................. NOISY
*PrimaryCPU........................... 5
*Program.............................. $SYSTEM.SYSTEM.OMP
```

4. Restart the SMF master process:

```
-> START $ZSMS
```

Deleting the SMF Master Process

The "DELETE MON Command" (page 249) removes the SMF master process from the system configuration database.

Considerations for DELETE MON

CAUTION: Using wild-card characters can affect the MON objects of other subsystems (QIO, for example).

- The DELETE MON command stops the MON process, deletes the reserved name from the destination control table, and removes configuration information from the system configuration database.
- If you delete the MON process, the catalog files remain on the disk. Remove these catalog files by using the SMFIXUP utility.
- Example

1. Before removing an SMF master process from the system configuration database, verify that:
   - The MON process is stopped.
   - TMF is running.
   - The CATALOGLOCATION volume for the SMF master process is started and enabled by TMF.
   - Stop the SMF master process:

     ```
     -> STOP $ZSMS
     ```

     The SMF master process finishes any current activity before entering the STOPPED state.
2. Delete the SMF master process:

   ```
   -> DELETE MON $ZSMS
   ```
3. Verify the deletion:

-> INFO $ZSMS

Managing the SMF Master Process

Management tasks for the SMF master process include:

- "Starting the SMF Master Process" (page 46)
- "Stopping the SMF Master Process" (page 46)
- "Resetting the SMF Master Process" (page 47)
- "Swapping Processors for the SMF Master Process" (page 47)

Starting the SMF Master Process

The "START MON Command" (page 275) starts the SMF master process and puts it in the STARTED state.

Considerations for START MON

- Before starting the SMF master process, verify that:
  - $ZSMS has a configuration record (as verified by the INFO command).
  - TMF is running on the system.
  - The CATALOGLOCATION volume for $ZSMS is started, and is enabled by TMF.
  - To take $ZSMS out of the SERVICING state, reset it and start it again.

Starting $ZSMS

1. Start the process:

   ```
   -> START $ZSMS
   ```
2. Verify the process is started:

   ```
   -> STATUS $ZSMS
   ```

Stopping the SMF Master Process

The "STOP MON Command" (page 291), like the "ABORT MON Command" (page 193), stops access to the SMF master process.

-> STOP $ZSMS

The SMF master process finishes any current activity before entering the STOPPED state.

Resetting the SMF Master Process

The "RESET MON Command" (page 271) puts the SMF master process into the STOPPED state, substate DOWN, reading for restarting.

1. Check the current status of the SMF master process:

   ```
   -> STATUS $ZSMS
   ```

2. If it is not in the STOPPED state, substate DOWN:

   ```
   -> RESET $ZSMS
   ```

3. Start the SMF master process:

   ```
   -> START $ZSMS
   ```

4. Verify the SMF master process is started:

   ```
   -> STATUS $ZSMS
   ```

Swapping Processors for the SMF Master Process

The "PRIMARY MON Command" (page 265) swaps the primary and backup processors for the SMF master process. The current primary processor of a specified device becomes the backup processor, and the backup processor becomes the primary processor, but the PRIMARYCPU and BACKUPCPU values stay the same.

You typically swap processors when load balancing the system or preparing for disk replacement.

Examples

- To execute the primary process of $ZSMS in processor 3 (assuming it is configured to run in processor 3):

  ```
  -> PRIMARY $ZSMS, 3
  ```

- To make the current backup process of $ZSMS the primary process:

  ```
  -> PRIMARY $ZSMS
  ```

5 Displaying Information About Disk Drives

This chapter describes displaying information about disk drives. For a description of the different types of disks that the storage subsystem supports, see "Types of Disks" (page 70).

Effective with the H06.23/J06.12 RVU, Solid State Drives (SSDs) are supported in addition to Hard Disk Drives (HDDs). SSDs are configured and displayed in the same manner as HDDs.

Physical disks share the object type of DISK with virtual disks (discussed in "Configuring and Managing Virtual Disks" (page 143)).

Displaying Information About Installed, Nonconfigured Internal Disks

When physically installing an internal SCSI disk, you should note manually the location of any disk that is not configured. If you physically install a new disk in an empty slot that is not configured for a disk, OSM or TSM displays the disk. However, the disk is not visible to SCF unless you have automatic configuration enabled. Likewise, if you delete a disk from the system configuration database, the disk becomes invisible to SCF, although OSM and TSM show it.

Finding Nonconfigured Internal Disks

1. Check the EMS messages. A nonconfigured internal disk generates an EMS message ("Disk is not configured") if one of these events occurs:
   - You physically install the disk into a slot.
   - You load the system.
   - You reload a processor.
2. To find and automatically configure nonconfigured internal disks by using SCF:
   a. Enable automatic configuration:

      ```
      -> ALTER SUBSYS, AUTOCONFIGURE ON
      ```

      This action allows the system to configure automatically all installed internal disks according to a default or custom profile (see "The PROFILE Object" (page 36)).

      For more information about how to use the AUTOCONFIGURE attribute, see "Automating Disk Configuration" (page 73).

   b. Use theNAMEMASK attribute of the profile for the enclosure to help you determine the names of automatically added disks.

      NOTE: To find nonconfigured model 45*xx* disks, refer to the completed installation planning forms for those disks. Neither SCF, OSM, nor TSM can display information about nonconfigured 45*xx* disks.

Displaying Information About Configured Disks

- “Displaying Disk Configuration Information” (page 50)
- “Displaying Disk Label Information” (page 52)
- “Displaying Profile Attributes” (page 54)
- “Displaying Information about Partitioned Disks” (page 56)
- “Displaying Information about Encrypted Disks” (page 58)

To display a report about all disks on the system:

```
-> INFO DISK $* , SUB MAGNETIC
```

An INFO Report for a Disk

```
-> INFO $AUDIT

STORAGE - Info Magnetic DISK configuration \COMM.$AUDIT
Primary      *Mirror      *Primary  *Backup  *DiskProcesses   *Pool
Location      Location     CPU       CPU
(1,1,15)      (1,1,16)     0         1        4                None
```

Explanation of Fields — INFO Report for a Disk	
*	An attribute whose value you can change by using the ALTER DISK command.
Primary Location	The physical location (group, module, slot) for the primary disk of a mirrored volume.
Mirror Location	The physical location (group, module, slot) for the mirror disk of a mirrored volume.
Primary CPU	The processor number of the primary disk process.
Backup CPU	The processor number of the backup disk process.
DiskProcesses	The number of disk I/O processes allocated for the disk.
Pool	The storage pool process with which the disk volume is associated.

Example 2: A Detailed INFO Report for a Disk

For disks, the INFO DISK attributes produce a report containing specific sections:

Topic	INFO Command Attribute
“Displaying Common Disk Information” (page 49)	DETAIL
“Displaying Disk Configuration Information” (page 50)	CONFIG
“Displaying Disk Cache Configuration Information” (page 59)	CACHE
“Displaying Disk Label Information” (page 52)	LABEL
“Displaying Bad Sector Information” (page 62)	BAD
“Displaying Profile Attributes” (page 54)	LOG

Displaying Common Disk Information

```
-> INFO $SYSTEM, DETAIL

STORAGE - Detailed Information Magnetic DISK \COMM.$SYSTEM
Common Disk Configuration Information:
*BackupCpu............................. 1
*HighPin............................... ON
*PrimaryCpu............................ 0
*Program............................... $SYSTEM.SYSTEM.TSYSDP2
*StartState............................ STARTED
```

The INFO DETAIL,and INFO CONFIG reports start with this display.

Explanation of Fields — INFO SYSTEM Report	
*	Indicates an attribute whose value you can change by using the ALTER DISK command.
BackupCpu	The processor number of the backup disk process.
HighPin	Shows whether the disk process can run at a high PIN (ON) or must run at a low PIN (OFF). The PIN is the process identification number.
PrimaryCpu	The processor number of the primary disk process.
Program	The object file name of the disk process.
StartState	Shows whether the disk process is configured to be STARTED (in the STARTED state) or STOPPED (in the STOPPED state, substate DOWN) when the system is loaded.

Displaying Disk Configuration Information

```
-> INFO $SYSTEM, CONFIG

Disk Type Specific Information:
*AuditTrailBuffer/SQLMXBuffer (MB)..... 0
*AutoRevive.......................... OFF
*AutoSelect.......................... n/a
*AutoStart........................... ON
*CapacityMismatch.................... OFF
*CBPoolLen........................... 1000
*FastBulkWrite....................... ON
*FSTCaching.......................... OFF
*FullCheckpoints..................... ENABLED
*HaltOnError......................... 1
*LKIDLongPoolLen..................... 8
*LKTableSpaceLen..................... 15
*MaxLocksPerOCB...................... 5000
*MaxLocksPerTCB...................... 5000
*NonAuditedInsert.................... OFF
*NumDiskProcesses.................... 6
*OSSCaching.......................... ON
*PhysvolSelect....................... n/a
*Pool................................ None
*ProtectDirectory.................... SERIAL
*RecoveryTimeout..................... 30
*ReviveBlocks........................ 1
*ReviveInterval...................... 100
*RevivePriority...................... 0
*ReviveRate.......................... 0
*SerialWrites........................ ENABLED
*WriteCache.......................... DISABLED

Primary Path Information:
 Adapter............................. $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50
 Disk Device ID / Port Name........... 0
 Disk Device LUN...................... 0
 Location (Group,Module,Slot)......... (1,1,11)
 SAC Name............................ $ZZSTO.#PMF.SAC-2.GRP-1.MOD-1.SLOT-50
 SAC Number.......................... 2

Backup Path Information:
 Adapter............................. $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-55
 Disk Device ID / Port Name........... 0
 Disk Device LUN...................... 0
 Location (Group,Module,Slot)......... (1,1,11)
 SAC Name............................ $ZZSTO.#PMF.SAC-2.GRP-1.MOD-1.SLOT-55
 SAC Number.......................... 2

Mirror Path Information:
 Adapter............................. $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-55
 Disk Device ID / Port Name............0
 Disk Device LUN...................... 0
```

```
 Location (Group,Module,Slot).......... (1,1,12)
 SAC Name............................. $ZZSTO.#PMF.SAC-1.GRP-1.MOD-1.SLOT-55
 SAC Number........................... 1

Mirror Backup Path Information:
 Adapter.............................. $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50
 Disk Device ID / Port Name........... 0
 Disk Device LUN...................... 0
 Location (Group,Module,Slot).......... (1,1,12)
 SAC Name............................. $ZZSTO.#PMF.SAC-1.GRP-1.MOD-1.SLOT-50
 SAC Number........................... 1
```

Explanation of Fields

Disk Type Specific Information

AuditTrailBuffer/ SQLMXBuffer (MB)	Shows one of: • The audit-trail buffer size in megabytes for a TMF audit-trail volume. This value is used to improve the performance of systems using the Remote Duplicate Database Facility. • The SQLMXBuffer attribute specifies the buffer size (in megabytes) for an SQL/MX session in DP2.
AutoRevive	Shows whether the storage subsystem manager automatically starts a revive operation on an internal SCSI or M8xxx disk that is inserted into a slot.
AutoSelect	(Physical disks only) shows whether a virtual disk process is allowed to automatically consider this physical volume when making file placement decisions.
AutoStart	(Internal SCSI and M8*xxx* disks only) specifies whether the disk process is automatically started when the disk is inserted.
CapacityMismatch	Specifies whether the source drive remains up or goes harddown when a revive completes for a mirrored volume consisting of drives of different capacities.
CBPoolLen	The maximum memory (in 128-KB units) that can be allocated for open-related data structures on the disk. The memory available for these structures limits the total number of concurrent opens allowed on the disk.
FastBulkWrite	Shows whether Fast Cache Bulk Writes are turned on. When ON, applications and utilities using bulk writes to unstructured files may have higher throughput. Using FastBulkWrite ON can result in lost data in unstructured files if the CPU running the primary disk process fails. This attribute affects all 512-byte-sector disks.
FSTCaching	Shows whether the free space table for a disk is updated in memory (ON) or on disk (OFF). FSTCaching ON can increase performance. The FST is always rebuilt from disk when the disk is started. The system disk always runs with FSTCACHING ON, regardless of the configured value.
FullCheckpoints	Shows whether the validity of data written to the disk is being protected by a full-block checkpoint (applies only to nondirectory structured files). Options are ENABLED, DISABLED, or FORCED. For mirrored volumes, see SERIALWRITES.
HaltOnError	Shows whether an internally detected disk-process failure should force a halt (code %11500) in the primary processor, backup processor, or both.
LKIDLongPoolLen	The memory (in 128-KB units) to be allocated for lock key space. This space stores keys larger than 16 bytes when locking records in key-sequenced files.
LKTableSpaceLen	The memory (in 128-kilobyte units) to be allocated for lock-related data structures. The memory available for these structures limits the total number of file and record locks allowed on the disk.
MaxLocksPerOCB	The maximum number of records that can be locked outside a transaction.
MaxLocksPerTCB	The maximum number of records and files a transaction can lock.
NonAuditedInsert	Indicates whether nonaudited insert mode is enabled.
NumDiskProcesses	The number of disk processes allocated for the disk.
OSSCaching	Shows whether caching for HP NonStop Open System Services (OSS) files is enabled.

PhysvolSelect	(Physical disks only) shows whether a virtual disk process is allowed to consider the physical volume for file placement.
Pool	Shows which storage pool process the disk is associated with.
ProtectDirectory	Shows the tape of protection to use for the disk volume directory: full-block checking (CHECKPOINT), serial writes (SERIAL), or disabled (OFF).
RecoveryTimeout	The number of seconds that the NonStop storage controller must wait for the device to respond to a recovery I/O. This attribute only applies to ESS (Enterprise Storage) disks.
ReviveBlocks	The number of revive blocks of disk data to be copied during each copy interval when the disk is being revived.
ReviveInterval	The interval of time, in 10-millisecond units, between each copy operation and the next when the disk is being revived
RevivePriority	The priority of a revive task that executes in the DP2 IOP. See also REVIVEPRIORITY.
ReviveRate	The amount of data to be revived between preemption checks. See also REVIVERATE.
SerialWrites	(Mirrored volumes only) specifies whether serial writes are used when updating files. ENABLED minimizes the potential data loss caused by certain processor error conditions. DISABLED maximizes the performance of mirrored volumes. When ENABLED, the IOP performs serial write operations when the mirror disk is up. If the mirror disk is not up and if the FullCheckpoints value is FORCED or ENABLED, full-block checkpoints are used to protect the validity of the data. See also FULLCHECKPOINTS.
WriteCache	Shows whether write caching is enabled.

Path Information (for each path)	
Adapter	The adapter type and physical location (group, module, slot) of the storage adapter that controls the disk. This value is retrieved from the system.
Disk Device ID / Port Name	For an internal disk, the SCSI address. For a model 45*xx* disk connected to an adapter, the slot number of the disk in the modular disk subsystem. This is the address used by the SAC to access the disk. For an M8*xxx* disk in a Fibre Channel disk module (FCDM), the (*shelf*, *bay*) of the disk drive. For a disk in an Enterprise Storage System (ESS), the worldwide name (WWN) of the disk.
Disk Device LUN	Always shows a value of zero (0).
Location (Group, Module, Slot)	The physical location of an internal disk (group, module, slot). For a model 45*xx* disk in a modular disk subsystem or a model M8xx disk in a FCDM, Location is the location of the storage adapter that controls the disk.
SAC Name	The name of the ServerNet addressable controller in the adapter. The name is determined by concatenating the adapter type, the physical location of the SAC in the adapter, and the adapter location values. Slots 50, 51, and 53 are on the X fabric. Slots 52, 54, and 55 are on the Y fabric.
SAC Number	The physical location of the SAC in the adapter. For an internal disk, this value is either 1 or 2. This value determines the SCSI bus used to access the internal disk. For devices connected to an adapter, a value of 1, 2, 3, or 4 depending on how many SACs are installed in the adapter.

Displaying Disk Label Information

```
-> INFO $SYSTEM, LABEL

Disk Label Information $SYSTEM:
 Disk Label Information $SYSTEM Primary:
 *Volume Name......................... $SYSTEM
 *Alternate Volume Name............... $SYSTEM
  Volume ID........................... 257
```

```
 Volume Label Verifier................ D2
 Volume Label Version................. 0
 Operating System Last Mounted Under.. G06 (Q06)
 Volume Label Last Written............ 16 May 2001, 11:21:36.583
 Disk Subtype......................... 42

     Section Name             Address       Length     Version   Parameter
     -------------------- ------------- ------------- ------- -------------
   0. Spare Tracks Table   000000.003000 000000.005000       0 000000.000000
   1. Boot                 000106.024000 000000.020000       0 000000.000000
   2. Free Space Table     002054.014000 000001.020000       0 000000.000000
   3. Directory Label      000000.110000 000000.010000       0 000000.000000
   4. Unused               000000.000000 000000.000000       0 000000.000000
   5. Undo                 000000.120000 000000.020000       0 000000.000000
   6. Pool                 000000.000000 000000.000000       0 000000.000000
  10. Pool-LSA             000000.000010 000407.061554       3 000000.000000
  11. Free Space Table-LSA 000002.013014 000000.000220       3 000000.000000
  12. Boot-LSA             000000.021424 000000.000020       0 000000.000000

Disk Label Information $SYSTEM Mirror:
*Volume Name.......................... $SYSTEM
*Alternate Volume Name................ $SYSTEM
 Volume ID............................ 257
 Volume Label Verifier................ D2
 Volume Label Version................. 0
 Operating System Last Mounted Under.. G06 (Q06)
 Volume Label Last Written............ 16 May 2001, 11:21:36.583
 Disk Subtype......................... 42

     Section Name             Address       Length     Version   Parameter
     -------------------- ------------- ------------- ------- -------------
   0. Spare Tracks Table   000000.003000 000000.005000       0 000000.000000
   1. Boot                 000106.024000 000000.020000       0 000000.000000
   2. Free Space Table     002054.014000 000001.020000       0 000000.000000
   3. Directory Label      000000.110000 000000.010000       0 000000.000000
   4. Unused               000000.000000 000000.000000       0 000000.000000
   5. Undo                 000000.120000 000000.020000       0 000000.000000
   6. Pool                 000000.000000 000000.000000       0 000000.000000
  10. Pool-LSA             000000.000010 000407.061554       3 000000.000000
  11. Free Space Table-LSA 000002.013014 000000.000220       3 000000.000000
  12. Boot-LSA             000000.021424 000000.000020       0 000000.000000
```

Explanation of Fields	
*	An attribute whose value you can change by using the ALTER DISK command.
Volume Name	The default disk name. To change this name, use the VOLNAME attribute of the “ALTER DISK Command” (page 228). To change the volume name and alternate volume name simultaneously, use the “RENAME DISK Command” (page 269).
Alternate Volume Name	The alternate volume name of the specified disk volume. To change this name, use the ALTNAME attribute of the “ALTER DISK Command” (page 228). To change the volume name and alternate volume name simultaneously, use the “RENAME DISK Command” (page 269) If this name is not the default volume name, and if the default volume name is currently being used by another device, the system starts the disk using the alternate name.
Volume ID	A value that is incremented each time the volume is started. System software uses this ID when performing program-file fixes.
Volume Label Verifier	Set to D2 for disks.
Volume Label Version	Always 0 for disks.
Operating System Last Mounted Under	The operating system version and (in parentheses) the TOS version that was running when the volume was last mounted.

Explanation of Fields	
Volume Label Last Written	The date and time when the volume label was last written to the volume.
Disk Subtype	The subtype of the disk on which the specified volume is located. This value is retrieved from the device.
Section Name table	The section name, address, length, version, and parameter for each section that resides on the disk volume label.

Displaying Profile Attributes

The "INFO PROFILE Command" (page 258) displays configured information about a disk profile.

Example of an INFO PROFILE Report

```
-> INFO PROFILE $ZZSTO.#INTERNAL-DISK

STORAGE - Detailed Information PROFILE \COMM.$ZZSTO.INTERNAL-DISK
*AuditTrailBuffer/SQLMXBuffer (MB)..... 0
*AutoLabel ............................ OFF
*AutoRevive............................ OFF
*AutoSelect............................ n/a
*AutoStart............................. ON
*CapacityMismatch...................... OFF
*CBPoolLen............................. 1000
*FastBulkWrite......................... ON
*FSTCaching............................ OFF
*FullCheckpoints....................... ENABLED
*HaltOnError........................... 1
*HighPin...............................ON
*LKIDLongPoolLen....................... 8
*LKTableSpaceLen....................... 15
*MaxLocksPerOCB........................ 5000
*MaxLocksPerTCB........................ 5000
*Mirroring............................. ON
*NameMask.............................. STANDARD
*NonAuditedInsert...................... OFF
*NumDiskProcesses...................... 6
*OSSCaching............................ ON
*PhysvolSelect......................... OFF
*Pool..................................
*Program............................... $SYSTEM.SYSTEM.TSYSDP2
*ProtectDirectory...................... SERIAL
*RecoveryTimeout....................... 30
*ReviveBlocks.......................... 1
*ReviveInterval........................ 100
*RevivePriority........................ 0
*ReviveRate............................ 0
*SerialWrites.......................... ENABLED
*StartState............................ STARTED
*WriteCache............................ DISABLED
```

Explanation of Fields — INFO PROFILE Report	
*	Indicates an attribute whose value you can change by using an ALTER PROFILE command.
AuditTrailBuffer/ SQLMXBuffer (MB)	Shows one of: • The audit-trail buffer size in megabytes for a TMF audit trail volume. This value is used to improve the performance of systems using the Remote Duplicate Database Facility. • The SQLMXBuffer attribute specifies the buffer size (in megabytes) for an SQL/MX session in DP2.

AutoLabel	Shows whether the storage subsystem manager should automatically write a label on an unlabeled disk that is inserted into a slot.
AutoRevive	Shows whether the storage subsystem manager automatically starts a revive operation on an internal SCSI or M8xxx disk that is inserted into a slot.
AutoSelect	(Physical disks only) shows whether a virtual disk process is allowed to automatically consider this physical volume when making file placement decisions.
AutoStart	(Internal SCSI and M8xxx disks only) specifies whether the disk process is automatically started when inserted.
CapacityMismatch	Specifies whether the source drive remains up or goes harddown when a revive completes for a mirrored volume consisting of drives of different capacities.
CBPoolLen	The maximum memory (in 128-KB units) that can be allocated for open-related data structures on the disk. The memory available for these structures limits the total concurrent opens allowed on the disk.
FastBulkWrite	Shows whether Fast Cache Bulk Writes are turned on. When ON, applications and utilities using bulk writes to unstructured files may have higher throughput. Using FASTBULKWRITE ON can result in lost data in unstructured files if the CPU running the primary disk process fails. This attribute affects all 512-byte-sector disks.
FSTCaching	Shows whether the free space table for a disk is updated in memory (ON) or on disk (OFF). When ON, FSTCaching can increase performance. The FST is always rebuilt from disk when the disk is started. The system disk always runs with FSTCACHING ON, regardless of the configured value.
FullCheckPoints	Shows whether the validity of data written to the disk is being protected by a full-block checkpoint (applies only to nondirectory structured files). Options are ENABLED (the default), DISABLED, or FORCED. For mirrored volumes, see also SERIALWRITES.
HaltOnError	Shows whether an internally detected disk-process failure should force a halt (code%11500) in the primary processor, backup processor, or both.
HighPin	The PIN range available to the disk process.
LKIDLongPoolLen	The memory (in 128-KB units) to be allocated for lock key space. This space is used to store keys larger than 16 bytes when locking records in key-sequenced files.
LKTableSpaceLen	The memory (in 128-kilobyte units) to be allocated for lock-related data structures. The memory available for these structures limits the total file and record locks allowed on the disk.
MaxLocksPerOCB	The maximum number of records which can be locked outside a transaction.
MaxLocksPerTCB	The maximum number of records and files a transaction can lock.
Mirroring	Shows whether the storage subsystem manager process should assign a mirror disk for an inserted internal disk.
NameMask	The naming mask used when auto-assigning a process name for a disk volume. The default value (STANDARD) is the factory naming convention.
NumDiskProcesses	The number of disk processes allocated for the disk.
OSSCaching	Shows whether caching for Open System Services (OSS) files is enabled.
PhysvolSelect	(Physical disks only) shows whether a virtual disk process is allowed to consider the physical volume for file placement.
Pool	The storage pool process with which the disk volume is associated.
Program	Shows the object file name of the disk process.

ProtectDirectory	The type of protection to use for the disk volume directory: full-block checking (CHECKPOINT), serial writes (SERIAL), or disabled (OFF).
RecoveryTimeout	The number of seconds that the NonStop storage controller must wait for the device to respond to a recovery I/O. This attribute only applies to ESS (Enterprise Storage) disks.
ReviveBlocks	The number of revive blocks of disk data to be copied during each copy interval when the disk is being revived.
ReviveInterval	The interval of time, in 10-millisecond units, between each copy operation and the next when the disk is being revived
RevivePriority	The priority of a revive task that executes in the DP2 IOP. See also REVIVEPRIORITY.
ReviveRate	The amount of data to be revived between preemption checks. See also REVIVERATE.
SerialWrites	(Mirrored volumes only) specifies whether serial writes are used when updating files. ENABLED minimizes the potential data loss caused by certain processor error conditions. DISABLED maximizes the performance of mirrored volumes. When ENABLED, the IOP performs serial write operations when the mirror disk is up. If the mirror disk is not up and if the FullCheckpoints value is FORCED or ENABLED, full-block checkpoints are used to protect the validity of the data. See also FULLCHECKPOINTS.
StartState	Shows whether the disk process is enabled (STARTED) or disabled (STOPPED) when the disk process is launched.
WriteCache	Shows whether write caching is enabled.

Displaying Information about Partitioned Disks

Effective with the H06.23/J06.12 RVU, Hard Disk Drives (HDDs) and Solid State Drives (SDDs) in Serial Attached SCSI (SAS) enclosures connected to CLIMs can be partitioned. SCF and OSM provides serviceability and manageability for SAS disk partitioning. See "Partitioning HDDs and SSDs" (page 92) in Chapter 6 for more information.

Use the "INFO PARTITION Command" (page 256) to display the disk partition information for a physical disk by providing the primary and backup CLIMs and the LUNs that correspond to that physical disk. The INFO DISK, CONFIG command provides the CLIMs and LUNs for the primary and mirror disks.

Here is an example of the `INFO PARTITION $ZZSTO` command:

```
->INFO PARTITION $ZZSTO, PRIMARYCLIM S1002533, PRIMARYLUN 101,
BACKUPCLIM S1002531

STORAGE - LUN PARTITION Info

PRIMARY CLIM \NBSTS01.$ZZSTO.#S1002533

 Configured Partitions: LUN 101
   Name      Partition     Size (GB)
   $DATA00-P 1           20
   $SWAP00-M 2           10
   None      3          30
   None      4          40

BACKUP CLIM \NBSTS01.$ZZSTO.#S1002531

 Configured Partitions: LUN 101
   Name      Partition     Size (GB)
   $DATA00-B 1           20
   $SWAP00-MB 2            10
```

```
   None     3          30
   None     4          40

LUN Usage Information:
   Unpartitioned Size (GB)... 40
   Total Size (GB)........... 140
```

Use the `INFO DISK $disk-name, [CONFIG | DETAIL]` command to display the partition number (if it exists) for each path. If the partition number is not displayed, the disk LDEV path is configured to a disk that is not partitioned.

Here is an example of the INFO DISK, CONFIG command:

```
> INFO DISK $SSD1, CONFIG

STORAGE - Detailed Info Magnetic DISK \NBSTS01.$SSD1
Common Disk Configuration Information:
*BackupCpu............................ 1
*HighPin.............................. ON
*PrimaryCpu........................... 0
*Program.............................. $SYSTEM.SYSTEM.TSYSDP2
*StartState........................... STARTED

Disk Type Specific Information:
*AuditTrailBuffer/SQLMXBuffer (MB)..... 0
*AutoRevive........................... OFF
*AutoSelect........................... n/a
*AutoStart............................ ON
*CapacityMismatch..................... OFF
*CBPoolLen............................ 1000
*FastBulkWrite........................ OFF
*FSTCaching........................... OFF
*FullCheckpoints...................... ENABLED
*HaltOnError.......................... 1
*LKIDLongPoolLen...................... 8
*LKTableSpaceLen...................... 15
*MaxLocksPerOCB....................... 5000
*MaxLocksPerTCB....................... 5000
*NonAuditedInsert..................... OFF
*NumDiskProcesses..................... 4
*OSSCaching........................... ON
*PhysvolSelect........................ n/a
*Pool................................. None
*ProtectDirectory..................... SERIAL
*RecoveryTimeout...................... 0
*ReviveBlocks......................... 10
*ReviveInterval....................... 100
*RevivePriority....................... 0
*ReviveRate........................... 0
*SerialWrites......................... ENABLED
*WriteCache........................... DISABLED

Primary Path Information:
 CLIM................................. S1002531
 LUN.................................. 101
 PARTITION............................ 2

Backup Path Information:
 CLIM................................. S1002533
 LUN.................................. 101
 PARTITION............................ 2

Mirror Path Information:
 CLIM................................. S1002533
```

```
 LUN................................. 201
 PARTITION........................... 2

Mirror Backup Path Information:
 CLIM................................ S1002531
 LUN................................. 201
 PARTITION........................... 2
```

Use the `INFO CLIM $ZZSTO.`*`clim-name`*`, DETAIL` command to display the partition number (if it exists) for each path. A disk LDEV path that is not configured to a partition will not display a partition number.

```
> INFO CLIM $ZZSTO.S1002533,DETAIL

STORAGE - Detailed Info CLIM \NBSTS01.$ZZSTO.#S1002533

 Configured Devices:
   Type  Name     Primary CPU Backup CPU Lun  Partition
   DISK  $SSD1-P  0           1          101   1
   DISK  $SSD2-B  2           3          101   2
   DISK  $SSD3-P  1           3          101   3
   DISK  $SSD4-P  1           3          202
   DISK  $SAS1-P  2           3          208
   DISK  $SAS2-B  0           1          102
```

Use the `STATUS CLIM $ZZSTO.`*`clim-name`*`, ENCRYPTION` command to display the partition number in addition to the lun. A disk LDEV path that is not configured to a partition will not display a partition number.

```
> STATUS CLIM $ZZSTO.C100271, ENCRYPTION
STORAGE - Encryption Status CLIM \JUNO1.$ZZSTO.#C100271
 Encrypted Devices:
   Name             Lun   Partition
   $DATA00-P        100     1
   $SWAP00-MB       120     2
   $G6D103-MB       203
```

Displaying Information about Encrypted Disks

Display information about a four-path disk with a change currently in progress on the primary path:

```
STORAGE - Status DISK $DATA03, ENCRYPTION
Primary path
  KeyName.......... VGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG_N_YYYYMMDDHHMMSS,
                    VGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG_N_YYYYMMDDHHMMSS
  KeyAlgorithm..... XTS-AES
  KeySize.......... 256
  ChangeStatus..... In progress at %H00020900 of %H0433DBE8 (10%)
  EncryptRate...... 50
  EncryptPriority.. 4

Backup path
  KeyName.......... VGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG_N_YYYYMMDDHHMMSS,
                    VGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG_N_YYYYMMDDHHMMSS
  KeyAlgorithm..... XTS-AES
  KeySize.......... 256
  ChangeStatus..... In progress on other CLIM

Mirror path
  KeyName.......... VGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG_N_YYYYMMDDHHMMSS,
                    VGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG_N_YYYYMMDDHHMMSS
  KeyAlgorithm..... XTS-AES
  KeySize.......... 256
```

```
  ChangeStatus..... No change in progress

Mirror Backup path
  KeyName.......... VGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG_N_YYYYMMDDHHMMSS,
                    VGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG_N_YYYYMMDDHHMMSS
  KeyAlgorithm..... XTS-AES
  KeySize.......... 256
  ChangeStatus..... No change in progress
```

Display information for a one-path disk with a change aborted by I/O error:

```
STORAGE - Status DISK $DATA04, ENCRYPTION
Primary path
  KeyName.......... VGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG_N_YYYYMMDDHHSS,
                    VGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG_N_YYYYMMDDHHSS
  KeyAlgorithm..... XTS-AES
  KeySize.......... 256
  ChangeStatus..... I/O error during key change - disk down
```

Display the status of a specified path on a non-encrypted disk:

```
STORAGE - Status DISK $DATA05-B, ENCRYPTION
Backup path
  Not encrypted
  ChangeStatus..... No change in progress
```

Display the status of a specified path on non-CLIM disk:

```
STORAGE - Status DISK $DATA06-M, ENCRYPTION
Mirror path
  Not encrypted (non-CLIM)
```

Explanation of Fields — STATUS DISK, ENCRYPTION Report

KeyName	The name of the encryption key in use for the media.
KeyAlgorithm	The key algorithm in use for the media.
KeySize	The key size.
ChangeStatus	Indicates if a key status change is in progress. Possible values are: • In progress at *location* of *location percent* • In progress on other CLIM • No change in progress
EncryptRate	If change is in progress, displays the encryption rate.
EncryptPriority	If change is in progress, displays the encryption priority.

Displaying Disk Cache Configuration Information

The "INFO DISK Command" (page 254) can display information about a disk's cache configuration.

```
1-> INFO $SYSTEM, CACHE
STORAGE - Cache Info Magnetic DISK \NBSTS01.$SYSTEM (Cn = config, Al = alloc)
Size______512_________1K_________2K_________4K________32K

Cn          4         -1        100        100         -1
Al        512        512        512       4096        512

 Bytes Allocated To Cache: 34560 KB
```

Explanation of Fields — Disk Cache Report

Size	Displays as column headers the configurable block sizes (512 bytes, 1024 bytes, 2048 bytes, 4096 bytes, and 32 K (32768 bytes)).

Cn	The number of blocks that either were requested by an ALTER DISK, CACHE configuration command or set by default for each block size. A value of -1 indicates that no default value is set for the number of blocks required.
Al	The number of blocks allocated to cache for each block size.
Bytes Allocated To Cache	The total memory allocated to cache. This value equals the number of blocks allocated for a particular block size, multiplied by the block size, and totaled for all block sizes.

Displaying Disk Cache Statistics

The "STATS DISK Command" (page 277) displays cache statistics about a disk.

```
1-> STATS $DATA00
STORAGE - Stats DISK \NBSTS01.$DATA00

Cache Statistics:
  Current Time............. 23 Jul 2013, 14:38:05.260
  Counters Reset Time...... 23 Jul 2013, 14:26:34.317
  Elapsed Time............. 0 days, 00:11:30

  Bytes Allocated To Cache. 34560 KB
  Writes/Control Point..... 0.00

Block Size___________512_________1K_________2K_________4K

Requested....         -1         -1         -1         -1
Allocated....        512        512        512       4096
Blocks In Use          0          0          0        208
Blocks Dirty.        0 %        0 %        0 %        0 %

Cache Reads..        1 %        0 %        0 %       82 %
Read Hits....       90 %        0 %      100 %       99 %
Read Misses..       10 %        0 %        0 %        1 %

Cache Writes.       99 %      100 %      100 %       18 %
Write Dirties        0 %        0 %        0 %        0 %
Write Cleans.       98 %       98 %       98 %       99 %
Write Misses.        2 %        2 %        2 %        1 %

Cache Calls..       1001        198        596       1725
Cache Faults.          0          0          0          0
Audit Forces.          0          0          0          0
```

Explanation of Fields — Disk Cache Statistics

Current Time	The date and time of this STATS DISK report.
Counters Reset Time	Shows when the system was last loaded or when the counters were reset. Use the STATS DISK, RESET command (see "STATS DISK Command" (page 277)) to reset the counters.
Elapsed Time	The total elapsed time since the counters were reset, the disk was started, or the system was loaded.
Bytes Allocated To Cache	The total memory allocated to cache (in kilobytes). This value equals the number of blocks allocated for a particular block size, multiplied by the block size, and totaled for all block sizes.
Writes/Control Point	The average number of buffers that had to be written to disk at each control point (the result of dividing the number of writes forced at a control point by the number of control points that were encountered since the counters were initialized).
Block Size	The specific block size (in bytes) for each cache block.
Requested	The default number of cache blocks that are requested.
Allocated	The number of cache blocks allocated.

Blocks In Use	The number of cache blocks containing a valid disk block in memory.
Blocks Dirty	The percentage of allocated blocks that are currently dirty (blocks in cache that have been changed but are not yet written back to disk).
Cache Reads	The percentage of cache calls that the disk process made for user read requests. (When added together, the percentages displayed for Cache Reads and Cache Writes equal 100 percent.)
Read Hits	The percentage of cache reads when the requested block was found in cache memory. (When added together, the percentages displayed for Cache Read Hits and Cache Read Misses equal 100 percent.) This value should be a high percentage, indicating that the requested blocks are frequently found in cache. If the percentage is low and you want to raise it, increase the size of cache by using the ALTER DISK, CACHE command. For more information, see "Configuring the Size of Disk Cache" (page 86).
Read Misses	The percentage of cache reads when the disk process could not find the requested block in cache and had to bring the block in from disk. A user write request does not affect the read counters even if the cache write request causes a disk read. If this percentage becomes very high, consider increasing the size of cache. For more information, see "Configuring the Size of Disk Cache" (page 86).
Cache Writes	The percentage of cache calls that the disk process made for user write requests and that resulted in cache writes. (When added together, the percentages displayed for Cache Reads and Cache Writes equal 100 percent.)
Write Dirties	The percentage of cache writes for which the block was found changed (dirty). When new data is inserted into a block and the disk process finds a dirty block, it does not have to perform a disk read but it must perform a disk write. Having a large value for Cache Write Dirties reduces the number of required physical I/O operations (that is, disk reads and disk writes).
Write Cleans	The percentage of cache writes for which the block was found unchanged in cache (not dirty). When new data is inserted into a block, the disk process does not have to perform a disk read but it must perform a disk write. This counter is also incremented when an end of file (EOF) is extended (an application appends to the end of a file).
Write Misses	The percentage of cache writes that resulted in the disk process not finding a block and having to read the block from the disk. For cache write misses, the disk process must perform both a disk read and a disk write. (When added together, the percentages displayed for Cache Write Cleans, Cache Write Misses, and Cache Write Dirties equal 100 percent.)
Cache Calls	The total number of cache reads and cache writes performed during the measurement interval. If a user request bypasses cache (for example, by specifying direct I/O), none of the cache counters are affected by that call.
Cache Faults	The number of times a cache call expected to find the block in cache but could not, due to limited memory space. This number should be very small or zero. If it is not, there is insufficient memory to satisfy the cache configuration. You must either add more memory or reconfigure cache to use memory more efficiently.
Audit Forces	The number of times an audited dirty block required an audit-trail write to disk to make room for a new block to be read from disk. (This value is the same as the AUDIT-BUF-FORCE counter in the disk report from the Measure product.)

Reconfiguring Cache to Resolve Performance Problems

Based on the statistics displayed by a STATS DISK report, you can reconfigure cache to resolve memory performance problems

Condition	**What to Do**
Cache Read Hits is low. Cache Read Misses is high.	Use the ALTER DISK, CACHE command to increase the number of cache blocks for that cache block size.
Cache Faults is large.	Use the ALTER DISK, CACHE command to reduce the number of cache blocks for that cache block size.

Condition	What to Do
Audit Forces is high (indicates insufficient cache memory and tends to take resources from other processes).	If the processor that controls the disk you are analyzing has enough available physical memory, use the ALTER DISK, CACHE command to allocate more memory to the cache (increase the number of blocks) for that cache block size.
Cache Faults is not close to zero.	Add more memory or reconfigure cache to use less memory.

Displaying Error and Current Status Information

The "STATUS DISK Command" (page 282) displays the current status of a disk.

Displaying Disk Errors

1. Run DSAP on the suspected volume.
2. DSAP identifies unspared sectors: if an SCF INFO DISK, BAD command confirms that sectors have not been spared, see "Sparing a Defective Sector" (page 104).

   DSAP identifies doubly allocated file extents. See "Correcting Doubly Allocated File Extents" (page 106).
3. Run these OSM or TSM actions to perform general tests and check for disk errors:
   - Test Verify

     CAUTION: This test can affect performance of the disk. Do not use this test when critical applications are using the disk.
   - Validate Checksum
   - Alarms

     Check these reports and resolve errors reported by OSM or TSM.
   - Use either the EMS Event Viewer to view messages from any log ($0, $ZLOG, an alternate collector), including event logs saved on the server. You can also download messages and save them in a file on the workstation.
   - If disk performance seems to be the source of problems:
   - Use DCOM to compress the free space of a disk volume to minimize file fragments. See the *Guardian Disk and Tape Utilities Reference Manual*.
   - Use the Measure product to collect performance statistics on system resources. You can use the data provided to balance work load and do capacity planning. For detailed information, see the *Measure Reference Manual*.

     If, after resolving all errors, you want to rebuild the free space table on the disk, see "Correcting Doubly Allocated File Extents" (page 106).

Displaying Bad Sector Information

If the disk is not installed or if it is in a state that prevents the system from getting the information, SCF indicates that the information is not available.

This display is an example of a disk that has no bad sectors.

```
-> INFO $SYSTEM, BAD

 Bad Sectors Information $SYSTEM Primary:
  No bad sectors found.

 Bad Sectors Information $SYSTEM Mirror:
  No bad sectors found.
```

This display indicates that a disk that has a bad sector, as recorded in the Spare Tracks Table.

-> INFO $DATA01, BAD

```
STORAGE - Bad Sector Information Magnetic DISK \GANESAN.$DATA01

 Bad Sectors Information $DATA01 Primary:

 No bad sectors found.

 Bad Sectors Information $DATA01 Mirror:

 Logical Sector Address                   Date Detected
 %H0000795C                               SEPTEMBER 22, 2000 15:25:12

 File Name                           File Address              Logical Sector
                                                               Address
\GANESAN.$DATA01.DLSYS42X.OSIMAGE    573440-573951             %H0000795C
```

To spare the bad sector, see "Sparing a Defective Sector" (page 104).

Explanation of Fields — Displaying Bad Sectors

Logical Sector Address	A list of logical addresses of all bad sectors. The address is an 8-digit hexadecimal number. A logical sector contains 512 bytes.
Date Detected	The date and time the bad sector was detected.
File Name	The file name of the file (if any) that is stored in the bad sector.
File Address	Specifies that portion of the file that lies in the bad sector. This value is the number of bytes (in decimal) from the beginning of the file to the start of the bad logical sector and from the beginning of the file to the end of the bad logical sector.

Displaying Defect Log Information

The defect log lists defective sectors that have been spared using the CONTROL DISK, SPARE command.

-> INFO $DATA01, LOG

```
Defect Log Information $DATA01 Primary:

  Lifetime defect log report on $DATA01-P 16 May 2001, 11:21:43.973

  Physical Sector Address            Source
  -----------------------            ------

  %H0029211                          Added defect map

  The defect log contains 1 defects of which 1 were displayed.

 Defect Log Information $DATA01 Mirror:

  Lifetime defect log report on $DATA01-M 16 May 2001, 11:21:44.017

  Physical Sector Address            Source
  -----------------------            ------

  %H0029829                          Added defect map
```

```
 The defect log contains 1 defects of which 1 were displayed.
```

Explanation of Fields — Defect Log Information

Physical Sector Address	List of physical sector addresses of defective sectors on the disk. The addresses are displayed as 8-digit hexadecimal numbers.
Source	Specifies the defect map that lists the defect. The Added Defect Map identifies bad sectors that were spared by the CONTROL DISK, SPARE command (see SPARE) and the $ZRD9 automatic sector reallocator ("Automatic Sector Reallocation" (page 104)), but not the manufacturer's defect log.

Examples of STATUS Reports for Disks

Example 1: Displaying the Status of a Mirrored Disk

To display the status of $SYSTEM:

```
-> STATUS $SYSTEM

STORAGE - Status DISK \COMM.$SYSTEM
LDev   Primary    Backup     Mirror     MirrorBackup    Primary   Backup
                                                        PID       PID
   6  *STARTED    STARTED   *STARTED    STARTED          0,257     1,257
```

Explanation of Fields — Status of a Mirrored Disk

*	Indicates the currently active paths to the disk.
LDev	The logical device number for the disk volume. This number is arbitrarily assigned to a device when you configure the device and every time the system is loaded.
Primary, Backup, Mirror, MirrorBackup	The current SCF state of the paths to the disk.
Primary PID Backup PID	The processor number and PIN of the primary and backup disk processes.

Example 2: Displaying the Status and State of All Paths to a Disk

To display the status and state of all paths to $SYSTEM:

```
-> STATUS $SYSTEM-*


STORAGE - Status DISK \COMM.$SYSTEM-*
LDev  Path           Status      State      Substate          Primary   Backup
                                                              PID       PID
   6  PRIMARY        ACTIVE      STARTED                       0,257     1,257
   6  BACKUP         INACTIVE    STARTED                       0,257     1,257
   6  MIRROR         ACTIVE      STARTED                       0,257     1,257
   6  MIRROR-BACKUP  INACTIVE    STARTED                       0,257     1,257
```

Explanation of Fields — Status and State of All Paths to a Disk

LDev	The logical device number for the disk volume. This number is arbitrarily assigned to a device when you configure the device and every time the system is loaded.
Path	The disk path assignment.
Status	Shows whether the disk path is the current path (ACTIVE) or not (INACTIVE).

State	The current SCF state of the disk path. For an explanation, see "Object States and Substates of Disks" (page 34).
Substate	The current SCF substate of the disk path.
Primary PID Backup PID	The processor number and PIN of the primary and backup disk processes.

Example 3: Displaying the Status of a Nonmirrored Disk

To display the status of the nonmirrored disk $DATA11:

```
-> STATUS $DATA11

STORAGE - Status DISK \COMM.$DATA11
LDev    Primary     Backup      Mirror      MirrorBackup   Primary    Backup
                                                           PID        PID
 210   *STARTED     STARTED                                 9,262      8,271
```

The explanation of fields is the same as for "Example 1: Displaying the Status of a Mirrored Disk" (page 64).

Example 4: Displaying STATUS DISK, CONSISTENCY Information

To display the consistency information about the mirrored disk volume $WENDY:

```
-> STATUS $WENDY, CONSISTENCY
```

This display shows that the system-configuration database, the two disk processes, and the two SIFM processes are consistent.

```
STORAGE - Status DISK \SYS.$WENDY, CONSISTENCY

Path Opinion (Grp,Mod,Slt) Sac  Device-id/Portname Lun

--------------------------------------------------------
-P   CONFIG  (1,1,1)        2   4                   0

-B   CONFIG  (1,1,1)        2   4                   0

-M   CONFIG  (1,1,2)        1   4                   0

-MB  CONFIG  (1,1,2)        1   4                   0
```

This display shows an inconsistency. The mirror disk for $WENDY is configured in the system-configuration database, but the SIFM tables in processor 0 do not have the mirror and mirror-backup paths configured.

```
6-> STATUS DISK $WENDY, CONSISTENCY

STORAGE - Status DISK \SYS.$WENDY, CONSISTENCY

Path Opinion (Grp,Mod,Slt) Sac Device-id/Portname Lun

-----------------------------------------------------

-P   CONFIG  (1,1,1)       2   4                  0

-B   CONFIG  (1,1,1)       2   4                  0

-M   CONFIG  (1,1,2)       1   4                  0

     SIFM-00 (0,0,0)      99   0                  0

-MB  CONFIG  (1,1,2)       1   4                  0

     SIFM-00 (0,0,0)      99   0                  0
```

For an example of the output for the STATUS DISK $PART01, CONSISTENCY command, refer to "STATUS DISK Examples For Physical Disks" (page 283).

Example of STATUS DISK Detailed Report for Disks

To display detailed status of $SYSTEM:

```
-> STATUS $SYSTEM, DETAIL

STORAGE - Detailed Status DISK \COMM.$SYSTEM

Disk Path Information:
 LDev  Path           Status       State      Substate         Primary  Backup
                                                               PID      PID
    6  PRIMARY        ACTIVE       STARTED                      0,257    1,257
    6  BACKUP         INACTIVE     STARTED                      0,257    1,257
    6  MIRROR         ACTIVE       STARTED                      0,257    1,257
    6  MIRROR-BACKUP  INACTIVE     STARTED                      0,257    1,257

General Disk Information:
  Device Type........... 3          Device Subtype........... 42
  Primary Drive Type.... 4608-1     Mirror Drive Type........ 4608-3
  Physical Record Size.. 4096       Priority................. 220
  Library File..........
  Program File.......... $SYSTEM.SYS01.OSIMAGE
  Protection............ AUDITED, MIRRORED

Usage Information:
  Capacity (MB)......... 8837.85    Free Space (MB).......... 2168.80(24.53%)
  Free Extents.......... 3369       Largest Free Extent (MB). 1425.43

Hardware Information:
  Device          Location             Power          Physical Status
                  (group,module,slot)
  PRIMARY         (1,1,11)             DUAL           PRESENT
  MIRROR          (1,1,12)             DUAL           PRESENT

Persistent Write Verify Information:
  Path           PWV Status
  PRIMARY        OFF
  BACKUP         OFF
  MIRROR         OFF
  MIRROR BACKUP  OFF
```

Explanation of Fields — STATUS DISK Detailed Report for Disks

Disk Path Information:

LDev	The logical device number for the disk volume. This number is arbitrarily assigned to a device when you configure the device and every time the system is loaded.
Path	The disk path assignments to the specified disk.
Status	Shows whether the disk path is the current path (ACTIVE) or not (INACTIVE).
State	The current SCF state of the disk path. For an explanation, see "Object States and Substates of Disks" (page 34)6.
Substate	The current SCF substate of each disk path.
Primary PID Backup PID	The processor number and PIN of the primary and backup disk processes.

General Disk Information:

Device Type	The device type. Physical disks are always type 3.
Device Subtype	The device subtype. This value varies by device model and is retrieved from the device.
Primary Drive Type	The product number of the primary disk retrieved from the device.

	The values of Primary Drive Type and Mirror Drive Type must be the same except when the disks are being upgraded to a larger capacity. For information about mismatching disk types while upgrading, see “Swapping Processors for a Disk” (page 109).
Mirror Drive Type	The product number of the mirror disk retrieved from the device.
Physical Record Size	The size of the physical records on the disk retrieved from the device.
Priority	The execution priority of the disk process.
Library File	The library file name of the disk process.
Program File	The program file name of the disk process.
Protection	the current protection status of the disk (a mirrored disk is MIRRORED, and a disk protected by the TMF product is AUDITED).

Usage Information:	
Capacity (MB)	The size of the disk in megabytes.
Free Space (MB)	The available space (in megabytes and as a percentage of the disk capacity)
Free Extents	The total number of extents available
Largest Free Extent (MB)	Rhe size, in megabytes, of the largest extent available

Hardware Information:	
Device	The path from the ServerNet addressable controller (SAC) to the disk. Values are PRIMARY and MIRROR.
Location (group, module, slot)	The location (group, module, slot) of the disk.
Power	The power supply status of the disk. Values can be: DUAL—Powered by two power supplies. SINGLE—Powered by one power supply. NONE—Does not have power.
Physical Status	The physical status of the disk. Values can be: PRESENT—disk is accessible in the location. ABSENT—disk is missing or not accessible in the location.

Persistent Write Verify Information:	
Path	The disk path.
PWV Status	The current, persistent, write-verify status for the specified disk path. When this value is ON, the SAC does additional verification to ensure that the data is written error-free. For more details, see WRITEVERIFY.

Pool Information

If the disk belongs to a pool, a STATUS, DETAIL command displays:

```
Pool Information:
  Autoselect............ ON          Pool.................... $POOL01
  Physvolselect......... ON
```

Autoselect	Shows whether Autoselect attribute of the disk is ON or OFF. When on, the physical volume is automatically considered for file placement when the placement decision is left up to the virtual disk process.
Pool	The name of the storage pool that the disk is in.
Physvolselect	Shows whether the Physvolselect attribute of the disk is set ON or OFF. When on, the virtual disk process is allowed to consider the physical volume for file placement.

Process Information

- Process information is displayed if the name of the object file for the running disk process is not the same as the name of the object file for the configured disk process. This difference indicates either that the alternate disk process was activated or that the active object file for the configured disk process was renamed.

In either case, a STATUS, DETAIL command displays:

```
Process Information:
  OSR state............. Active     OSR substate............. Active
  Installed............. 07 Jul 2001, 13:12:25.369
  Committed............. n/a
  Configured program.... $SYSTEM.SYSTEM.TSYSDP2
  Alternate program..... $SYSTEM.SYS77.TSYSTEST
  Primary executes...... $SYSTEM.SYS77.TSYSTEST
  Backup executes....... $SYSTEM.SYS77.TSYSTEST
```

OSR State	The current state of an online software replacement (OSR) operation: Abort—The alternate disk process is being replaced by the configured disk process. Active—The alternate disk process has replaced the configured disk process and is currently running. Install—The operation to replace the disk process has begun.n/a—No state information is available. Revert—The alternate disk process is being replaced by the configured disk process.
OSR substate	The current substate of the online software replacement (OSR) operation: Active—The OSR operation is complete: the alternate disk process is running as both the primary and backup process. Phase 1—The backup disk process is being replaced. Phase 2—The primary disk process is being replaced. n/a—No substate information is available. Either the replacement operation did not complete or the object file for the running disk process was renamed. Preparing—The online software replacement (OSR) operation is starting.
Installed	The date and time of the last successful online software replacement (OSR) operation for this disk process. n/a—The replacement operation did not complete or the object file for the running disk process was renamed.
Committed	The date and time when the alternate program was committed. n/a—The disk alternate disk process has not been committed, or the object file for the running disk process was renamed.
Configured program	The object-file name in the system configuration database; this is the configured disk process of this disk volume.
Alternate program	The object-file name of the alternate disk process stored in the SYS*nn*.CONFALT file.

Primary executes	The object-file name of the disk process currently running as the primary process. The OSR state and OSR substate fields indicate whether the configured disk process was replaced by the alternate disk process. If the alternate disk process replaced the configured disk process but was automatically abandoned, the configured object-file name is displayed in this field. If you rename the object-file for the disk process after it is running, the name displayed in this field could be a unique name that does not match either the name of the configured object-file name of the disk process or the object-file name of the alternate disk process.
Backup executes	The object-file name of the disk process currently running as the backup process. The detailed description for the Primary Executes also applies to this field.

Revive Information

If the disk is in the middle of a revive operation (from the "START DISK Command" (page 274)), a STATUS, DETAIL command displays:

```
Revive Information:
  Current Logical Sector............ %HFDC8 of %H7E5B71 (0%)
  Blocks Per Interval.............. 1
  Interval......................... 100 * 10 milliseconds
```

Explanation of Fields — Revive Information	
Current Logical Sector	Which logical sector is being revived. The value has the format `s of l (p %)`, where: *s*—the current sector being revived, in hexadecimal. *l*—the last sector of the disk, in hexadecimal. *p*—the percentage of disk revived so far.
Blocks Per Interval	The number of blocks copied during each interval (from the system configuration database)
Interval	The maximum interval (in 10-millisecond units) between each copy operation when the disk is being revived. The range is 1 (10 milliseconds) through 32767 (more than 5 minutes). The default value is 100 units (one second).

6 Configuring Disks

This chapter describes configuring disks. Disks share the object type of DISK with virtual disks. For information about disks and their states, see “The DISK Object” (page 33) and “Object States and Substates of Disks” (page 34). For commands that affect disks, see “Storage Subsystem Commands” (page 190). For configuration information, see “Configuring and Managing Virtual Disks” (page 143) . For information about disk load balancing, see “Disk Load Balancing” (page 118). For a list of supported disks, see the *NonStop S-Series Planning and Configuration Guide* and the *NonStop NS-Series Planning Guide*. This chapter describes:

Types of Disks

You can configure several types of disks for NonStop servers:

NOTE: The terms “disk” and “physical disk” include both magnetic and solid state disks.

Type	Description
Internal SCSI	SCSI disks that reside in slots 1 through 18 of NonStop S-series system enclosures.
45xx	Fibre Channel disks that reside in slots 0 through 7 of a modular disk subsystem outside of NonStop S-series system enclosures. The disk subsystem connects to the server through a ServerNet/DA Adapter (SNDA).

Type	Description
M8xxx	Model M8*xxx* disks that reside in slots 1 through 14 of a Fibre Channel disk module (FCDM) connected to FCSAs in an IOAM enclosure.
ESS	Disks within an Enterprise Storage System.
CLIM-attached disks	Disks that attach to the system through CLIMs, such as Serial Attached SCSI (SAS) disk drives in a SAS disk enclosure.

Configuring Paths for Different Disk Types

The values you specify when configuring a disk vary depending on the type of disk and the NonStop server model. For example, the valid ranges for group, module, and slot of an M8*xxx* disk on an Integrity NonStop NS-series server differ from those of an internal disk on a NonStop S-series server

Table 9 lists the possible attribute values for the different disk types.

Table 9 Disk Path Attributes

	LOCATION						
	GROUP	MODULE	SLOT	SAC	DEVICEID	PORTNAME	LUN
Internal SCSI Disk	1-89	1	1-18	1-2 *	0-9 *	NA	NA
45*xx* Disk	1-89	1	51-54 (SNDA)	1-4	0-7	NA	NA
M8*xxx* Disk							
NonStop S-Series	11-89	2-3 (IOAM)	1-5 (FCSA)	1-2	Shelf 1-4 Bay 1-14	NA	NA
NonStop NS-Series	100 or 110-125	2-3 (IOAM)	1-5 or 7 (FCSA)	1-2	Shelf 1-4 Bay 1-14	NA	NA
ESS Disk							
NonStop S-Series	11-89	2-3	1-5	1-2	NA	64-bit WWN	0-32767
NonStop NS-Series	100 or 110-125	2-3	1-5 or 7	1-2	N/A	64-bit WWN	0-32767
CLIM-Attached Disks	N/A	N/A	N/A	N/A	N/A	N/A	0-32767

* For Internal SCSI disks, SAC and DEVICEID are not required. SCF derives these values from the slot.LOCATION represents PRIMARYLOCATION, BACKUPLOCATION, MIRRORLOCATION, and MBACKUPLOCATION.

SAC represents PRIMARYSAC, BACKUPSAC, MIRRORSAC, and MBACKUPSAC.

DEVICEID represents PRIMARYDEVICEID, BACKUPDEVICEID, MIRRORDEVICEID, and MBACKUPDEVICEID.

PORTNAME represents PRIMARYPORTNAME, BACKUPPORTNAME, MIRRORPORTNAME, and MBACKUPPORTNAME.

LUN represents PRIMARYLUN and MIRRORLUN.

For a CLIM-attached disk, tape, or Open SCSI device, path attributes are the CLIM name and the LUN.

Considerations for M8*xxx* Fibre Channel Disks

- In an FCDM containing M8*xxx* disks, at least one disk must be added for the enclosure to be recognized by the storage subsystem.
- Before you add the first M8*xxx* disk in an enclosure to the storage subsystem, the primary and backup paths must be physically connected.
- Before you add a mirrored pair of M8*xxx* disks to the storage subsystem, both paths to the primary disk and the mirror disk must be physically connected.

- The primary disk and mirror disk of a mirrored pair of M8*xxx* disks must be in different FCDMs, connected to a different pair of Fibre Channel loops.
- On NonStop S-series servers the system disk and its mirror cannot be M8*xxx* Fibre Channel disks.
- During normal operations, a disk should be mirrored with another disk of the same type. If you revive an IOMF disk to a newer type of disk, the system will use a migratory revive that will down the source drive after the revive is complete. For information about the migratory revive operation, ask your service provider to refer to the *Modular I/O Installation and Configuration Guide*.
- SCF requires that you specify a primary and a backup path when configuring an M8*xxx* disk.
- AUTOCONFIGURE is not recommended for M8*xxx* disks. Confirm that this attribute is OFF before adding M8*xxx* disks.
- New M8*xxx* disks that are being used for the first time must be initialized after they have been added.

Considerations for ESS Disks

- The primary and mirror logical devices (LDEVs) in the ESS must be in different physical array groups.
- The primary LUN on the primary and backup paths must be the same.
- The mirror LUN on the mirror and mirror backup paths must be the same.
- For ESS disks connected to Storage CLIMs, ESS or XP LUN 0 must be defined.

For more ESS configuration information, see the *Modular I/O Installation Guide* or the NonStop planning guide for your particular system.

Considerations for CLIM-Attached Disks

- The primary and mirror disks must be in different disk enclosures. It is better if they are on different Storage CLIM pairs.

System Disk Configuration

When you configure the system disk, these restrictions exist:

- You cannot use SCF to configure the system disk. The system generation program configures the system disk. If the system disk is configured improperly, the system halts during system load, and you must use the system generation program to recover.
- You can specify an alternate system disk and perform a system load from it to create an alternate system configuration. For more information, see the *NonStop S-Series Hardware Installation and FastPath Guide*.
- For NonStop S-series systems, enclosure interleaving is not supported for the system disks. Both halves of the system disk must be in enclosure 1.
- Beginning with G06.06, on NonStop S-series systems, the system disk can reside in other slots in group 01 besides 11 and 12. For more information, refer to the *NonStop S-Series Planning and Configuration Guide*.

- On NonStop S-series servers, the system disk and its mirror cannot be M8*xxx*Fibre Channel disks.
- These attributes cannot be changed:
  - The system disk always runs with FSTCACHING ON.
  - You cannot decrease these attribute values online because the disk must be stopped to decrease them. The configured value will can decreased, but the IOP will continue to run with the original, larger value until it is stopped and restarted.
    - CBPOOLLEN
    - MAXLOCKSPEROCB
    - MAXLOCKSPERTCB
    - NUMDISKPROCESSES

Automating Disk Configuration

To automate the configuration of internal disks when you insert the disk into a slot (or, if the system is down, when the next system load occurs), use the "ALTER SUBSYS Command" (page 240) to set AUTOCONFIGURE { ON | OFF }.

Automatic configuration enables the system to automatically configure an internal disk when initially detected after a system load, processor reload, or disk insertion

NOTE: HP does not recommend using AUTOCONFIGURE with M8*xxx* Fibre Channel disks.

When the AUTOCONFIGURE attribute is set to on, the system configures the disk using the profile stored in the system configuration database. The disk profile used is $ZZSTO.INTERNAL-DISK, unless you have configured a custom profile for a specific enclosure. See "Configuring Custom Profiles" (page 75).

Plug and play not only configures a disk automatically but also automatically labels an unlabeled disk, starts the disk, and revives a mirrored disk.

Limitations of Automatic Configuration

- You cannot automatically configure model 45*xx* disks in a modular disk subsystem.
- You cannot automatically configure virtual disks.
- You cannot use AUTOCONFIGURE with enclosure interleaving if either of these is true:
  - Two internal disks of a mirrored volume are installed in separate enclosures.
  - The adapters that are connected to the two 45*xx* disks of a mirrored disk volume are installed in separate enclosures.

    For more information about enclosure interleaving, see the *NonStop S-Series Planning and Configuration Guide*.

Enabling Automatic Disk Configuration

1. View the automatic configuration attributes that are enabled for the system:

```
-> INFO SUBSYS $ZZSTO

STORAGE - Info SUBSYS $ZZSTO
 AutoConfigure   AutoRevive   Autostart   BulkIO   LabelTape   UPS
 OFF             OFF          OFF         ON       ON          OFF
```

2. To automate configuration for internal disks:

```
-> ALTER $ZZSTO, AUTOCONFIG ON
```

The storage subsystem looks for a custom profile (see “Creating a Custom Profile” (page 75)) for the enclosure. If no custom profile exists, the storage subsystem uses the default profile (see “Example of an INFO PROFILE Report” (page 54)) to configure the disk.

To make internal disk configuration plug and play, enable these attributes:

```
-> ALTER $ZZSTO, AUTOCONFIG ON, AUTOREVIVE ON, AUTOSTART ON
```

3. After configuring the disk, the storage subsystem checks the disk label:
   - If the disk is not labeled and AUTOLABEL for $ZZSTO is ON, it labels the disk.
   - If the disk is not labeled and AUTOLABEL for $ZZSTO is OFF, see “Naming a Disk” (page 88).
4. If AUTOSTART for $ZZSTO and the disk is ON, it starts the disk process. If AUTOSTART is OFF for either $ZZSTO or the disk, see “Starting a Disk” (page 97).
5. If the disk is mirrored and all of these statements are true, a revive is started from the already installed disk to the just inserted disk:
   - AUTOREVIVE for $ZZSTO is ON.
   - The slot is already configured as the mirror of an installed disk.
   - Both disks are physically present.

Configuring Internal Disks to Start Automatically

NOTE: HP does not recommend using AUTOSTART with M8*xxx* Fibre Channel disks.

The AUTOSTART attribute affects the start behavior of internal disks differently depending on the object type specified:

- AUTOSTART for the SUBSYS object type:

  ```
  -> ALTER SUBSYS $ZZSTO, AUTOSTART ON
  ```

  For the storage subsystem, the AUTOSTART attribute controls whether disks are automatically started after disk insertion. AUTOSTART for the SUBSYS overrides AUTOSTART for the PROFILE.

- AUTOSTART for the DISK object type:

  ```
  -> ALTER DISK $DISK1, AUTOSTART ON
  ```

  For an individual disk, the AUTOSTART attribute controls whether a disk is automatically started after system load, processor reload, or disk insertion.

  These attributes interact as follows:

	DISK AUTOSTART ON	DISK AUTOSTART OFF
SUBSYS AUTOSTART ON	Disk automatically starts after: ◦ System load **Yes** ◦ Disk insertion **Yes**	Disk automatically starts after: ◦ System load **No** ◦ Disk insertion **No**
SUBSYS AUTOSTART OFF	Disk automatically starts after: ◦ System load **Yes** ◦ Disk insertion **No**	Disk automatically starts after: ◦ System load **No** ◦ Disk insertion **No**

You might want to use this configuration scheme:

- Configure your system to have disks start automatically (by setting AUTOSTART ON for the subsystem).
- Make exceptions where you want some disks to not start automatically (by setting those disks to AUTOSTART OFF).

Configuring Custom Profiles

You can automatically configure internal disks by using a custom or default set of attributes, called a profile..

This set of attributes has the object type of PROFILE.

- "The PROFILE Object" (page 36)
- "Storage Subsystem Commands" (page 190)
- "Creating a Custom Profile" (page 75)
- "Adding a Similar Disk to Another System" (page 84)
- "Altering a Profile" (page 76)
- "Deleting a Custom Profile" (page 76)

Creating a Custom Profile

The "ADD PROFILE Command" (page 221) adds a custom profile to the system configuration database. See "PROFILE Attributes" (page 222).

Creating a custom profile enables you to speed the automatic configuration of internal disks or customize the attributes of internal disks in a specific enclosure:

1. Using the default INFO PROFILE display (see "Example of an INFO PROFILE Report" (page 54)), identify the attributes you want to change from the default profile.
2. Configure a custom profile for a specific enclosure, specifying the attributes you want to differ from the default profile. For example, if you want all the internal disks in enclosure 03 to be plug and play (automatically labeled and revived upon insertion):

```
-> ADD PROFILE $ZZSTO.INTERNAL-DISK-3, AUTOLABEL ON, &
-> AUTOREVIVE ON
```

Adding a Similar Custom Profile Using the OBEYFORM Attribute

To configure the same or a similar profile, create a command file by using the OBEYFORM attribute of the INFO DISK command. You can copy this file to another system or add it to different configuration file on the current system.

This sequence of actions changes the default attributes for internal disks in enclosure 3:

1. Capture the existing default profile:

```
-> INFO / OUT LOG / PROFILE $ZZSTO.#INTERNAL-DISK, OBEYFORM

== STORAGE - Obeyform Information PROFILE \COMM.$ZZSTO.INTERNAL-DISK
ADD PROFILE \COMM.$ZZSTO.INTERNAL-DISK , &
    AUDITTRAILBUFFER 0 , &
    AUTOLABEL OFF , &
    AUTOREVIVE OFF , &
    AUTOSELECT OFF , &
    AUTOSTART ON , &
    CBPOOLLEN 1000 , &
    FSTCACHING OFF , &
    FULLCHECKPOINTS ENABLED , &
    HALTONERROR 1 , &
    HIGHPIN ON , &
    LKIDLONGPOOLLEN 8 , &
    LKTABLESPACELEN 15 , &
    MAXLOCKSPEROCB 5000 , &
    MAXLOCKSPERTCB 5000 , &
    MIRRORING ON , &
    NAMEMASK * , &
    NUMDISKPROCESSES 4 , &
    OSSCACHING ON , &
    PHYSVOLSELECT OFF , &
    POOL  , &
    PROGRAM $SYSTEM.SYSTEM.TSYSDP2 , &
    PROTECTDIRECTORY SERIAL , &
    RECOVERYTIMEOUT 0 , &
    REVIVEBLOCKS 0 , &
    REVIVEINTERVAL 0 , &
    REVIVEPRIORITY 0 , &
    REVIVERATE 0 , &
    SERIALWRITES ENABLED, &
    STARTSTATE STARTED
```

2. Edit the resulting log file to change attributes as desired for a specific enclosure.
3. Edit the first line of the log file to add the enclosure number for this profile:

```
ADD PROFILE \COMM.$ZZSTO.INTERNAL-DISK-3 , &
```

4. Enter the edited log file contents either by copying and pasting into an SCF command line or by using the log file as a command file.

Altering a Profile

The "ALTER PROFILE Command" (page 236) alters a custom or default profile in the system configuration database.

Examples

- Modify the standard default profile for all internal disks in the system:

```
-> ALTER PROFILE $ZZSTO.#INTERNAL-DISK, NAMEMASK DATA
```

- Change a profile for all disks inserted into group 03:

```
-> ALTER PROFILE $ZZSTO.#INTERNAL-DISK-3, MIRRORING OFF
```

Deleting a Custom Profile

The "DELETE PROFILE Command" (page 251) removes a custom profile from the system configuration database.

Considerations for DELETE PROFILE

- You cannot delete the standard profile $ZZSTO.#INTERNAL-DISK.
- After you delete a custom profile, autoconfiguration uses the default profile.

Examples

To delete the custom profile for all disks inserted into group 01:

```
-> DELETE PROFILE $ZZSTO.#INTERNAL-DISK-1
```

Mirrored Disk Placement

Mirroring Without NonStop S-Series Enclosure Interleaving

For proper fault tolerance, always install the two disks of a mirrored volume on different SCSI buses:

- For internal disks in a NonStop S-series system enclosure, if you install the primary disk in an odd-numbered slot, install the mirror disk in an even-numbered slot.
- For model 45*xx* disks, configure the primary disk and mirror disk of a mirrored volume in different disk modules.
- Mirroring With NonStop S-Series Enclosure Interleaving

NOTE: Enclosure interleaving does not apply to Integrity NonStop NS-series systems.

Enclosure interleaving improves the fault tolerance of your NonStop S-series system by allowing you to install the disks of a mirrored volume (or configure the adapters for the two disks of a mirrored volume) in separate enclosures within the same topology branch. Configuring paths across multiple enclosures increases availability, because no enclosure is a single point of failure preventing access to the storage device.

For internal disks, install the halves of a mirrored volume in separate enclosures. The disks that make up a mirrored volume need not be in an even-numbered and odd-numbered slot. The -P and -B paths must be in one enclosure. The -M and -MB paths are in another enclosure in the same topology branch.

For model 45*xx* disks, install their adapters in separate enclosures.

For more information about enclosure interleaving, see the NonStop S-Series Planning and Configuration Guide.

Mirroring With M8*xxx* Fibre Channel Disks

For model M8*xxx* disks, configure the primary disk and mirror disk of a mirrored volume in different FCDMs, connected to a different pair of Fibre Channel loops.

If FCDMs are daisy-chained, the primary disk and mirror disk of a mirrored volume must be in different sets of daisy-chained enclosures.

Mirroring With CLIM-Attached Disks

The ADD DISK, MIRRORCLIM attribute enables you to mirror a CLIM-attached disk, and the MIRRORPARTITION attribute enables you to mirror the partition of a SAS HDD or SSD. Primary and mirror disks cannot both be on the same disk enclosure.

You can configure the DP2 processes in any available NonStop CPU pairs. A single disk can have the DP2 processes (disk LDEVs) associated with its individual partitions configured in different NonStop CPU pairs.

For data migration purposes only, you can mirror a SAS SSD with a different type of disk.

For a partition capacity upgrade, you can temporarily mirror partitions of different sizes. If it is desired to keep both mirrored disks of different sizes up, the SCF CAPACITYMISMATCH option must be turned ON for the LDEV. Normally mirroring partitions of different sizes is used temporarily as part of an online disk capacity upgrade operation.

To simplify disk replacement, a mirrored disk partition must be on two physical disks that are partitioned in the same configuration. During disk replacement, the ADD PARTITION, LIKE command can be used to configure the replacement disk to be like the mirror disk.

$SYSTEM cannot be configured (via OSM Low-Level Link) to a disk partition. Only an unpartitioned disk pair can be used as $SYSTEM. Any other NonStop disk LDEV can be configured to a disk partition.

Configuring Mirrored Disks

The mirroring examples in this section are based on internal SCSI disks. If you are mirroring model 45*xx* or M8*xxx* disks, you also need to specify PRIMARYDEVICEID, BACKUPDEVICEID, MIRRORDEVICEID, and MBACKUPDEVICEID attributes.

For considerations for the type of disk you are mirroring, see “Mirrored Disk Placement” (page 77) and “Considerations for M8*xxx* Fibre Channel Disks” (page 71)

Configuring Internal Mirrored Disks to Revive Automatically

The AUTOREVIVE attribute affects the start behavior of internal mirrored disks differently depending on the object type specified:

- AUTOREVIVE for the SUBSYS object type:

  ```
  -> ALTER SUBSYS $ZZSTO, AUTOREVIVE ON
  ```

  If AUTOREVIVE is ON for the storage subsystem, automatic revive operations are allowed. If AUTOREVIVE is OFF for the storage susbsytem, automatic revive operations are not allowed.

  AUTOREVIVE for the SUBSYS overrides AUTOREVIVE for the PROFILE.

- AUTOREVIVE for the DISK object type:

  ```
  -> ALTER DISK $DATA2, AUTOREVIVE ON
  ```

  For an individual disk, this attribute controls whether a disk is automatically revived after insertion or at system load or processor reload. AUTOREVIVE for disk only occurs if AUTOREVIVE for SUBSYS is also turned ON

These attributes interact as follows:

	DISK AUTOREVIVE ON	**DISK AUTOREVIVE OFF**
SUBSYS AUTOREVIVE ON	Autorevive after: • System load **Yes** • Disk insertion **Yes**	Autorevive after: • System load **No** • Disk insertion **No**
SUBSYS AUTOREVIVE OFF	Autorevive after: • System load **No** • Disk insertion **No**	Autorevive after: • System load **No** • Disk insertion **No**

You might want to use this configuration scheme:

- Configure your system to have disks revive automatically (by setting AUTOREVIVE ON for the subsystem).
- Make exceptions where you want some disks to not revive automatically (by setting those disks to AUTOREVIVE OFF).

Changing a Mirrored Volume Into Two Nonmirrored Disks

This procedure converts a mirrored, internal volume into two nonmirrored disks without destroying files on either disk.

1. Review the "Considerations for RENAME DISK" (page 89).
2. Stop both halves of the mirrored volume:
   ```
   -> STOP $DATA01
   ```
3. Delete the mirror from the system configuration database:
   ```
   -> DELETE $DATA01-M
   ```
4. Restart the disk process:
   ```
   -> START $DATA01
   ```
5. Add the deleted mirror to the system configuration database as a nonmirrored disk by specifying a new name and its location:
   ```
   -> ADD $SPARE00, SENDTO STORAGE, PRIMARYLOCATION (1,1,2)
   ```
6. Start the disk process in the SERVICING state, substate SPECIAL in preparation for changing its name:
   ```
   -> START $SPARE00, SPECIAL
   ```
7. Change the disk name on the disk label to make the configuration record equal the label, and restart the disk:
   ```
   -> RENAME $SPARE00, $SPARE00
   ```

Changing Two Nonmirrored Disks Into a Mirrored Volume

This example converts two adjacent, internal disks into a mirrored, internal volume. The nonmirrored disk $DATA02 is in group 01, module 1, slot 3. The disk to become the mirror disk is in slot 4 in the same enclosure. Automatic configuration is off.

1. Review information about the MIRRORLOCATION attribute.
2. Stop the disk:
   ```
   -> STOP $DATA02
   ```
3. Identify the unconfigured disk in slot 4 as the mirrored volume of $DATA02 in slot 3:
   ```
   -> ALTER $DATA02, MIRRORLOCATION (1,1,4)
   ```
4. Start the primary half of the mirrored volume in slot 3:
   ```
   -> START $DATA02-P
   ```
5. Revive the mirrored half in slot 4:
   ```
   -> START $DATA02-M
   ```

Remirroring Disks Online

You can perform these tasks without stopping the entire disk volume:

- Change the disk configuration from an unmirrored configuration to a mirrored configuration
- Relocate both the mirror disk and primary disk to new slots
- Unmirror a disk volume
- Switch the roles of the two disks of a mirror disk volume

Throughout the performance of these tasks, a primary disk drive is always available to applications. Thus, applications need not be taken out of service. For an example of disk relocation, see "Example of Online Disk Remirroring" (page 81). Before you consult this example, read these considerations.

Considerations for ALTER DISK, MIRRORLOCATION and Online Disk Remirroring

- You cannot specify the attribute MIRRORLOCATION and the attribute SWAPMIRROR in the same ALTER DISK command.
- You can issue this command when all paths to the primary disk drive are in a STARTED state. However, the paths to the mirror disk drive must be in a STOPPED state.
- Effective with the H06.23/J06.12 RVU, you can use the attribute NOSTART with the ALTER DISK command to specify that the configured mirror disk should not be started.
- “Example of Online Disk Remirroring” (page 81) deals with internal disks in a NonStop S-series system enclosure. To remirror other types of disks, you need to specify additional attributes whenever MIRRORLOCATION is specified:
  - For M8*xxx* disks in an FCDM, you must specify MIRRORLOCATION, MIRRORDEVICEID, and MIRRORSAC. For the backup path, specify MBACKUPLOCATION and MBACKUPSAC.
  - For model 45*xx* disks in a modular disk subsystem, you must specify MIRRORLOCATION, MIRRORDEVICEID, and MIRRORSAC. For the backup path, specify MBACKUPLOCATION and MBACKUPSAC.
  - For ESS disks, you must specify MIRRORLOCATION, MIRRORSAC, MIRRORPORTNAME, and MIRRORLUN. For the backup path, specify MBACKUPLOCATION and MBACKUPSAC, and MBACKUPPORTNAME.

Considerations for ALTER DISK, SWAPMIRROR, and Online Disk Remirroring

- The SWAPMIRROR attribute causes the two disk drives of a mirrored disk volume to switch roles:
  - The primary disk drive becomes the mirror disk drive.
  - The mirror disk drive becomes the primary disk drive.
- To delete the paths to a primary disk drive when you perform online disk remirroring, you must first use an ALTER DISK, SWAPMIRROR command so that the primary disk drive becomes the mirror disk drive. You cannot DELETE the paths to a disk drive that is currently serving as the primary disk drive.
- For the ALTER DISK, SWAPMIRROR command to work:
  - The disk volume must be mirrored.
  - Both disk drives must have the same number of paths; that is, both disks either have, or do not have, backup paths.
- You cannot specify both the attribute SWAPMIRROR and the attribute MIRRORLOCATION in the same ALTER DISK command.
- The SWAPMIRROR command is allowed only when the disk paths are in one of these combinations of states:

-P	-B	-M	-MB
STARTED	STARTED	STARTED	STARTED
STARTED	STOPPED	STARTED	STOPPED
STOPPED	STARTED	STOPPED	STARTED
STOPPED	STOPPED	STARTED	STARTED
STARTED	STARTED	STOPPED	STOPPED

Considerations for DELETE DISK and Online Disk Remirroring

- You must stop all paths to the mirror disk drive (-M and -MB) before you issue this DELETE command:

  ```
  ->DELETE DISK $disk-volume-name -M
  ```

- To delete a disk drive currently serving as the primary disk drive, you must first change its role by using the SWAPMIRROR command.

Example of Online Disk Remirroring

NOTE: OSM automatically recognizes online remirrored disks. TSM does not. Anytime you use TSM in conjunction with online disk remirroring, observe these rules:

- Before issuing a sequence of SCF commands that implements online disk remirroring, always stop the TSM process.
- After you have issued the commands and before you exit SCF, make sure to restart the TSM process.

For example:

- Use the SCF ABORT command to stop the TSM server:

In this example, SCF is used interactively to move both the primary disk drive and the mirror disk drive to new slots. The mirror disk drive is moved to location (1,1,2), and the primary disk drive is moved to location (1,1,1). The ALTER DISK, SWAPMIRROR command is used twice. When the task is complete, the disk drive originally serving as the primary serves as the primary once again, and the original mirror disk drive again plays the role of mirror.

The example assumes that:

- The disks are internal disks.
- The disk volume has the AUTOREVIVE attribute turned on.

To perform this remirroring task:

1. Stop the -M and -MB paths:

   ```
   ->STOP DISK $DATA1-M

   ->STOP DISK $DATA1-MB
   ```

2. Delete the mirror disk path (-M):

   ```
   ->DELETE DISK $DATA1-M
   ```

3. Issue an ALTER DISK $`disk-volume-name` command whose MIRRORLOCATION attribute assigns the mirror disk drive to location (1,1,2):

   ```
   ->ALTER $DATA1, MIRRORLOCATION (1,1,2)
   ```

   A prompt gives you the option of reviving the disk drive.

4. To change the location of the primary disk drive, you must first stop and delete it. However, SCF cannot perform such operations on a disk drive currently serving as the primary. You must therefore use the SWAPMIRROR attribute so that the primary disk drive becomes the mirror disk drive:

   ```
   -> ALTER DISK $DATA1, SWAPMIRROR
   ```

   This command converts the original mirror disk drive into the primary, thus making it available to applications.

5. After the disk drive originally serving as the primary has become the mirror, you can issue these commands:

```
-> STOP DISK $DATA1-M
-> STOP DISK $DATA1-MB
-> DELETE DISK $DATA1-M
```

NOTE: Allow 20 seconds to ensure the DELETE DISK command completes before proceeding to the ALTER DISK command in the next step. When using a TACL obey file or SCF infile, put the DELETE DISK commands in a different obey file or infile from the ALTER command.

6. Then issue an ALTER DISK $*disk-volume-name* command whose MIRRORLOCATON attribute assigns the mirror disk drive, which was once the primary disk drive, to location (1,1,1):

   ```
   -> ALTER $DATA1, MIRRORLOCATION (1,1,1)
   ```

   A prompt then gives you the option of reviving the disk drive.

   NOTE: When using SCF with an infile, SCF will start reviving if the disk is already initialized. Use the attribute NOSTART with the ALTER DISK command to specify that the configured mirror disk should not be started.

7. To return both disk drives to their original roles, use the SWAPMIRROR attribute again:

   ```
   -> ALTER DISK $DATA1, SWAPMIRROR
   ```

Adding a Disk

The “ADD DISK Command” (page 194) adds one or more disks to the system configuration database.

- You can insert the disk before or after entering the ADD command, unless you are adding the first disk in an M8*xxx* FCDM daisy chain.
- For daisy-chained M8*xxx* FCDMs, there must be at least one disk somewhere in the daisy chain, so that the storage subsystem can verify that the two paths specified in the ADD command go to the same daisy chain.
- For internal disks, if AUTOCONFIGURATION and AUTOSTART are on, the disk process automatically starts when you insert the disk. If AUTOCONFIGURATION and AUTOSTART are off, insert the disk and then enter the ADD command. For more information, see “Automating Disk Configuration” (page 73).

  NOTE: HP does not recommend AUTOCONFIGURATION for M8*xxx* disks.

- New M8*xxx* disks that are being used for the first time must be initialized after they have been added.
- When automatic configuration is enabled, internal disks are automatically added and optionally started. You do not need to enter an ADD command.

Steps to Add a Disk

Use this procedure for internal disks if autoconfiguration is not enabled.

1. For the PRIMARYLOCATION attribute, note the group, module, and slot location where you install the disk.
2. Add the disk:

   ```
   -> ADD DISK $DATA02, SENDTO STORAGE, PRIMARYLOCATION (1,1,3)
   ```

3. Verify the attributes recorded in the system configuration database:

   ```
   -> INFO $DATA02, DETAIL
   ```

4. See “Starting a Disk” (page 97).

ADD DISK Examples for Different Disk Types

- To add a mirrored internal disk:

```
-> ADD DISK $DATA01, SENDTO STORAGE, &
-> PRIMARYLOCATION (1,1,1), MIRRORLOCATION (1,1,2)
```

- To add a nonmirrored 45*xx* disk connected to a pair of 6760 adapters:

```
-> ADD DISK $DISK01, SENDTO STORAGE,&
-> PRIMARYLOCATION (1,1,53), PRIMARYSAC 1,
-> PRIMARYDEVICEID 0, BACKUPLOCATION (1,1,54), BACKUPSAC 1, &
-> BACKUPDEVICEID 0
```

- To add a mirrored M8*xxx* disk to an Integrity NonStop NS-series server:

```
-> ADD DISK $FCDISK8, SENDTO STORAGE, &
-> PRIMARYLOCATION (111,2,1), PRIMARYSAC 1, &
-> PRIMARYDEVICEID (1,8), &
-> BACKUPLOCATION (111,3,1), BACKUPSAC 1 &
-> MIRRORLOCATION (112,2,2), MIRRORSAC 1 &
-> MIRRORDEVICEID (1,4) &
-> MBACKUPLOCATION (112,3,2), MBACKUPSAC 1
```

- To add a mirrored ESS disk to an Integrity NonStop NS-series server:

```
-> ADD DISK $ESS02, SENDTO STORAGE, &
-> PRIMARYCPU 01 &
-> BACKUPCPU 02 &
-> PRIMARYLOCATION (112,2,5), &
-> PRIMARYSAC 1, &
-> BACKUPLOCATION (112,3,5), &
-> BACKUPSAC 1, &
-> MIRRORLOCATION (112,3,5), &
-> MIRRORSAC 2, &
-> MBACKUPLOCATION (112,2,5), &
-> MBACKUPSAC 2 &
-> PRIMARYPORTNAME 50060E8003501213, &
-> BACKUPPORTNAME 50060E8003501225 &
-> PRIMARYLUN 16 &
-> MIRRORPORTNAME 50060E8003501241, &
-> MBACKUPPORTNAME 50060E8003501243, &
-> MIRRORLUN 17
```

- To add a mirrored SAS disk drive, connected to a pair of CLIMs, to an Integrity NonStop NS-series server:

```
-> ADD DISK $-> ADD DISK $BLNSK, SENDTO STORAGE, &
-> PRIMARYCLIM C1002531, &
-> BACKUPCLIM C1002533, &
-> MIRRORCLIM C1002533, &
-> MBACKUPCLIM C1002531, &
-> PRIMARYLUN 101, &
-> MIRRORLUN 201, &
-> PRIMARYCPU 0, &
-> BACKUPCPU 1
```

Adding a Similar Disk to This System

To add another disk that is similar to an existing disk, use the LIKE attribute. If the disk is an internal disk, specify a unique name and the group and slot number:

```
-> ADD DISK $DATA03, SENDTO STORAGE, LIKE $DATA02,&
-> PRIMARYLOCATION (1,1,4)
```

Adding a Similar Disk to Another System

To configure the same or a similar disk on another system, create a command file by using the OBEYFORM attribute of the INFO DISK command. You can copy this file to another system or add it to a different configuration file on the current system.

1. Capture the configuration for an existing disk:

```
-> INFO / OUT LOG / $MAG1, OBEYFORM

== STORAGE - Obeyform Information Magnetic DISK \COMM.$MAG1
ADD DISK $MAG1 , &
    SENDTO STORAGE , &
    BACKUPCPU 0 , &
    HIGHPIN ON , &
    PRIMARYCPU 1 , &
    PROGRAM $SYSTEM.SYSTEM.TSYSDP2 , &
    STARTSTATE STARTED, &
    PRIMARYLOCATION (1,1,3) , &
    PRIMARYSAC $ZZSTO.#PMF.SAC-2.GRP-1.MOD-1.SLOT-50, &
    MIRRORLOCATION (1,1,4) , &
    MIRRORSAC $ZZSTO.#PMF.SAC-1.GRP-1.MOD-1.SLOT-55, &
    AUDITTRAILBUFFER 0 , &
    AUTOREVIVE OFF, &
    AUTOSTART ON, &
    CBPOOLLEN 1000 , &
    FSTCACHING ON , &
    FULLCHECKPOINTS ENABLED , &
    HALTONERROR 1 , &
    LKIDLONGPOOLLEN 8 , &
    LKTABLESPACELEN 15 , &
    MAXLOCKSPEROCB 5000 , &
    MAXLOCKSPERTCB 5000 , &
    NUMDISKPROCESSES 6 , &
    OSSCACHING OFF , &
    PROTECTDIRECTORY CHECKPOINT , &
    RECOVERYTIMEOUT 0 , &
    REVIVEBLOCKS 1 , &
    REVIVEINTERVAL 100 , &
    REVIVEPRIORITY 0 , &
    REVIVERATE 0 , &
    SERIALWRITES DISABLED, &
    WRITECACHE DISABLED
```

2. Optionally edit the resulting log file to specify:
   - A unique disk name
   - Different slot numbers for PRIMARYLOCATION and MIRRORLOCATION
   - Different SAC numbers for PRIMARYSAC and MIRRORSAC if the new locations use different SACs

     For 45xx, M8xxx, and ESS disks, PRIMARYLOCATION and MIRRORLOCATION are the storage adapter locations instead of disk locations, so the adapter location attributes do not automatically determine the SAC and DEVICEID attributes. You must make sure that every path for the new disk differs in some way (group, module, slot, SAC number, or device ID) from every other path configured to every device on the system.
3. Enter the log file contents either by copying and pasting into an SCF command line or by using the log file as a command file.

Altering Disk Attribute Values

Use the “ALTER DISK Command” (page 228) to change configured attributes for disks.

Considerations for ALTER DISK and Disks

- To change these attributes, first put the disk in the STOPPED state by using the “STOP DISK Command” (page 290):

ALTNAME	MBACKUPLOCATION	PRIMARYLOCATION
AUDITTRAILBUFFER	MBACKUPPORTNAME	PRIMARYLUN
BACKUPDEVICEID	MBACKUPSAC	PRIMARYPARTITION
BACKUPCPU	MIRRORDEVICEID	PRIMARYPORTNAME
BACKUPDEVICEID	MIRRORLOCATION	PRIMARYSAC
BACKUPLOCATION	MIRRRORLUN	PROTECTDIRECTORY
BACKUPPORTNAME	MIRRORPARTITION	PROGRAM
BACKUPSAC	MIRRORPORTNAME	SERIALWRITES
HIGHPIN	MIRRORSAC	SQLMXBUFFER
LABEL	PRIMARYCPU	VOLNAME
MBACKUPDEVICEID	PRIMARYDEVICEID	

- If a nonmirrored disk is in STARTED state, you can add a mirror drive by altering the appropriate MIRROR and MBACKUP path attributes.
- When changing these attributes, the IOP is not forced to stop:

CAPACITYMISMATCH	LKIDLONGPOOLLEN	NUMDISKPROCESSES
CBPOOLLEN	LKTABLESPACELEN	OSSCACHING
FASTBULKWRITE	MAXLOCKSPEROCB	RECOVERYTIMEOUT
FSTCACHING	MAXLOCKSPERTCB	WRITECACHE
HALTONERROR	NONAUDITEDINSERT	

- To change the VOLNAME or ALTNAME attributes, see “Naming a Disk” (page 88).
- When specifying attributes to add a physical volume to a storage pool:
  - The pool process and CATALOGLOCATION volume must be in the STARTED state.
  - The CATALOGLOCATION volume must be protected by TMF.
  - When using the AUTOSELECT and PHYSVOLSELECT attributes:
  - The pool process and CATALOGLOCATION volume must both be in the STARTED state.
  - The CATALOGLOCATION volume must be enabled by TMF.
  - The physical volume must be in the STARTED state.

Changing the Values of the Attributes for Disks

1. Display detailed information about the configured attributes of the disk you want to alter:

   ```
   -> INFO $DATA00, DETAIL
   ```

   Displayed fields are defined under “Example 2: A Detailed INFO Report for a Disk” (page 49). An asterisk (*) indicates the attributes you can alter.

2. Change one or more of the "Disk Attributes for the ALTER DISK Command" (page 230). For example, this command changes the revive attributes for the disk:

   ```
   -> ALTER $DATA00, REVIVERATE 10, REVIVEPRIORITY 100
   ```

3. Verify the change is entered into the system configuration database:

   ```
   -> INFO $DATA00, DETAIL
   ```

4. See "Resetting a Disk" (page 103).

Configuring the Size of Disk Cache

The CACHE attribute specifies the disk cache configuration for an in-use volume.

CAUTION: Configuring the cache too small or too large can cause severe performance problems.

Considerations for ALTER DISK, CACHE and Disks

- Using the ALTER DISK, CACHE command causes all disk caches to be flushed when the cache configuration is changed.
- A cache that is too large can degrade system performance, causing:
  - Excessive page faulting in the processor
  - A significant number of read faults on the cache
  - File-system errors indicating lack of memory (such as error 35 or 36)
  - Processor halts due to lack of memory (such as halt %11501)
- Repeat this procedure as often as necessary to achieve:
  - Cache Read Hits: High
  - Cache Read Misses: Low
  - Cache Faults: Low
  - Audit Forces: Low
- There can be up to eight disk processes for each volume. Each disk process in the volume disk-process group requires a minimum configuration for each cache block size. These minimums are determined by the disk process during initialization (that is, after a system-load or when a volume is brought up). The minimum values are based upon a number of factors, such as the physical memory page size and the number of disk processes. If the specified number of blocks is below the minimum, the disk process uses the calculated minimum number.
- To avoid wasting unused space within a physical memory page, all allocations round up to the next whole physical page boundary. For example, assuming a 16K physical page size, if you specify 100 for the 512-byte cache, the actual value is rounded up to 128 (8 physical pages). While the pages underlying the cache are not swappable, they are available to the memory manager when not in use.
- The maximum tested cache allocation is 900 MB. If you allocate more cache than can be supported by the available physical and processor memory, the disk process overrides the requested values by setting the cache sizes to default values (.5K,18), (1K,18), (2K,18), (4K,1024).
- On occasion, the memory needs of the rest of the system might prevent the disk process from actually using the allocated cache.

Configuring Disk Cache

1. Review information about the CACHE attribute.
2. List current disk cache configuration information:

```
1-> INFO $DATA00, CACHE
STORAGE - Cache Info Magnetic DISK \NBSTS01.$DATA00 (Cn = config, Al = alloc)
Size______512_________1K_________2K_________4K________32K

Cn          -1          -1          -1          -1          -1
Al         512         512         512        4096         512

 Bytes Allocated To Cache: 34560 KB
```

3. Save accumulated statistical data before flushing it:

```
1-> STATS / OUT STATSLOG / $DATA00
SCF W20052 Creating file \NBSTS01.$DATA00.STATSLOG.STATSLOG

==SCF - T9082H01 - (23JUN11) (02MAY11) - 07/23/2013 14:39:17 System \NBSTS01
==(C) 1986 Tandem (C) 2006 Hewlett Packard Development Company, L.P.
STORAGE - Stats DISK \NBSTS01.$DATA00

Cache Statistics:
  Current Time............. 23 Jul 2013, 14:39:17.511
  Counters Reset Time...... 23 Jul 2013, 14:26:34.317
  Elapsed Time............. 0 days, 00:12:43

  Bytes Allocated To Cache. 34560 KB
  Writes/Control Point..... 0.00

Block Size___________512_________1K_________2K_________4K

Requested....         -1         -1         -1         -1
Allocated....        512        512        512       4096
Blocks In Use          0          0          0        208
Blocks Dirty.        0 %        0 %        0 %        0 %

Cache Reads..        1 %        0 %        0 %       82 %
Read Hits....       90 %        0 %      100 %       99 %
Read Misses..       10 %        0 %        0 %        1 %

Cache Writes.       99 %      100 %      100 %       18 %
Write Dirties        0 %        0 %        0 %        0 %
Write Cleans.       98 %       98 %       98 %       97 %
Write Misses.        2 %        2 %        2 %        3 %

Cache Calls..       1001        198        596       1847
Cache Faults.          0          0          0          0
Audit Forces.          0          0          0          0
```

4. For an explanation of the display fields and for help determining if memory allocation is adequate, see "Displaying Disk Cache Configuration Information" (page 59).
5. Change the 4K cache block size to increase the number of blocks allocated to 2000:

```
-> ALTER $DATA10, CACHE ( 4K, 2000 )
```

6. View the new cache configuration information:

```
1-> INFO $DATA00, CACHE
STORAGE - Cache Info Magnetic DISK \NBSTS01.$DATA00 (Cn = config, Al = alloc)
Size___________512___________1K___________2K_________4K__________32K

Cn               -1            -1            -1          -1            -1
Al              512           512           512        4096           512

 Bytes Allocated To Cache: 34560 KB
```

7. Reset all counters for the disk:

   ```
   -> STATS $DATA10, RESET
   ```

   Resetting the statistics after a cache configuration change lets you see how the new values affect performance. For more information, see “Reconfiguring Cache to Resolve Performance Problems” (page 61).

Naming a Disk

All disk volumes have two names: a default volume name and an alternate volume name. Both can be assigned the same name when you label a volume (by using either the RENAME DISK or ALTER DISK, LABEL command):

- Use the RENAME DISK command when you want to retain use of all the files that reside on the disk you are renaming.
- Use the ALTER DISK, LABEL command when you want to erase all files on the disk

CAUTION: Do not rename a volume unless you are sure that other products or applications (such as SMF, SQL, or TMF) will not search for files on that volume using the previous volume name:

- To determine if a specific disk is associated with a pool:

  ```
  > SCF STATUS DISK $disk, DETAIL
  ```

- To determine whether the disk contains SQL files:

  ```
  > DSAP $diskk, SQL, NEWFORMAT
  ```

- To determine if a specific disk is protected by TMF:

  ```
  > TMFCOM STATUS DATAVOLS $disk
  ```

You can name a disk in several ways:

- “Changing the Volume Name and Alternate Volume Name” (page 88)
- “Changing the Volume Name and Alternate Volume Name (Deleting Files)” (page 91)
- “Changing the Volume Name (and Keeping Files)” (page 90)
- “Changing the Alternate Volume Name (and Keeping Files)” (page 90)

Resolving Disk-Naming Conflicts

When a disk is started, the storage subsystem verifies that the configured name of the disk matches one of the names on the disk label and, if necessary, changes the configured name. Before changing the configured name, the storage subsystem must also verify the new name is not being used by another process. The sequence of events is:

1. The configured name is checked to see if it matches first the volume name and then the alternate volume name. If the configured name matches either name, the disk is started using the configured name.
2. If the configured name does not match the volume name or the alternate volume name, you are asked to confirm that the disk rename itself.
3. If you reply OK, the disk process is started and renamed, and the system configuration database is updated to reflect this.
4. If both the volume name and alternate volume name are in use by other processes, the disk remains in the STOPPED state, substate DOWN, and the configured name is left unchanged.

Changing the Volume Name and Alternate Volume Name

This example uses the “RENAME DISK Command” (page 269)to change the volume name and alternate volume name from $DATA02 to $SPARE00 without destroying the files on the volume.

Considerations for RENAME DISK

- When you issue the RENAME DISK command, the disk must be in one of these states:
  - STOPPED state, substate DOWN
  - SERVICING state, substate SPECIAL
- During a rename operation, SCF puts the disk (or both halves of a mirrored volume) in the SERVICING state, substate SPECIAL, so no other processes can access the disk. When the operation finishes, SCF puts the disk in the STARTED state.
- After the RENAME DISK command is executed, the default volume and alternate volume names are changed to the $`$`*`new-disk`* name. The disk configuration record is updated to reflect the change. The old disk names are deleted from the system and replaced by the new disk names.
- If the time stamps in the volume labels are different on the two disks of a mirrored volume, only the newer disk is renamed. An error message alerts you that one half of the mirrored volume is inconsistent with the other half.
- You cannot rename a disk that resides in a storage pool. If you do, the pool process and virtual disks are not informed of the name change. Therefore, before letting you rename a disk in a storage pool, the storage subsystem manager issues a warning regarding the possible effects on the storage pool process and virtual disks.
- During system load or a START DISK command, the storage subsystem uses the default volume name to bring up the disk. However, if the default volume name is already in use, the alternate volume name is used. If the alternate volume name is also in use, the volume name as recorded in the system configuration database is used.

  If both the default and alternate names are already being used, you can resolve duplicate name conflicts by using one of these SCF commands to assign a new name that is not in use:
  - RENAME DISK
  - ALTER DISK, VOLNAME
  - ALTER DISK, ALTNAME

    These commands let you rename and bring up a disk volume that has the same name as another volume currently active on the system. For more details, see "Naming a Disk" (page 88).

Renaming a Disk

1. Stop the disk you want to rename:

   ```
   -> STOP DISK $DATA00
   ```
2. Verify the disk is stopped:

   ```
   -> STATUS $DATA00
   ```

   If the disk is not stopped, you can use the STOP command with the FORCED attribute:

   ```
   -> STOP $DATA00, FORCED
   ```
3. Rename the disk:

   ```
   -> RENAME DISK $DATA00, $SPARE00
   ```

   The disk volume is automatically started with the new names, $SPARE00-P and $SPARE00-M, even if AUTOSTART is not configured.
4. Verify the volume name of the disk has changed:

   ```
   -> STATUS $SPARE00
   ```

5. View the new volume name and alternate volume name:

```
-> INFO $SPARE00, LABEL
```

Changing Either the Volume Name or Alternate Volume Name

The VOLNAME and ALTNAME attributes of the ALTER DISK command change the volume name and alternate volume name, respectively, of a stopped disk.

When a disk is started, SCF uses the name in the system configuration database to bring up the disk. The disk process reads the disk label and might change its process name to match either the VOLNAME or ALTNAME. If both VOLNAME or ALTNAME are in use by other processes, the disk process continues running with the configured name but does not enter the STARTED state.

When you change just the volume name or alternate volume name, only the volume label is changed. After you change either volume name or alternate volume name, the disk process might attempt to change the disk name to either the volume name or alternate volume name. If the disk name changes, the disk process tells the storage subsystem manager to change its system configuration database records to match the new disk name. For details about how disk names are determined when a disk is started, see "Naming a Disk" (page 88).

Changing the Volume Name (and Keeping Files)

This example uses the VOLNAME attribute of the ALTER DISK command to change the volume name from $DATA02 to $DEF00 without destroying the files on the volume

CAUTION: Do not rename a volume unless you are sure that other products or applications (such as SMF, SQL, or TMF) will not search for files on that volume using the previous volume name.

Stop the disk you want to rename:

```
-> STOP DISK $DATA02
```

1. Verify the disk is stopped:

```
-> STATUS $DATA02
```

2. Change the volume name:

```
-> ALTER $DATA02, VOLNAME $DEF00
```

3. View the new volume name:

```
-> INFO $DATA02, LABEL
```

The volume name in the display should be $DEF00.

4. Start the disk:

```
-> START $DATA02
```

Starting the disk changes the name of the $DATA02 disk to $DEF00 in the system configuration database.

Subsequent commands to the disk should use the new name:

```
-> INFO $DEF00
```

Changing the Alternate Volume Name (and Keeping Files)

This example uses the ALTNAME attribute of the ALTER DISK command to change the alternate volume name from $DATA02 to $ALT00 without destroying the files on the volume

CAUTION: Do not rename a volume unless you are sure that other products or applications (such as SMF, SQL, or TMF) will not search for files on that volume using the previous volume name.

Stop the disk you want to rename:

```
-> STOP DISK $DATA02
```

1. Verify the disk is stopped:

```
-> STATUS $DATA02
```

2. Change the alternate volume name:

```
-> ALTER $DATA02, ALTNAME $ALT00
```

3. View the new alternate volume name:

```
-> INFO $DATA02, LABEL
```

   The alternate volume name in the display should be $ALT00.

4. Start the disk:

```
-> START $DATA02
```

Because you changed only the alternate volume name and the volume name is still $DATA02, the disk name does not change.

Changing the Volume Name and Alternate Volume Name (Deleting Files)

The LABEL attribute of the ALTER DISK command changes the volume name and deletes all files from the volume.

Considerations for LABEL and Disks

CAUTION:

- The ALTER DISK, LABEL command changes both the default and alternate volume names and erases any existing files on the disks you are renaming.
- During a label operation, SCF puts the disk (or both halves of a mirrored volume) in the SERVICING state, substate SPECIAL, so no other processes can access the disk. When the operation finishes, SCF puts the disk in the STARTED state.
- You can label the accessible half of a mirrored volume if the other half is inaccessible (for instance, if one half is in the STOPPED state, substate HARDDOWN).
- The default (or alternate) volume name remains with the volume until it is changed by one of:
  - Another ALTER DISK, LABEL command
  - An ALTER DISK, VOLNAME (or ALTNAME) command
  - A RENAME DISK command
- When you use the START DISK command on a volume (or when you load the system from tape), SCF uses the default volume name to bring up the disk unless this name is being used by some other device or process. If the default volume name is in use, the storage subsystem automatically uses the alternate volume name.
- If the alternate volume name is also in use, the storage subsystem uses the name for this disk as stored in the system configuration database. The disk is left in the STOPPED state, substate DOWN. To change the name (or alternate volume name) of the disk, use one of these SCF commands to assign a new name:

```
-> RENAME DISK
-> ALTER DISK, VOLNAME
-> ALTER DISK, ALTNAME
```

Relabeling a Volume

1. Stop the disk you want to relabel:

```
-> STOP DISK $DATA01
```

2. Verify the disk is down:

```
-> STATUS $DATA01
```

3. Rename the disk $DATA01 to $BLANK01:

   ```
   -> ALTER $DATA01, LABEL $BLANK01
   ```

   The disk is automatically started using the new names, $BLANK01-P and $BLANK01-M, even if AUTOSTART is not configured.

4. Verify the volume name of the disk has changed:

   ```
   -> STATUS $BLANK01
   ```

5. View the new volume name and alternate volume name:

   ```
   -> INFO $BLANK01, LABEL
   ```

Relabeling and Initializing a Disk

The "INITIALIZE DISK Command" (page 260) re-creates the label and initializes disk information such as the spare-tracks table.

CAUTION: The INITIALIZE command removes directory information so that files are not accessible to normal software. The file data remains on the disk and could be read by specialized low level disk access utilities.

Considerations for INITIALIZE DISK

- If you do not specify the LABEL attribute, the disk is automatically relabeled using the current name (if the volume name and alternate volume name in the disk label have the same name).
- After the INITIALIZE command completes, the disk is started.

Example of Relabeling With the Current Name

1. Initialize $DATA12 and relabel it with the current volume name:

   ```
   -> INITIALIZE DISK $DATA12
   ```

2. Find and spare any bad sectors. (See "Sparing a Defective Sector" (page 104).)

Example of Relabeling With a New Name

1. Initialize $DATA12 and relabel it with the name $SPARE00:

   ```
   -> INITIALIZE DISK $DATA12, LABEL $SPARE00
   ```

2. Find and spare any bad sectors. See "Sparing a Defective Sector" (page 104).

Partitioning HDDs and SSDs

Effective with the H06.23/J06.12 RVU, you can partition some Hard Disk Drives (HDDs) and all Solid State Drives (SSDs) in Serial Attached SCSI (SAS) enclosures connected to CLIMs. To determine which hardware supports SSD and disk partitioning, refer to the planning guide for your system.

CAUTION: Disk partitioning requires the J06.13 or later J-series RVU. If you must fall back to an earlier RVU, you cannot partition SSDs. If you partitioned SSDs before falling back, that disk will appear as uninitialized to the system although the data is still on the disk.

if you want to reuse a disk that's been partitioned, you can't initialize it until you've deleted the partitions. Then you can initialize.

You can use SCF commands to add, configure, delete, and view the partitions on a SAS disk. These commands are:

- The “ADD PARTITION Command” (page 216) adds a new partition on a physical disk specified by the CLIM and LUN.
- The “INFO PARTITION Command” (page 256) displays the disk partition information for the physical disk specified by the CLIM and LUN.
- The “DELETE PARTITION Command” (page 250) deletes the last partition or all partitions on the physical disk specified by the CLIM and LUN.
- The “ADD DISK Command” (page 194) allows you to configure a disk LDEV to a specific partition on a physical disk.
- The “ALTER DISK Command” (page 228) allows you to alter a disk LDEV's partition either offline or online.
- The “INFO DISK Command” (page 254) displays the partition number (if it exists) for each path. The INFO DISK, CONFIG command provides the location of the mirror disk.
- The “INFO CLIM Command” (page 253) displays the partition number in addition to the LUN.
- The “STATUS CLIM Command” (page 280) displays the partition number in addition to the LUN.

Partition configuration commands are separate from NonStop disk LDEV configuration commands, so you would partition disks in this sequence:

1. Use the ADD PARTITION command to configure partitions on the physical disk. The ADD PARTITION command with the LIKE attribute will partition a disk just like another disk.
2. Use the ADD DISK command to configure a NonStop disk LDEV to utilize each partition.

Since partition configuration commands are issued before the disk partition is configured to a NonStop disk LDEV, the configuration commands take a physical disk location (such as primary and backup CLIM names and LUNs) rather than a NonStop disk LDEV name.

The object of the PARTITION commands, ADD PARTITION, DELETE PARTITION and INFO PARTITION, is a physical drive that is indicated by a pair of CLIMs and LUNs. The PRIMARYCLIM and BACKUPCLIM attributes indicate the pair of CLIMs that are connected to the physical drive. They can be either NonStop paths –P and –B or –M and –MB.

CAUTION:

- A mirrored disk partition must reside on drives that are partitioned exactly the same.
- The OBEYFORM option creates an obey file for configuring partitions on the disk. You can use that obey file to recreate partitions on the replacement disk.
- When you partition a disk, be sure to make a record of your partitioning scheme if you want to restore the partitioning scheme you used to partition the disk.

For information and considerations on mirroring partitions, refer to “Mirroring With CLIM-Attached Disks” (page 77).

Deleting a Disk

The “DELETE DISK Command” (page 248) removes a disk from the system configuration database.

Considerations for DELETE DISK and Disks

- A disk volume must be STOPPED or not running before you can delete it.
- You can delete a mirror half of a disk volume while the disk volume is in a STARTED state.
- You cannot delete only the primary half of a mirrored volume.

- Deleting the mirror volume also deletes the mirror backup, if it exists.
- Always note the location of the disk that you delete. Once you delete a disk from the system configuration database, it becomes unavailable to SCF. However:
  - Installing an unconfigured disk in a slot generates an error message either at that time or when the system is restarted.
  - You can use OSM or TSM to view disks that are physically currently installed in system slots. For more information, see the OSM or TSM online help for more information.
  - For additional details, see "Displaying Information About Installed, Nonconfigured Internal Disks" (page 48).
  - Issue a START command to restart the IOP after deleting the mirror volume.

Example

1. Stop any new disk activity:
   ```
   -> STOPOPENS $DATA14
   ```
2. Stop the disk:
   ```
   -> STOP $DATA14
   ```
   If any files on the disk are currently open, they are identified in the return message.
3. Delete $DATA14 from the system configuration database:
   ```
   -> DELETE $DATA14
   ```
4. Confirm that the disk has been deleted:
   ```
   -> INFO $DATA14
   ```

Write Caching

The WRITECACHE disk attribute controls whether write caching is performed for disk writes.

If write caching is disabled, when a disk write operation completes, disk status is returned only after data is written to the drive media. If you enable write caching, data is written to the drive's cache and completion status is returned immediately (before the data is written to the drive media). This allows for faster writes to SAS disks.

To enable write cache, set the WRITECACHE disk attribute to ENABLED. The default is DISABLED.

After changing the write cache setting, use the STATUS DISK command to determine the setting that is in effect on the drive.

Considerations for WRITECACHE

- This attribute is available only on J06.03 and subsequent J-series RVUs.
- This attribute only affects CLIM-connected SAS disk drives. It is ignored for all other disks.
- The SUBSYS attribute UPS should be ON either before WRITECACHE is enabled or before the IOP is started. Otherwise, the IOP will run with WRITECACHE disabled.
- If you set WRITECACHE to ENABLED for SAS HDDs or SSDs, you must have an HP rack mount UPS to protect data during power interruptions.
- When adding a new disk on disk partitions, the writecache setting of the new disk must match the writecache setting of the existing disk partitions.

- Altering a WRITECACHE setting of a disk partition sets the writecache setting of all the disk partitions that are configured on the same physical disk. Changes to the other disk partitions set the writecache setting of all their mirror disks.
- Altering WRITECACHE setting must be done only when the Storage CLIMs are free.

CAUTION: In order to prevent data loss, if a volume is configured with WRITECACHE enabled, HP strongly recommends that the volume be protected by a cabinet or HP rack mounted UPS, which will give HDD or SSD drives enough time to write cached data to the media after the processor stops writing in the event of power loss.

An HP rack mounted UPS contains batteries that are auxiliary to the energy storage in the site UPS system. Batteries have a lifespan which is likely to be shorter than that of the system the UPS supports and they must be maintained so that they are replaced when their usable lifespan has elapsed. The HP rack mounted UPS system includes self test capabilities which must be monitored to insure the UPS remains capable of functioning in the event of a power loss.

The HP rack mounted UPS must be configured in OSM Service Connection. This ensures that system processing is halted once the HP rack mounted UPS has been in use for a period of time that is configured in SCF. The configuration should allow for ten additional minutes of rack mounted UPS power following the issuance of the halt.

7 Managing Disks

This chapter describes managing disks. Disks share the object type of DISK with virtual disks. See "Configuring and Managing Virtual Disks" (page 143). For information about disk load balancing, see "Disk Load Balancing" (page 118). This chapter describes:

Managing Disks

When managing disks, review these special considerations:

Managing the System Disk

You cannot stop access to the system disk because important processes have their object files and swap files on it. Therefore, you cannot stop the last path to the system disk. You must take special precautions when performing some procedures:

- If you have to replace the disk bootstrap program on the system disk, make sure it is mirrored. See "Changing Two Nonmirrored Disks Into a Mirrored Volume" (page 79).
- Do not issue a "STOPOPENS Command" (page 293) on the system disk.

Recovering From the Loss of a Path to a Disk

If the X fabric or Y fabric fails, internal disk paths using the failed fabric go down. After repairing a fabric failure, these failed disk paths are not automatically restarted. They remain HARDDOWN until you restart the disk process. The storage subsystem never attempts to use the failed path, which creates a potential single point of failure.

To restore the paths to all disks that use the repaired fabric, see "Resetting a Group of Disks" (page 104).

Troubleshooting Disks

For more information about troubleshooting disk drives, see the Service and Support Library of the NonStop Technical Library (NTL).

Starting a Disk

The "START DISK Command" (page 274) makes a stopped disk accessible to user processes. When finished, the disk is in the STARTED state.

Considerations for START DISK

- New M8*xxx* disks that have been added and are being used for the first time must be initialized before they can be started.
- If you are starting a mirrored volume, see "Considerations When Reviving a Mirrored Volume" (page 98).
- If you receive a DUPLICATE VOLUME message in response to a START command, you must rename the disk. See "Naming a Disk" (page 88).
- When a system is loaded or when you use the START DISK command, SCF uses the default volume name. If the default volume name is already in use, the volume is started using the alternate volume name. If the alternate volume name is also in use, the storage subsystem uses the volume name in the system configuration database and the disk is left in the DOWN state. For more information, see "Naming a Disk" (page 88).
- The START DISK command can introduce discrepancies between the disk file labels for SQL files and the catalog descriptions of the SQL files, thereby making the SQL dictionary inconsistent. These inconsistencies make the SQL database unusable.
- Use the RESET DISK command followed by the START DISK command to bring up a disk path that is in any of these states:
  - STOPPED state, substate HARDDOWN
  - SERVICING state, substate TEST
  - SERVICING state, substate SPECIAL

Starting an Unmirrored Disk

1. Check the current status of the disk:

   ```
   -> STATUS $DISK00
   ```

2. If the disk is not in the STOPPED state, substate DOWN, see "Resetting a Disk" (page 103).
3. If the disk is in the STOPPED state, substate DOWN, start the disk:

   ```
   -> START $DISK00
   ```

4. Confirm that the disk has started:

   ```
   -> STATUS $DISK00

   STORAGE - Status DISK \COMM.$DISK00
   LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                          PID       PID
    212  *STARTED    STARTED   *STARTED    STARTED          9,262     8,271
   ```

5. If the START DISK command fails, see the information on troubleshooting disk drives in the Service and Support Library of the NonStop Technical Library (NTL).

Reviving a Mirrored Disk

The "START DISK Command" (page 274) makes a stopped disk accessible to user processes and revives the second half of a mirrored volume.

Use this procedure if you have stopped one half of a mirrored volume and need to synchronize the two halves when restarting the disk. This situation can occur if:

- You do something that affects only one disk of a mirrored volume. For an example, see "Replacing the Bootstrap Program" (page 107).
- You back up disk files by replacing the mirror drive with another drive instead of using BACKUP and RESTORE.
- You are "Swapping Processors for a Disk" (page 109).
- One half of a mirrored pair experiences a physical media failure, and you have to stop and remove it.

Considerations When Reviving a Mirrored Volume

- Do not stop the revive operation by issuing a STOP DISK command unless you want to force the next revive operation to restart from the beginning. See "Stopping a Revive Operation" (page 100).
- If a nonfatal error occurs, the revive operation does not proceed but stalls or goes into a loop. The START DISK command continues to retry at the current address until either the revive operation is successful or the revive operation is suspended or stopped. For more details, see "Stopping a Revive Operation" (page 100).
- When SCF must revive the information on the volume before starting the volume, the revive operation can seriously affect system performance, especially for users of the volume being revived. You can use the RESET DISK command to temporarily suspend the revive operation when needed. See "Temporarily Stopping a Revive Operation" (page 101).

Starting a Mirrored Disk

1. Verify the current state of the disk volume:

```
-> STATUS $DSMSCM-*

STORAGE - Detailed Status DISK \ALM171.$DSMSCM

Disk Path Information:
 LDev  Path           Status       State      Substate          Primary  Backup
                                                                 PID      PID
   98  PRIMARY        ACTIVE       STOPPED    DOWN               0,21     1,15
   98  BACKUP         INACTIVE     STOPPED    DOWN               0,21     1,15
   98  MIRROR         ACTIVE       STOPPED    DOWN               0,21     1,15
   98  MIRROR-BACKUP  INACTIVE     STOPPED    DOWN               0,21     1,15
```

2. If all paths are not in the STOPPED state, substate DOWN:

```
-> RESET $DSMSCM
```

3. Start the disk process:

```
-> START $DSMSCM
```

   The disk revive operation does the following:

   a. The disk process copies the percentage of the total number or sectors that have been specified by the REVIVERATE attribute from the disk with the newer timestamp to the disk that is being revived (the disk with the older timestamp).
   b. The disk process checks for other work to be done. If there are no other requests pending, the disk process starts the disk revive operation again.
   c. The disk process handles any waiting requests.

d. When the copy interval expires, the disk process finishes the request that it is currently handling and continues the disk revive operation again.

e. To preserve the older disk and copy it to the disk that has the newer time stamp, start a path to the older disk first. This action causes an update to its timestamp, making it more recent. Then you can start the rest of the volume. To monitor the progress of the revive operation, periodically issue this command:

```
-> STATUS $DSMSCM, DETAIL

STORAGE - Detailed Status DISK \ALM171.$DSMSCM

Disk Path Information:
 LDev  Path           Status      State      Substate         Primary  Backup
                                                               PID      PID
   98  PRIMARY        ACTIVE      STARTED                      0,21     1,15
   98  BACKUP         INACTIVE    STARTED                      0,21     1,15
   98  MIRROR         ACTIVE      STARTING   REVIVE            0,21     1,15
   98  MIRROR-BACKUP  INACTIVE    STARTING   REVIVE            0,21     1,15

General Disk Information:
  Device Type........... 3         Device Subtype........... 41
  Primary Drive Type.... 4604-1    Mirror Drive Type........ 4604-1
  Physical Record Size.. 4096      Priority................. 220
  Library File..........
  Program File.......... $SYSTEM.SYS00.TSYSDP2
  Protection............ AUDITED, MIRRORED

Revive Information:

  Revive Type.......................... Delta
  Revive Status........................ Active
  Current Logical Sector............... %H00012900
  Current Revive Progress.............. %H00012900 of %H00DC0BBC (0%)
  Current Revive Priority.............. 50
  Current Revive Rate.................. 50
  Estimated completion at current load... 13 minutes

Usage Information:
  Capacity (MB)......... 4238.96    Free Space (MB).......... 1524.41 (35.96%)
  Free Extents.......... 105        Largest Free Extent (MB). 1504.11

Hardware Information:
  Device          Location             Power          Physical Status
                  (group,module,slot)
  PRIMARY         (1,1,13)             DUAL           PRESENT
  MIRROR          (1,1,14)             DUAL           PRESENT
```

The above display shows:

- The mirror disk is being revived.
- The Revive Priority is 50.
- The Revive Rate is 50. Thus, half the total number sectors are revived between preemption checks. The estimated completion time is 13 minutes.

4. To change the speed of a revive operation, see “Changing the Speed of a Revive Operation” (page 99).
5. To stop the revive operation, see “Example of an INFO POOL Report” (page 135).
6. When no more tracks are left to be copied, the revive operation is complete. At that time, the revived disk is in the STARTED state.

Changing the Speed of a Revive Operation

While a revive operation is in progress, you can use the ALTER command to change its speed.

Considerations for Changing the Speed of a Revive Operation

- The speed of all future revive operations is also changed.
- During system installation, the system administrator should tailor the REVIVEPRIORITY and REVIVERATE attributes for your system. The default values provided by SCF are acceptable for most environments:
  - REVIVEPRIORITY 50
  - REVIVERATE 100 second between copies

    The default values minimize potential interference with system performance but could result in revives that take too long to finish. (The longer the revive operation takes, the longer your mirrored disks have dissimilar data.)
- To speed up the revive operation (even though this change might slow system performance), increase the REVIVEPRIORITY value and/or increase the REVIVERATE value.
- If you change these values while a revive operation is in progress, the disk process does not restart the revive operation from the beginning but continues from the point at which you entered the new values.

For more information, see REVIVERATE.

Example of Changing the Speed of a Revive Operation

This command establishes a revive priority of 60 and specifies that 90 megabytes of data be revived between preemption checks while a revive operation is in progress:

```
-> ALTER $DATA01, REVIVEPRIORITY 60, REVIVERATE 90
```

Stopping a Revive Operation

It is seldom necessary to stop a revive operation (with a STOP DISK command on the disk being revived) unless you want to force the revive operation to restart from the beginning.

You might want to adjust a revive operation if:

- System performance is degraded. See “Changing the Speed of a Revive Operation” (page 99).
- A media error has occurred, causing a defective sector.

  If you have enabled automatic sector reallocation, the system spares the sector and the revive operation resumes. If automatic sector reallocation is disabled, see “Temporarily Stopping a Revive Operation” (page 101).

  Nonfatal checksum errors have stalled a revive operation or put it into a loop. Symptoms are:
  - The revive interval in a STATUS, DETAIL display has become large.
  - The Current Logical Sector remains unchanged.
  - EMS messages are generated. Research these messages with the *Event Management Service (EMS) Analyzer User’s Guide and Reference Manual*.

    The revive operation continues to retry at the current address until it is successful, suspended, or stopped.

    After each retry, the revive interval doubles in length. In this way, the retries do not consume system resources or produce too many operator messages. The interval continues to double up to a maximum of one hour; thereafter retries occur indefinitely. If a retry succeeds, the revive interval returns to its original configured value.

    See “Temporarily Stopping a Revive Operation” (page 101).

Temporarily Stopping a Revive Operation

1. Temporarily stop the disk being revived:

   ```
   -> RESET $DATA01-M
   ```

   The disk remains in the STARTING state, REVIVE substate.
2. While the revive operation is suspended, do what needs to be done to resolve the problem. For example, spare the defective sector.
3. Resume the revive operation:

   ```
   -> START $DATA01-M
   ```

The revive operation resumes from the point at which it was stopped.

Stopping a Disk

These two commands stop access to a disk:

- STOP DISK command—stops access in an orderly manner
- ABORT DISK command—stops access immediately

Stopping a Disk With the STOP DISK Command

The "STOP DISK Command" (page 290) stops access to a disk in an orderly manner. When finished, the disk is in the STOPPED state, substate DOWN. The disk remains configured in the system configuration database.

Considerations for STOP DISK

- To determine the current status of disk paths you want to stop, use the STATUS DISK command.
- If you attempt to stop a disk and the backup disk path is detected as being unavailable, the disk is not stopped.
- If you do not use the FORCED attribute, SCF asks you to confirm any request to stop a disk when it is the last path to that volume.
- When the last path to a disk is stopped, an implicit refresh operation is performed.
- TMF considerations:
  - You cannot stop all paths to a disk enabled by TMF.
  - You cannot stop a volume containing active audit trails.
  - You cannot stop the last path to the system disk.
- KMSF considerations:
  - Stopping any volume that contains active Kernel-Managed Swap Facility (KMSF) swap files may cause processor halts or process abends.
  - To determine if KMSF swap files are configured in the volume, enter `NSKCOM` at a TACL prompt. If such files exist, you may need to reconfigure the KMSF swap files before you can stop the volume. For more details, see the *Kernel-Managed Swap Facility (KMSF) Manual*.

Stopping a Disk

1. Check the state of the disk:

   ```
   -> STATUS $AUDIT
   ```
2. Stop the disk:

   ```
   -> STOP $AUDIT
   ```

3. Confirm that the disk is in the STOPPED state, substate DOWN:

   ```
   -> STATUS $AUDIT
   ```

If the STOP command does not work and you need to stop the disk, use the ABORT command.

Stopping a Disk With the ABORT DISK Command

The "ABORT DISK Command" (page 193) stops access to a disk when the disk or path to the disk either:

- Is malfunctioning but has not been brought down by the disk
- Must be removed from the system (but the system currently cannot be reconfigured)

Considerations for ABORT DISK

- Do not used the ABORT DISK, FORCED option on the system disk.
- Do not abort a volume that has open object files or swap files for currently executing processes. First close these files by stopping the processes that are using them.
- If you omit the FORCED option and the volume has files open, SCF asks you to confirm the abort request.
- If you omit the FORCED option when aborting the last available path, SCF asks you to confirm the request before it aborts that path.
- You cannot abort a disk containing active TMF trails.
- When the last path to a disk is stopped, an implicit refresh operation is also performed. The refresh operation is a general cleanup operation to prevent the device from having any changed buffers or file control blocks still outstanding.
- Before restarting the process, you must use the "RESET DISK Command" (page 271). The ABORT command leaves configured device paths in the STOPPED state, substate HARDDOWN. The process remains in the system configuration database file. Any attempt to access a path that is in the STOPPED state, substate HARDDOWN, fails with file-system error 66.

Aborting a Disk

When finished, the disk is in the STOPPED state, substate HARDDOWN.

Check the state of the disk:

```
-> STATUS $AUDIT
```

1. Force the disk to stop:

   ```
   -> ABORT $AUDIT
   ```

2. Confirm that the disk is in the STOPPED state, substate HARDDOWN:

   ```
   -> STATUS $AUDIT-*

   STORAGE - Status DISK \COMM.$AUDIT-*
   LDev  Path            Status       State       Substate         Primary   Backup
                                                                   PID       PID
    395  PRIMARY         INACTIVE     STOPPED     HARDDOWN          2,266     3,279
    395  BACKUP          INACTIVE     STOPPED     HARDDOWN          2,266     3,279
    395  MIRROR          INACTIVE     STOPPED     HARDDOWN          2,266     3,279
    395  MIRROR-BACKUP   INACTIVE     STOPPED     HARDDOWN          2,266     3,279
   ```

Resetting a Disk

The "RESET DISK Command" (page 271) puts a disk into the STOPPED state, substate DOWN, ready for restarting.

- Use the RESET command to prepare a device to be started if:
  - A disk is in the STOPPED state, substate HARDDOWN.
  - A hardware error has occurred.
  - You stopped the disk with an ABORT command.
  - The disk was stopped for service.
  - A STOP or ABORT command fails to put the device into the STOPPED state, substate DOWN.

    See "Resetting One Disk" (page 103).
- You can reset and start more than one disk at a time:
  - After installing or replacing a component like a disk, SEB, MSEB, PMF CRU, IOMF CRU, or ServerNet/DA.
  - After repairing a fabric failure

    When a fabric fails, the storage subsystem automatically switches the disk paths, if possible, so that the disks remains operational.

    After repairing a fabric failure, failed disk paths are not automatically restarted. They remain HARDDOWN until you restart the disk process. The storage subsystem never attempts to use the failed path, which creates a potential single point of failure.

    See "Resetting a Group of Disks" (page 104).

Considerations for RESET DISK

- The RESET command is ignored if the process is started.
- If the disk is in the STARTING state, substate REVIVE, the disk process is suspended in that state until either another START command restarts the revive operation or a STOP command terminates the revive operation.

Resetting One Disk

1. Check the current status of the disk:

   ```
   -> STATUS $DISK00-*
   ```
2. If any disk paths are in one of these states:
   - SERVICING state, SPECIAL substate
   - SERVICING state, TEST substate
   - STOPPED state, HARDDOWN substate

     Put those paths into a STOPPED state, substate DOWN:

     ```
     -> RESET $DISK00
     ```

     Paths that are in the STARTED state are unaffected by the RESET command.
3. To prevent a specified disk path from starting:

   ```
   -> ABORT DISK $disk00-MB
   ```
4. Start the disk:

   ```
   -> START $DISK00
   ```

5. Confirm that the disk has started:

```
-> STATUS $DISK00
```

Resetting a Group of Disks

1. Identify disks that have paths that are not running:

```
-> STATUS DISK $*-*, SEL NOT STARTED, SUB MAGNETIC
```

2. Prepare all nonrunning disk paths for a subsequent START command:

```
-> RESET DISK $*
```

3. To prevent a START DISK command with a wild-card disk name from failing if some disk paths do not start:

```
-> ALLOW ALL ERRORS
```

4. To prevent a specified disk path from starting:

```
-> ABORT DISK $disk00-P
```

5. Start all startable disks:

```
-> START DISK $*, SUB MAGNETIC
```

Sparing a Defective Sector

Sector sparing is the process of moving the data on a physically defective sector to a different sector and preventing data from using that defective sector again. A disk sector can be spared in two ways:

- "Enabling Automatic Sector Reallocation" (page 104)
- "Sparing a Sector Manually" (page 105)

Automatic Sector Reallocation

If automatic sector reallocation is enabled (the default), the system automatically spares defective disk sectors when detected. The INFO, LOG command displays spared sectors.

Automatic sector reallocation is controlled by the $ZRD9 process. This generic process starts automatically when you load the system and continuously searches for media error events on every sector on every disk in the system.

When automatic sector reallocation finds a media error event, it starts the $ZARS process. $ZARS spares the bad sector, updates the system's internal database, and continues to run as long as media error events are reported. When no more media error events are reported, $ZARS stops.

Displaying Information About Automatic Sector Reallocation

To determine whether automatic sector reallocation is enabled on your system, at the TACL prompt:

```
> STATUS $ZRD9

System \COMM

Process              Pri  PFR %WT Userid   Program file                 Hometerm
$ZRD9        0,381   150      015 255,255  $SYSTEM.SYS00.EMSDIST        $ZHOME
                          Swap File Name:  $SYSTEM.#0
       Current Extended Swap File Name:  $SYSTEM.#0
$ZRD9     B  1,372   150      001 255,255  $SYSTEM.SYS00.EMSDIST        $ZHOME
                          Swap File Name:  $SYSTEM.#0
       Current Extended Swap File Name:  $SYSTEM.#0
```

Enabling Automatic Sector Reallocation

To enable automatic sector reallocation:

```
-> START PROCESS $ZZKRN.#ROUTING-DIST
```

Disabling Automatic Sector Reallocation

If you want to disable automatic sector reallocation, you must disable it for all disks; you cannot enable or disable it for specific disks.

To disable automatic sector reallocation:

```
> STOP $ZRD9
```

You might want to disable $ZRD9 if you choose to manually spare a sector.

Controlling Whether $ZRD9 Starts Automatically After a System Load

- To ensure that $ZRD9 does not start automatically after a system load, do one of the following:
  - Stop the process and reconfigure it to be started manually:

    ```
    -> ABORT PROCESS $ZZKRN.#ROUTING-DIST

    -> ALTER PROCESS $ZZKRN.#ROUTING-DIST, STARTMODE MANUAL
    ```

  - Stop the process and delete it:

    ```
    -> ABORT PROCESS $ZZKRN.#ROUTING-DIST

    -> DELETE PROCESS $ZZKRN.#ROUTING-DIST
    ```

- To reenable $ZRD9 automatic startup after system load:
  - Reverse the ALTER command and start the process:

    ```
    -> ALTER PROCESS $ZZKRN.#ROUTING DIST, &
    -> STARTMODE APPLICATION

    -> START PROCESS $ZZKRN.#ROUTING-DIST
    ```

  - Reverse the DELETE command and start the process:

    ```
    -> ADD PROCESS $ZZKRN.#ROUTING-DIST, AUTORESTART 0, &
    -> BACKUPCPU 1, DEFAULTVOL $SYSTEM.ZSERVICE, &
    -> HIGHPIN ON, HOMETERM $ZHOME, NAME $TSMRD, &
    -> PRIMARYCPU 0, PRIORITY 150, &
    -> PROGRAM $SYSTEM.SYSTEM.TACL, &
    -> INFILE $SYSTEM.ZTSM.INITRD, STARTMODE APPLICATION, &
    -> TYPE OTHER, USERID SUPER-ID

    -> START PROCESS $ZZKRN.#ROUTING-DIST
    ```

Sparing a Sector Manually

If automatic sector reallocation is disabled or if it fails to spare a defective sector because the error is intermittent, you might have to spare the sector manually.

If you get an error message reporting the address of a bad sector, try to verify that the message is caused by a disk media error before manually sparing the sector by using the CONTROL DISK, SPARE command

CAUTION: Do not spare sector %H2 or sector %H3 before talking to your service provider.

Confirm that the address reported in the error message also appears in an INFO DISK, BAD report:

```
-> INFO $DATA00, BAD
```

1. Spare the sector containing the address identified in the BAD report.

   ```
   -> CONTROL $DATA00-M, SPARE %H795C
   ```

   The CONTROL command assigns an alternate sector for the data in the spared sector and makes an entry into the defect log.

   If either of these events occur during a spare operation, the specified sector is not spared, and an error message is displayed:

   - All spare sectors on the disk are in use.
   - The added defect map is full.
   - In either case, you must replace the disk.

2. Verify the defective sector has been replaced:

   ```
   -> INFO $DATA00, BAD
   ```

   If the defective sector was spared, it no longer appears in the detailed report.

Correcting Doubly Allocated File Extents

The REBUILDDFS attribute of the CONTROL DISK command rebuilds the disk free space table. This process also gets rid of doubly allocated file extents.

File extent overlaps (doubly allocated file extents) are caused by a hardware or software error. A doubly allocated file extent is an error condition that occurs when the same page is allocated more than once to a single file, to two different files, or to a file and available free space. Report such errors to your service provider. Use the procedures described next to resolve the problem.

Considerations for the REBUILDDFS attribute

Use this attribute if one of these events occurs:

- File-system error 58 has occurred (the disk free space table is marked bad).
- The results from a DSAP command indicate that free space has been lost over time.
- You have resolved a problem with doubly allocated file extents (by purging one of the two conflicting files).

  If DSAP reports that a file has doubly allocated file extents, and if the message “(SQL Shadow)” appears on the same line as the file name, the file is an SQL table that has been dropped, but the drop is not yet committed. Such a file is not available to a TACL FILES command or FUP INFO command. The DSAP DETAIL or SQLCI FILEINFO report can also show the SQL shadow label.

Identifying Doubly Allocated File Extents

To identify any doubly allocated file extents:

```
-> DSAP $DATA00, FREESPACE
```

Rebuilding the Disk Free Space Table

1. Stop all applications using the disk.
2. Verify the specified volume is in the STARTED state.
3. Before initiating the REBUILDDFS operation, you must resolve any disk errors, such as unspared sectors or doubly allocated file extents:
   - “Sparing a Defective Sector” (page 104)
   - “Correcting Doubly Allocated File Extents” (page 106)
4. To rebuild the disk free space table:

   ```
   -> CONTROL $DATA00, REBUILDDFS
   ```

Troubleshooting Doubly Allocated File Extents

1. Repeat "Identifying Doubly Allocated File Extents" (page 106) to see whether any doubly allocated file extents remain:
   - If none remain, you are finished with the procedure and you can restart applications.
   - If doubly allocated file extents remain, continue to the next step.
2. Use FUP DUP or BACKUP to copy to another volume the files identified in the DSAP report that contain doubly allocated extents.
3. Purge these files from the original volume.

   If purging a file causes file-system errors, resolve them by repeating Step 4 of "Rebuilding the Disk Free Space Table" (page 106).
4. Repeat "Identifying Doubly Allocated File Extents" (page 106) to ensure that all doubly allocated file extents have been corrected. If none remain, continue with the next step. Otherwise, repeat from Step 2.
5. Use FUP DUP or RESTORE to copy the files back to the original volume.
6. Restart applications.

Replacing the Bootstrap Program

NOTE: The REPLACEBOOT attribute does not apply to Integrity NonStop NS-series servers. Instead, the bootstrap program is installed through firmware update into flash memory.

The REPLACEBOOT attribute of the CONTROL DISK command replaces the disk bootstrap program. You must replace the disk bootstrap program on a disk in any of these cases:

- You migrate the system to a new software RVU that has a new bootstrap program.
- You plan to install a system disk that is incompatible with your current bootstrap program.
- You want to create an alternate system-load volume.
- You cannot perform a system load from a disk that has been used on other occasions to load the system. This error indicates that the bootstrap program is corrupted or incompatible with the format of the target SYS*nn*.OSIMAGE file.

If a system failure occurs during the CONTROL DISK, REPLACEBOOT operation, the disk bootstrap program could be unusable. This procedure can help you recover from such a situation.

Replacing the Disk Bootstrap Program on a Mirrored Volume (NonStop S-Series Servers)

CAUTION: If you are replacing the disk bootstrap program on the system disk, use the procedure described here. Do not attempt to replace the bootstrap program on both disks of a mirrored volume simultaneously.

This procedure replaces the disk bootstrap program on the primary disk of a mirrored volume and then verifies that the disk is functional before replacing the bootstrap program on the mirror disk.

1. Review information about the REPLACEBOOT attribute.
2. If the disk volume you plan to replace the bootstrap program on is not already started, start it. This command executes a revive operation if the primary and mirror disks are not current with each other.

   ```
   -> START $SYSTEM
   ```
3. Verify the disk is started:

   ```
   -> STATUS $SYSTEM-*
   ```

If the disk is in the STARTING state, substate REVIVE, wait until the revive operation has finished before continuing to the next step. Both halves must be in the STARTED state before proceeding.

4. Replace the bootstrap program on the primary disk:

```
-> CONTROL DISK $SYSTEM, REPLACEBOOT $SYSTEM.SYSnn.SYSDISC
```

5. To facilitate recovery of a mirrored volume in the event of failure, SCF asks, `Do you want to ABORT $SYSTEM-M?` Type yes

CAUTION: If you type `no`, the REPLACEBOOT operation updates the diskboot file on both disks simultaneously. Do not attempt to replace the bootstrap program on both disks of a mirrored volume simultaneously.

In response to a yes reply, SCF places the mirror disk in the STOPPED state, substate HARDDOWN and replaces the bootstrap program on the primary disk.

6. Do not start the stopped mirror disk. Load the system using the disk that has the new bootstrap program. If the system loads properly, go to Step 7.

   If the system load fails in any way:

   a. Use the halt codes that are displayed in the Processor Status dialog box in either the OSM or TSM Low-Level Link application to determine what to do next. For descriptions of the halt codes, see the *Processor Halt Codes Manual*.
   b. If you still cannot perform a system load after following the halt code instructions, you can use the unchanged mirror disk to load the system.

7. Reset and start the unchanged mirror disk:

```
-> RESET $SYSTEM
-> START $SYSTEM
```

SCF automatically revives the unchanged mirror disk, updating the old bootstrap program from the new version copied earlier to the primary disk.

Replacing the Disk Bootstrap Program on a Nonmirrored Volume (NonStop S-Series Servers)

This procedure replaces the disk bootstrap program on an unmirrored volume of a NonStop S-series server.

The REPLACEBOOT attribute does not apply to Integrity NonStop NS-series servers. Instead, the bootstrap program is installed through firmware update into flash memory

CAUTION: This procedure assumes that the nonmirrored disk is not a system disk. Do not, under any circumstances, attempt to replace the disk bootstrap program on a system disk that is not mirrored. If you attempt to do this and the CONTROL DISK, REPLACEBOOT operation fails or if the bootstrap program is unusable and no other system disk exists, contact your service provider and refer to the information about restoring a system disk in the NonStop S-Series Operations Guide.

Review information about the REPLACEBOOT attribute.

1. As a safety measure, use BACKUP VOLUME MODE to create a backup tape of the disk on which you are replacing the bootstrap program.
2. Replace the bootstrap program on disk $DATA01:

```
-> CONTROL DISK $DATA01, REPLACEBOOT $SYSTEM.SYSnn.SYSDISC
```

3. Reset and start the disk:

```
-> RESET $DATA01
-> START $DATA01
```

Enabling and Disabling File Opens on a Disk

Preventing File Opens on a Disk

The "STOPOPENS DISK Command" (page 293) prevents applications from opening files on a specific disk.

Considerations for STOPOPENS DISK

CAUTION: Do not issue a STOPOPENS DISK command on the system disk. Issuing a STOPOPENS DISK command on the system disk makes it inaccessible to file-open attempts. If you do issue a STOPOPENS DISK command on the system disk, do not exit SCF. Immediately enter an ALLOWOPENS DISK, SUPERONLY command. Otherwise, the system disk becomes inaccessible when you exit SCF.

- The disk must be in the STARTED state.
- When you issue the STOPOPENS command, SCF provides a message indicating the number of currently open files.
- If the disk to which you issue a STOPOPENS command is in a storage pool, SCF issues a warning message. Problems can occur because the storage pool process is not advised of this change.
- If the disk to which you issue a STOPOPENS command contains SMF catalog files that are not currently not open, the SMF process that uses those catalogs cannot be started.
- Use the "ALLOWOPENS Command" (page 227) to reverse the action of the STOPOPENS command.

Examples

1. Before issuing the STOPOPENS command, verify the disk is in the STARTED state.
2. Stop file-system opens:
   - To prevent new file-system opens on a disk:

     ```
     -> STOPOPENS $DATA00
     ```

   - To prevent new file-system opens on all disks in the STARTED state:

     ```
     -> STOPOPENS DISK $*, SEL STARTED
     ```

Allowing File Opens on a Disk

The "ALLOWOPENS DISK Command" (page 227) lets applications open files on a specific disk.

Examples

- To allow files on a disk to be opened by applications:

  ```
  -> ALLOWOPENS $DATA00
  ```

- To prevent files on a disk from being opened by anyone except the super ID (255, 255):

  ```
  -> ALLOWOPENS $DATA14, SUPERONLY
  ```

Swapping Processors for a Disk

The "PRIMARY DISK Command" (page 264) swaps the primary and backup processors for a disk. The current primary processor becomes the backup processor, and the backup processor becomes the primary processor, but the PRIMARYCPU and BACKUPCPU values stay the same

NOTE: To change the primary or backup processor values for a disk, stop all the disk paths (STOP DISK command) or terminate the disk process (RESET DISK, FORCED). Then you can alter the disk configuration with new PRIMARYCPU and BACKUPCPU values.

You typically swap processors for a disk when load balancing the system or preparing for disk replacement.

PRIMARY DISK Consideration

For G06.10 and earlier RVUs, using the PRIMARY command to swap a disk to its backup processor can affect the other disk processes using the same SACs. See:

- “PRIMARY Command Function in G06.00 Through G06.10” (page 125)
- “PRIMARY Command Function in G05.00 and Earlier” (page 128)

An Example for G06.11 and Later RVUs

On G06.11 and later RVUs, SACs in the same topology branch are owned by both processors. The PRIMARY command can change which processor actively uses a SAC, but it does not change SAC ownership.

Because both processors in the same topology branch can use the same SAC, the PRIMARY command does not create an access problem for the other processor.

In the figures that follow:

- SACs and disks controlled by processor 2 are shaded.
- Active paths, fabrics, and SACs are in bold.
- Backup disk processes are represented with a prime mark: for example, $D0201'.

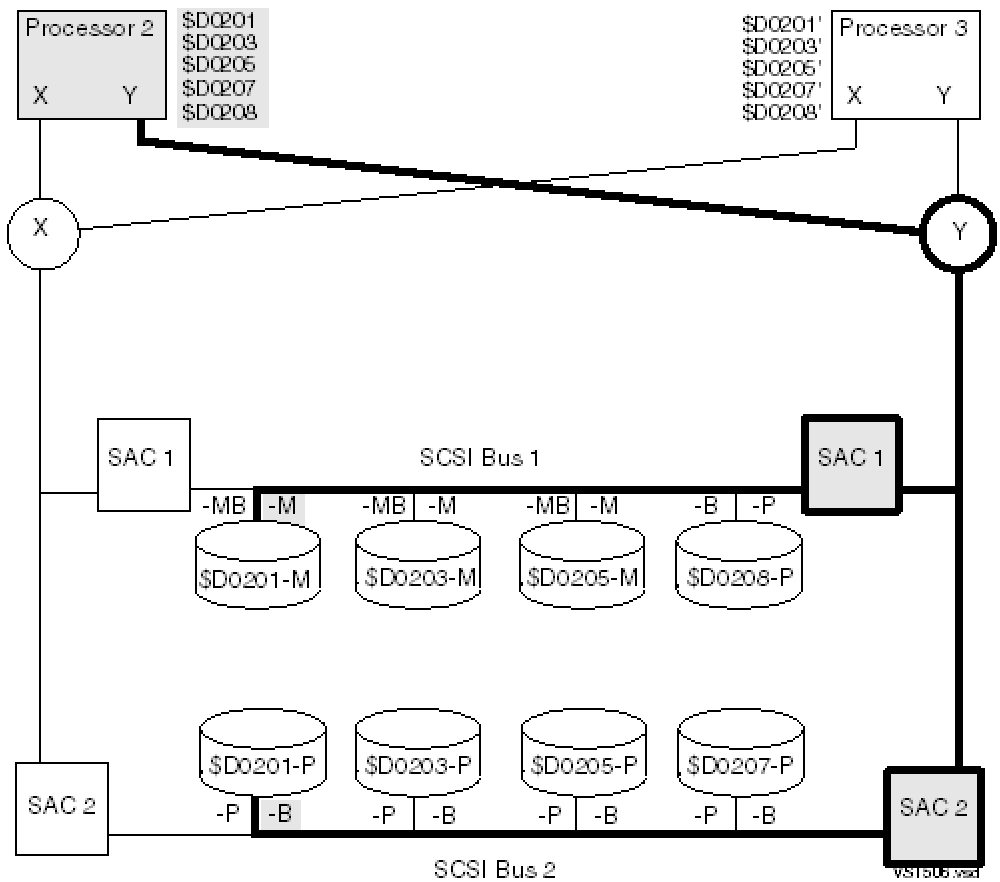
1. Display the current status of the disk on which you intend to swap the processors:

```
-> STATUS DISK $D0201-*

STORAGE - Status DISK \WAGER.$D0201-*

LDev    Path            Status        State      Substate    Primary   Backup
                                                             PID       PID

104     PRIMARY         ACTIVE        STARTED                 2,107     3,35
104     BACKUP          INACTIVE      STARTED                 2,107     3,35
104     MIRROR          ACTIVE        STARTED                 2,107     3,35
104     MIRROR-BACKUP   INACTIVE      STARTED                 2,107     3,35
```



2. Move the primary disk process to processor 3:

```
-> PRIMARY $D0201, 3
```

3. Display the changed status:

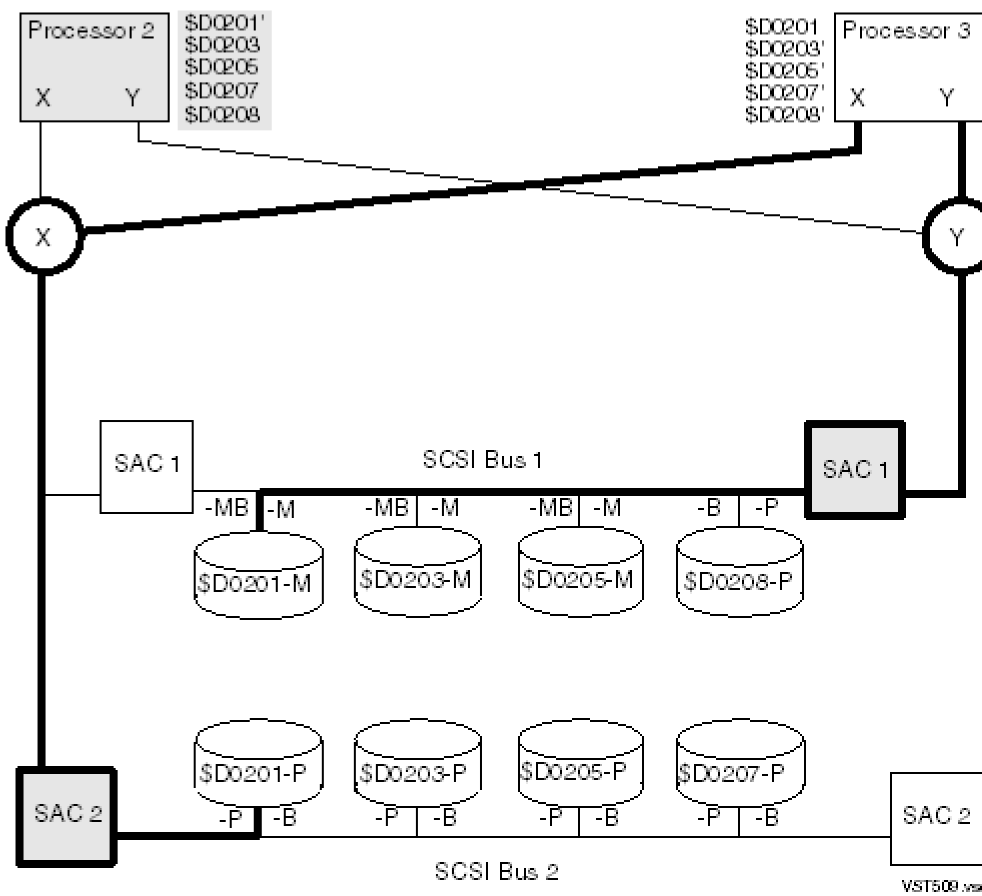
```
-> STATUS DISK $D0201-*

STORAGE - Status DISK \WAGER.$D0201-*

LDev    Path            Status        State      Substate    Primary   Backup
                                                             PID       PID

104     PRIMARY         ACTIVE        STARTED                  3,35      2,107
104     BACKUP          INACTIVE      STARTED                  3,35      2,107
104     MIRROR          ACTIVE        STARTED                  3,35      2,107
104     MIRROR-BACKUP   INACTIVE      STARTED                  3,35      2,107
```

Processor 3 now controls the primary disk process for $D0201, and processor 2 controls the backup disk process.



Changing the Active Path for a Disk

The "SWITCH DISK Command" (page 295) designates the active path to a disk, determines whether the -P or -B path is active and, if the disk is mirrored, whether the -M or -MB path is active.

You should configure the -P and -M paths to be active because they are on opposite ServerNet fabrics.

Considerations for SWITCH DISK

- If you attempt to switch the primary and backup paths but the backup path is unavailable, the paths are not switched.
- If the specified path is already active, the SWITCH command has no effect.
- Use the STATUS DISK command to verify the path switch.
- Switching the path to the backup controller can create a SAC ownership conflict in G06.10 and earlier RVUs. See “Understanding SAC Ownership” (page 123).

Switching the Active Path for a Disk

1. Display the current status of the disk paths:

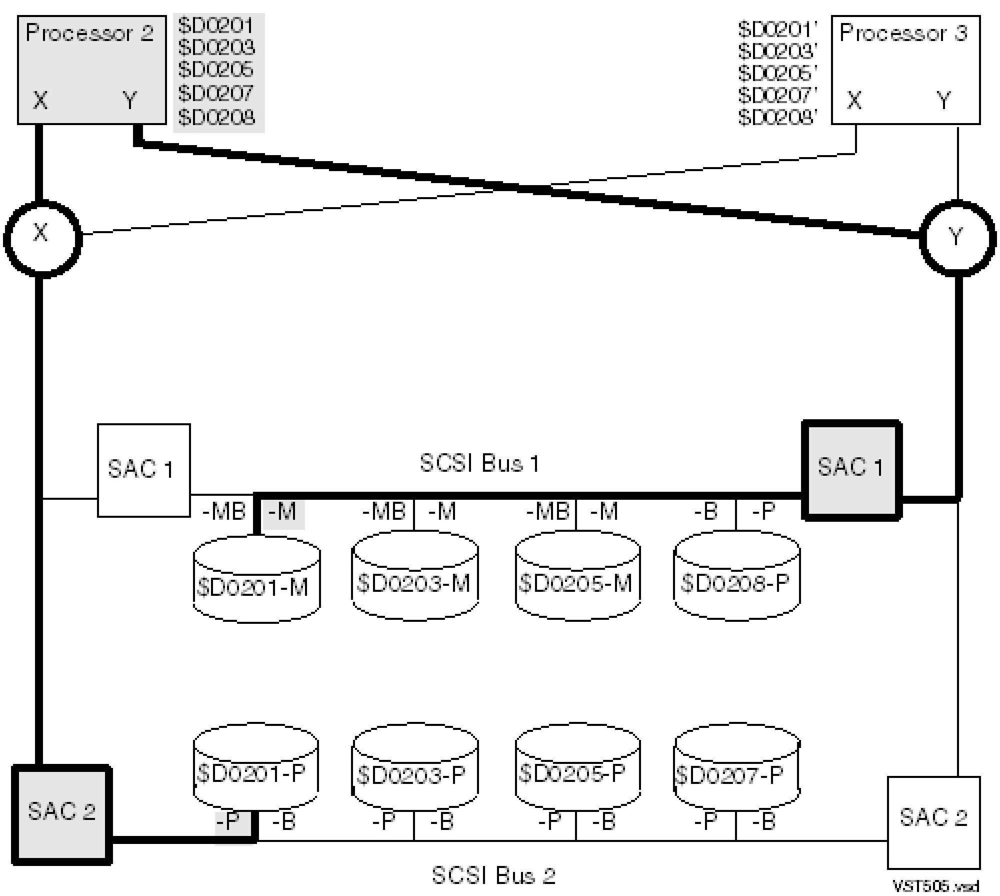
```
-> STATUS $D0201-*

STORAGE - Status DISK \WAGER.$D0201-*

LDev    Path             Status       State       Substate    Primary  Backup
                                                              PID      PID

104     PRIMARY          ACTIVE       STARTED                  2,107    3,35
104     BACKUP           INACTIVE     STARTED                  2,107    3,35
104     MIRROR           ACTIVE       STARTED                  2,107    3,35
104     MIRROR-BACKUP    INACTIVE     STARTED                  2,107    3,35
```

- The $D0201 disk process is still executing in processor 2.
- The $D0201 disk is now accessed through SACs in the same PMF CRU.
- The disk process is now using the backup path to access the primary disk.
- The mirror path is unaffected by the switch.
- Both paths now use the Y fabric.



- The $D0201 disk process is executing in processor 2.
- The $D0201 disk is accessed through SACs in two different PMF CRUs.
- The disk process is using the primary path to access the primary disk.
- The paths use the X and Y fabrics.

2. Change the active path accessing the primary disk:

```
-> SWITCH $D0201-B
```

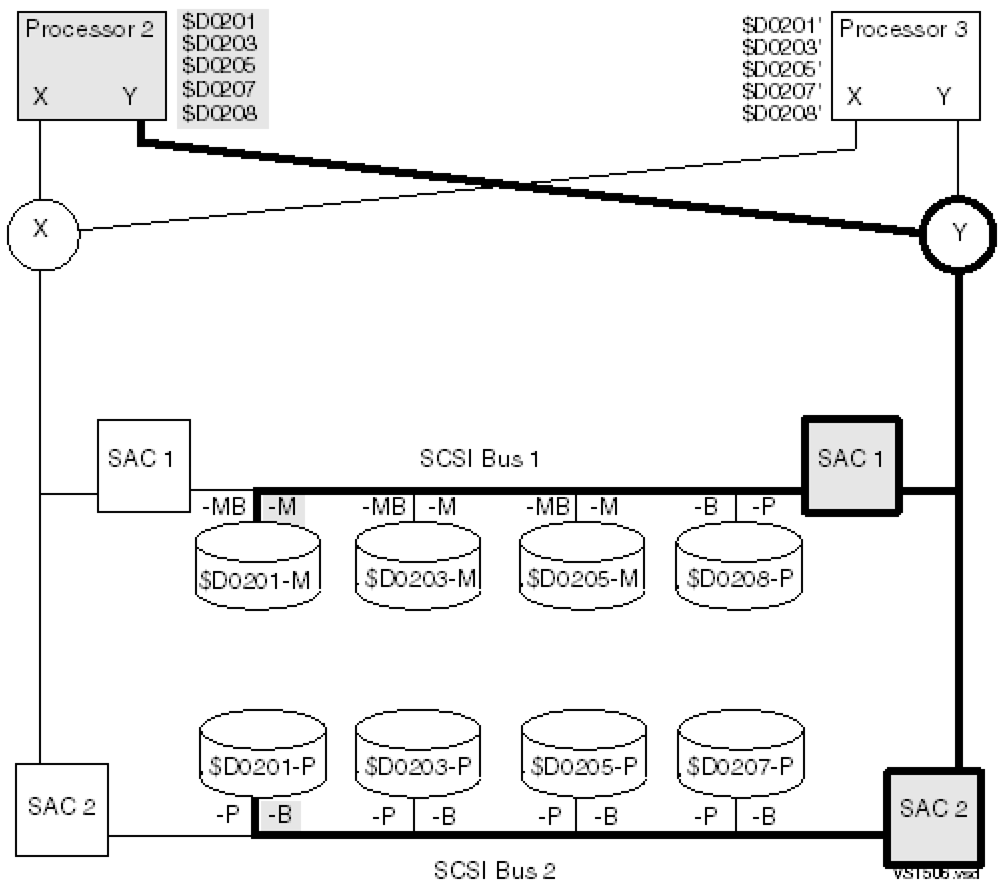
3. Display the changed status:

```
-> STATUS $D0201-*

STORAGE - Status DISK \WAGER.$D0201-*

LDev    Path            Status       State      Substate    Primary   Backup
                                                            PID       PID

104     PRIMARY         INACTIVE     STARTED                 2,107     3,35
104     BACKUP          ACTIVE       STARTED                 2,107     3,35
104     MIRROR          ACTIVE       STARTED                 2,107     3,35
104     MIRROR-BACKUP   INACTIVE     STARTED                 2,107     3,35
```



4. Switch the active path accessing the mirror disk:

```
-> SWITCH $D0201-MB
```

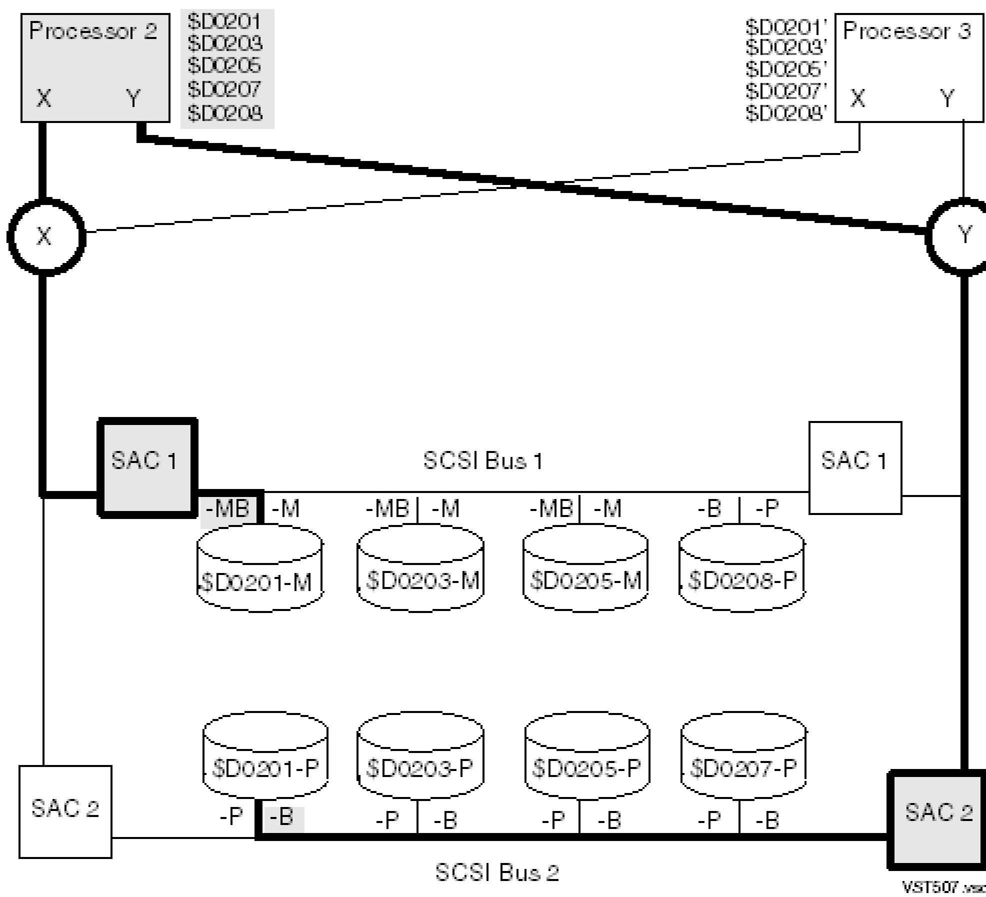
5. Display the changed status:

```
-> STATUS DISK $D0201-*

STORAGE - Status DISK \WAGER.$D0201-*

LDev   Path           Status      State      Substate  Primary  Backup
                                                       PID      PID

104    PRIMARY        INACTIVE    STARTED               2,107    3,35
104    BACKUP         ACTIVE      STARTED               2,107    3,35
104    MIRROR         INACTIVE    STARTED               2,107    3,35
104    MIRROR-BACKUP  ACTIVE      STARTED               2,107    3,35
```



Managing Encrypted Disk Drives

SCF supports encryption of data-at-rest for disk objects that are connected with CLIMs. Encryption uses keys generated and stored by the HP Enterprise Secure Key Manager (ESKM). Customers use the ALTER, INITIALIZE, and STATUS commands to manage encryption on disks.

Only members of the SAFEGUARD security officer group on the local system can perform an ALTER DISK command with the CLEARENCRYPTKEY and NEWENCRYPTKEY attributes or an INITIALIZE command with NEWENCRYPTKEY. Only security officers can initiate a revive from an encrypted disk to a non-encrypted disk. Security officers must also be members of the Guardian SUPER.* group. Keys and system security should be managed by customer security officers, not system administrators.

For details about encryption, see the *NonStop Volume Level Encryption Guide*.

Displaying Information about Encrypted Disks

To display information about encrypted disks, see "Displaying Information about Encrypted Disks" (page 58).

Changing the Speed of Encryption Key Rotation

While an encryption key rotation operation is in progress, you can use the ALTER command to change its speed.

Considerations for Changing the Speed of an Encryption Key Rotation

- The defaults are 50 for ENCRYPTRATE and 4 for ENCRYPTPRIORITY. The default values limit potential interference with system performance.
- To speed up the encryption key rotation (even though this change might slow system performance), increase the ENCRYPTPRIOITY value and/or increase the ENCRYPTRATE value.
- You may change these values only while an encryption key rotation is in progress. The new values affect the ongoing encryption key rotation from the point at which you entered the new values. They have no effect on future encryption key rotations.

Example of Changing the Speed of an Encryption Key Rotation

This command establishes an encryption priority of 6 and an encryption rate of 70:

```
ALTER $ENCM21-P, ENCRYPTIONPRIORITY 6, ENCRYPTRATE 70
```

Handling Power Failures

Hard Disk Drives (HDDs) and Solid State Drives (SSDs)

When WRITECACHE is enable, the HDD and SSD DRAM caches need to be written to disk media after a power fail.

CAUTION: HP strongly recommends that the drive be protected by a cabinet or an HP rack mounted UPS.

For information about caching in the event of power failure, refer to "Considerations for WRITECACHE" (page 94).

Storage CLIMs

A Storage CLIM with the write cache enable option selected and a local HP rack mount UPS on that rack can preserve power long enough for the cached data to be flushed to disk.

CAUTION: If the WRITECACHE disk attribute is ENABLED and there is no HP rack mounted UPS, data in the disk cache might not be written to the disk media and you could lose data.

8 Disk Load Balancing

Disk load balancing seeks to distribute the disk work load across all fabrics and processors in the system. This chapter describes load balancing between the X and Y fabrics and pairs of processors in a NonStop S-series processor enclosure. A processor enclosure and its I/O enclosures are called a topology branch.

The principles that apply to the internal disks in these examples also apply to 45*xx* disks in a modular disk subsystem.

Using reports from products like Guardian Performance Analyzer and Measure, you can assign each disk in a topology branch to run its primary processes in a specific processor. For more information, see the *GPA Manual* and *Measure User's Guide*. This chapter describes:

Disk Load Balancing Between Fabrics

Because a storage adapter in a NonStop S-series server is permanently assigned to a fabric (adapters in slots 50, 51, and 53 use the X fabric; adapters in slots 52, 54, and 55 use the Y fabric), you can examine a detailed STATUS ADAPTER report to see whether the mirrored disks on an storage adapter are balanced between the X and Y fabrics.

Examine this display:

```
-> STATUS ADAPTER $ZZSTO.#*.GRP-3*, DETAIL
```

The disks are balanced if the primary (-P or -B) disk or the mirror (-M or -MB) disk has an asterisk.

For example, if $OPER on the PMF CRU in slot 50 is balanced between the X and Y fabrics, the primary (-P or -B) disk OR the mirror (-M or -MB) disk has an asterisk in a detailed INFO ADAPTER display:

```
Slot 50:
2   $OPER-MB       STARTED                      0,298     1,269
 ...
1   $OPER-P       *STARTED                      0,298     1,269
```

Correspondingly, if $OPER on the PMF CRU in slot 55 is balanced between X and Y, the primary (-P or -B) disk OR the mirror (-M or -MB) disk has an asterisk:

```
Slot 55:
2   $OPER-M       *STARTED                     0,298     1,269
 …
1   $OPER-B        STARTED                     0,298     1,269
```

However, these asterisks illustrate an unbalanced situation:

```
Slot 50:
2   $OPER-MB       STARTED                     0,298     1,269
 …
1   $OPER-P        STARTED                     0,298     1,269

Slot 55:
2   $OPER-M       *STARTED                     0,298     1,269
 …
1   $OPER-B       *STARTED                     0,298     1,269
```

Both halves of the mirrored volume are running on one fabric (in this case, the Y fabric).

To restore the balance between fabrics, switch the primary process to $OPER-P:

```
-> SWITCH $OPER-P
```

For detailed guidelines about how to use the SWITCH command, see "Changing the Active Path for a Disk" (page 112).

Disk Load Balancing Between Processors

This subsection describes principles of disk load balancing between processors and how to achieve an ideal balance. The technique for obtaining an ideal balance depends on the RVU:

RVU
"Guidelines for All G-Series RVUs" (page 119)
"Guidelines for G06.11 and Later RVUs" (page 119)
"Guidelines for G06.10 and Earlier RVUs" (page 123)
"PRIMARY Command Function in G06.00 Through G06.10" (page 125)
"PRIMARY Command Function in G05.00 and Earlier" (page 128)

Guidelines for All G-Series RVUs

Regardless of RVU:

- When you change a disk path or primary processor, the change does not actually take place until the next I/O action on the disk.
- The active paths to any mirrored disk should be either the primary and mirror paths or the backup and mirror-backup paths. This distribution ensures fault-tolerance because the disks are accessed through separate system components.
- The same load-balancing principles apply to both internal disks and 45*xx* disks.

Most of the examples in this section use the naming conventions documented in the *NonStop S-Series Planning and Configuration Guide*.

Guidelines for G06.11 and Later RVUs

- A processor can access any disk in its topology branch. Both processors in the processor enclosure jointly own each SAC in each enclosure in the topology branch.
- The PRIMARY command specifies the processor in which the IOP should run its primary process; it has no effect on the paths being used by other disk processes in the same topology branch.

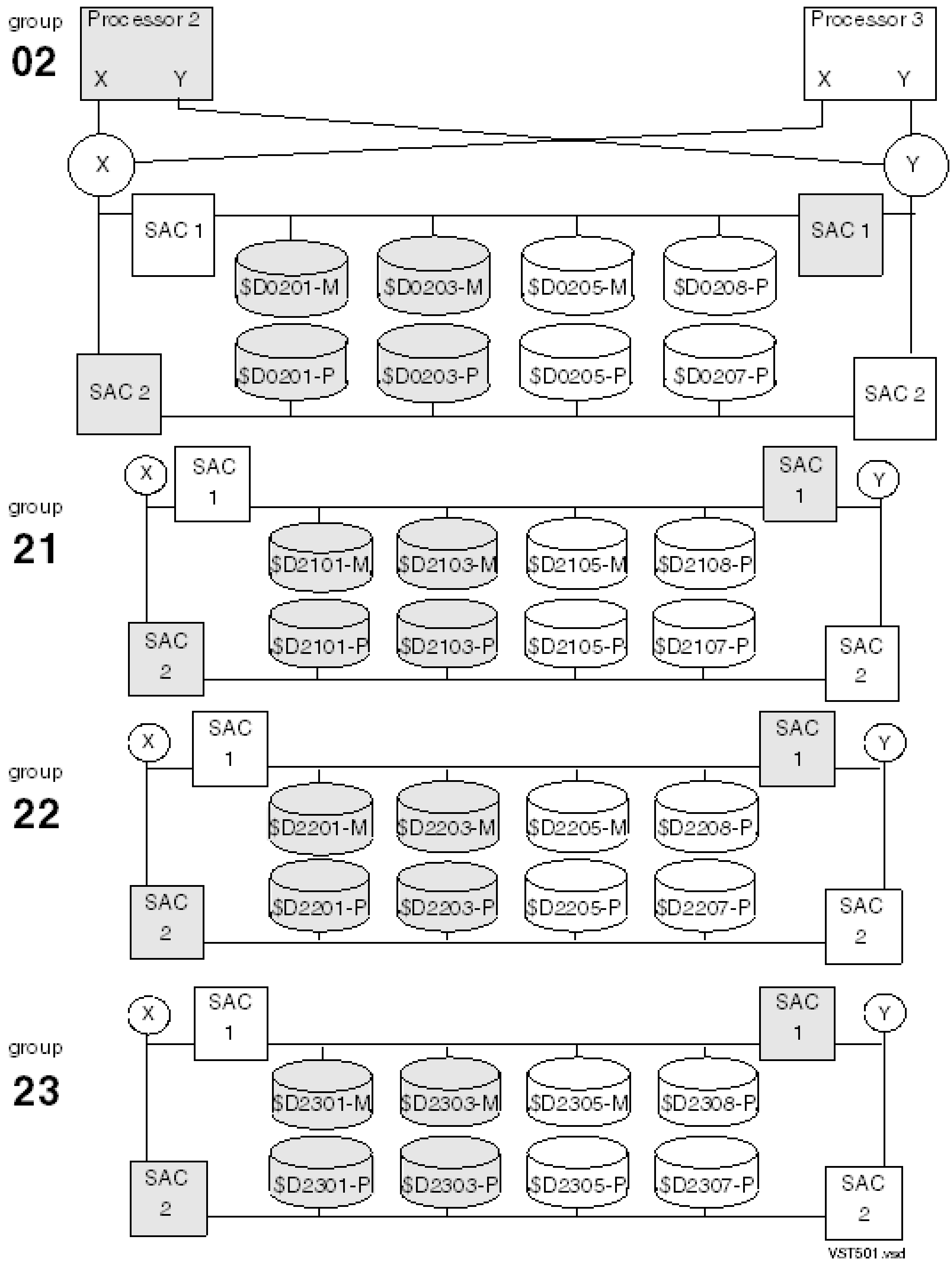
For detailed guidelines about how to use the PRIMARY command, see "Swapping Processors for a Disk" (page 109).

The disk processing load within each enclosure should be evenly balanced between the processor pair in that enclosure. Assuming that each disk handles the same processing load, each enclosure should have one processor managing half of the primary disk processes and the other processor managing the other half. If one disk has to support a heavier processing load, you need to primary fewer disks from that processor.

Topology Branch Example for G06.11 and Later RVUs

The group 02 topology branch includes processors 2 and 3 in the group 02 processor enclosure and the three I/O enclosures attached to it (groups 21, 22, and 23). In the figure below, the disk processes are configured as follows:

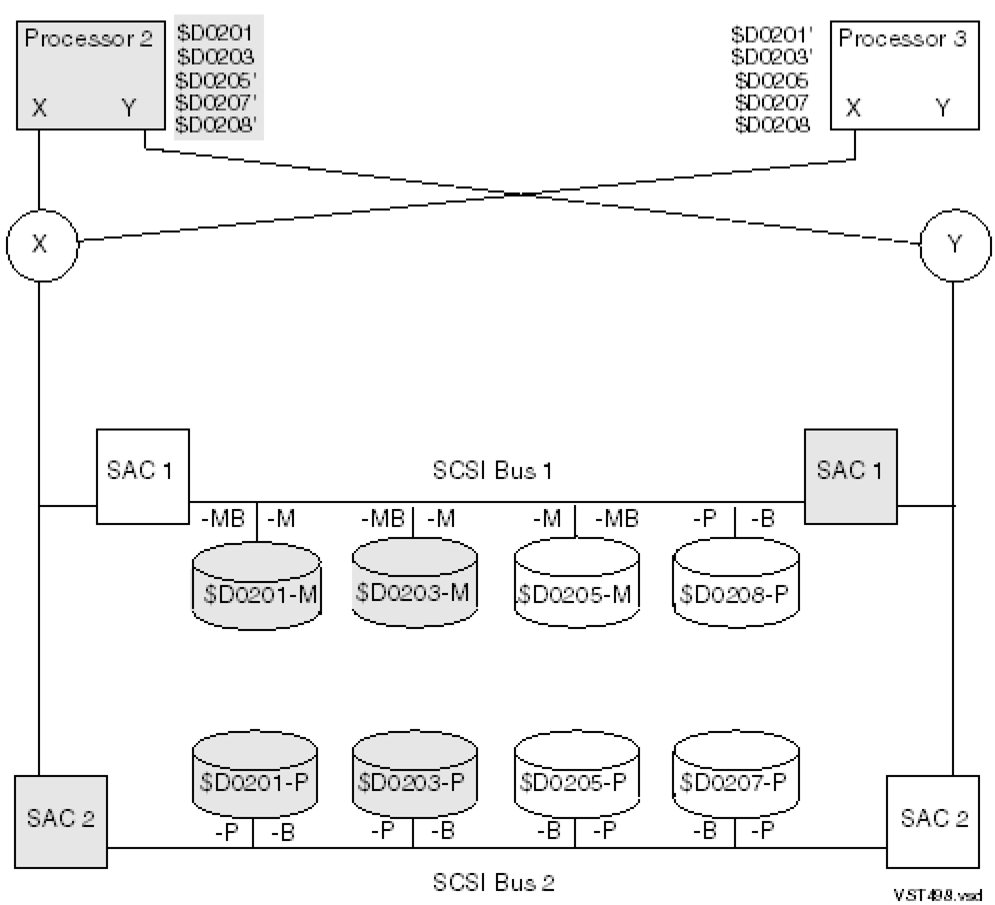
- Processors 2 and 3 jointly own all SACS in the same topology branch (groups 02, 21, 22, and 23).
- Processor 2 manages the primary disk processes using the **shaded** SACs and disks.
- Processor 3 manages the primary disk processes using the **white** SACs and disks.
- For simplicity, four disks are shown connected to each communication bus. The enclosure can have up to eight disks per bus (16 disks in each enclosure).


group
02
Processor 2
X
Y
Processor 3
X
Y
X
Y
SAC 1
SAC 1
$D0201-M
$D0203-M
$D0205-M
$D0208-P
$D0201-P
$D0203-P
$D0205-P
$D0207-P
SAC 2
SAC 2
group
21
X
SAC 1
SAC 1
Y
$D2101-M
$D2103-M
$D2105-M
$D2108-P
$D2101-P
$D2103-P
$D2105-P
$D2107-P
SAC 2
SAC 2
group
22
X
SAC 1
SAC 1
Y
$D2201-M
$D2203-M
$D2205-M
$D2208-P
$D2201-P
$D2203-P
$D2205-P
$D2207-P
SAC 2
SAC 2
group
23
X
SAC 1
SAC 1
Y
$D2301-M
$D2303-M
$D2305-M
$D2308-P
$D2301-P
$D2303-P
$D2305-P
$D2307-P
SAC 2
SAC 2
VST501.vsd

A Closer Look at Eight Disks

Looking more closely at the disks in the group 02 processor cabinet, this configuration is an ideal situation for G06.11 and later RVUs:

- Processor 2 controls these disks (**shaded**):
  - The primary disk processes for mirrored disks $D0201 and $D0203
  - The backup disk process (represented with a prime mark, ') for mirrored disk $D0205'
  - The backup disk processes for nonmirrored disks $D0207' and $D0208'
- Processor 3 controls these disks (**white**):
  - The backup disk processes (represented with a prime mark, ') for mirrored disks $D0201' and $D0203'
  - The primary disk process for mirrored disk $D0205
  - The primary disk process for nonmirrored disks $D0207 and $D0208
- Each processor controls one SAC.



Disk Load Balancing on G06.10 and Earlier RVUs

Understanding SAC Ownership

On G06.10 and earlier RVUs each SAC is owned by a processor. A mirrored volume can communicate with its processors through up to four SACs. These events can cause SAC ownership to change:

- A hardware failure along an active data path to that SAC.
- A PRIMARY command on a started disk.
- A processor failure or halt on a primary processor.

How a disk process responds to a loss of SAC ownership depends on the RVU:

RVU	Response
G06.11 and later	SACS are no longer exclusively owned. This is no longer an issue.
G06.00 through G06.10	The disk process switches its primary process to the alternate processor. See “PRIMARY Command Function in G06.00 Through G06.10” (page 125)
G05.00 and earlier	The disk process tries to switch to the alternate path. See “PRIMARY Command Function in G05.00 and Earlier” (page 128)

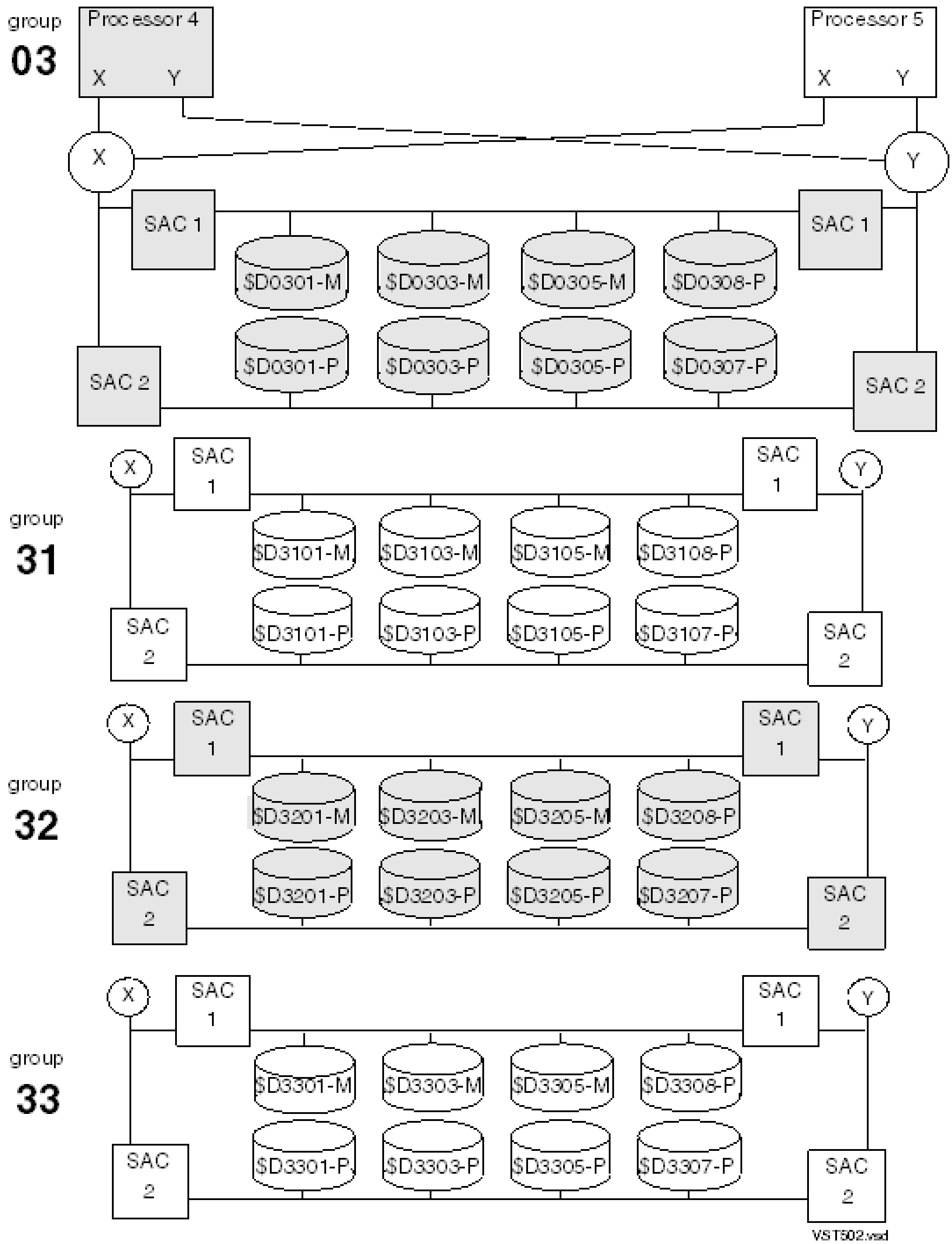
Guidelines for G06.10 and Earlier RVUs

- The disk load balance within an enclosure should be one-sided. Ideally, each enclosure should have one primary processor that:
  - Runs all disk processes managing disk volumes in that enclosure
  - Owns all SACs in that enclosure
- When you use the PRIMARY DISK command and it results in a SAC ownership change, all the other disks that are actively using that SAC are affected.

Topology Branch Example for G06.10 and Earlier RVUs

The group 03 topology branch includes processors 4 and 5 in the group 03 processor enclosure and the three I/O enclosures attached to it (groups 31, 32, and 33). The disk processes are configured as follows:

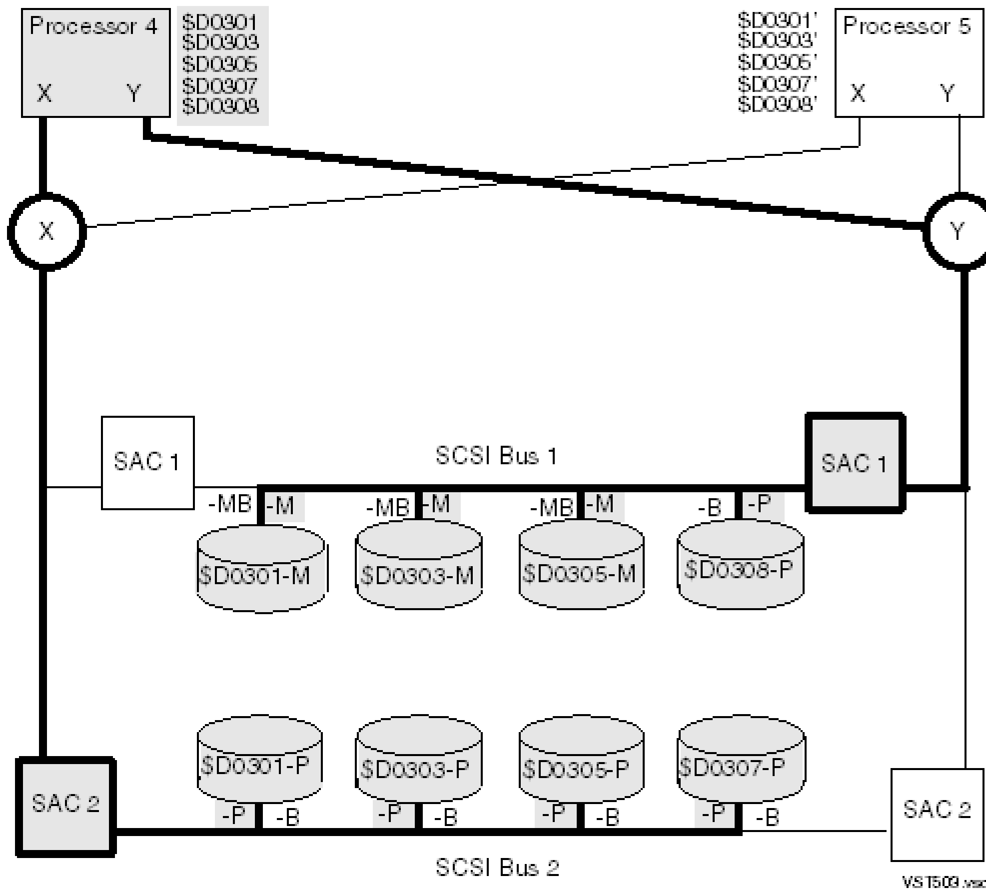
- Processor 4 owns the shaded SACS and manages the primary disk processes that are running in the shaded disks.
- Processor 5 owns the white SACs and manages the primary disk processes that are running in the white disks.



A Closer Look at Eight Disks

Looking more closely at the disks in the group 03 processor cabinet, this configuration is the ideal situation for G06.10 and earlier RVUs:

- All primary disk processes run in processor 4.
- All backup disk processes, represented with a prime mark (') run in processor 5.
- Processor 4 owns all four SACs.



PRIMARY Command Function in G06.00 Through G06.10

On G05.00 through G06.10, the PRIMARY command changes processor ownership of active paths. (The FORCED option forces all SACs to change ownership.) This change in ownership in turn causes other IOPs in the same topology branch to perform additional path switches or processor swaps to recover from the loss of SAC ownership.

1. Before the PRIMARY command is issued, the status of $D0203 is:

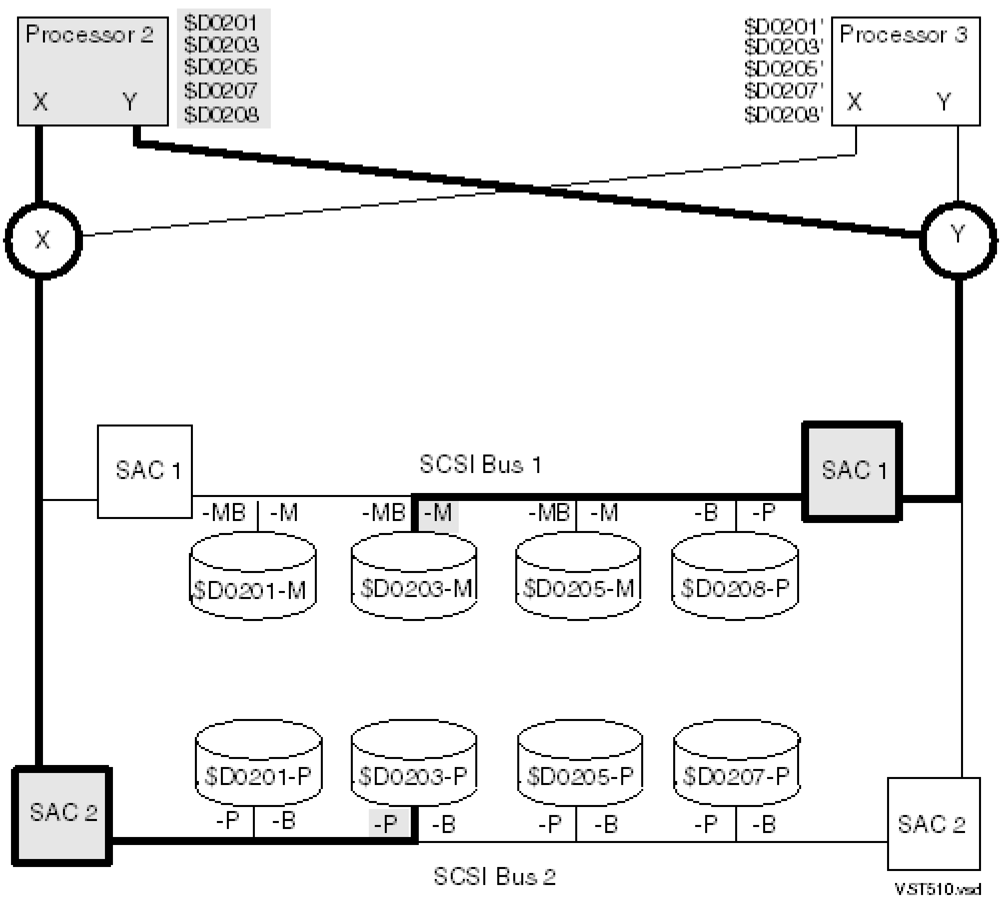
```
STORAGE - Status DISK \WAGER.$D0203-*

LDev    Path            Status        State      Substate    Primary  Backup
                                                             PID      PID

105     PRIMARY         ACTIVE        STARTED                 2,22     3,16
105     BACKUP          INACTIVE      STARTED                 2,22     3,16
105     MIRROR          ACTIVE        STARTED                 2,22     3,16
105     MIRROR-BACKUP   INACTIVE      STARTED                 2,22     3,16
```

In the figures that follow:

- SACs owned by processor 2 are shaded.
- Disk processes that are primaried in processor 2 are shaded.
- Primary paths are the bold lines.
- Backup disk processes are represented with a prime mark, such as $D0201'.



2. Move the primary disk process of **another** disk in the same topology branch to processor 3:

```
-> PRIMARY $D0201, 3
```

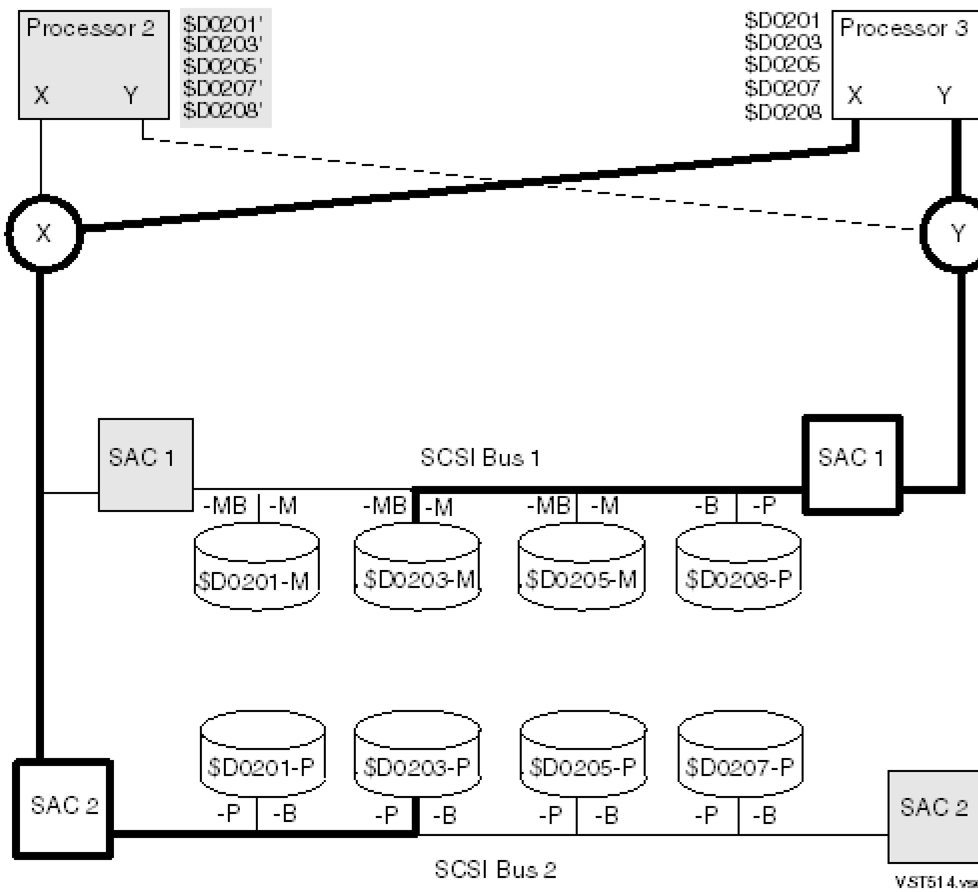
3. Display the changed status:

```
-> STATUS DISK $D0203-*

STORAGE - Status DISK \WAGER.$D0203-*

LDev    Path            Status        State      Substate    Primary  Backup
                                                              PID      PID

104     PRIMARY         ACTIVE        STARTED                  3,35     2,107
104     BACKUP          INACTIVE      STARTED                  3,35     2,107
104     MIRROR          ACTIVE        STARTED                  3,35     2,107
104     MIRROR-BACKUP   INACTIVE      STARTED                  3,35     2,107
```



$D0203 also changes primary processors in because:

1. The disk process executing in processor 2 attempts to access $D0203-P.
2. The path to $D0203-P on the X fabric leads through SAC 2. Because of the PRIMARY command on the other disk, this SAC is now owned by processor 3. Because the disk process is executing in processor 2, the disk access fails.
3. The disk process detects the loss of SAC ownership, changes processors, and tries to access $D0203 through processor 3. Because this SAC is now owned by processor 3, the disk access succeeds.
4. The disk process, now executing in processor 3, attempts to access $D0203-M.

   The path to $D0203-M on the Y fabric leads through SAC 1. Because this SAC is also owned by processor 3, the disk access succeeds.

This process affects all disk processes that use these SACs.

NOTE: When ownership of both SACs is lost, a read from one or both paths detects a loss of ownership. However, a read from one path whose SAC ownership was not lost would not detect the loss of ownership of the SAC on the path to the other disk.

PRIMARY Command Function in G05.00 and Earlier

On G05.00 and earlier RVUs, the PRIMARY command changes SAC ownership.

Before the PRIMARY command is issued, the status of $D0203 is:

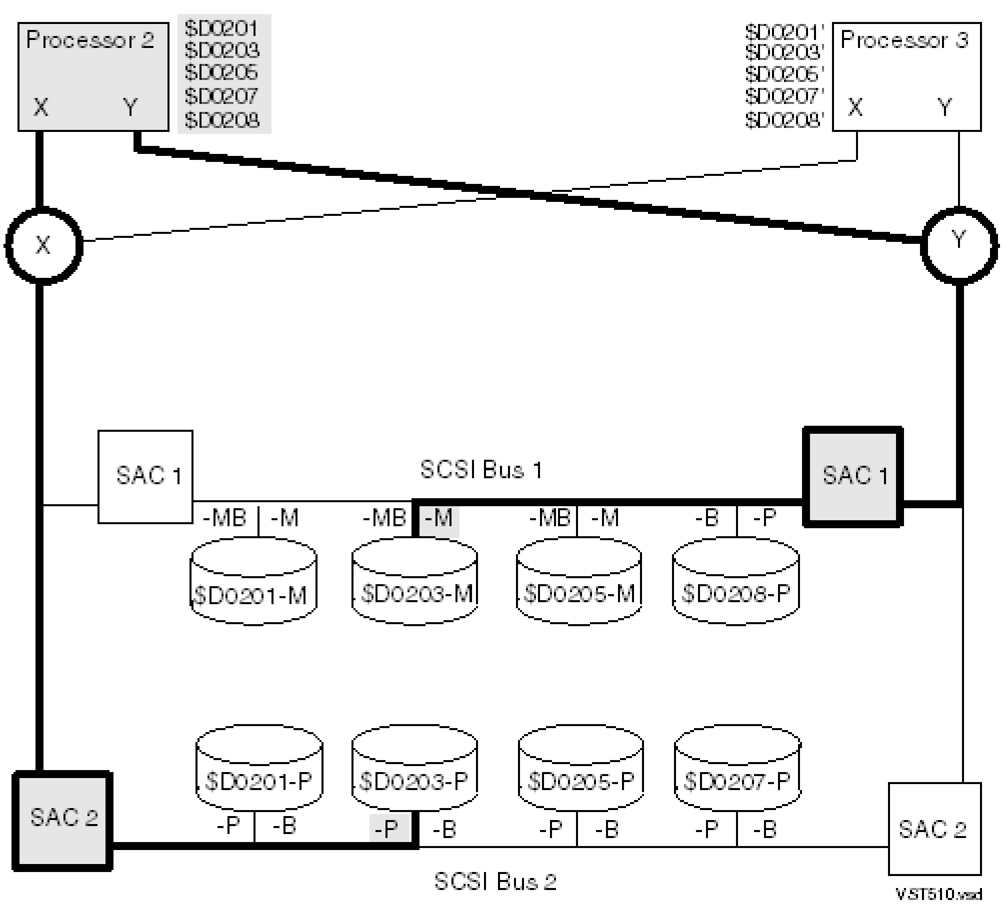
```
STORAGE - Status DISK \WAGER.$D0203-*

LDev    Path             Status        State      Substate    Primary  Backup
                                                              PID      PID

105     PRIMARY          ACTIVE        STARTED                 2,22     3,16
105     BACKUP           INACTIVE      STARTED                 2,22     3,16
105     MIRROR           ACTIVE        STARTED                 2,22     3,16
105     MIRROR-BACKUP    INACTIVE      STARTED                 2,22     3,16
```

In the figures that follow:

- SACs owned by processor 2 are shaded.
- Disk processes that are primaried in processor 2 are shaded.
- Primary paths are the bold lines.
- Backup disk processes are represented with a prime mark, such as $D0201'.



1. Move the primary disk process of **another** disk in the same topology branch to processor 3:

```
-> PRIMARY $D0201, 3
```

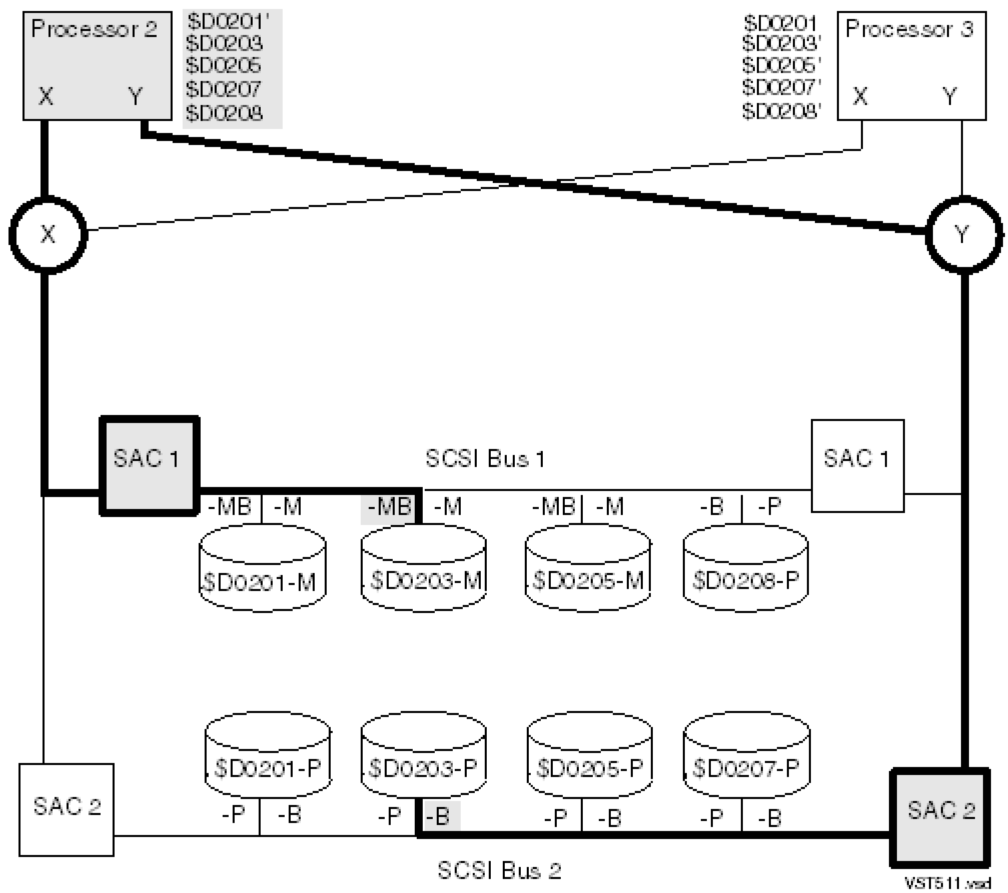
2. Display the changed status:

```
-> STATUS DISK $D0203-*

STORAGE - Status DISK \WAGER.$D0203-*

LDev   Path           Status       State      Substate   Primary  Backup
                                                         PID      PID

105    PRIMARY        INACTIVE     STARTED                2,22     3,16
105    BACKUP         ACTIVE       STARTED                2,22     3,16
105    MIRROR         INACTIVE     STARTED                2,22     3,16
105    MIRROR-BACKUP  ACTIVE       STARTED                2,22     3,16
```



$D0203 changes paths to access the disk in because:

1. The disk process executing in processor 2 attempts to access $D0203-P.
2. The path to $D0203-P on the X fabric leads through SAC 2. This SAC is now owned by processor 3. Because the disk process is executing in processor 2, the disk access fails.
3. The disk process changes paths and tries to access $D0203-B.

   The path to $D0203-B on the Y fabric leads through SAC 2. Because this SAC is still owned by processor 2, the disk access succeeds.

4. The disk process, still executing in processor 2, attempts to access $D0203-M.

   The path to $D0203-M on the Y fabric leads through SAC 1. This SAC is now owned by processor 3. Because the disk process is executing in processor 2 but the SAC is owned by processor 3, the disk access fails.

   The disk process changes paths and tries to access $D0203-MB.

   The path to $D0203-MB on the X fabric leads through SAC 1. Because this SAC is still owned by processor 2 the disk access succeeds.

This process affects all disk processes that use these SACs.

NOTE: Loss of SAC ownership is detected only when an I/O is attempted. When ownership of both SACs is lost, a read from one or both paths detects a loss of ownership. However, a read from one path whose SAC ownership was not lost would not detect the loss of ownership of the SAC on the path to the other disk.

Establishing a Disk Load Balance in RVUs Prior to G06.11

In RVUs prior to G06.11, you can simplify disk load balancing if you first establish a starting point for SAC ownership so you know the owner of all SACs.

You enforce SAC ownership by using the PRIMARY DISK, FORCED command, which produces this behavior:

- A specified disk primary process must run in the specified processor.
- All SACs that are used to access the disk become owned by that processor.
- All disk processes using those SACs must use that processor.
- Force all mirrored disks in the group 02 topology branch to run in processor 2:

  ```
  -> PRIMARY $D02*, 2, FORCED
  ```

1. Change SAC ownership for all group 02 disks to processor 2:

   ```
   -> INFO DISK $D02*, LABEL
   ```

2. Again swap the processors for these same disks, this time to processor 3:

   ```
   -> PRIMARY $D0201, 3, FORCED
   ```

3. Change SAC ownership to processor 3:

   ```
   -> INFO DISK $D02*-*, LABEL
   ```

4. If you want all these disks to run in processor 2, enter one final PRIMARY command:

   ```
   -> PRIMARY $D02*, 2, FORCED
   ```

5. Change SAC ownership to processor 2:

   ```
   -> INFO DISK $D02*, LABEL
   ```

6. The status shows all primary processes in processor 2:

   ```
   -> STATUS $D02*-*

   STORAGE - Status DISK \WAGER.$D0201-*

   LDev   Path           Status      State     Substate   Primary  Backup
                                                          PID      PID
   104    PRIMARY        ACTIVE      STARTED               2,107    3,35
   104    BACKUP         INACTIVE    STARTED               2,107    3,35
   104    MIRROR         ACTIVE      STARTED               2,107    3,35
   104    MIRROR-BACKUP  INACTIVE    STARTED               2,107    3,35


   STORAGE - Status DISK \WAGER.$D0203-*

   LDev   Path           Status      State     Substate   Primary  Backup
                                                          PID      PID
   ```

```
105    PRIMARY        ACTIVE       STARTED                 2,22     3,16
105    BACKUP         INACTIVE     STARTED                 2,22     3,16
105    MIRROR         ACTIVE       STARTED                 2,22     3,16
105    MIRROR-BACKUP  INACTIVE     STARTED                 2,22     3,16


STORAGE - Status DISK \WAGER.$D0205-*

LDev   Path           Status       State      Substate   Primary  Backup
                                                         PID      PID
105    PRIMARY        ACTIVE       STARTED                 2,112    3,45
105    BACKUP         INACTIVE     STARTED                 2,112    3,45
105    MIRROR         ACTIVE       STARTED                 2,112    3,45
105    MIRROR-BACKUP  INACTIVE     STARTED                 2,112    3,45


STORAGE - Status DISK \WAGER.$D0207-*

LDev   Path           Status       State      Substate   Primary  Backup
                                                         PID      PID
112    PRIMARY        ACTIVE       STARTED                 2,118    3,237
112    BACKUP         INACTIVE     STARTED                 2,118    3,237


STORAGE - Status DISK \WAGER.$D0208-*

LDev   Path           Status       State      Substate   Primary  Backup
                                                         PID      PID
114    PRIMARY        ACTIVE       STARTED                 2,122    3,86
114    BACKUP         INACTIVE     STARTED                 2,122    3,86
```

7. Now balance the disk work load across fabrics: switch the disk paths to ensure that all four SACs are used for I/O operations. These SWITCH commands force half the disk processes to use different paths to access the same disk:

```
-> SWITCH ($D0201-P, $D0201-M)
-> SWITCH ($D0203-B, $D0203-MB)
-> SWITCH ($D0205-P, $D0205-M)
-> SWITCH ($D0207-B, $D0208-B)
```

8. Display the changed status:

```
-> STATUS $D02*-*

STORAGE - Status DISK \WAGER.$D0201-*

LDev   Path           Status       State      Substate   Primary  Backup
                                                         PID      PID
104    PRIMARY        ACTIVE       STARTED                 2,107    3,35
104    BACKUP         INACTIVE     STARTED                 2,107    3,35
104    MIRROR         ACTIVE       STARTED                 2,107    3,35
104    MIRROR-BACKUP  INACTIVE     STARTED                 2,107    3,35


STORAGE - Status DISK \WAGER.$D0203-*

LDev   Path           Status       State      Substate   Primary  Backup
                                                         PID      PID
105    PRIMARY        INACTIVE     STARTED                 2,22     3,16
105    BACKUP         ACTIVE       STARTED                 2,22     3,16
105    MIRROR         INACTIVE     STARTED                 2,22     3,16
105    MIRROR-BACKUP  ACTIVE       STARTED                 2,22     3,16


STORAGE - Status DISK \WAGER.$D0205-*

LDev   Path           Status       State      Substate   Primary  Backup
                                                         PID      PID
105    PRIMARY        ACTIVE       STARTED                 2,112    3,45
105    BACKUP         INACTIVE     STARTED                 2,112    3,45
105    MIRROR         ACTIVE       STARTED                 2,112    3,45
105    MIRROR-BACKUP  INACTIVE     STARTED                 2,112    3,45
```

```
STORAGE - Status DISK \WAGER.$D0207-*

LDev   Path           Status       State      Substate    Primary  Backup
                                                          PID      PID
112    PRIMARY        INACTIVE     STARTED                 2,118    3,237
112    BACKUP         ACTIVE       STARTED                 2,118    3,237


STORAGE - Status DISK \WAGER.$D0208-*

LDev   Path           Status       State      Substate    Primary  Backup
                                                          PID      PID
114    PRIMARY        INACTIVE     STARTED                 2,122    3,86
114    BACKUP         ACTIVE       STARTED                 2,122    3,86
```

9. Move the corresponding disk processes to the other processor to match the switched paths. This PRIMARY command forces half the disk processes to run in processor 3:

   ```
   -> PRIMARY ($D0203, $D0207, $D0208), 3
   ```

10. Display the changed status:

   ```
   -> STATUS $D02*-*

   STORAGE - Status DISK \WAGER.$D0201-*

   LDev   Path           Status       State      Substate    Primary  Backup
                                                             PID      PID
   104    PRIMARY        ACTIVE       STARTED                 2,107    3,35
   104    BACKUP         INACTIVE     STARTED                 2,107    3,35
   104    MIRROR         ACTIVE       STARTED                 2,107    3,35
   104    MIRROR-BACKUP  INACTIVE     STARTED                 2,107    3,35


   STORAGE - Status DISK \WAGER.$D0203-*

   LDev   Path           Status       State      Substate    Primary  Backup
                                                             PID      PID
   105    PRIMARY        INACTIVE     STARTED                 3,16     2,22
   105    BACKUP         ACTIVE       STARTED                 3,16     2,22
   105    MIRROR         INACTIVE     STARTED                 3,16     2,22
   105    MIRROR-BACKUP  ACTIVE       STARTED                 3,16     2,22


   STORAGE - Status DISK \WAGER.$D0205-*

   LDev   Path           Status       State      Substate    Primary  Backup
                                                             PID      PID
   105    PRIMARY        ACTIVE       STARTED                 2,112    3,45
   105    BACKUP         INACTIVE     STARTED                 2,112    3,45
   105    MIRROR         ACTIVE       STARTED                 2,112    3,45
   105    MIRROR-BACKUP  INACTIVE     STARTED                 2,112    3,45


   STORAGE - Status DISK \WAGER.$D0207-*

   LDev   Path           Status       State      Substate    Primary  Backup
                                                             PID      PID
   112    PRIMARY        INACTIVE     STARTED                 3,237    2,118
   112    BACKUP         ACTIVE       STARTED                 3,237    2,118


   STORAGE - Status DISK \WAGER.$D0208-*

   LDev   Path           Status       State      Substate    Primary  Backup
                                                             PID      PID
   114    PRIMARY        INACTIVE     STARTED                 3,86     2,122
   114    BACKUP         ACTIVE       STARTED                 3,86     2,122
   ```

   This is the desired load-balancing scheme for RVUs prior to G06.11.

Maintaining a Disk Load Balance

- For fabrics, verify each disk IOP is correctly balancing its load between the X and Y fabrics. See "Disk Load Balancing Between Fabrics" (page 118).
- For processors, once you have achieved the desired load-balancing scheme, use the STATUS command periodically to monitor that balance.

  If you have configured all SACs in group 02 to be owned by processor 02, type:

```
-> STATUS ADAPTER $ZZSTO.#*.GRP-2*, DETAIL

STORAGE - Detailed Status ADAPTER \COMM.$ZZSTO.#PMF.GRP-2.MOD-1.SLOT-50
 …
   SAC \COMM.$ZZSTO.#PMF.SAC-1.GRP-2.MOD-1.SLOT-50:
 …
     Configured Devices ( group: 2 , module: 1  ):
       Slot  Name           State      Substate         Primary  Backup
                                                        PID      PID
          2  $AOL2-MB       STARTED                      2,269    3,286
          4  $VIRCFG-MB     STARTED                      2,288    3,267
 …
```

  If the Primary PID for any IOP is processor 3, an unwanted processor change has occurred. Use a PRIMARY command to swap processors for that IOP.

9 Configuring and Managing Storage Pools for Disks

A storage pool is a collection of disks that are managed by the Storage Management Foundation (SMF). Each storage pool is associated with virtual disks. A storage pool has the object type of POOL. For descriptions of this object and its states, see "The POOL Object" (page 36)and "POOL Object States" (page 36). For commands that affect POOL objects, see "Storage Subsystem Commands" (page 190). This chapter describes:

Displaying Information About Storage Pools

To display information about storage pools for disks:

Displaying Storage Pool Configuration Information

The "INFO POOL Command" (page 258)displays configured information about the storage pool.

Considerations for INFO POOL

- If the SMF master process ($ZSMS) is not started, the Configuration State in the INFO report shows that the "SMF master process is Down."
- If a configuration record for a storage pool process exists, but the SMF master process does not have a definition for it, no information is displayed for attribute values that are supplied by the SMF master process. The Configuration State in the INFO MON report displays: No Definition with SMF master process.

Example of an INFO POOL Report

```
-> INFO POOL $POOL3

STORAGE - Info POOL \COMM.$POOL3
*Audited    CatalogLocation              *Magnetic   *Mirrored   *Updatestats
 ALLOWED    $VIRCFG.POOL3CAT              REQUIRED    REQUIRED    1
```

Explanation of Fields — INFO POOL Report

*	Indicates an attribute whose value you can change by using the SCF ALTER POOL command.
Audited	Shows whether audited volumes, protected by the TMF product, are allowed or required in the storage pool.
CatalogLocation	The location of the catalog for the storage pool process.
Physical Mirrored	Shows whether physical or mirrored volumes are allowed or required in the storage pool.
Updatestats	The time interval, in minutes, during which the storage pool process should collect information about the physical volumes in the storage pool.

Example of a Detailed INFO POOL Report

```
-> INFO POOL $POOL3, DETAIL

STORAGE - Detailed Info POOL \COMM.$POOL3
*Audited............................... ALLOWED
*BackupCPU............................. 7
 CatalogLocation....................... $VIRCFG.POOL3CAT
*DiskInterval.......................... 2%
*DiskThreshold......................... 90%
*ExtentInterval........................-1
*ExtentThreshold.......................-1
*HighPIN............................... OFF
*Magnetic.............................. REQUIRED
*Mirrored.............................. REQUIRED
*Mode.................................. QUIET
 Monitor............................... $ZSMS
*PrimaryCPU............................ 6
*Program............................... $SYSTEM.SYSTEM.OPP
*StartState............................ STARTED
*UpdateStats........................... 1
 Configuration State................... Fully Defined

Physical Volumes:
$P3D03  $P3D02  $P3D01

Virtual Volumes:
$NSDOM1  $NODE    $NMS     $NE     $MYSTIC  $MOSS    $MINE    $MHSRL2  $MHSREL
$MEZRPT  $LOGICJ  $LOGICI  $LOGICG  $LOGICF  $LOGICD  $LOGICB  $LOGICA  $LIME
$LANX    $LANO    $LANN    $LANM    $LANL    $LANF    $LAND    $ASAP    $FOXGT
```

Many fields of the INFO POOL display different values if the SMF master process is not started (Configuration State is SMF Master Process is Down).

Explanation of Fields — INFO POOL

*	Indicates an attribute whose value you can change by using the ALTER POOL command.
Audited	Shows whether audited volumes (protected by the TMF product) are allowed or required in the storage pool.
BackupCPU	The backup processor in which the storage pool process should start its backup process.
CatalogLocation	The location of the catalog for the storage pool process.
DiskInterval	The interval percentage at which the disk processes associated with the physical volumes in the storage pool process should generate additional disk-full EMS messages.

DiskThreshold	The threshold percentage of disk space usage at which the disk processes associated with the physical volume in the storage pool should generate an EMS warning message.
ExtentInterval	The extent interval at which the disk processes associated with the physical volumes in the storage pool should generate additional file-extent usage EMS messages.
ExtentThreshold	Shows the threshold for remaining extend allocation space at which the disk processes associated with the physical volume in the storage pool should generate an EMS warning message.
HighPin	The desired PIN range for the storage pool process.
Physical Mirrored	Shows whether physical or mirrored volumes are allowed or required in the storage pool.
Mode	Shows whether the storage pool process generates additional EMS messages.
Monitor	$ZSMS, the name of the SMF master process.
PrimaryCPU	The preferred processor in which the storage pool process should start its primary process.
Program	The object file name of the storage pool process.
StartState	Shows whether the storage pool process is created in a STARTED or STOPPED state after a system load or processor load.
UpdateStats	The time interval, in minutes, at which the storage pool process should collect information about the physical volumes in the storage pool.
Configuration State	Shows the configuration state for the storage pool process. The value displayed is: **Fully Defined**—The storage pool process has a configuration record and is defined by the SMF master process ($ZSMS). **No Definition with SMF Master Process**—Although there is a system configuration database record, the SMF master process indicates it has no definition for the storage pool process. If this value is displayed, use the SCF DELETE command to remove the configuration record. **SMF Master Process Down**—-Information expected to come from the SMF master process is not available because the SMF master process is not available; that is, there is no way to know whether it exists.
Physical Volumes	Shows all disk volumes associated with the storage pool.
Virtual Volumes	Shows all virtual disk volumes associated with the storage pool.

Displaying Storage Pool Status Information

The “STATUS POOL Command” (page 285) displays current status information about the storage pool.

Example of a STATUS POOL Report

To display current status information about $POOL3:

```
-> STATUS $POOL3

STORAGE - Status POOL \COMM.$POOL3
LDev   State        Primary    Backup    Type  Subtype
                    PID        PID
 204   STARTED       6,13       4,141    25    0
```

Example of a Detailed STATUS POOL Report

To display detailed current status information about $POOL3:

```
-> STATUS $POOL3, DETAIL

STORAGE - Detailed Status POOL \COMM.$POOL3

Pool Process Information:
 LDev   State        Primary    Backup    Type  Subtype
                     PID        PID
  204   STARTED       6,13       4,141    25    0
```

```
Pool I/O Process Information:
  Library File.......
  Program File....... $SYSTEM.SYS01.OPP
```

Explanation of Fields — Detailed STATUS POOL Report	
LDev	The logical device number for the storage pool process.
State	The object state of the storage pool process.
Primary PID Backup PID	The processor number and PIN of the current primary and backup storage pool processes.
Type	The type number of the storage pool process is 25
Subtype	The subtype number of the storage pool process is 0.
Pool I/O Process Information:	
Library File	The library file name of the storage pool process.
Program File	The program file name of the storage pool process.

Configuring a Storage Pool for Disk Drives

Configuration tasks for storage pools include:

- "Adding a Storage Pool" (page 137)
- "Adding a Disk to a Storage Pool" (page 138)
- "Removing a Disk From a Storage Pool" (page 138)
- "Altering Storage Pool Attribute Values" (page 139)
- "Deleting a Storage Pool" (page 140)

Adding a Storage Pool

Use the "ADD POOL Command" (page 218) to add a storage pool to the system configuration database.

Considerations for ADD POOL

- TMF must be running on the system when adding a storage pool.
- The $ZSMS SMF Master Process must be in the STARTED state.

Example

1. Configure the pool process:

```
-> ADD POOL $POOL00, SENDTO STORAGE, PRIMARYCPU 6, &
-> BACKUPCPU 7
```

2. Verify the configuration:

```
-> INFO $POOL00
```

3. Start the pool. See "Starting a Storage Pool".

Example: Adding a Similar Pool Process Using the LIKE Attribute

To add another pool process similar to an existing process, use the LIKE attribute:

```
-> ADD POOL $POOL01, SENDTO STORAGE, LIKE $POOL00
```

Example Adding a Similar Pool Process Using the OBEYFORM Attribute

To configure one or more pool processes, you can create a command file by using the OBEYFORM attribute of the INFO DISK command. You can copy this file to another system or add it to different configuration file on the current system.

1. Capture the configuration for an existing pool process:

   ```
   -> INFO / OUT LOG / $POOL1, OBEYFORM

   == STORAGE - Detailed Info POOL in obeyform: \COMM.$POOL1
   ADD POOL $POOL1 , &
       SENDTO STORAGE , &
       AUDITED ALLOWED, &
       BACKUPCPU 5 , &
       CATALOGLOCATION $VIRCFG.POOL1CAT, &
       DISKINTERVAL  2, &
       DISKTHRESHOLD  90, &
       EXTENTINTERVAL -1, &
       EXTENTTHRESHOLD -1, &
       HIGHPIN OFF, &
       MAGNETIC REQUIRED, &
       MIRRORED REQUIRED, &
       MODE QUIET, &
       PRIMARYCPU 4 , &
       PROGRAM $SYSTEM.SYSTEM.OPP, &
       STARTSTATE STARTED, &
       UPDATESTATS 1
   ```

2. Optionally edit the resulting log file to specify:
   - A unique process name
   - Remove the subvolume name from CATALOGLOCATION
3. Enter the log file contents either by copying and pasting into an SCF command line or by using the log file as a command file.

Adding a Disk to a Storage Pool

The “ALTER DISK Command” (page 228) adds a disk to an SMF storage pool.

Considerations for Adding a Disk to a Storage Pool

- All the disks in a storage pool must be on the same system.
- Up to 144 disk volumes can be placed in a pool.
- A disk volume can be in only one storage pool at a given time.
- When you add a disk to a storage pool:
  - The disk must be in the STARTED state.
  - The POOL object must be the STARTED state.
  - The disk cannot be a member of another storage pool.

Example

```
-> ALTER $DATA, POOL $POOL1
```

Removing a Disk From a Storage Pool

The “ALTER DISK Command” (page 228) removes a disk to an SMF storage pool.

Considerations for Removing a Disk From a Storage Pool

When you remove a disk from a storage pool:

- The POOL name is removed from the disk label if the pool process for the disk is not defined in the system.
- SCF asks you to confirm the command in these cases:
  - The POOL process for the disk is not in the STARTED state.
  - Virtual disks associated with the pool are not in the STARTED state.
  - The physical volume has storage-managed files on it.
  - Avoid overriding any safety checks whenever possible. Excluding a disk from a storage pool even though a discrepancy exists could have severe consequences. For example, storage-managed files on a physical volume are still accessible after the physical volume has been excluded from a storage pool, but the catalog for the storage pool is no longer in a consistent state.
- You might be able to correct discrepancies created by safety overrides by using the SMFIXUP utility. (See the *Storage Management Foundation User's Guide*.)

Example

To remove a disk from a storage pool:

```
-> ALTER $DATA, POOL EXCLUDE
```

Altering Storage Pool Attribute Values

Use the "ALTER POOL Command" (page 233) to change the attributes of a storage pool.

Considerations for ALTER POOL

- TMF must be running on the system when you alter a storage pool.
- The SMF Master Process ($ZSMS) must be in the STARTED state.
- These attributes require that the storage pool process be in the STOPPED state or not running:

BACKUPCPU	PRIMARYCPU	STARTSTATE
HIGHPIN	PROGRAM	

  Changes take effect when you restart the process.

  All other storage pool process attributes can be altered regardless of whether the storage pool process is running and regardless of its state.

Example

1. If necessary, stop the storage pool:

   ```
   -> STOP $POOL00
   ```

2. Change one or more "ALTER POOL Attributes" (page 234). For example, to change how often $POOL00 collects information about physical volumes in the pool:

   ```
   -> ALTER $POOL00, UPDATESTATS 3
   ```

3. Verify the configuration change:

   ```
   -> INFO $POOL00, DETAIL
   ```

4. Restart the storage pool:

   ```
   -> START $POOL00
   ```

Deleting a Storage Pool

Use the "DELETE POOL Command" (page 251) to delete a storage pool.

Considerations for DELETE POOL

- Before deleting a storage pool process from the system configuration database, you must delete the virtual disks associated with the pool and reconfigure the physical volumes.
- The storage pool process you are deleting:
  - Must be stopped
  - Cannot contain any physical volumes
  - Cannot have any virtual disks associated with it
- The storage pool catalogs must be available
- TMF must be running.
- The SMF master process ($ZSMS) must be started.
- The CATALOGLOCATION volume for the MON process must be started, and it must be enabled by TMF.
- The CATALOGLOCATION volume for the storage pool process must be started, and it must be enabled by TMF.
- If, for any reason, the SMF master process does not give permission to delete the storage pool process, you are prompted to continue deletion, even though this might cause discrepancies in the SMF catalog.

Example

1. Identify the physical and virtual volumes associated with the pool:

```
-> INFO $POOL5, DETAIL

STORAGE - Detailed Info POOL \COMM.$POOL5
*Audited............................. ALLOWED
*BackupCPU........................... 9
 CatalogLocation..................... $VIRCFG.POOL5CAT
*DiskInterval........................ 2%
*DiskThreshold....................... 90%
*ExtentInterval......................-1
*ExtentThreshold.....................-1
*HighPIN............................. OFF
*Magnetic............................ REQUIRED
*Mirrored............................ REQUIRED
*Mode................................ QUIET
 Monitor............................. $ZSMS
*PrimaryCPU.......................... 8
*Program............................. $SYSTEM.SYSTEM.OPP
*StartState.......................... STARTED
*UpdateStats......................... 1
 Configuration State................. Fully Defined

Physical Volumes:
$P5D04  $P5D03  $P5D02  $P5D01

Virtual Volumes:
$VDISK01 $VDISK02 $VDISK03 $VDISK04 $VDISK05 $VDISK06 $VDISK07 $VDISK08 $VDISK09
$VDISK11 $VDISK12 $VDISK13 $VDISK14 $VDISK15 $VDISK16 $VDISK17 $VDISK18 $VDISK19
```

2. Delete the virtual disks associated with $POOL5:

```
-> DELETE DISK $*, SUB VIRTUAL, POOL $POOL5
```

3. Remove the following physical volumes from the storage pool:

```
-> ALTER $P5D04, POOL EXCLUDE
-> ALTER $P5D03, POOL EXCLUDE
-> ALTER $P5D02, POOL EXCLUDE
-> ALTER $P5D01, POOL EXCLUDE
```

4. Stop the storage pool:

   ```
   -> STOP $POOL5
   ```

   The storage pool enters the STOPPED state.
5. Delete the storage pool:

   ```
   -> DELETE $POOL5
   ```

6. Verify the storage pool process has been removed from the system configuration database:

   ```
   -> INFO $POOL5
   ```

Managing a Storage Pool

Management tasks for storage pools include:

- "Starting a Storage Pool" (page 141)
- "Stopping a Storage Pool" (page 141)
- "Resetting a Storage Pool" (page 142)
- "Swapping Processors for a Pool Process" (page 142)

Starting a Storage Pool

The "START POOL Command" (page 275) puts the storage pool into the STARTED state.

Considerations for START POOL

- The storage pool process name must have a configuration record and a definition with the SMF master process (as displayed by the INFO or STATUS commands).
- TMF must be running on the system.
- These objects must be started:
  - The SMF master process ($ZSMS).
  - The CATALOGLOCATION volume for the MON process (must also be enabled by TMF).
  - The CATALOGLOCATION volume of the storage pool process (must also be enabled by TMF).
  - To take a storage pool process out of the SERVICING state, reset it and start it again.

Example

1. Start the storage pool process:

   ```
   -> START $POOL00
   ```

2. Check the status of the process:

   ```
   -> STATUS $POOL00
   ```

Stopping a Storage Pool

The "STOP POOL Command" (page 291), like the "ABORT POOL Command" (page 194), makes a storage pool inaccessible to user requests.

-> STOP $POOL5

The storage pool finishes any current activity before it enters the STOPPED state. Virtual disks that use the stopped pool encounter an error.

Resetting a Storage Pool

The “RESET POOL Command” (page 272)puts a storage pool into the STOPPED state, substate DOWN, reading for restarting.

1. Check the current status of the storage pool:

   ```
   -> STATUS $POOL01
   ```

2. If it is not in the STOPPED state, substate DOWN:

   ```
   -> RESET $POOL01
   ```

3. Start the storage pool process:

   ```
   -> START $POOL01
   ```

4. Verify the storage pool process is started:

   ```
   -> STATUS $POOL01
   ```

Swapping Processors for a Pool Process

The “PRIMARY POOL Command” (page 265) swaps the primary and backup processors for a pool process. The current primary processor becomes the backup processor, and the backup processor becomes the primary processor, but the PRIMARYCPU and BACKUPCPU values stay the same. You typically swap processors when load balancing the system or preparing for disk replacement.

Examples

- To execute the primary process of $POOL01 in processor 3 (assuming it is configured to run in processor 3):

  ```
  -> PRIMARY $POOL01, 3
  ```

- To make the current backup process of $POOL01 the primary process:

  ```
  -> PRIMARY $POOL01
  ```

10 Configuring and Managing Virtual Disks

Virtual disks share the object type of DISK with disks (discussed in “Configuring Disks” (page 70)). A virtual disk is created by configuring one or more disks in a storage pool as part of “Storage Management Foundation (SMF)” (page 26). For information about disks and their states, see “The DISK Object” (page 33)and “Object States and Substates of Disks” (page 34). This chapter describes:

Overview of Virtual Disks

To understand virtual disks:

How Virtual Disks Relate to Physical Disks

A virtual disk appears to application programs to be a physical disk in most respects. However, some SCF commands affect only physical disks, not virtual disks, and other commands affect only virtual disks, not physical disks.

Any number of virtual disks can be associated with a pool. The files on a virtual disk are not necessarily placed by SMF on a single physical disk. The following examples summarize the relationships between virtual disks and physical disks:

- One physical volume might be 12 virtual disks, allocated to 12 users who share space on the same disk.
- Fifty physical disks might be 10 virtual disks, allocated to 10 users requiring a large amount of storage for completing their development work.
- Five physical disks might be one virtual disk, as one way of handling temporary space management.

Functions of the Virtual Disk Process

The virtual disk process:

- Performs file placement management; that is, the mapping between external and internal file names that supports location-independent naming
- Maintains a name catalog that contains the name-mapping information between the external and internal names of files that reside on the physical volumes it uses
- Gathers statistical information on the status of physical disks in the storage pool so you can determine the best location for the files on the disks

Location-Independent Naming

Virtual disks take advantage of location-independent naming. As a result, a file's external or logical name can differ from its internal name. In earlier RVUs, the external file name indicated the physical location of the data; that is, each file name indicated the node, volume, and subvolume where the file was located. For example, the file OVERDUE on system \SYSA, volume $DATA00, and subvolume CURRENT, is named \SYSA.$DATA00.CURRENT.OVERDUE.

With location-independent naming, the file \SYSA.$DATA00.CURRENT.OVERDUE can reside on any physical volume. With the exception of the node name, the name of the file managed by SMF is independent of its location.

SMF controls the mapping of the external name to the internal name. The internal name, which identifies the file's physical location, can change when a file is moved to a different location, while the external name remains the same to applications.

Displaying Information

To display information about virtual disks:

- "Displaying Configuration Information for Virtual Disks" (page 144)
- "Displaying Current Status Information" (page 146)

Displaying Configuration Information for Virtual Disks

The "INFO DISK Command" (page 254) displays information about a virtual disk, as recorded in the system configuration database.

Considerations for INFO DISK and Virtual Disks

- If the SMF master process ($ZSMS) is not started, the Configuration State in the INFO report shows that the "SMF master process is Down."
- If a configuration record for a virtual disk exists, but the SMF master process does not have a definition for it, no information is displayed for attribute values that are supplied by the SMF master process (the Configuration State in the INFO report displays "No Definition with SMF master process").

Examples

- To display information about a virtual disk:

  ```
  -> INFO $VDISK00
  ```

  The display is shown and explained under "Example of an INFO DISK Summary Report" (page 145).

- To display detailed information about a virtual disk:

  ```
  -> INFO $VDISK00, DETAIL
  ```

  The display is shown and explained under "Example of an INFO DISK Detailed Report" (page 146).

- To display a report about all virtual disks on the system:

  ```
  -> INFO DISK $* , SUB VIRTUAL
  ```

Example of an INFO DISK Summary Report

```
-> INFO $VDISK00

STORAGE - Info Virtual DISK \COMM.$VDISK00
*ANT       ANT                        *Cache  PendOps                     Pool
 Capacity  Location                    Size   Location
 1000000   $P1D02.ZYS00000.A0001MRG    30000  $P1D02.ZYS00000.A0001MRH    $POOL1
```

Explanation of Fields — INFO DISK Summary Report	
*	Indicates an attribute whose configured value you can change by using an ALTER DISK command.
ANT Capacity	The maximum size (in number of entries) of the audited name table of the virtual disk.
ANT Location	The location of the audited name table of the virtual disk.
Cache Size	The size (in number of entries) of the name cache of the virtual disk process.
PendOps Location	The location of the pending operations log file of the virtual disk.
Pool	The storage pool that the virtual disk is associated with.

Example of a Missing Data Report

If the $ZSMS Storage Management Foundation monitor process is stopped, the report can look like this:

```
-> INFO $VDISK00

STORAGE - Info Virtual DISK \ALM171.$VDISK00
 Configuration State
 SMF Master Process Down
```

Configuration State	
Fully Defined	the virtual disk has a configuration record and is defined by the SMF master process ($ZSMS).
No Definition With SMF master process	although a system configuration database record exists, the SMF master process indicates that the record has no definition for the virtual disk. If this value is displayed, use the "DELETE DISK Command" (page 248)to remove the configuration record.
SMF Master Process Down	information expected from the SMF master process is not available because the SMF master process is not available.

Example of an INFO DISK Detailed Report

For virtual disks, the INFO DISK, DETAIL command produces a short report (compared to the report for physical disks).

```
-> INFO DISK $DATA1, DETAIL

STORAGE - Detailed Info Virtual DISK \COMM.$DATA1
*ANTCapacity.......................... 1000000
 ANTLocation.......................... $P1D02.ZYS00000.A0001MRG
*BackupCPU............................ 5
*CacheSize............................ 30000
*Highpin.............................. OFF
*Mode................................. QUIET
 Monitor.............................. $ZSMS
 PendOpsLocation...................... $P1D02.ZYS00000.A0001MRH
 Pool................................. $POOL1
*PrimaryCPU........................... 4
*Program.............................. $SYSTEM.SYSTEM.OVDP
*StartState........................... STARTED
 Configuration State.................. Fully Defined
```

Explanation of Fields — INFO DISK Detailed Report

*	Indicates an attribute whose value you can change by using an ALTER DISK command.
ANT Capacity	Shows the maximum size (in number of entries) of the audited name table of the virtual disk.
ANT Location	Shows the location of the audited name table of the virtual disk.
BackupCPU	Shows the backup processor in which the disk process should start its backup process.
Cache Size	Shows the size (in number of entries) of the name cache of the virtual disk process.
HighPin	Shows the PIN range available for the virtual disk process.
Mode	Shows whether the virtual disk process generates additional EMS messages.
Monitor	Shows the name of the Storage Management Foundation (SMF) master process.
PendOpsLocation	Shows the location of the pending operations log file of the virtual disk.
Pool	Shows the storage pool process associated with the virtual disk.
PrimaryCPU	The processor number of the processor in which the primary virtual disk process should run.
Program	Shows the object file name of the virtual disk process.
StartState	Shows whether the virtual disk process is enabled (STARTED state) or disabled (STOPPED state) when the system is loaded.
Configuration State	Shows the state of the configuration for the virtual disk. For a description of possible states, see “Example of a Missing Data Report” (page 145).

Displaying Current Status Information

The “STATUS DISK Command” (page 282)displays the current status of a virtual disk.

Example of a STATUS DISK Report for Virtual Disks

```
-> STATUS $DATA1

STORAGE - Status VIRTUAL DISK \COMM.$DATA1
LDev    State        Primary    Backup   Type   Subtype
                     PID        PID
 386    STARTED       4,105      5,93    3      36
```

Example of a Detailed STATUS DISK Report for Virtual Disks

```
-> STATUS $DATA1, DETAIL

STORAGE - Detailed Status VIRTUAL DISK \COMM.$DATA1

Virtual Disk Process Information:
 LDev    State        Primary   Backup   Type   Subtype
                      PID       PID
  386    STARTED       4,105     5,93    3      36

Virtual Disk I/O Process Information:
  Library File.......
  Program File....... $SYSTEM.SYS01.OVDP
```

Explanation of Fields — STATUS DISK report for virtual disks	
LDev	The logical device number for the disk volume, arbitrarily assigned to a device when you configure the device and every time the system is loaded.
State	The current state of the disk.
Primary PID Backup PID	The processor number and PIN of the primary and backup disk processes.
Type	The device type is always type 3.
Subtype	The device subtype is always type 36.
Virtual Disk I/O Process Information:	
Library File	The library file name of the disk process.
Program File	The program file name of the disk process.

Configuring a Virtual Disk

Virtual disk configuration tasks include:

- "Adding a Virtual Disk" (page 147)
- "Troubleshooting Problems With Adding a Virtual Disk" (page 148)
- "Altering Virtual Disk Attribute Values" (page 149)
- "Deleting a Virtual Disk" (page 150)

Adding a Virtual Disk

The "ADD DISK Command" (page 194) adds a virtual disk to the system configuration database.

Considerations for ADD DISK and Virtual Disks

Before adding a virtual disk verify that:

- The TMF product is running on the system.
- These objects are in the STARTED state:
  - The physical disk volume used as a catalog by the SMF master process
  - The SMF master process ($ZSMS)
  - The storage pool process to be specified in the POOL attribute
  - The disk volume where the CATALOGLOCATION for the storage pool process resides

Example

1. Add the virtual disk to the system configuration database and associate it with a storage pool:

   ```
   ->
   ADD DISK $VDISK00, SENDTO STORAGE, ANTLOCATION $SYSTEM, &
   -> POOL $POOL01, PENDOPSLOCATION $SYSTEM, TYPE VIRTUAL
   ```

2. Verify the attributes recorded in the system configuration database:

   ```
   -> INFO DISK $VDISK00
   ```

3. See “Starting a Virtual Disk” (page 150).

Example: Adding a Similar Disk Using the LIKE Attribute

To add another disk similar to an existing disk on the same system, you can use the LIKE attribute. If the disk is an internal disk, specify a unique name and the group and slot number:

```
-> ADD DISK $VDISK02, SENDTO STORAGE, LIKE $VDISK00, &
-> TYPE VIRTUAL, ANTLOCATION $DATA00, PENDOPSLOCATION $DATA00
```

Example Adding a Similar Disk Using the OBEYFORM Attribute

To configure the same or a similar disk on another system, create a command file by using the OBEYFORM attribute of the INFO DISK command. You can copy this file to another system or add it to a different configuration file on the current system.

1. Capture the configuration for an existing virtual disk:

   ```
   -> INFO / OUT LOG / $TECH, OBEYFORM

   == STORAGE - Detailed Info Virtual DISK in obeyform: \COMM.$TECH
   ADD DISK $TECH , &
       SENDTO STORAGE , &
       TYPE VIRTUAL , &
       ANTCAPACITY 1000000 , &
       ANTLOCATION $P6D01.ZYS00000.A0000QBC, &
       BACKUPCPU 8 , &
       CACHESIZE 30000 , &
       HIGHPIN OFF, &
       MODE QUIET, &
       PENDOPSLOCATION $P6D01.ZYS00000.A0000QBD, &
       POOL $POOL6, &
       PRIMARYCPU 9 , &
       PROGRAM $SYSTEM.SYSTEM.OVDP, &
       STARTSTATE STARTED
   ```

2. Optionally edit the resulting log file to:
   - Specify a unique disk name
   - Remove the subvolume and file name from ANTLOCATION and PENDOPSLOCATION
3. Enter the log file contents either by copying and pasting into an SCF command line or by using the log file as a command file.

Troubleshooting Problems With Adding a Virtual Disk

1. Use an INFO or STATUS command to detect problems when adding a virtual disk:
   - The INFO command displays a process that has a configuration record but the Configuration State is “No Definition With SMF master process.”
   - The STATUS command also displays the Configuration State “No Definition With SMF master process.”
2. Check the EMS event messages for the storage subsystem.

The ADD DISK Command Fails

An ADD DISK command can fail if a record for the given name exists or if the SMF Master Process already has a definition for a virtual disk of the given name.

For example, if SCF successfully adds the configuration record but then cannot define the process by using the SMF Master Process, SCF attempts to delete the configuration records. If this action fails, SCF puts out a message to warn that configuration records remain although there is no definition in the SMF Master Process.

This inconsistency can also occur if a processor failure happens after the configuration records are added but before the defining with the SMF Master Process is done.

To recover, delete the incorrect entry. See "Deleting a Virtual Disk" (page 150). If the DELETE command fails, alter the STARTSTATE attribute to STOPPED to ensure that the process does not unexpectedly start.

Storage Subsystem Message 9026

If you get storage subsystem message 9026 while adding a storage pool process or virtual disk, the error is caused by an inconsistency between SMF and the system configuration database (the storage pool or virtual disk is defined in SMF but is not defined in the system configuration database). In this case, you can:

- Add the storage pool or virtual disk without using the SMF attributes.
- Add the storage pool or virtual disk using the same attributes that are defined in SMF.

Altering Virtual Disk Attribute Values

The "ALTER DISK Command" (page 228) changes configured attributes for a virtual disk.

Considerations for ALTER DISK and Virtual Disks

- The process must be in the STOPPED state or not running before you can change its configuration.
- Changes take effect when you restart the process.
- All configuration changes related to SMF require that:
  - TMF must be running on the system.
  - The SMF master process ($ZSMS) must be in the STARTED state.
  - The CATALOGLOCATION volume of the SMF Master Process must be in the STARTED state and enabled by TMF.

Example

1. Display detailed information about the configured attributes of the disk you want to alter:

   ```
   -> INFO $VDISK00, DETAIL
   ```

   An asterisk (*) marks those attributes you can alter.

2. Stop the disk process:

   ```
   -> STOP $VDISK00
   ```

3. Change one or more Virtual Disk Attributes for the "ALTER DISK Command" (page 228). For example, this command changes the maximum size of the audited name table:

   ```
   -> ALTER $VDISK00, ANTCAPACITY 1000000
   ```

4. Verify the change is entered into the system configuration database:

   ```
   -> INFO $VDISK00, DETAIL
   ```

5. Restart the disk process:

```
-> START $VDISK00
```

Deleting a Virtual Disk

The "DELETE DISK Command" (page 248) removes a virtual disk from the system configuration database.

Considerations for DELETE DISK and Virtual Disks

- For the DELETE DISK command to succeed:
  - The process must be stopped or not running before you can delete it.
  - TMF must be running.
  - $ZSMS must be in the STARTED state.
  - The storage pool process specified by the POOL attribute must be started.
  - The CATALOGLOCATION volume for the MON process must be started and must be enabled by TMF.
  - The CATALOGLOCATION volume for the storage pool process must be started and enabled by TMF.
  - The ANTLOCATION and PENDOPSLOCATION volumes for the virtual disk must be started, they must be enabled in TMF, and they must be configured to be in a storage pool.
  - If a virtual disk has a system configuration database record, you can delete it even if the SMF master process ($ZSMS) has no knowledge of that record.
- If, for any reason, the virtual disk process cannot be started or the SMF Master process does not give permission to delete the virtual disk, you are prompted to continue with the delete, even though deletion might cause discrepancies in SMF catalogs.
- Example

1. Verify the disk is in the STOPPED state, substate DOWN:

   ```
   -> STATUS $VDISK00
   ```

   If the disk is not stopped, see "Stopping a Virtual Disk" (page 151).
2. Delete the disk from the system configuration database:

   ```
   -> DELETE DISK $VDISK00
   ```
3. Verify the disk is deleted:

   ```
   -> INFO $VDISK00
   ```

Managing a Virtual Disk

Virtual disk management tasks include:

- "Starting a Virtual Disk" (page 150)
- "Resetting a Virtual Disk" (page 151)
- "Stopping a Virtual Disk" (page 151)
- "Swapping Processors for a Virtual Disk" (page 152)

Starting a Virtual Disk

The "START DISK Command" (page 274) puts the virtual disk process into the STARTED state.

Considerations for START DISK and Virtual Disks

- The disk specified must have a configuration record and a definition with the SMF master process (as verified by the INFO and STATUS commands).
- TMF must be running on the system.
- These items must be started:
  - The SMF master process ($ZSMS)
  - The CATALOGLOCATION volume for the MON process (must also be enabled by TMF)
  - The storage pool process specified by the POOL attribute
  - The CATALOGLOCATION volume of the storage pool process (must also be enabled by TMF)
  - The ANTLOCATION and PENDOPSLOCATION volumes for the virtual disk (must also be enabled by TMF and configured to be in a storage pool)

Example

1. Start the virtual disk process:

   ```
   -> START $VDISK00
   ```

2. Verify the status of the started disk:

   ```
   -> STATUS $VDISK00
   ```

Resetting a Virtual Disk

The "RESET DISK Command" (page 271) puts a virtual disk into the STOPPED state, substate DOWN, ready to be started.

Check the current status of the disk:

```
-> STATUS $VDISK00
```

1. If the disk is in the HARDDOWN state or SERVICING state, substate SPECIAL:

   ```
   -> RESET $VDISK00
   ```

2. Start the disk:

   ```
   -> START $VDISK00
   ```

3. Confirm that the disk is started:

   ```
   -> STATUS $VDISK00
   ```

Stopping a Virtual Disk

The "STOP DISK Command" (page 290) stops access to a virtual disk. Although you can use ABORT DISK for virtual disks, the command that is actually executed is the STOP DISK command. When the STOP DISK command finishes, the disk is left in a STOPPED state, substate DOWN and remains configured in the system configuration database.

Example

1. Check the state of the disk:

   ```
   -> STATUS $VDISK00
   ```

2. Stop the disk:

   ```
   -> STOP $VDISK00
   ```

3. Confirm that the disk is in the STOPPED state, substate DOWN:

   ```
   -> STATUS $VDISK00
   ```

Swapping Processors for a Virtual Disk

The"PRIMARY DISK Command" (page 264) swaps the primary and backup processors for a virtual disk. The current primary processor becomes the backup processor, and the backup processor becomes the primary processor, but the PRIMARYCPU and BACKUPCPU values stay the same. You typically swap processors when load balancing the system or preparing for disk replacement.

Examples

- To execute the primary process of $VDISK00 in processor 3 (assuming it is configured to run in processor 3):

  ```
  -> PRIMARY $VDISK00, 3
  ```

- To make the current backup process of $VDISK00 the primary process:

  ```
  -> PRIMARY $VDISK00
  ```

11 Configuring and Managing ServerNet Storage Adapters

A ServerNet storage adapter has the object type of ADAPTER. For a list of supported storage adapters, see the *Storage Management Foundation User's Guide*. For information about ADAPTER objects and their states, see "The ADAPTER Object" (page 32) and "Object States and Substates" (page 31). For commands that relate to this object, see "Storage Subsystem Commands" (page 190).

Displaying Storage Adapter Information

To display information about storage adapters:

Displaying Storage Adapter Configuration Information

The "INFO ADAPTER Command" (page 253) displays configured information about storage adapters and connected devices:

- To display information about the adapter in the PMF CRU in group 1, slot 50:

  ```
  -> INFO ADAPTER $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50
  ```

  The display is shown and explained under "Example of an INFO PMF ADAPTER Report" (page 154).

- To display detailed configuration information about the same adapter in the PMF CRU:

  ```
  -> INFO ADAPTER $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50, DETAIL
  ```

  The display is shown and explained under "Example of a Detailed INFO PMF ADAPTER Report" (page 154).

- To display configuration information about the adapter in the IOMF CRU in group 11, slot 50:

  ```
  -> INFO ADAPTER $ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-50
  ```

  The display is shown under "Example of an INFO IOMF ADAPTER Report" (page 155).

- To display detailed configuration information about the same adapter in the IOMF CRU:

```
-> INFO ADAPTER $ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-50, DETAIL
```

The display is shown under "Example of a Detailed INFO IOMF ADAPTER Report" (page 155).

- To display configuration information about the 6760 ServerNet device adapter in group 11, slot 51:

  ```
  -> INFO ADAPTER $ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-51
  ```

  The display is shown under "Example of an INFO SNDA ADAPTER Report" (page 156).

- To display detailed configuration information about the same 6760 ServerNet device adapter:

  ```
  -> INFO ADAPTER $ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-51, DETAIL
  ```

  The display is shown under "Example of a Detailed INFO SNDA ADAPTER Report" (page 156).

- To display configuration information about a Fibre Channel ServerNet adapter (FCSA):

  ```
  -> INFO ADAPTER $ZZSTO.#FCSA.GRP-11.MOD-2.SLOT-1
  ```

  The display is shown under "Example of an INFO FCSA ADAPTER Report" (page 157).

- To display detailed configuration information about a Fibre Channel ServerNet adapter (FCSA):

  ```
  -> INFO ADAPTER $ZZSTO.#FCSA.GRP-11.MOD-2.SLOT-1, DETAIL
  ```

  The display is shown under "Example of a Detailed INFO FCSA Adapter Report" (page 157).

Example of an INFO PMF ADAPTER Report

```
-> INFO ADAPTER $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50

STORAGE - Info ADAPTER \COMM.$ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50
 Location      Access List
 (1,1,50)      0,1
```

Explanation of Fields — INFO PMF ADAPTER Report

Location	Where the storage adapter is located (group, module, and slot).
Access List	The numbers of the processors configured to access this storage adapter.

Example of a Detailed INFO PMF ADAPTER Report

```
-> INFO ADAPTER $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50, DETAIL

STORAGE - Detailed Info ADAPTER \COMM.$ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50
 Access List................... 0,1
 Location (Group,Module,Slot).. (1,1,50)

 ServerNet Addressable Controllers:

  SAC \COMM.$ZZSTO.#PMF.SAC-1.GRP-1.MOD-1.SLOT-50:

     Configured Devices ( group: 1, module: 1 ):
       Slot  Type    Name           Primary  Backup  Device
                                    CPU      CPU     Target ID  Lun
          2  DISK    $OPER-MB        0        1       4          0
          4  DISK    $MAG1-MB        1        0       5          0
          8  DISK    $AOL1-MB        0        1       9          0
         12  DISK    $SYSTEM-MB      0        1       0          0
         14  DISK    $DSMSCM-MB      0        1       1          0
         16  DISK    $AUDIT-MB       0        1       2          0
         18  DISK    $NEWSYS-MB      0        1       3          0

  SAC \COMM.$ZZSTO.#PMF.SAC-2.GRP-1.MOD-1.SLOT-50:

     Configured Devices ( group: 1, module: 1 ):
       Slot  Type    Name           Primary  Backup  Device
```

```
                                     CPU       CPU      Target ID  Lun
      1  DISK    $OPER-P              0         1        4          0
      3  DISK    $MAG1-P              1         0        5          0
      7  DISK    $AOL1-P              0         1        9          0
     11  DISK    $SYSTEM-P            0         1        0          0
     13  DISK    $DSMSCM-P            0         1        1          0
     15  DISK    $AUDIT-P             0         1        2          0
     17  DISK    $NEWSYS-P            0         1        3          0

 SAC \COMM.$ZZSTO.#PMF.SAC-3.GRP-1.MOD-1.SLOT-50:

    Configured Devices ( group: 1, module: 1 ):
      Slot  Type    Name              Primary  Backup   Device
                                      CPU      CPU      Target ID
       50  TAPE    $DL2            0         1        5
```

This report shows that the storage adapter in a PMF CRU :

- Has mirrored disks on SAC 1 and SAC 2
- Has a tape drive on SAC 3

Explanation of Fields — Detailed INFO PMF ADAPTER Report	
Access List	The numbers of the processors that can access this storage adapter.
Location (Group, Module, Slot)	Where the storage adapter is located.
ServerNet Addressable Controllers:	
SAC	The name of the SAC, determined automatically by the storage adapter hardware.
Configured Devices (group: n, module: n)	
Slot	For internal disks, this is the slot number of the device. For 45*xx* disks and tape drives, this is the slot number of the storage adapter controlling the device. This value is determined automatically from the physical location of the hardware.
Type	The configured device type of the device, determined by the physical device installed.
Name	The configured name of the device. The name for a disk includes a suffix that denotes the path used by the disk.
Primary CPU Backup CPU	The configured primary and backup processors for the device.
Device Target ID	The configured ID of the device.
Lun	The logical unit number of the device.

Example of an INFO IOMF ADAPTER Report

```
-> INFO ADAPTER $ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-50

STORAGE - Info ADAPTER \COMM.$ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-50
 Location      Access List
 (11,1,50)     0,1
```

See Explanation of Fields — Detailed INFO PMF ADAPTER Report.

Example of a Detailed INFO IOMF ADAPTER Report

```
-> INFO ADAPTER $ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-50, DETAIL

STORAGE - Detailed Info ADAPTER \COMM.$ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-50
 Access List................... 0,1
 Location (Group,Module,Slot).. (11,1,50)
```

```
 ServerNet Addressable Controllers:

  SAC \COMM.$ZZSTO.#IOMF.SAC-1.GRP-11.MOD-1.SLOT-50:

     Configured Devices ( group: 11, module: 1 ):
       Slot   Type     Name            Primary  Backup   Device
                                       CPU      CPU      Target ID   Lun
          2   DISK     $ESD001-MB       0        1        4          0
          4   DISK     $ESD002-MB       0        1        5          0
          6   DISK     $D11105-MB       0        1        8          0
          8   DISK     $MAIL-MB         0        1        9          0
         12   DISK     $P1D04-MB        0        1        0          0
         14   DISK     $P5D04-MB        0        1        1          0
         16   DISK     $P6D04-MB        0        1        2          0
         18   DISK     $D11117-MB       0        1        3          0

  SAC \COMM.$ZZSTO.#IOMF.SAC-2.GRP-11.MOD-1.SLOT-50:

     Configured Devices ( group: 11, module: 1 ):
       Slot   Type     Name            Primary  Backup   Device
                                       CPU      CPU      Target ID   Lun
          1   DISK     $ESD001-P        0        1        4          0
          3   DISK     $ESD002-P        0        1        5          0
          5   DISK     $D11105-P        0        1        8          0
          7   DISK     $MAIL-P          0        1        9          0
         11   DISK     $P1D04-P         0        1        0          0
         13   DISK     $P5D04-P         0        1        1          0
         15   DISK     $P6D04-P         0        1        2          0
         17   DISK     $D11117-P        0        1        3          0

  SAC \COMM.$ZZSTO.#IOMF.SAC-3.GRP-11.MOD-1.SLOT-50:

     Configured Devices ( group: 11, module: 1 ):
       Slot   Type     Name            Primary  Backup   Device
                                       CPU      CPU      Target ID
         50   SCSI     $L700C16-P       1        0        0
```

This report shows that the storage adapter in an IOMF CRU:

- Has mirrored disks on SAC 1 and SAC 2
- Has an Open SCSI device on SAC 3

See Explanation of Fields - Detailed INFO PMF ADAPTER Report.

Example of an INFO SNDA ADAPTER Report

```
-> INFO ADAPTER $ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-51

STORAGE - Info ADAPTER \COMM.$ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-51
 Location     Access List
 (11,1,51)    0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
```

This report shows that the storage adapter is accessed by processors 0 through 15.

See Explanation of Fields — Detailed INFO PMF ADAPTER Report.

Example of a Detailed INFO SNDA ADAPTER Report

```
-> INFO ADAPTER $ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-51, DETAIL

STORAGE - Detailed Info ADAPTER \COMM.$ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-51
 Access List.................. 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
 Location (Group,Module,Slot).. (11,1,51)

 ServerNet Addressable Controllers:

  SAC \COMM.$ZZSTO.#SNDA.SAC-1.GRP-11.MOD-1.SLOT-51:
```

```
    Configured Devices ( group: 11, module: 1 ):
      Slot  Type    Name           Primary  Backup  Device
                                   CPU      CPU     Target ID  Lun
        51  DISK    $LANA-P         0        1       0         0
        51  DISK    $LANA-MB        0        1       2         0

SAC \COMM.$ZZSTO.#SNDA.SAC-2.GRP-11.MOD-1.SLOT-51:

SAC \COMM.$ZZSTO.#SNDA.SAC-3.GRP-11.MOD-1.SLOT-51:

SAC \COMM.$ZZSTO.#SNDA.SAC-4.GRP-11.MOD-1.SLOT-51:
```

This report shows that the 6760 ServerNet device adapter has two disks on SAC 1.

See Explanation of Fields — Detailed INFO PMF ADAPTER Report.

Example of an INFO FCSA ADAPTER Report

```
-> INFO ADAPTER $ZZSTO.#FCSA.GRP-11.MOD-2.SLOT-1

STORAGE - Info ADAPTER \IO.$ZZSTO.#FCSA.GRP-11.MOD-2.SLOT-1
 Location     Access List
 (11,2,1)     2,3,0,1,4,5,6,7,8,9,10,11,12,13,14,15
```

This report shows that the storage adapter is accessed by processors 0 through 15.

Example of a Detailed INFO FCSA Adapter Report

```
-> INFO ADAPTER $ZZSTO.#FCSA.GRP-11.MOD-2.SLOT-1, DETAIL

STORAGE - Detailed Info ADAPTER \IO.$ZZSTO.#FCSA.GRP-11.MOD-2.SLOT-1

  ServerNet Addressable Controllers:

  SAC \IO.$ZZSTO.#FCSA.SAC-1.GRP-11.MOD-2.SLOT-1:
     Configured Devices ( group: 11, module: 2 ):

       Slot  Type    Name           Primary  Backup  Device ID
                                    CPU      CPU     /Port Name        Lun
          1  DISK    $PT00-B         3        2      50060E8003506012  0
          1  DISK    $PT01-B         3        2      50060E8003506012  1
          1  DISK    $PT02-B         3        2      50060E8003506012  2
          1  DISK    $PT03-B         3        2      50060E8003506012  3

  SAC \IO.$ZZSTO.#FCSA.SAC-2.GRP-11.MOD-2.SLOT-1:

     Configured Devices ( group: 11, module: 2 ):
       Slot  Type    Name           Primary  Backup  Device ID
                                    CPU      CPU     /Port Name        Lun
          1  DISK    $PT00-M         3        2      50060E8003506011  0
          1  DISK    $PT01-M         3        2      50060E8003506011  1
          1  DISK    $PT02-M         3        2      50060E8003506011  2
          1  DISK    $PT03-M         3        2      50060E8003506011  3
```

Displaying Storage Adapter Status Information

The “STATUS ADAPTER Command” (page 279) displays current information about storage adapters and connected devices:

- To display status of the adapter in the PMF CRU in group 1, slot 50:

  ```
  -> STATUS ADAPTER $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50
  ```

  The display is shown and explained under “Example of a STATUS PMF ADAPTER Report” (page 158).

- To display detailed status of the same adapter in the PMF CRU:

```
-> STATUS ADAPTER $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50, DETAIL
```

The display is shown and explained under "Example of a Detailed STATUS PMF ADAPTER Report" (page 158).

- To display status of the adapter in the IOMF CRU in group 11, slot 50:

  ```
  -> STATUS ADAPTER $ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-50
  ```

  The display is shown and explained under "Example of a STATUS IOMF ADAPTER Report" (page 160).

- To display detailed status of the same adapter in the IOMF CRU:

  ```
  -> STATUS ADAPTER $ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-50, DETAIL
  ```

  The display is shown and explained under "Example of a Detailed STATUS IOMF ADAPTER Report" (page 160).

- To display status of the 6760 ServerNet device adapter in group 11, slot 51:

  ```
  -> STATUS ADAPTER $ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-51
  ```

  The display is shown and explained under "Example of a STATUS SNDA ADAPTER Report" (page 161).

- To display detailed status of the same 6760 ServerNet device adapter:

  ```
  -> STATUS ADAPTER $ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-51, DETAIL
  ```

  The display is shown and explained under "Example of a Detailed STATUS SNDA ADAPTER Report" (page 161).

Example of a STATUS PMF ADAPTER Report

```
-> STATUS ADAPTER $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50

STORAGE - Status ADAPTER \COMM.$ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50
 Location     Status            POST    Power-1  Power-2  SACs
 (1,1,50)     PRESENT           PASSED  ON       ON       3
```

Explanation of Fields — STATUS PMF ADAPTER Report

Location	The physical location of the adapter (group, module, slot).
Status	Shows whether the adapter is present.
POST	Shows whether the power-on self-test (POST) passed.
Power-1	Shows whether power rail 1 has power on.
Power-2	Shows whether power rail 2 has power on.
SACs	The number of SACs present in the adapter (three SACs for a PMF CRU or IOMF CRU, up to four SACs for a ServerNet/DA).

Example of a Detailed STATUS PMF ADAPTER Report

```
-> STATUS ADAPTER $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50, DETAIL

STORAGE - Detailed Status ADAPTER \COMM.$ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50
 Adapter Type..... PMF
 Location......... (1,1,50)        Number of SACs... 3
 Part ID.......... 142184          POST Result...... PASSED
 Power-1.......... ON              Power-2.......... ON
 Revision Level... A03-05          Status........... PRESENT
 Tracking Number.. G25XC8          Vendor ID........

 ServerNet Addressable Controllers:

   SAC \COMM.$ZZSTO.#PMF.SAC-1.GRP-1.MOD-1.SLOT-50:
```

```
 SAC Number..... 1                Firmware.......... T1067AAO
 POST Result.... PASSED           SAC SSCI ID....... 6
 SAC Type....... SCSI             Side.............. X-Fabric
 Status......... PRESENT

  Configured Devices ( group: 1 , module: 1  ):
    Slot  Name           State      Substate         Primary  Backup
                                                     PID      PID
       2  $OPER-MB       STARTED                      0,298    1,269
       4  $MAG1-MB       STARTED                      0,295    1,272
       8  $AOL1-MB       STARTED                      0,287    1,280
      12  $SYSTEM-MB     STARTED                      0,257    1,257
      14  $DSMSCM-MB     STARTED                      0,291    1,276
      16  $AUDIT-MB      STARTED                      0,288    1,279
      18  $NEWSYS-MB     STARTED                      0,297    1,270

SAC \COMM.$ZZSTO.#PMF.SAC-2.GRP-1.MOD-1.SLOT-50:
 SAC Number.... 2                 Firmware.......... T1067AAOi
 POST Result.... PASSED           SAC SCSI ID....... 6
 SAC Type....... SCSI             Side.............. X-Fabric
 Status......... PRESENT

  Configured Devices ( group: 1 , module: 1  ):
    Slot  Name           State      Substate         Primary  Backup
                                                     PID      PID
       1  $OPER-P       *STARTED                      0,298    1,269
       3  $MAG1-P       *STARTED                      0,295    1,272
       7  $AOL1-P       *STARTED                      0,287    1,280
      11  $SYSTEM-P     *STARTED                      0,257    1,257
      13  $DSMSCM-P     *STARTED                      0,291    1,276
      15  $AUDIT-P      *STARTED                      0,288    1,279
      17  $NEWSYS-P      STOPPED    HARDDOWN          0,297    1,270

SAC \COMM.$ZZSTO.#PMF.SAC-3.GRP-1.MOD-1.SLOT-50:
 SAC Number....  3                Firmware........   T1067AAO
 POST Result.... PASSED           SAC SCSI ID...     6
 SAC Type....... SCSI             Side.............. X-Fabric
 Status......... PRESENT

  Configured Devices ( group: 1 , module: 1  ):
    Slot  Name           State      Substate         Primary  Backup
                                                     PID      PID
      50  $DLT22         STARTED                  0,284    1,284
```

This report shows that the PMF adapter:

- Has mirrored disks on SAC 1 and SAC 2
- Has a device on SAC 3

Explanation of Fields — Detailed STATUS PMF ADAPTER Report

Adapter Type	The type of adapter. Possible values: PMF, PMF2, IOMF, IOMF2, SNDA, and FCSA
Location	The location of the adapter (group, module, slot).
Number of SACs	The number of SACs present in the adapter (3 SACs for a PMF CRU or IOMF CRU, up to 4 SACs for an SNDA, and 2 SACs for an FCSA).
Part ID	The vendor number stored in the adapter.
POST Result	Shows whether the power-on self-test (POST) passed.
Power-1	Shows whether power rail 1 has power on.
Power-2	Shows whether power rail 2 has power on.

Revision Level	The revision level stored in the adapter.
Status	Shows whether the adapter is present.
Tracking Number	The vendor ID stored in the adapter.
Vendor ID	The manufacturer of the adapter (when available).
ServerNet Addressable Controllers:	
SAC Number	The SAC within the adapter.
Firmware	The part number of the firmware.
POST Result	Shows whether the power-on self-test (POST) passed.
SAC Type	The protocol type of the SAC.
SAC SSCI ID	The SAC ID, which is used to locate a specific SAC in the adapter.
Side	Shows whether the SAC is accessed by the X fabric or Y fabric.
Status	Shows whether the SAC is present.
Configured Devices (group: *n*, module: *n*):	
Slot	For internal disks, this is the slot number of the device. For 45*xx* disks or tape drives, this is the location of the adapter controlling the 45*xx* disk or tape drive.
Name	The configured name of the device accessed by this SAC.
State	The current SCF object state of the device (an asterisk (*) indicates that the disk or Open SCSI path is in use).
Substate	The current SCF substate of the device, if available.
Primary PID Backup PID	The processor number and PIN of the primary and backup I/O processes.

Example of a STATUS IOMF ADAPTER Report

```
-> STATUS ADAPTER $ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-50

STORAGE - Status ADAPTER \COMM.$ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-50
 Location     Status            POST     Power-1   Power-2   SACs
 (11,1,50)    PRESENT           PASSED   ON        ON        3
```

See Explanation of Fields — Detailed STATUS PMF ADAPTER Report.

Example of a Detailed STATUS IOMF ADAPTER Report

```
-> STATUS ADAPTER $ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-50, DETAIL

STORAGE - Detailed Status ADAPTER \COMM.$ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-50
 Adapter Type..... IOMF
 Location......... (11,1,50)       Number of SACs... 3
 Part ID.......... 129574          POST Result...... PASSED
 Power-1.......... ON              Power-2.......... ON
 Revision Level... A10-15          Status........... PRESENT
 Tracking Number.. G23LAJ          Vendor ID........

 ServerNet Addressable Controllers:

   SAC \COMM.$ZZSTO.#IOMF.SAC-1.GRP-11.MOD-1.SLOT-50:
    SAC Number..... 1              Firmware.......... T1067AAO
    POST Result.... PASSED         SAC SSCI ID....... 6
    SAC Type....... SCSI           Side.............. X-Fabric
    Status......... PRESENT
```

```
  Configured Devices ( group: 11 , module: 1  ):
    Slot  Name           State      Substate         Primary  Backup
                                                     PID      PID
       2  $ESD001-MB     STARTED                      0,292    1,275
       4  $ESD002-MB     STARTED                      0,293    1,274
       6  $D11105-MB     STOPPED    HARDDOWN          0,289    1,278
       8  $MAIL-MB       STARTED                      0,296    1,271
      12  $P1D04-MB      STARTED                      0,299    1,268
      14  $P5D04-MB      STARTED                      0,300    1,267
      16  $P6D04-MB      STARTED                      0,301    1,265
      18  $D11117-MB     STOPPED    HARDDOWN          0,290    1,277

SAC \COMM.$ZZSTO.#IOMF.SAC-2.GRP-11.MOD-1.SLOT-50:
 SAC Number..... 2              Firmware.......... T1067AAO
 POST Result.... PASSED         SAC SSCI ID....... 6
 SAC Type....... SCSI           Side.............. X-Fabric
 Status......... PRESENT

  Configured Devices ( group: 11 , module: 1  ):
    Slot  Name           State      Substate         Primary  Backup
                                                     PID      PID
       1  $ESD001-P     *STARTED                      0,292    1,275
       3  $ESD002-P     *STARTED                      0,293    1,274
       5  $D11105-P      STOPPED    HARDDOWN          0,289    1,278
       7  $MAIL-P       *STARTED                      0,296    1,271
      11  $P1D04-P      *STARTED                      0,299    1,268
      13  $P5D04-P      *STARTED                      0,300    1,267
      15  $P6D04-P      *STARTED                      0,301    1,265
      17  $D11117-P      STOPPED    HARDDOWN          0,290    1,277

SAC \COMM.$ZZSTO.#IOMF.SAC-3.GRP-11.MOD-1.SLOT-50:
 SAC Number..... 3              Firmware.......... T1067AAO
 POST Result.... PASSED         SAC SSCI ID....... 6
 SAC Type....... SCSI           Side.............. X-Fabric
 Status......... PRESENT

  Configured Devices ( group: 11 , module: 1  ):
    Slot  Name           State      Substate         Primary  Backup
                                                     PID      PID
      50  $L700C16-P    *STARTED                      1,282    0,286
```

See Explanation of Fields — Detailed STATUS PMF ADAPTER Report.

Example of a STATUS SNDA ADAPTER Report

```
-> STATUS ADAPTER $ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-51

STORAGE - Status ADAPTER \COMM.$ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-51
 Location     Status            POST     Power-1   Power-2   SACs
 (11,1,51)    PRESENT           PASSED   ON        ON        4
```

See Explanation of Fields — Detailed STATUS PMF ADAPTER Report.

Example of a Detailed STATUS SNDA ADAPTER Report

```
-> STATUS ADAPTER $ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-51, DETAIL
STORAGE - Detailed Status ADAPTER \COMM.$ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-51
 Adapter Type..... SNDA
 Location......... (11,1,51)        Number of SACs... 4
 Part ID.......... 123201           POST Result...... PASSED
 Power-1.......... ON               Power-2.......... ON
 Revision Level... B02-04           Status........... PRESENT
 Tracking Number.. VONRAS           Vendor ID........

 ServerNet Addressable Controllers:
```

```
SAC \COMM.$ZZSTO.#SNDA.SAC-1.GRP-11.MOD-1.SLOT-51:
 SAC Number..... 1               Firmware.......... T0054AAF
 POST Result.... PASSED          SAC SSCI ID....... 0
 SAC Type....... FIBER           Side.............. X-Fabric
 Status......... PRESENT

  Configured Devices ( group: 11 , module: 1  ):
    Slot  Name            State      Substate          Primary  Backup
                                                       PID      PID
      51  $LANA-P        *STARTED                       0,294    1,273
      51  $LANA-MB        STARTED                       0,294    1,273

SAC \COMM.$ZZSTO.#SNDA.SAC-2.GRP-11.MOD-1.SLOT-51:
 SAC Number..... 2               Firmware.......... T0054AAF
 POST Result.... PASSED          SAC SSCI ID....... 0
 SAC Type....... FIBER            Side............. X-Fabric
 Status......... PRESENT

  Configured Devices ( group: 11 , module: 1  ):
    Slot  Name            State      Substate          Primary  Backup
                                                       PID      PID

SAC \COMM.$ZZSTO.#SNDA.SAC-3.GRP-11.MOD-1.SLOT-51:
 SAC Number..... 3               Firmware.......... T0054AAF
 POST Result.... PASSED          SAC SSCI ID....... 0
 SAC Type....... FIBER           Side.............. X-Fabric
 Status......... PRESENT

  Configured Devices ( group: 11 , module: 1  ):
    Slot  Name            State      Substate          Primary  Backup
                                                       PID      PID

SAC \COMM.$ZZSTO.#SNDA.SAC-4.GRP-11.MOD-1.SLOT-51:
 SAC Number..... 4               Firmware.......... T0054AAF
 POST Result.... PASSED          SAC SSCI ID....... 0
 SAC Type....... FIBER           Side.............. X-Fabric
 Status......... PRESENT

  Configured Devices ( group: 11 , module: 1  ):
    Slot  Name            State      Substate          Primary  Backup
                                                       PID      PID
```

See Explanation of Fields — Detailed STATUS PMF ADAPTER Report.

Example of a STATUS FCSA ADAPTER Report

```
 -> STATUS ADAPTER $ZZSTO.#FCSA.GRP-11.MOD-2.SLOT-1

Location     Status        POST       Power-1  Power-2  SACs
(11,2,1)    PRESENT       PASSED     ON        ON        2
```

See Explanation of Fields — Detailed STATUS PMF ADAPTER Report.

Example of a STATUS FCSA ADAPTER, SACS

```
-> STATUS ADAPTER $ZZSTO.#FCSA.GRP-11.MOD-2.SLOT-1, SACS

STORAGE - Detailed Status ADAPTER \IO.$ZZSTO.#FCSA.GRP-11.MOD-2.SLOT-1
 Adapter Type..... FCSA

 Flash Boot....... T0612G06         Flash Update Active.... Done
 Flash Firmware... T0630G06         Flash Update Result.... Init
 Location......... (11,2,1)         Number of SACs......... 2
 Part ID.......... 526217-001       POST Result............ PASSED
 Power-1.......... ON               Power-2................ ON
 Revision Level... A01-01           Status................. PRESENT
```

```
Tracking Number.. MP0008          Vendor ID.............

ServerNet Addressable Controllers:

  SAC \IO.$ZZSTO.#FCSA.SAC-1.GRP-11.MOD-2.SLOT-1:
   SAC Number..... 1              Firmware.......... T0630G06
   POST Result.... PASSED         SAC SCSI ID.......
   SAC Type....... FC             Side.............. Both
   Status......... PRESENT

   Connection..... F port         Node Name......... 20060B00001CE5F8
   Port Id........ 65792          Port Name......... 50060B00001CE5F8
   SAC State...... Ready          SAC Subtype....... 1

  SAC \IO.$ZZSTO.#FCSA.SAC-2.GRP-11.MOD-2.SLOT-1:
   SAC Number..... 2              Firmware.......... T0630G06
   POST Result.... PASSED         SAC SCSI ID.......
   SAC Type....... FC             Side.............. Both
   Status......... PRESENT

   Connection..... Loop           Node Name......... 20060B00001CE5FA
   Port Id........ 1              Port Name......... 50060B00001CE5FA
   SAC State...... Ready             SAC Subtype....... 1
```

Explanation of Fields — STATUS FCSA ADAPTER, SACS

Flash BootT	The name of the flash boot file.
FirmwareT	The name of the flash firmware file
Flash Update Active	The status of the flash update operation
Flash Update Result	The state resulting from the flash update operation

ServerNet Addressable Controllers:

Connection	Indicates the Fibre Channel connections type.
Port Id	The controller identifier from the Fabric Switch Name Server. (For the Loop connection type, Port Id is equal to controller AL_PA.)
SAC State	Indicates the state of the SAC:

INIT	Initial state, not downloaded
READY	Downloaded, ready to accept commands
NO-HW	No hardware, SAC not installed
DOWNLOADING	Downloading
FW_RR	SAC firmware error
HW_ERR	SAC hardware error
MAX_DOWNLOAD	Downloads exceed limit

Node Name	The worldwide name (WWN) of the controller node.
Port Name	The worldwide name (WWN) of the controller port.
SAC Subtype	Indicates the type of SAC. The numeral one (1) indicates a two-port, 2 GB Fibre Channel SAC.

For an explanation, see Explanation of Fields — Detailed STATUS PMF ADAPTER Report and "Example of a Detailed STATUS FCSA ADAPTER Report" (page 164).

Example of a STATUS FCSA ADAPTER, SERVERNET Report

```
-> STATUS ADAPTER $ZZSTO.#FCSA.GRP-11.MOD-2.SLOT-1, SERVERNET

STORAGE - ServerNet Status ADAPTER \IO.$ZZSTO.#FCSA.GRP-11.MOD-2.SLOT-1
     CPU|  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
 fabric |

 -------+-----------------------------------------------------------------
 X-from | UNK UNK UP  UP  DWN DWN DWN DWN DWN DWN DWN DWN DWN DWN DWN DWN
 X-to   | UNK UNK UP  UP  DWN DWN DWN DWN DWN DWN DWN DWN DWN DWN DWN DWN
 -------+-----------------------------------------------------------------
 Y-from | UNK UNK UP  UP  DWN DWN DWN DWN DWN DWN DWN DWN DWN DWN DWN DWN
 Y-to   | UNK UNK UP  UP  DWN DWN DWN DWN DWN DWN DWN DWN DWN DWN DWN DWN
```

Explanation of Fields — STATUS FCSA ADAPTER, SERVERNET Report	
x-from	Iindicates the ServerNet X-fabric status on the link from each CPU to the specified FCSA.
y-to	Indicates the ServerNet Y-fabric status on the link to each CPU from the specified FCSA.

Example of a STATUS FCSA ADAPTER, VPROCS Report

```
-> STATUS ADAPTER $ZZSTO.#FCSA.GRP-11.MOD-2.SLOT-1, VPROCS

STORAGE - Vprocs Status ADAPTER \IO.$ZZSTO.#FCSA.GRP-11.MOD-2.SLOT-1
 SAC-1 Firmware.......T0630G06^25JUN2004^20APR2004^15
 SAC-2 Firmware.......T0630G06^25JUN2004^20APR2004^15
 Flash Firmware.......T0630G06^25JUN2004^23APR2004^16
 Flash Boot..........T0612G06_04JUN2004_13APR2004
```

Explanation of Fields — STATUS FCSA ADAPTER, VPROCS Report	
SAC-1 Firmware	The SAC-1 operational firmware downloaded from the NonStop system to the adapters random access memory.
SAC-2 Firmware	The SAC-2 operational firmware downloaded from the NonStop system to the adapters random access memory.
Flash Firmware	The copy of the operational firmware in the FCSA flash memory.
Flash Boot	The copy of the flash boot firmware in the FCSA flash memory.

Example of a Detailed STATUS FCSA ADAPTER Report

```
-> status adapter $ZZSTO.#FCSA.GRP-11.MOD-2.SLOT-1, DETAIL

STORAGE - Detailed Status ADAPTER \IO.$ZZSTO.#FCSA.GRP-11.MOD-2.SLOT-1
 Adapter Type..... FCSA

 Flash Boot....... T0612G06        Flash Update Active.... Done
 Flash Firmware... T0630G06        Flash Update Result.... Init
 Location......... (11,2,1)        Number of SACs......... 2
 Part ID.......... 526217-001      POST Result............ PASSED
 Power-1.......... ON              Power-2................ ON
 Revision Level... A01-01          Status................. PRESENT
 Tracking Number.. MP0008          Vendor ID..............

 ServerNet Addressable Controllers:


   SAC \IO.$ZZSTO.#FCSA.SAC-1.GRP-11.MOD-2.SLOT-1:
    SAC Number..... 1              Firmware.......... T0630G06
    POST Result.... PASSED         SAC SCSI ID.......
    SAC Type....... FC             Side.............. Both
    Status......... PRESENT
```

```
    Connection..... F port          Node Name......... 20060B00001CE5F8
    Port Id........ 65792           Port Name......... 50060B00001CE5F8
    SAC State...... Ready           SAC Subtype....... 1

     Configured Devices ( group: 11, module: 2 ):

       Slot  Name           State      Substate          Primary  Backup
                                                         PID      PID
          1  $PT00-B        STARTED                       2,272    3,264
          1  $PT01-B        STARTED                       3,283    2,273
          1  $PT02-B        STARTED                       3,282    2,274
 SAC \IO.$ZZSTO.#FCSA.SAC-2.GRP-11.MOD-2.SLOT-1:
    SAC Number..... 2               Firmware.......... T0630G06
    POST Result.... PASSED          SAC SCSI ID.......
    SAC Type....... FC              Side.............. Both
    Status......... PRESENT

    Connection..... Loop            Node Name......... 20060B00001CE5FA
    Port Id........ 1               Port Name......... 50060B00001CE5FA
    SAC State...... Ready           SAC Subtype....... 1


     Configured Devices ( group: 11, module: 2 ):

       Slot  Name           State      Substate          Primary  Backup
                                                         PID      PID
          1  $PT00-M       *STARTED                       2,272    3,264
          1  $PT01-M       *STARTED                       3,283    2,273
          1  $PT02-M       *STARTED                       3,282    2,274
```

Explanation of Fields — Detailed STATUS FCSA ADAPTER Report

Flash Boot	The name of the flash boot file.
Firmware	The name of the flash firmware file
Flash Update Active	Indicates the status of the flash update operation
Flash Update Result	Indicates the state resulting from the flash update operation

ServerNet Addressable Controllers:

Connection	Indicates the Fibre Channel connections type.
Port Id	The controller identifier from the Fabric Switch Name Server. (For the Loop connection type, Port Id is equal to controller AL_PA.)
SAC State	indicates the state of the SAC:

INIT	Initial state, not downloaded
READY	Downloaded, ready to accept commands
NO-HW	No hardware, SAC not installed
DOWNLOADING	Downloading
FW_RR	SAC firmware error
HW_ERR	SAC hardware error
MAX_DOWNLOAD	Downloads exceed limit

Node Name	The worldwide name (WWN) of the controller node.

Port Name	The worldwide name (WWN) of the controller port.
SAC Subtype	Indicates the type of SAC. The numeral one (1) indicates a two-port, 2 GB Fibre Channel SAC

For an explanation of the other fields, see Explanation of Fields — Detailed STATUS PMF ADAPTER Report.

Configuring a Storage Adapter

Configuration tasks for storage adapters include:

- "Checking the Automatic Configuration of Storage Adapters" (page 166)
- "Deleting an Adapter" (page 166)

Checking the Automatic Configuration of Storage Adapters

When you physically install an adapter, it is automatically added to the system configuration database. The system assigns the fabric based on the slot number:

- X fabric: slot 50, 51, or 53
- Y fabric: slot 52, 54, or 55

Naming Conventions for Storage Adapters

The system derives the name of the adapter by combining the name of the storage subsystem manager process ($ZZSTO) with the adapter type (PMF, IOMF, or SNDA) and the physical location of the adapter.

Example

1. Physically install the device.
2. Use the "INFO ADAPTER Command" (page 253) to determine the name of the storage adapter. For example:
   - A PMF CRU located in group 04, module 1, slot 50 is on the X fabric and is assigned the name:

     ```
     $ZZSTO.#PMF.GRP-4.MOD-1.SLOT-50
     ```
   - An IOMF CRU located in group 21, module 1, slot 52 is on the Y fabric and is assigned the name:

     ```
     $ZZSTO.#IOMF.GRP-21.MOD-1.SLOT-52
     ```
   - A ServerNet/DA CRU located in group 01, module 1, slot 53 is on the X fabric and is assigned the name:

     ```
     ADAPTER $ZZSTO.#SNDA.GRP-1.MOD-1.SLOT-53
     ```
3. Use the "STATUS ADAPTER Command" (page 279) to verify the adapter is available and ready to use.
4. If a problem occurs, see "Troubleshooting Adapter Installation" (page 167).

Deleting an Adapter

All ServerNet addressable controller (SAC) records associated with the adapter are deleted.

Considerations for DELETE ADAPTER

You must physically remove the storage adapter before deleting the adapter from the system configuration database. If you do not physically remove the deleted CRU from the system, the system automatically adds an installed adapter back to the system configuration database in any of these cases:

- You remove the adapter and then insert it back into the slot.
- You restart the system.
- You power off and on the adapter or its enclosure.
- Example

1. Identify all the devices currently using the adapter:

   ```
   -> STATUS $ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-51, DETAIL
   ```

   See "Example of a Detailed INFO SNDA ADAPTER Report" (page 156).
2. Stop each of the devices or device paths that use the adapter.
3. If any disks belong to a storage pool, you get an error message. You must first remove them from the pool. (See the *Storage Management Foundation User's Guide*.)
4. Delete all paths connected to this adapter.
5. Physically remove the adapter.
6. Delete the adapter from the system configuration database:

   ```
   -> DELETE $ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-51
   ```

Managing a Storage Adapter

Management tasks for storage adapters include:

- "Troubleshooting Adapter Installation" (page 167)
- "Changing the Active Path for a Storage Adapter" (page 167)

Troubleshooting Adapter Installation

If the slot that you want to install an adapter in was previously configured for another adapter, you get an EMS message (storage message 1041) and the adapter is not configured. To remove the previous configuration:

1. Find out what storage or LAN adapter is currently configured for that slot:

   ```
   -> INFO ADAPTER $*.*
   ```
2. If that slot is configured for a LAN adapter, stop and delete all subordinate objects like LIFs and PIFs.
3. Delete the adapter:

   ```
   -> DELETE ADAPTER $ZZSTO.#type.GRP-n.MOD-1.SLOT-n

   -> DELETE ADAPTER $ZZLAN.#type.GRP-n.MOD-1.SLOT-n
   ```
4. Physically install the new adapter.
5. Verify the adapter is available and ready to use:

   ```
   -> STATUS ADAPTER $ZZSTO.#type
   ```

Changing the Active Path for a Storage Adapter

The "SWITCH ADAPTER Command" (page 294) moves all device paths from and to a SAC on an adapter.

Use this command before replacing a device that contains a storage adapter and dual paths.

Considerations for SWITCH ADAPTER

- The SWITCH ADAPTER command replaces a series of SWITCH commands for individual paths and devices in the topology branch.
- At the completion of the command, the SWITCH ADAPTER command returns a count of the number of paths that did not switch. Handle any exceptions individually.
- You can use this command only on devices that are configured with dual paths.
- Use the STATUS ADAPTER command to verify the path switch.
- Example

1. Stop all data paths to the storage adapter before removing a PMF CRU:

```
-> SWITCH ADAPTER $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-55, &
-> AWAY, FORCED
```

2. Verify the status of the storage adapter:

```
-> STATUS ADAPTER $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-55
```

3. Replace the PMF CRU.
4. Restore data paths to the storage adapter after inserting the replacement PMF CRU:

```
-> SWITCH ADAPTER $ZZSTO.#SNDA.GRP-21.MOD-1.SLOT-54, DEFAULT
```

Downloading New Firmware to Adapters and SACs

The “REPLACE ADAPTER Command” (page 269) downloads new firmware to adapters and SACs.

Considerations for REPLACE ADAPTER

- If the REPLACE ADAPTER command is entered correctly, the storage subsystem manager generates an EMS message. This message names the command, reports the time at which the command was executed, specifies the terminal from which the command was entered, and gives the group and user numbers of the user issuing the command.
- Operational firmware downloaded through a REPLACE command does not remain in effect beyond the next system load. A system load reverts to the standard firmware file names in the specified SYS*nn*.
- Flash firmware replacement is permanent until the next replacement.
- For operational firmware, you can use the CPU attribute or the ABANDON attribute to have different SACs use different versions of the same firmware type. After determining which version you want, use the REPLACE command to download the chosen version to all SACs that use that firmware type.
- For operational firmware, a complete download from all CPUs on a 16-processor system takes at least 8 minutes.

Displaying Information About Connections to SACs on an FCSA

You can use the STATUS SAC command to display current status information about connections to a SAC on an FCSA, including worldwide names and available LUNs. For details, see “STATUS SAC Command” (page 285).

You can use the STATS SAC command to display statistical information about the connections to a SAC on an FCSA. You can get information about a port in the Fibre Channel link or the SAC itself. For details, see “STATS SAC Command” (page 278).

Testing Connections to the SACs on an FCSA

You can use the PROBE SAC command to test the connection to a SAC on an FCSA. The connection can be either an arbitrated loop or a link to a Fibre Channel switch. For details, see "PROBE SAC Command" (page 268).

12 Configuring and Managing Open SCSI Devices

An Open SCSI device is a device that obeys the ANSI standard protocol for the small computer system interface (SCSI) to communicate with the system through the Open SCSI I/O process. An Open SCSI device has the object type of SCSI.

The SCSI object can also represent Fibre Channel devices that use the SCSI protocol to communicate with the system. These devices are called Open SCSI Fibre Channel devices.

Unless stated otherwise, references to Open SCSI devices also refer to Open SCSI Fibre Channel devices. For details about Open SCSI devices, see "The SCSI Object" (page 38) and "SCSI Object States" (page 38). For commands that affect SCSI devices, see "SCF Commands and Object Types" (page 190). This chapter describes:

Configuring Paths for Open SCSI Devices

The values you specify when configuring an Open SCSI device vary depending on the type of connection and the NonStop server model. For example, the valid ranges for group, module, and slot of an Open SCSI device connected to an FCSA on an Integrity NonStop NS-series server differ from those of an Open SCSI device connected to an ServerNet/DA on a NonStop S-series server

NOTE: When a NonStop S-series I/O enclosure is attached to an Integrity NonStop NS-series system, the enclosure retains the same group number it had on the NonStop S-series system. However, if the appropriate port for the group number is not available on the Integrity NS-series server, you must change the group number of the I/O enclosure. For more information, see the *NonStop NS-Series Hardware Installation Manual.*

"Open SCSI Path Attributes" (page 171) lists the possible attribute values for the different types of Open SCSI connections.

Table 10 Open SCSI Path Attributes

	SNDA (S-pic) Connection	PMF/IOMF Connection	FCSA Connection Integrity NonStop NS-series
LOCATION			
Group	1-89	1-89	110-125
Module	1	1	2-3 (IOAM)
Slot	51-54	50 or 55	1-5 (FCSA)
SAC	1-4	3	1-2
SCSIID	0-15 (except 6-7)	0-15 (except 6-7)	NA
PORTNAME	NA	NA	64-bit WWN
LUN	0-7	0-7	0-32767

LOCATION represents PRIMARYLOCATION and BACKUPLOCATION.
SAC represents PRIMARYSAC and BACKUPSAC.
PORTNAME represents PRIMARYPORTNAME and BACKUPPORTNAME.

Displaying Information About an Open SCSI Device

Displaying Configuration Information

The "INFO SCSI Command" (page 259) displays configured information about an Open SCSI device.

Example of an INFO SCSI Report

```
-> INFO SCSI $S11500

STORAGE - Info SCSI configuration \COMM.$S11500
Primary     Backup      *SCSI ID / Port     *LUN    *Program
Location    Location
(11,1,50)               0                    0      $SYSTEM.SYSTEM.TDSCSI
```

Explanation of Fields — INFO SCSI Report	
*	Indicates an attribute whose value you can change by using an ALTER SCSI command.
Primary Location Backup Location	The primary and backup physical locations of the adapter where the device is attached
SCSI ID / Port	For Open SCSI devices, the SCSI ID that the IOP uses to access the device. For Open SCSI Fibre Channel devices, the worldwide name (WWN) that the IOP uses to access the device.
LUN	The logical unit number (LUN) of the device used by the IOP to select any additional devices that are attached to the device.
Program	The object file name of the IOP.

Examples of Detailed INFO SCSI Reports

This example shows an INFO DETAIL report for an Open SCSI device connected to an IOMF CRU:

```
-> INFO SCSI $S11500, DETAIL

STORAGE - Detailed Info SCSI configuration \COMM.$S11500
```

```
I/O Process Information:
*BackupCpu........................... 0
*HighPin............................. ON
*PrimaryCpu.......................... 1
*Program............................. $SYSTEM.SYSTEM.TDSCSI
*RecSize............................. 57344
*StartState.......................... STARTED

SCSI Device Settings:
*MaxOpens............................ 4
*NumIO............................... 8
*StructAreaSize...................... 24
*TraceBufLen......................... 33
*TraceLevel.......................... 65535

Primary Path Info:
 Adapter Name........................ $ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-50
 Adapter Location (group,module,slot).. (11,1,50)
*LUN................................. 0
 SAC Name............................ IOMF.SAC-3.GRP-11.MOD-1.SLOT-50
 SAC Number.......................... 3
*SCSI ID / Portname.................. 0
```

This example shows an INFO DETAIL report for an Open SCSI Fibre Channel device connected to an FCSA on an Integrity NonStop NS-series server:

```
-> INFO SCSI $EXT8, DETAIL

STORAGE - Detailed Info SCSI configuration \NSCOMM.$EXT8
I/O Process Information:
*BackupCpu........................... 1
*HighPin............................. ON
*PrimaryCpu.......................... 1
*Program............................. $SYSTEM.SYSTEM.TDSCSI
*RecSize............................. 57344
*StartState.......................... STARTED

SCSI Device Settings:
*MaxOpens............................ 4
*NumIO............................... 8
*StructAreaSize...................... 24
*TraceBufLen......................... 33
*TraceLevel.......................... 65535

Primary Path Info:
 Adapter Name........................ $ZZSTO.#FCSA.GRP-111.MOD-2.SLOT-5
 Adapter Location (group,module,slot).. (111,2,5)
*LUN................................. 0
 SAC Name............................ FCSA.SAC-1.GRP-111.MOD-2.SLOT-5
 SAC Number.......................... 1
*SCSI ID / Portname.................. 2000003E00452211
```

Explanation of Fields

I/O Process Information:

*	Indicates an attribute whose value you can change by using an ALTER SCSI command.
BackupCpu	The processor number of the backup IOP.
HighPin	Shows whether the IOP can run at a high PIN (ON) or a low PIN (OFF). The PIN is the process identification number.
PrimaryCpu	The processor number of the primary IOP.
Program	The object file name of the IOP.

RecSize	The configured record size (in bytes) for the device.
StartState	Shows whether the IOP is configured to be in the STARTED or state when the system is loaded.
SCSI Device Settings:	
MaxOpens	The maximum number of concurrent opens allowed by the IOP.
NumIO	The maximum number of concurrent I/O operations the IOP can have outstanding; used by the SCSI IOP to determine the size of allocated data-buffer area.
StructAreaSize	The memory (in 1-kilobyte units) to be allocated for data structures by the IOP.
TraceBufLen	The memory (in 1-kilobyte units) to be allocated for internal trace data by the IOP.
TraceLevel	The internal IOP trace level.
Path Information:	
Adapter Name	The name of the PMF CRU, IOMF CRU, or ServerNet/DA that controls the access path to the device, determined by the location of the adapter in the system.
Adapter Location (Group, Module, Slot)	The group, module, and slot number of the adapter.
LUN	The logical unit number (LUN) of the device used by the IOP to select any additional devices that are attached to the device.
SAC Name	The name of the SAC on the adapter, determined by the location of the adapter in the system.
SAC Number	The subdevice number of the SAC on the adapter which determines which SCSI bus accesses the device.
SCSI ID / Portname	For Open SCSI devices, the SCSI ID that the IOP uses to access the device. For Open SCSI Fibre Channel devices, the worldwide name (WWN) that the IOP uses to access the device.

Displaying Status Information

"STATUS SCSI Command" (page 288) displays current status information about an Open SCSI device.

Example of a STATUS SCSI Report

```
-> STATUS $S11500

STORAGE - Status SCSI \COMM.$S11500
LDev   Primary    Backup     Primary   Backup
                              PID       PID
 316  *STARTED                1,282     0,286
```

Example of a Detailed STATUS SCSI Report

```
-> STATUS $S11500, DETAIL

STORAGE - Detailed Status SCSI \COMM.$S11500

SCSI Path Information:
 LDev   Primary    Backup     Primary   Backup
                               PID       PID
  316  *STARTED                1,282     0,286

SCSI Specific Information:
  Pending I/Os......... 0                High Pending I/Os..... 1
  Opens................ 1                Max Opens............. 4
  Open Paths........... 1                Max I/O Requests...... 8
  SIM Queue Status..... Normal           Max Transfer Length... 32767
  Tracing Level........ 65535
```

```
SCSI I/O Process Information:
  Device Type.......... 8                Device Subtype........ 0
  Physical Record Size. 4096             Priority.............. 220
  Library File.........
  Program File......... $SYSTEM.SYS01.TDSCSI
```

Explanation of Fields

SCSI Path Information:

LDev	The logical device number for the device. This number is arbitrarily assigned to a device when you configure the device and every time the system is loaded.
Path	The Open SCSI device path assignment.
Status	Shows whether the device path is the current path (ACTIVE) or not (INACTIVE).
State	The current state of the device path.
Primary PID Backup PID	The processor number and PIN of the current primary and backup IOP.

SCSI-Specific Information:

Pending I/Os	The current number of pending I/O requests against the device.
High Pending I/Os	The highest number of pending I/O requests that have been outstanding against the device since the IOP was started.
Opens	The current number of requesters that have the IOP open.
Max Opens	The maximum number of open requests allowed by the IOP.
Open Paths	The number of paths currently opened by the IOP.
Max I/O Requests	The maximum number of I/O requests allowed by the IOP.

SCSI I/O Process Information:

Device Type	8, the Open SCSI device type.
Device Subtype	0, the Open SCSI device subtype.
Physical Record Size	The size of the physical records (in bytes) on the device (retrieved from the SCSI IOP).
Priority	The current execution priority of the IOP.
Library File	Always blank.
Program File	The program file name of the IOP.

Configuring an Open SCSI Device

Adding an Open SCSI Device

The “ADD SCSI Command” (page 222) adds an Open SCSI device to the system configuration database:

1. Physically install the device.
2. Based on the manufacturer documentation, verify the SCSI ID of the device is the same as the configured SCSIID value.
3. Add the device to the system, specifying its group, module, and slot:

   ```
   -> ADD SCSI $S11500, SENDTO STORAGE, &
   -> PRIMARYLOCATION (11,1,50)
   ```

   For Open SCSI devices, the name convention is `$S`*`ggssi`*, where *gg* is the group number, *ss* is the slot number, and *i* is the SCSI ID.
4. Verify the configuration:

```
-> INFO $S11500
```

5. See “Starting an Open SCSI Device” (page 177).

Adding a Similar Open SCSI Device to This System

To add another Open SCSI device similar to an existing device, use the LIKE attribute:

```
-> ADD SCSI $S11501, LIKE $S11500, PRIMARYLOCATION (11,1,51)
```

Adding a Similar Open SCSI Device to Another System

To configure the same or a similar Open SCSI device on another system, create a command file by using the OBEYFORM attribute of the INFO DISK command. You can copy this file to another system or add it to different configuration file on the current system.

1. Capture the configuration for an existing Open SCSI device:

```
-> INFO / OUT LOG / $S11502, OBEYFORM

== STORAGE - Detailed Info SCSI in obeyform: \COMM.$S11502
ADD SCSI $S11502 , &
    SENDTO STORAGE , &
    BACKUPCPU 0 , &
    HIGHPIN ON , &
    LUN 0 , &
    MAXOPENS 4 , &
    NUMIO 8 , &
    PRIMARYCPU 1 , &
    PRIMARYLOCATION (11,1,50) , &
    PRIMARYSAC 5 , &
    PROGRAM $SYSTEM.SYSTEM.TDSCSI , &
    RECSIZE 57344 , &
    SCSIID 2 , &
    STARTSTATE STARTED , &
    STRUCTAREASIZE 24 , &
    TRACEBUFLEN 33 , &
    TRACELEVEL 65535
```

2. Optionally edit the resulting log file to specify:
   - A unique device name
   - Based on physical location, change at least one of:
     - PRIMARYLOCATION and, optionally, MIRRORLOCATION slot numbers
     - PRIMARYSAC and, optionally, MIRRORSAC SAC numbers, if the new locations use different SACs
     - SCSIID value
3. Enter the log file contents either by copying and pasting into an SCF command line or by using the log file as a command file.

Adding an Open SCSI Fibre Channel Device

To add an Open SCSI Fibre Channel device, you do not specify a SCSI ID. Instead, you specify the PRIMARYPORTNAME (WWN) and LUN of the device.

This command adds an Open SCSI Fibre Channel device to the system:

```
-> ADD SCSI $EXT8, SENDTO STORAGE, PRIMARYLOCATION (111,2,5) &
-> PRIMARYSAC 1, PRIMARYPORTNAME 2000003E00452211, LUN 1
```

NOTE: You can optionally specify a backup path consisting of BACKUPLOCATION, BACKUPSAC, and BACKUPPORTTNAME. However, there is no backup LUN. The same LUN is used for both the primary and backup paths.

Altering Open SCSI Attribute Values

The “ALTER SCSI Command” (page 237) alters the attributes of an Open SCSI device in the system configuration database. For a description of alterable Open SCSI attributes, see “ALTER SCSI Attributes” (page 238).

Considerations for ALTER SCSI

- The process must be in the STOPPED state or not running before you can change its configuration.
- Changes take effect when you restart the process.
- Verify the SCSI ID of the device is the same as the configured SCSIID value.
- Changing the Values of the Attributes for an Open SCSI Device

1. Stop the device:
   ```
   -> STOP $S11500
   ```
2. Verify the device is in the STOPPED state:
   ```
   -> STATUS $S11500
   ```
3. Change one or more attributes. For example:
   ```
   -> ALTER $S11500, STARTSTATE STOPPED
   ```
4. Verify the change took place:
   ```
   -> INFO $S11500, DETAIL
   ```
5. Start the device:
   ```
   -> START $S11500
   ```
   When the START SCSI command finishes successfully, the device is left in the STARTED state.

Deleting an Open SCSI Device

The “DELETE SCSI Command” (page 252) removes an Open SCSI device from the system configuration database:

1. Stop the device:
   ```
   -> STOP $S11500
   ```
2. Verify the device is in the STOPPED state:
   ```
   -> STATUS $S11500
   ```
3. Remove the device from the system configuration database:
   ```
   -> DELETE $S11500
   ```
4. Verify the device is removed from the system configuration database:
   ```
   -> INFO $S11500
   ```

Managing an Open SCSI Device

Resetting an Open SCSI Device

The “RESET SCSI Command” (page 272) puts an Open SCSI device into the STOPPED state, substate DOWN, reading for restarting:

1. Check the current status of the Open SCSI device:
   ```
   -> STATUS $S11500
   ```
2. If it is not in the STOPPED state, substate DOWN:
   ```
   -> RESET $S11500
   ```

3. Start the Open SCSI device:

```
-> START $S11500
```

4. Verify the process is started:

```
-> STATUS $S11500
```

Starting an Open SCSI Device

The "START SCSI Command" (page 276) puts the storage pool into the STARTED state:

1. Start the device:

```
-> START $S11500
```

2. Check the status of the process:

```
-> STATUS $S11500
```

Stopping an Open SCSI Device

The "STOP SCSI Command" (page 292) places an Open SCSI device in the STOPPED state:

1. Stop the device:

```
-> STOP $S11500
```

2. Verify the device is in the STOPPED state:

```
-> STATUS $S11500
```

Changing the Active Data Path for an Open SCSI Device

The "SWITCH SCSI Command" (page 296) switches the primary and backup paths to an Open SCSI device. This command also designates the preferred SAC path for any device accessible through dual paths.

Considerations for SWITCH SCSI

- You can use this command only on Open SCSI devices that are configured with dual paths (by using PRIMARYLOCATION and BACKUPLOCATION attributes in the ADD SCSI command).
- Use the STATUS SCSI command to verify the path switch.

Example

1. Switch the primary and backup paths of the device:

```
-> SWITCH $S11500-P
```

2. Verify the status of the switch:

```
-> STATUS $S11500
```

Swapping Processors for an Open SCSI Device

The "PRIMARY SCSI Command" (page 266) swaps the primary and backup processors for an Open SCSI device. The current primary processor of a specified device becomes the backup processor, and the backup processor becomes the primary processor, but the PRIMARYCPU and BACKUPCPU values stay the same. You typically swap processors when load balancing the system or preparing for device replacement.

Examples

- To execute the primary process of $DEV1 in processor 3 (assuming it is configured to run in processor 3):

```
-> PRIMARY $DEV1, 3
```

- To make the current backup process of $DEV1 the primary process:

```
-> PRIMARY $DEV1
```

13 Configuring and Managing Tape Drives

This chapter describes configuring tape drives. The tape drive processes on the system have the object type of TAPE. For details about this device, see “The TAPE Object” (page 39)and “TAPE Object States” (page 39). For commands that can affect this device, see “Storage Subsystem Commands” (page 190). This chapter describes:

Configuring Paths for Tape Devices

The values you specify when configuring a tape device depend on the type of connection and the NonStop server model. For example, the valid ranges for group, module, and slot of a tape device connected to an FCSA on an Integrity NonStop NS-series server differ from those of a tape device connected to an IOMF CRU on a NonStop S-series server.

NOTE: When a NonStop S-series I/O enclosure is attached to an Integrity NonStop NS-series system, the enclosure retains the same group number it had on the NonStop S-series system. However, if the appropriate port for the group number is not available on the Integrity NonStop NS-series server, you must change the group number of the I/O enclosure. For more information, see the *NonStop NS-Series Hardware Installation Manual.*

“Tape Device Path Attributes” (page 179) lists the possible attribute values for the different types of tape device paths.

Table 11 Tape Device Path Attributes

	SNDA Connection	PMF/IOMF Connection	FCSA Connection Integrity NonStop NS-Series
LOCATION			
Group	1-89	1-89	110-125

Table 11 Tape Device Path Attributes *(continued)*

	SNDA Connection	PMF/IOMF Connection	FCSA Connection Integrity NonStop NS-Series
Module	1	1	2-3 (IOAM)
Slot	51-54	50 or 55	1-5 (FCSA)
SAC	1-4	3	1-2
DEVICEID	0-5 (S-pic) 4-5 (F-pic)	0-5	NA
PORTNAME	NA	NA	64-bit WWN
LUN	0	0	0-31

Fibre Channel Tape Connections

To configure tape drives that are connected to the system through an FCSA follow these rules:

- If a Fibre Channel tape drive is connected directly to an FCSA, you must specify the PORTNAME and LUN of the tape drive.
- You can use a 9800FC Fibre Channel converter to connect a Fibre Channel tape drive to a NonStop S-series server. You add the converter to the storage subsystem as an Open SCSI device. For details, see the *9800FC Fibre Channel Converter Installation and User's Guide*.
- If a SCSI tape drive is connected to an FCSA through an M8201 SCSI to Fibre Channel router, you must specify the PORTNAME and LUN associated with the M8201 port to which the tape drive is connected. For details, see the *M8201 Fibre Channel to SCSI Router Installation and User's Guide*.

Displaying Information

To display information about tape drives:

- "Displaying Configuration Information" (page 180)
- "Displaying Status Information" (page 182)

Displaying Configuration Information

The "INFO TAPE Command" (page 260) displays configuration information about a tape drive.

Example of an INFO TAPE Report

```
-> INFO $DLT23

STORAGE - Info TAPE configuration \COMM.$DLT23
*Density      *MaxOpens    *RecSize *Compression *Adapter   *DeviceID/Portname
                                                   Location
 6250         4             57344    ON            1,1,55     5
```

Explanation of Fields — INFO TAPE Report	
*	Indicates an attribute whose value you can change by using an ALTER TAPE command.
Density	The configured tape density in bits per inch (bpi) of a 5170 tape drive. This attribute applies only to 5170 tape drives.
MaxOpens	The maximum concurrent open files allowed for the tape drive.
RecSize	The configured record size (in bytes) for the tape drive.

Compression	For cartridge tape drives, shows whether compression is configured. This setting is ignored by 5194 tape drives because compression is always on. For other tape drives, this field is not applicable (N/A).
Adapter Location	The physical location (group, module, and slot) of the adapter where the tape drive is attached.)
DeviceID/PortName	For SCSI devices, the device ID that is configured for the device. This ID must match the SCSI ID that is physically set in the device. Normally, tape device IDs are either 4 or 5. For Fibre Channel devices, the worldwide port name (WWPN) that is configured for the path to the device. The number is a 16 character hexadecimal number, without a leading %H.

Examples of INFO TAPE Detailed Reports

This example shows an INFO DETAIL report for a tape drive connected to a PMF CRU:

```
-> INFO TAPE $DLT23, DETAIL

STORAGE - Detailed Info TAPE configuration \COMM.$DLT23
 Adapter Name......................... $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-55
*Adapter Location (Group,Module,Slot).. (1,1,55)
*BackupCpu........................... 0
*Compression......................... ON
*Density............................. 6250
*DeviceID/PortName.................... 5
*HighPin............................. ON
*LUN................................. 0
*MaxOpens............................ 4
*PrimaryCpu.......................... 1
*Program............................. $SYSTEM.SYSTEM.OTPPROCP
*RecSize............................. 57344
*SAC Name............................ $ZZSTO.#PMF.SAC-3.GRP-1.MOD-1.SLOT-55
*StartState.......................... STARTED
```

This example shows an INFO DETAIL report for a tape drive connected directly to an FCSA on an Integrity NonStop NS-series server:

```
-> INFO TAPE $LTO3, DETAIL

STORAGE - Detailed Info TAPE configuration \OSMQA3.$LTO3
 Adapter Name......................... $ZZSTO.#FCSA.GRP-110.MOD-3.SLOT-4
*Adapter Location (Group,Module,Slot).. (110,3,4)
*BackupCpu........................... 1
*Compression......................... ON
*Density............................. 6250
*DeviceID/PortName.................... 210000E08B1755DC
*HighPin............................. ON
*LUN................................. 0
*MaxOpens............................ 4
*PrimaryCpu.......................... 0
*Program............................. $SYSTEM.SYSTEM.OTPPROCP
*RecSize............................. 57344
*SAC Name............................ $ZZSTO.#FCSA.SAC-2.GRP-110.MOD-3.SLOT-4
*StartState.......................... STARTED
```

Explanation of Fields — INFO TAPE Report

*	Indicates an attribute whose value you can change by using an ALTER TAPE command.
AdapterName	The name of the adapter where the tape drive is attached.
Adapter Location (Group, Module, Slot)	The physical location of the adapter where the tape drive is attached.
BackupCpu	The processor number of the backup tape process.
Compression	For cartridge tape drives, shows whether compression is configured. This setting is ignored by a 5194 tape drive because compression is always on. For other tape drives, this field is not applicable (N/A).

Density	The configured tape density in bpi of a 5170 tape drive.
DeviceID/PortName	For SCSI devices, the device ID that is configured for the device. This ID must match the SCSI ID that is physically set in the device. Normally, tape device IDs are either 4 or 5. For Fibre Channel devices, the WWPN that is configured for the path to the device. The number is a 16 character hexadecimal number, without a leading %H.
HighPin	Shows whether the tape process is allowed to run at a high PIN (ON) or only a low PIN (OFF).
LUN	Always shows a value of zero (0).
MaxOpens	The maximum concurrent open files allowed.
PrimaryCpu	The processor number of the primary tape process.
Program	The object file name of the tape process.
RecSize	The configured record size (in bytes) for the tape drive.
SAC Name	The name of the SAC where the tape drive is attached. This value is determined by the SAC number and the name of the adapter.
StartState	Shows whether the tape process is available to other processes (in the STARTED state) or unavailable (in the STOPPED state) when the system is loaded.

Displaying Status Information

The "STATUS TAPE Command" (page 289) displays current status information about a tape drive.

Example of a STATUS TAPE Report

```
-> STATUS $DLT24

STORAGE - Status TAPE \COMM.$DLT24
LDev   State      Primary   Backup    DeviceStatus
                  PID       PID
 370   STARTED     1,283     0,285    ONLINE, BOT
```

Example of a Detailed STATUS TAPE Report

```
-> STATUS $DLT24, DETAIL

STORAGE - Detailed Status TAPE \COMM.$DLT24

Tape Process Information:
 LDev   State      Primary   Backup    DeviceStatus
                   PID       PID
  370   STARTED     1,283     0,285    ONLINE, BOT

Tape I/O Process Information:
  Library File.......
  Program File....... $SYSTEM.SYS01.OTPPROCP

Current Settings:
  ACL................ NOT INSTALLED      Buffer Level...... RECORD
  Checksum Mode...... NORMAL I/O        *Compression....... ON
 *Density........... 38000               Media Type........ 18-TRACKS
  Opens.............. 0                 *RecSize........... 57344
  Short Write Mode... ALLOWED, PADDED    SubType........... 9
  Volume Switching... TRANSPARENT
```

Explanation of Fields — Detailed STATUS TAPE Report	
LDev	The logical device number for the tape drive. This number is arbitrarily assigned to a device when you configure the device and every time the system is loaded.

State	The current object state of the tape drive. For a list of tape drive states, see "TAPE Object States" (page 39).
Primary PID	The processor number and PIN of current primary tape process.
Backup PID	The processor number and PIN of current backup tape process.
DeviceStatus	The current status of the device. Values can be:

NOT READY	The tape drive is not ready to be used.
ONLINE	The tape drive is online and away from beginning of tape.
ONLINE, BOT	The tape drive is online and at beginning of tape.

Tape I/O Process Information

Library File	Always blank.
Program File	The program file name of the tape process.

Current Settings:

*	Indicates an attribute whose value you can change by using an ALTER TAPE command.
ACL	Shows whether an automatic cartridge loader (ACL) is installed.
Buffer Level	The level of buffering provided by the tape process. Buffering can increase the performance of a write operation when no errors occur. Values can be:

UNSUPPORTED	The buffering level is unsupported or unknown.
RECORD	No buffering is performed on tape writes (every record written to the tape process is passed to the tape drive immediately).
FILE	File buffering is performed on tape writes (records are passed to the tape drive only when a file mark is written). Therefore, an application might have to rewrite all records written since the last file mark if an unrecoverable error occurs during a write to tape.
REEL	Reel buffering is performed on tape writes (even file marks are buffered). Therefore, that an application might have to rewrite the whole reel if an unrecoverable error occurs during a write.

Checksum Mode	Shows the current checksum method in use. Values can be:

CHECKSUM	Checksum processing is performed when a record is read (the checksum of the record is appended to the record after it is read). The tape process computes the checksum of the record and appends the checksum to the end of the record. On a successful read, an FESUMREADOK message is returned (instead of an FEOK).
NORMAL I/O	No special checksum processing is performed.

Compression	For cartridge tape drives, shows whether compression is configured. This setting is ignored by 5194 tape drives because compression is always on. For other tape drives, this field is not applicable (N/A).
Density	The configured tape density in bits per inch (bpi) of a 5170 tape drive.
Media Type	The type of media in the tape drive. The type of media can be UNKNOWN, 9-TRACK, 18-TRACK, or 36-TRACK.
Opens	The maximum concurrent opens to the tape process.
RecSize	The configured record size for the tape drive (in bytes).

Short Write Mode	Shows how the tape process handles write requests of less than 24 bytes. Values can be: ALLOWED, NONPADDED — Short writes are allowed (no data padding is performed). ALLOWED, PADDED — Short writes are allowed (but data is padded with zeros to a length of 24 bytes). NOT ALLOWED — Short writes are not allowed. UNSUPPORTED — Short writes are not supported for this tape drive.
SubType	Shows the device subtype of the tape drive, which varies by tape-drive model and function. Subtypes are described under FILE_GETINFOLIST_ Procedure in the *Guardian Procedure Calls Reference Manual*.
Volume Switching	Shows how the tape process handles end-of-tape (EOT) messages. Values can be: EOT — The user of the volume is notified of a tape switch by file-system error 150 (EOT). TRANSPARENT — Volume switching is transparent. The user is not notified when a volume ends.

Example of a STATUS TAPE, ENCRYPTION Report

- This example shows encrypted media in an encrypted drive:

```
STORAGE - Status TAPE $TAPE07, ENCRYPTION, DETAIL
Media
  KeyName.......... VGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG_BBBBBBBB_YYMMDDHHMM
  KeyAlgorithm..... GCM-AES
  KeySize.......... 256
  KeyAccess........ OK

Drive
  MasterKeyName.... VGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
  KeyAlgorithm..... GCM-AES
  KeySize.......... 256
  KeyGenPolicy..... KeyPerTape
  KeyAccess........ OK
```

- This example shows an encrypted tape drive without media:

```
STORAGE - Status TAPE $TAPE09, ENCRYPTION, DETAIL
Media
  Media not present or encryption status unknown

Drive
  MasterKeyName.... VGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
  KeyName.......... VGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG_YYYYMMDDHHSS
  KeyAlgorithm..... GCM-AES
  KeySize.......... 256
  KeyGenPolicy..... KeyPerDrive
  KeyAccess........ OK
```

- This example shows unencrypted media in an non-encrypted drive:

```
STORAGE - Status TAPE $TAPE08, ENCRYPTION, DETAIL
Media
  Not encrypted

Drive
  Not encrypted
```

The display would be the same for non-encrypted tape media in encrypted, but disabled, drive.

- This example shows a non-CLIM tape drive:

```
STORAGE - Status TAPE $TAPE11, ENCRYPTION
Drive
  Not encrypted (non-CLIM)
```

Explanation of Fields — STATUS TAPE, ENCRYPTION Report

Media	If no media is present or encryption status is unknown, field will display "Not present or encryption status unknown". If media is unencrypted, field will display "Not encrypted".
KeyName	The name of the encryption key in use for the media.
KeyAlgorithm	The key algorithm in use for the media.
KeySize	The key size.
KeyAccess	The current status of access between the CLIM and the ESKM (Enterprise Storage Key Manager. Display with be "OK" or an error message.
Drive	If drive is unencrypted, field will display "Not encrypted".
MasterKeyName	The name of the encryption key in use for the drive.
KeyAlgorithm	The key algorithm in use for the drive.
KeySize	The key size.
KeyGenPolicy	The key generation policy. Values can be: KEYPERTAPE — Generate a key for each tape. KEYPERDRIVE — Generate one key for all tapes written by this drive. NOENCRYPTION — Do not encrypt tapes.
KeyAccess	The current status of access between the CLIM and the ESKM (Enterprise Storage Key Manager. Display with be "OK" or an error message.

Configuring a Tape Drive

Configuration tasks for tape drives include:

- "Adding a Tape Drive" (page 185)
- "Altering Tape Drive Attribute Values" (page 187)
- "Deleting a Tape Drive" (page 187)

Adding a Tape Drive

The "ADD TAPE Command" (page 225) adds a tape drive to the system configuration database.

Considerations for ADD TAPE

- Before adding a SCSI tape drive, verify the SCSI interface is properly terminated.
- If tape performance is a concern, do not connect more than one tape drive to the same adapter.
- Certain attributes such as compression and density are applicable only for specific types of tape drives. Consult the manual for the tape drive for information about these attributes. Other devices ignore these attribute settings.

- The tape IOP overrides any value provided in the ADD TAPE command that is not valid for that specific tape device.
- For tape drives connected to the system through an FCSA, see "Fibre Channel Tape Connections" (page 180)

Adding One Tape Drive

1. Physically install the tape drive and note the group, module, and slot of its adapter.
2. Add the tape drive to the system, specifying the group, module, and slot:

   ```
   -> ADD TAPE $TAPE0, SENDTO STORAGE, LOCATION (1,1,55)
   ```

3. Verify the configuration:

   ```
   -> INFO $TAPE0, DETAIL
   ```

Example: Adding a Similar Tape Drive Using the LIKE Attribute

To add another tape drive similar to an existing tape drive on the same system, you can using the LIKE attribute:

```
-> ADD TAPE $TAPE1, SENDTO STORAGE, LIKE $TAPE0, &
-> LOCATION (1,1,54)
```

Example: Adding a Similar Tape Drive Using the OBEYFORM Attribute

To configure the same or a similar tape drive, you can create a command file by using the OBEYFORM attribute of the INFO TAPE command. You can copy this file to another system or add it to different configuration file on the current system.

1. Capture the configuration for an existing tape drive:

   ```
   -> INFO / OUT LOG / $TAPE0, OBEYFORM

   == STORAGE - Detailed Info TAPE in obeyform: \COMM.$TAPE0
   ADD TAPE $TAPE0 , &
       SENDTO STORAGE , &
       BACKUPCPU 3 , &
       COMPRESSION ON , &
       DENSITY 6250 , &
       DEVICEID 5 , &
       HIGHPIN ON , &
       LOCATION (2,1,55) , &
       MAXOPENS 4 , &
       PRIMARYCPU 2 , &
       PROGRAM $SYSTEM.SYSTEM.OTPPROCP , &
       RECSIZE 57344 , &
       SAC 3 , &
       STARTSTATE STARTED
   ```

2. Optionally edit the resulting log file to specify:
   - A unique device name
   - Based on physical location, change at least one of:
     - LOCATION slot number
     - SAC number
     - SCSIID value
3. Enter the log file contents either by copying and pasting into an SCF command line or by using the log file as a command file.

Adding a Fibre Channel Tape Drive

To add a Fibre Channel tape device that is directly connected to an FCSA, you specify the PORTNAME (WWN) and LUN of the tape device. This command adds a Fibre Channel tape device to the system:

```
-> ADD TAPE $LTO3, SENDTO STORAGE, LOCATION (110,3,4) &
-> SAC 2, PORTNAME 210000E08B1755DC, LUN 0
```

Altering Tape Drive Attribute Values

The "ALTER TAPE Command" (page 241) alters the attributes of a tape drive in the system configuration database. For a description of alterable tape attributes, see "ALTER TAPE Attributes" (page 242).

Considerations for ALTER TAPE

- The process must be in the STOPPED state or not running before you can change its configuration.
- Changes take effect when you restart the process.
- Changing the Values of Tape Drive Attributes

1. Stop the tape drive:

   ```
   -> STOP $TAPE0
   ```

2. Verify the tape drive is in the STOPPED state:

   ```
   -> STATUS $TAPE0
   ```

3. Change one or more attributes. For example:

   ```
   -> ALTER $TAPE0, COMPRESSION OFF
   ```

4. Verify the change took place:

   ```
   -> INFO $TAPE0, DETAIL
   ```

5. Start the tape drive:

   ```
   -> START $TAPE0
   ```

   When the START TAPE command finishes successfully, the tape drive is left in the STARTED state.

Deleting a Tape Drive

The "DELETE TAPE Command" (page 252) removes a tape drive from the system configuration database.

Considerations for DELETE TAPE

- Stop the tape drive before deleting it.
- If the DELETE TAPE command does not work, try using the RESET TAPE, FORCED command and then issue the DELETE command.
- Example

1. Verify the tape drive is in the STOPPED state:

   ```
   -> STATUS $TAPE0
   ```

2. Remove the tape drive from the system configuration database:

   ```
   -> DELETE $TAPE0
   ```

3. Verify the tape drive is removed from the system configuration database:

   ```
   -> INFO $TAPE0
   ```

Managing a Tape Drive

Management tasks for tape drives include:

- "Enabling or Disabling Labeled-Tape Processing" (page 188)
- "Resetting a Tape Drive" (page 188)
- "Starting a Tape Drive" (page 189)
- "Stopping a Tape Drive" (page 189)

For information about I/O commands for managing tape drives, see the *Guardian Procedure Calls Reference Manual*.

Enabling or Disabling Labeled-Tape Processing

The "ALTER SUBSYS Attributes" (page 240) of the ALTER command toggles labeled-tape processing. Although this labeled-tape processing setting overrides the CONFTEXT entry, it is in turn overridden by the BLPCHECK or NLCHECK options of the MEDIACOM ALTER TAPEDRIVE command (described in the *DSM/Tape Catalog Operator Interface (MEDIACOM) Manual*).

To enable labeled-tape processing:

1. Stop all available tape drives on the system:

   ```
   -> STOP TAPE $*
   ```

2. Enable labeled-tape processing for the system:

   ```
   -> ALTER $ZZSTO, LABELTAPE ON
   ```

3. Restart all available tape drives:

   ```
   -> START TAPE $*
   ```

4. Exit SCF, then start the $ZSVR tape server process:

   ```
   -> EXIT
   > ZSERVER /NAME $ZSVR, NOWAIT, CPU primary-cpu / backup-cpu
   ```

To turn off labeled-tape processing:

1. Stop the $ZSVR tape server process:

   ```
   -> STOP $ZSVR
   ```

2. Start SCF, then stop all available tape drives on the system:

   ```
   > SCF
   -> STOP TAPE $*
   ```

3. Disable labeled-tape processing for the system:

   ```
   -> ALTER $ZZSTO, LABELTAPE OFF
   ```

4. Restart all available tape drives:

   ```
   -> START TAPE $*
   ```

Resetting a Tape Drive

The "RESET TAPE Command" (page 273) puts a tape drive into the STOPPED state, substate DOWN, reading for restarting:

1. Check the current status of the tape drive:

   ```
   -> STATUS $S11500
   ```

2. If it is not in the STOPPED state, substate DOWN:

   ```
   -> RESET $TAPE0
   ```

3. Start the tape drive:

   ```
   -> START $TAPE0
   ```

4. Verify the process is started:

```
-> STATUS $TAPE0
```

Starting a Tape Drive

The “START TAPE Command” (page 276) makes a tape drive accessible to user requests.

Considerations for START TAPE

- To start a tape drive, the tape drive must be connected to the external port of the adapter and the tape drive must be powered on.
- If the tape process does not start, reset it with the FORCED option; then start it again.

Starting One Tape Drive

1. Start the tape drive:

```
-> START $TAPE0
```

2. Verify the tape drive is started:

```
-> STATUS $TAPE0
```

Stopping a Tape Drive

The “STOP TAPE Command” (page 292) makes a tape drive inaccessible to user requests:

1. Stop the tape drive:

```
-> STOP $TAPE0
```

2. Verify the tape drive is in the STOPPED state:

```
-> STATUS $TAPE0
```

Managing Encrypted Tape Drives

SCF supports encryption of data-at-rest for LTO-4 and LTO-5 tape objects that are connected with CLIMs. Encryption uses keys generated and stored by the HP Enterprise Secure Key Manager (ESKM). Customers use the ALTER and STATUS commands to manage encryption on tapes.

Only members of the SAFEGUARD encryption officer group on the local system can perform ALTER TAPE with the NEWENCRYPTKEY or KEYGENPOLICY attributes. Encryption officers must also be members of the Guardian SUPER.* group. Keys and system security should be managed by customer security officers, not system administrators.

For details about encryption, see the *NonStop Volume Level Encryption Guide*.

14 Storage Subsystem Commands

This chapter provides the syntax and description of SCF commands.

Overview of Storage Subsystem Commands

Table 12 Overview of Storage Subsystem Commands

Command	Page	Description
ABORT	192	Terminates the operation of an object without regard to the current state of its operation.
ADD	194	Defines an object to the subsystem.
ALLOWOPENS	227	Allows an object to once again accept opens; reverses the effect of the STOPOPENS command.
ALTER	228	Changes the values of one or more attributes of an object.
CONTROL	244	Allows you to: Calculate a checksum Power a disk on or off Rebuild the free-space table Refresh cache pages Replace the bootstrap program Manually spare a sector
DELETE	248	Removes an object from the subsystem.
INFO	253	Displays the configured attributes of an object.
INITIALIZE	260	Prepares a physical disk for use.
NAMES	261	Displays the names described by an object-name template.
PRIMARY	263	Causes the backup processor to become the primary processor and the primary processor to become the backup processor.
PROBE	268	Tests the Fibre Channel connection to a SAC on an FCSA.
RENAME	269	Changes the name of an object.
REPLACE	269	Downloads new firmware to adapters and SACs.
RESET	270	Moves an object to a state from which it can be started.
START	273	Initiates the operation of an object previously defined to a subsystem.
STATS	277	Displays the accumulated statistics for an object and optionally resets them.
STATUS	279	Displays the status of an object.
STOP	290	Terminates the operation of an object in an orderly manner.
STOPOPENS	293	Prevents any additional opens to an object.
SWITCH	294	Designates the active a path to a device
VERSION	297	Displays the version level of the current subsystem.

SCF Commands and Object Types

Table 13 Object Types for Storage Subsystem Commands

Command	ADAPTER	CLIM	DISK	MON	POOL	PROFILE	SAC	SCSI	SUBSYS	TAPE
ABORT	—	—	X	X	X	—	—	—	—	—
ADD	—	—	X	X	X	X	—	X	—	X

Table 13 Object Types for Storage Subsystem Commands *(continued)*

Command	ADAPTER	CLIM	DISK	MON	POOL	PROFILE	SAC	SCSI	SUBSYS	TAPE
ALLOWOPENS	—		X	—	—	—	—	—	—	—
ALTER	—		X	X	X	X	—	X	X	X
CONTROL	—		X	—	—	—	X	—	—	—
DELETE	X		X	X	X	X	—	X	—	X
EJECT	—		X	—	—	—	—	—	—	—
INFO	X	X	X	X	X	X	—	X	X	X
INITIALIZE	—		X	—	—	—	—	—	—	—
LABEL	—		X	—	—	—	—	—	—	—
NAMES	X		X	X	X	X	—	X	X	X
PROBE							X			
PRIMARY	—		X	X	X	—	—	X	X	X
RENAME	—		X	—	—	—	—	—	—	—
REPLACE	X		X	—	—	—	X	—	—	—
RESET	—		X	X	X	—	—	X	—	X
START	—		X	X	X	—	—	X	—	X
STATS	—		X	—	—	—	X	—	—	—
STATUS	X	X	X	X	X	—	X	X	X	X
STOP	—		X	X	X	—	—	X	—	X
STOPOPENS	—		X	—	—	—	—	—	—	—
SWITCH	X	X	X	—	—	—	—	X	—	—
VERSION	—	—	—	—	—	—	—	—	X	—

Commands That Behave Differently When Used in a Command File

Some commands that display prompts for user action when used in interactive mode (at the command prompt) behave differently when used in noninteractive mode (in a command file). As a result, these commands cannot be used in a command file except where noted

Command	Special Notes
ALTER DISK, CACHE	
ALTER DISK, LABEL	
CONTROL DISK, REPLACEBOOT	
RESET DISK, FORCED	
RESET TAPE, FORCED	
START DISK	Can be used in a command file, but fails if the disk labels are inconsistent or if the disk will rename itself.

Command	Special Notes
STATS DISK, RESET	If you use the FORCED option, the command behaves the same way in a command file as interactively.
STOP DISK	Can be used in a command file, but fails if you attempt to stop the last path on an audited volume or if the volume has open files. If you use the FORCED option, the command behaves the same way in a command file as interactively.

Command Timeouts

If a device error occurs during execution of a command and the error delays completion of the command beyond the default SCF timeout value, you can change that timeout value. For example, you can change the timeout value from the default of 90 seconds to 120 seconds:

```
-> TIMEOUT 120
```

The TIMEOUT command is described in the *SCF Reference Manual for G-Series RVUs*.

Sensitive Commands

Only these users can issue a sensitive command:

- A super-group user (255,*n*)
- The owner of the subsystem
- A member of the group of the owner of the subsystem

  The owner of a subsystem is the user who started that subsystem.

Sensitive commands for the storage subsystem are:

ABORT	DELETE	PROBE	STOP
ADD	EJECT	RENAME	STOPOPENS
ALLOWOPENS	INITIALIZE	REPLACE	SWITCH
ALTER	LABEL	RESET	
CONTROL	PRIMARY	START	

Each sensitive command generates an EMS message that reports the command, the time it was executed, the terminal from which the command was entered, and the group and user names of the user issuing the command.

Nonsensitive Commands

Nonsensitive commands are:

INFO	PROBE	STATUS
NAMES	STATS	VERSION

ABORT Command

The ABORT command stops access to an object as quickly as possible.

Supported objects are:

- "ABORT DISK Command" (page 193)
- "ABORT MON Command" (page 193)
- "ABORT POOL Command" (page 194)

ABORT is a sensitive command.

ABORT DISK Command

The ABORT DISK command puts a disk into the STOPPED state, substate HARDDOWN, but leaves the disk process running. The syntax is:

```
ABORT  [ / OUT file-spec / ] DISK $disk[-P |-B |-M |-MB ]
    [ , FORCED ]  [ , POOL $pool ]  [ , SEL state ]
    [ , SUB { ALL | MAGNETIC | VIRTUAL } ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

DISK $*disk*[-P |-B |-M |-MB]

is the disk name and the path (primary, backup, mirror, or mirror backup). A single path specification is required for physical disks. There is no path specification for virtual disks.

FORCED

specifies that the command be executed without any interaction with the user, even if the command stops the last path to the device or files are open on the device.

If you use this attribute, you must first stop all processes that use the disk to store object code (programs) or swap files. Otherwise, a %5113 halt could occur.

POOL $*pool*

specifies that the command is performed only on disks associated with the specified storage pool.

SEL *state*

specifies that the command affects only devices in the specified state.

SUB { ALL | MAGNETIC | VIRTUAL }

specifies that the command affects only disks of the specified type. The default is ALL.

See "Attribute Descriptions for Disk Commands" (page 198) for descriptions of all attributes for disk commands.

ABORT DISK Examples

- To immediately stop the mirror path of $DATA00:

  ```
  -> ABORT $DATA00-M
  ```

- To immediately stop the mirror paths of $DATA00 and $DATA01, even if files are open or if all other paths to the disk are down:

  ```
  -> ABORT ($DATA00-M, $DATA01-M), FORCED
  ```

  See "Stopping a Disk With the ABORT DISK Command" (page 102).

ABORT MON Command

The ABORT MON command stops access to the SMF master process (same as the "STOP MON Command" (page 291)). The syntax is:

```
ABORT [ / OUT file-spec / ] MON $ZSMS
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

MON $ZSMS

is the SMF master process.

ABORT MON Consideration

Use the "RESET Command" (page 270) to prepare the process for restarting.

ABORT MON Example

```
-> ABORT MON $ZSMS
```

ABORT POOL Command

The ABORT POOL command makes a storage pool inaccessible to user requests (same as "STOP POOL Command" (page 291)). The syntax is:

```
ABORT [ / OUT file-spec / ] POOL $pool
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

POOL $*pool*
: is the storage pool process.

ABORT POOL Consideration

Use the "RESET Command" (page 270) to prepare the process for restarting.

ABORT POOL Example

```
-> ABORT POOL $ZSMS
```

ADD Command

The ADD command defines an object in the storage subsystem and adds the object to the system configuration database.

Supported objects are:

- "ADD DISK Command" (page 194)
- "ADD MON Command" (page 215)
- "ADD PARTITION Command" (page 216)
- "ADD POOL Command" (page 218)
- "ADD PROFILE Command" (page 221)
- "ADD SCSI Command" (page 222)
- "ADD TAPE Command" (page 225)

ADD is a sensitive command.

ADD DISK Command

The ADD DISK command adds a physical or virtual disk to the system configuration. The syntax is:

```
ADD [ / OUT file-spec / ] DISK $disk , SENDTO STORAGE
[ , LIKE object ]  [ , attribute-spec ]...
```

Wild-card characters are not supported for this command.

OUT *file-spec*
: directs all SCF output to the specified file.

DISK $*disk*

is the name of the disk.

SENDTO STORAGE

directs the command to the storage subsystem. This attribute is required unless you specified SENDTO STORAGE in a previous ASSUME command.

LIKE *object*

identifies an existing object whose attribute values are copied to the object affected by this command. The object name must currently exist in the system configuration, and the object type must match that of the object affected by this command. If you specify LIKE, any attribute values that you explicitly specify override the attribute values of the LIKE object.

The location, SAC, and device ID values are not copied from the LIKE object. To assign values to these attributes, you must explicitly specify them.

attribute-spec

is one or more of:

- "Disk Attributes for the ADD Command" (page 195)
- "Virtual Disk Attributes for the ADD Command" (page 196)

Disk Attributes for the ADD Command

These attributes are valid for the ADD DISK command. For a description of these attributes, see "Attribute Descriptions for Disk Commands" (page 198)

```
[ , AUDITTRAILBUFFER number ]
[ , AUTOREVIVE { ON | OFF } ]
[ , AUTOSELECT { ON | OFF } ]
[ , AUTOSTART { ON | OFF } ]
[ , BACKUPCLIM clim-name
[ , BACKUPCPU number ]
[ , BACKUPDEVICEID { number | ( shelf, bay ) }]
[ , BACKUPLOCATION {group,module,slot) ]
[ , BACKUPPORTNAME number ]
[ , BACKUPSAC { number | name } ]
[ , CAPACITYMISMATCH { ON | OFF } ]
[ , CBPOOLLEN number ]
[ , FASTBULKWRITE { ON | OFF } ]
[ , FORCED ]
[ , FSTCACHING { ON | OFF | ENABLED} ]
[ , FULLCHECKPOINTS { DISABLED | ENABLED | FORCED } ]
[ , HALTONERROR number ]
[ , HIGHPIN { ON | OFF } ]
[ , LKIDLONGPOOLLEN number ]
[ , LKTABLESPACELEN number ]
[ , MAXLOCKSPEROCB number ]
[ , MAXLOCKSPERTCB number ]
[ , MBACKUPCLIM clim-name
[ , MBACKUPDEVICEID { number | ( shelf, bay ) } ]
[ , MBACKUPLOCATION {group,module,slot) ]
[ , MBACKUPPORTNAME number ]
[ , MBACKUPSAC { number | name } ]
[ , MIRRORCLIM clim-name
[ , MIRRORDEVICEID { number | ( shelf,bay ) } ]
[ , MIRRORLOCATION ( group,module,slot ) ]
[ , MIRRORLUN number ]
[ , MIRRORPARTITION mirror-part-num
[ , MIRRORPORTNAME number ]
[ , MIRRORSAC { number | name } ]
[ , NONAUDITEDINSERT { ON | OFF } ]
[ , NUMDISKPROCESSES number ]
[ , OSSCACHING { ON | OFF } ]
```

```
[ , PHYSVOLSELECT { ON | OFF } ]
[ , POOL { $pool | EXCLUDE } ]
[ , PRIMARYCLIM clim-name
[ , PRIMARYCPU number ]
[ , PRIMARYDEVICEID { number | ( shelf,bay ) } ]
[ , PRIMARYLOCATION ( group,module,slot )
[ , PRIMARYLUN number }
[ , PRIMARYPARTITION primary-part-num
[ , PRIMARYPORTNAME number ]
[ , PRIMARYSAC { number | name } ]
[ , PROGRAM [[$vol.]subvol.]fileid ]
[ , PROTECTDIRECTORY { CHECKPOINT | OFF | SERIAL } ]
[ , RECOVERYTIMEOUT number
[ , REVIVEBLOCKS number ]
[ , REVIVEINTERVAL number ]
[ , REVIVEPRIORITY number ]
[ , REVIVERATE number ]
[ , SENDTO STORAGE ]
[ , SERIALWRITES { ENABLED | DISABLED } ]
[ , SQLMXBUFFER number ]
[ , STARTSTATE { STARTED | STOPPED } ]
[ , TYPE { MAGNETIC | VIRTUAL } ]
[ , WRITECACHE { DISABLED | ENABLED } ]
```

Virtual Disk Attributes for the ADD Command

These virtual disk attributes are valid for the ADD DISK command. For a complete description of these attributes, see "Attribute Descriptions for Disk Commands" (page 198).

```
[ , ANTCAPACITY number ]
  , ANTLOCATION { $vol | $vol.subvol.fileid }
[ , BACKUPCPU number ]
[ , CACHESIZE number ]
[ , HIGHPIN { ON | OFF } ]
[ , MODE { NOISY | QUIET } ]
  , PENDOPSLOCATION { $vol | $vol.subvol.fileid}
  , POOL $pool
[ , PRIMARYCPU number ]
[ , PROGRAM [[$vol.]subvol.]fileid ]
[ , STARTSTATE { STARTED | STOPPED } ]
  , TYPE { VIRTUAL }
```

ADD DISK Examples for Physical Disks

See the procedure and considerations for "Adding a Disk" (page 82).

- To add a new nonmirrored disk named $DATA04 in group 01, slot 07:

  ```
  -> ADD DISK $DATA04, SENDTO STORAGE, PRIMARYLOCATION (1,1,7)
  ```

- To add two disks as a mirrored volume named $DATA04 in group 01, slots 7 and 8:

  ```
  -> ADD DISK $DATA04, SENDTO STORAGE, PRIMARYLOCATION(1,1,7),&
  -> MIRRORLOCATION (1,1,8)
  ```

- To add two 45*xx* disks as a mirrored volume named $DISK02:

  ```
  -> ADD DISK $DISK02, SENDTO STORAGE,&
  -> PRIMARYLOCATION (1,1,53), PRIMARYSAC 1, &
  -> PRIMARYDEVICEID 0, BACKUPLOCATION (1,1,54), BACKUPSAC 1, &
  -> MIRRORLOCATION (1,1,54), MIRRORSAC 2, &
  -> MIRRORDEVICEID 0, MBACKUPLOCATION (1,1,53), MBACKUPSAC 2
  ```

- To add a 45*xx* nonmirrored disk named $DISK00 using SAC 1 of the adapter in slot 53 in group 01:

  ```
  -> ADD DISK $DISK00, SENDTO STORAGE, &
  -> PRIMARYLOCATION (1,1,53), PRIMARYSAC 1, PRIMARYDEVICEID 0
  ```

- To add a disk volume that uses a Fibre Channel ServerNet adapter to communicate with an Enterprise Storage System:

```
-> ADD DISK $DISK02, SENDTO STORAGE,&
-> PRIMARYCPU 01 &
-> BACKUPCPU 02 &
-> PRIMARYLOCATION (11,2,5), &
-> PRIMARYSAC 1, &
-> BACKUPLOCATION (11,3,5), &
-> BACKUPSAC 1, &
-> MIRRORLOCATION (11,3,5), &
-> MIRRORSAC 2, &
-> MBACKUPLOCATION (11,2,5), &
-> MBACKUPSAC 2 &
-> PRIMARYPORTNAME 50060E8003501213, &
-> BACKUPPORTNAME 50060E8003501225 &
-> PRIMARYLUN 16 &
-> MIRRORPORTNAME 50060E8003501241, &
-> MBACKUPPORTNAME 50060E8003501243, &
-> MIRRORLUN 17
```

- This example creates an obey file, from which you can use the ADD DISK command to replicate the attributes from one disk to another disk. In this particular example, partitions are added and WRITECACHE is enabled. See "Partitioning HDDs and SSDs" (page 92) and "Write Caching" (page 94) for a description of these features.

```
-> info disk $DISK121, obey
== STORAGE - Obeyform Info Magnetic DISK \JUNO1.$DISK121
ADD DISK $DISK121 , &
    SENDTO STORAGE , &
    BACKUPCPU 3, &
    HIGHPIN ON , &
    PRIMARYCPU 2, &
    PROGRAM $SYSTEM.SYSTEM.TSYSDP2 , &
    STARTSTATE STARTED, &
    PRIMARYCLIM C100271 , &
    PRIMARYLUN 121, &
    PRIMARYPARTITION 1, &
    BACKUPCLIM C100273 , &
    BACKUPLUN 121, &
    MIRRORCLIM C100273 , &
    MIRRORLUN 216, &
    MIRRORPARTITION 1, &
    MBACKUPCLIM C100271 , &
    MBACKUPLUN 216, &
    AUDITTRAILBUFFER 0 , &
    AUTOREVIVE OFF, &
    AUTOSTART ON, &
    CAPACITYMISMATCH OFF , &
    CBPOOLLEN 1000 , &
    FASTBULKWRITE OFF , &
    FSTCACHING OFF , &
    FULLCHECKPOINTS ENABLED , &
    HALTONERROR 1, &
    LKIDLONGPOOLLEN 8 , &
    LKTABLESPACELEN 15 , &
    MAXLOCKSPEROCB 5000 , &
    MAXLOCKSPERTCB 5000 , &
    NONAUDITEDINSERT OFF , &
    NUMDISKPROCESSES 8, &
    OSSCACHING ON , &
    PROTECTDIRECTORY SERIAL , &
    RECOVERYTIMEOUT 0 , &
    REVIVEBLOCKS 10 , &
```

```
REVIVEINTERVAL 100 , &
REVIVEPRIORITY 0 , &
REVIVERATE 0 , &
SERIALWRITES ENABLED, &
WRITECACHE ENABLED
```

ADD DISK Example for Virtual Disks

See the procedure and considerations for “Adding a Virtual Disk” (page 147).

To add the virtual disk $VDISK00 to the storage pool $POOL01:

```
-> ADD DISK $VDISK00, SENDTO STORAGE, ANTLOCATION $DATA00, &
-> POOL $POOL01, PENDOPSLOCATION $DATA00, TYPE VIRTUAL
```

Attribute Descriptions for Disk Commands

This list describes all the attributes that can be used to define physical disks, virtual disks, and disk profiles. Attributes that are not valid for certain commands, the PROFILE object, or virtual disks are indicated accordingly.

ALTNAME $*vol*

is an alternate volume name for the disk.

This attribute changes the alternate volume name on the label of the disk but does not change the system configuration database.

If the storage subsystem tries to start a volume and the default volume name is already in use, the volume is started using the alternate volume name. If the alternate volume name is also in use, the storage subsystem uses the volume name in the system configuration database and the disk is left in the DOWN state.

To use this attribute, see “Naming a Disk” (page 88)

NOTE: This attribute is not valid for the ADD DISK command and the PROFILE object.

ANTCAPACITY *number*

is the maximum number of entries in the virtual-disk, audited-name table.

NOTE: This attribute is only valid for virtual disks.

number is in the range 0 through 2500000. The default is 12288.

The value of this attribute must be large enough to hold not only the temporary and permanent files to be created on the virtual disk but also the normal overhead associated with key-sequenced files, as documented in the *Enscribe Programmer’s Guide*.

ANTLOCATION { $*vol* | $*vol.subvol.fileid* }

is the disk location of the audited-name table for the virtual disk

NOTE: This attribute is only valid for virtual disks and cannot be used with the ALTER DISK command.

This attribute is required for the ADD command and has no default (not even when you use the LIKE attribute).

If you use the *$vol.subvol.fileid* format, *fileid* must exist and must reside in a subvolume whose name begins with ZYS.

If you use the *$vol* format, $*vol* must be in the STARTED state, enabled in TMF, and configured to be in a storage pool. The virtual disk determines the subvolume name and file ID for the audited-name table it creates.

AUDITTRAILBUFFER *number*

(TMF audit-trail volumes only) specifies the number of megabytes to be allocated for the audit-trail buffer of an audit volume. This value improves the performance of systems that use the Remote Duplicate Database Facility. *number* is in the range 0 through 768.

If you do not configure a value for AUDITTRAILBUFFER, or if the value you specify is 0, DP2 uses a value of 1. However, if you do not configure a value or if you specify 0, the value is displayed as 0, not 1.

If the value you specify is greater than 768, DP2 uses a value of 128.

NOTE: From J06.17 RVU onwards, if the value specified for AUDITTRAILBUFFER is:

- Between 0 (zero) and 64, then DP2 uses the value 64.
- Between 64 and 768, then DP2 uses the specified value.
- Greater than 768, then DP2 uses the value 768.

Otherwise, DP2 uses the value you specified.

Memory for the audit-trail buffer is locked while the audit volume is active.

Before changing an audit volume to a data volume that will be used for SQL/MX, set SQLMXBUFFER to the value desired for that attribute, or zero. Conversely, before changing a data volume used for SQL/MX objects to an audit volume, set AUDITTRAILBUFFER to the desired value. If that value is zero, AUDITTRAILBUFFER will be configured to 1 MB when the disk is started as a TMF AuditTrail.

AUTOLABEL { ON | OFF }

specifies whether the disk should be automatically labeled when it is inserted into its slot. The default is OFF.

NOTE: This attribute is only valid for the PROFILE object.

This attribute is used only when you insert a nonlabeled disk into a slot where no disk is currently configured.

If a disk was already configured in the slot, the inserted disk is not automatically labeled. It can eventually receive a label in these ways:

- The slot is half of a mirrored pair, the other half has a label, the other half is up, AUTOSTART is ON, and AUTOREVIVE is ON. the label is copied from the other half during the automatic revive operation.
- The slot is half of a mirrored pair and the other half has a label, but one of the other conditions in case 1 is not true. If you start the disk and respond “yes” to the resulting question or questions, the resulting revive operation copies the label from the other half.
- You use the INITIALIZE DISK, LABEL command on the disk.

For information about using this attribute, see “Considerations for LABEL and Disks” (page 91).

AUTOREVIVE { ON | OFF }

(mirrored disks only) specifies whether to automatically start a revive operation on a mirrored volume either when either a new disk is inserted or when the system is loaded and a mirrored volume is not synchronized

`ON`	Automatically starts a revive operation using the values for the REVIVEPRIORITY and REVIVERATE attributes.
`OFF`	Does not automatically start a revive operation (default).

For information about using this attribute, see “Configuring Internal Mirrored Disks to Revive Automatically” (page 78).

AUTOSELECT { ON | OFF }

specifies whether a virtual disk process is allowed to automatically consider this physical volume when making file-placement decisions. This attribute is valid only when this physical volume is a member of a storage pool

`ON`	Enable automatic selection (default).
`OFF`	Disable automatic selection.

AUTOSTART { ON | OFF }

(internal and M8*xxx* disks only) specifies whether to automatically start the disk process when the disk is inserted.

`ON`	Enable automatic starting (default).
`OFF`	Disable automatic starting.

DISK AUTOSTART ON is ignored if SUBSYS AUTOSTART is OFF.

For information about using this attribute, see “Mirrored Disk Placement” (page 77).

BACKUPCLIM *clim-name*

is the name of the CLIM that controls the backup path to the disk.

NOTE: This attribute is not valid for the PROFILE object.

BACKUPCPU *number*

is the processor in which the backup IOP starts

NOTE: This attribute is not valid for the PROFILE object.

The process must be in the STOPPED state or not running for you to alter this attribute.

BACKUPDEVICEID { *number* | (*shelf, bay*) }

(45*xx* and M8*xxx* Fibre Channel disks only) is the device ID of the disk accessed on the backup path to the disk

NOTE: This attribute is not valid for the PROFILE object.

This attribute is optional. If specified, it must be the same device ID as that specified for PRIMARYDEVICEID.

For 45*xx* disks:

number	is the unit number of the disk. *number* is in the range 1 through 7.

For M8*xxx* Fibre Channel disks:

shelf	is the Fibre Channel disk module (FCDM) shelf number. *shelf* is in the range 1 through 4.
bay	is the number of the disk. *bay* is in the range 1 through 14.

If you configure a disk volume for use with an FCSA connected to an ESS, you cannot specify BACKUPDEVICEID.

BACKUPLOCATION (*group, module, slot*)

(45*xx* and M8*xxx* disks only) is the location of the adapter that controls the backup path to the device

NOTE: This attribute is not valid for the PROFILE object.

This attribute is optional if BACKUPSAC is given in *name* format but required if BACKUPSAC is given in *number* format.

BACKUPPORTNAME *number*

specifies the Fibre Channel port name (WWN) used by the backup path to a disk volume on the Enterprise Storage System (ESS)

NOTE: This attribute is not valid for the PROFILE object.

The ESS administrator must give you the WWN so that you can specify it in this attribute.

Enter *number* as a 16 character hexadecimal number, without a leading %H.

BACKUPSAC { *number* | *name* }

is the SAC that controls the backup path to the device

NOTE: This attribute is not valid for the PROFILE object.

number	is the SAC subdevice number on the adapter. If you specify BACKUPSAC in *number* format, you must also specify BACKUPLOCATION.
name	is the full name of the SAC location. For example:`SNDA.SAC-1.GRP-1.MOD-1.SLOT-53FCSA.SAC-1.GRP-11.MOD-2.SLOT-5`Use this form to override the default selection of which adapter gets the -P path and which gets the -B path.An FCSA must be located in either MOD- 2 or MOD-3 of an IOAM enclosure. There are five slots in each IOAM module

CACHE { (*block-size*, *num-blocks*) | ((*block-size*, *num-blocks*), (*block-size*, *num-blocks*)) }

specifies the disk cache configuration for an in-use volume.

NOTE: This attribute is not valid for the ADD DISK command and the PROFILE object.

Disk cache configuration is the number and size of blocks read from disk and stored in the processor for use as virtual memory. If you do not specify a value, SCF uses values set by the disk process, which might cause performance problems

block-size	is a sector size in bytes or kilobytes: 512 or .5K 1024 or 1K 2048 or 2K 4096 or 4K 32768 or 32K
num-blocks	is the number of blocks to be allocated in cache. The maximum number of blocks is 2,097,152 regardless of *block-size*. The minimum number of blocks is 18 multiplied by the number of disk processes configured for the volume. Use the INFO DISK, CONFIG command to display this value in the NumDiskProcesses field.

Before using this attribute, see "Considerations for ALTER DISK, CACHE and Disks" (page 86).

CACHESIZE *number*

is the number of entries of the name cache of the virtual disk process

NOTE: This attribute is only valid for virtual disks.

The name cache is used to hold entries in the audit-name table. `number` is in the range 0 through 50000. The default is 0.

Avoid making the cache too small, which can adversely affect file-management operations (such as FILE_OPEN_ and file information requests) serviced by a virtual disk process.

CAPACITYMISMATCH { ON | OFF },

when a revive completes for a mirrored volume consisting of drives of different capacities, CAPACITYMISMATCH specifies whether the source drive remains UP (ON) or goes HARDDOWN (OFF).

For the ADD command, the default value for CAPACITYMISMATCH is OFF.

CAPACITYMISMATCH can be changed online or offline.

CBPOOLLEN `number`

is the maximum memory (in 128-KB units) that can be allocated for open-related data structures for the disk. The memory available for these structures limits the total number of concurrent opens allowed on the disk. `number` is in the range 0 through 1000. The default is 1000.

You can increase the value of this attribute while the disk is in the STARTED state. However, to decrease the value of this attribute, the disk must be in the STOPPED state. As a result, this attribute value cannot be decreased for the system disk unless you load the system from a saved system configuration database that contains a smaller value for this attribute.

CLEARENCRYPTKEY

clears encryption on an encrypted disk.

This attribute changes the encryption status for a disk (single drive) while that drive is DOWN. Its mirror may be UP. The disk volume will be online if the mirror is UP. The drive is initialized. The drive's volume label is left blank. If data needs to be copied from the other mirror, you must start the REVIVE manually after the INITIALIZE command.

You must specify a path.

NOTE:

- This attribute is not valid for the ADD DISK command and the PROFILE object.
- Only members of the SAFEGUARD security officer group on the local system can perform an ALTER DISK command with the CLEARENCRYPTKEY attribute. Also, only security officers can initiate a revive from an encrypted disk to a non-encrypted disk.

ENCRYPTPRIORITY `encrypt-priority`

sets the priority for the disk key rotation process.

`encrypt-priority` is a value between 1 and 100.

If you do not specify this value, the default is 4 for EncryptPriority.

NOTE: This attribute is not valid for the ADD DISK command and the PROFILE object.

ENCRYPTRATE `encrypt-rate`

sets the encryption rate for disk key rotation.

`encrypt-rate` is a value between 0 and 8.

If you do not specify this value, the default is 50 for EncryptRate.

NOTE: This attribute is not valid for the ADD DISK command and the PROFILE object.

FASTBULKWRITE { ON | OFF },

(effective for all 512-byte-sector disks) specifies whether the system is using Fast Cache Bulk Writes.

ON	Cache Bulk Write is done in Fast mode.
OFF	Cache Bulk Write is done in Traditional I/O mode.

The default is OFF.

When ON, applications and utilities using bulk writes to unstructured files may have higher throughput. Using FASTBULKWRITE ON can result in lost data in unstructured files if the CPU running the primary disk process fails. After a CPU failure, EMS event 5052 will report unstructured files that have become broken as a result of the failure, and applications will receive error 59 attempting to write to such files.

FASTBULKWRITE can be changed online or offline.

NOTE: These restrictions apply to partitioned unstructured files on an XP storage array and on H-series, J-series, and S-series internal disks:

- All extents must be a multiple of 14 pages.
- All partitions must have identical extent sizes and maximum extents, because partitioning depends on each partition having the same size.

FORCED

specifies that the disk attribute values be changed without any interaction with the user. SCF does not prompt for confirmation.

FSTCACHING { ON | OFF }

specifies that the free-space table (FST) for a disk is updated in memory (ON), that the FST is updated on disk (OFF) or that DP2 decides whether to keep the FST in memory or on disk (ENABLED). The default is OFF. FSTCACHING ON can increase performance. The FST is always rebuilt from disk when the disk is started.

The system disk always runs with FSTCACHING ON regardless of the configured value.

FULLCHECKPOINTS { ENABLED | DISABLED | FORCED }

(nondirectory structured files only) specifies when to perform full-block checkpoints

DISABLED	Never use a full-block checkpoint to protect the data on the disk. Using DISABLED maximizes disk performance, but risks corrupting data blocks on the disk.
ENABLED	Use a full-block checkpoint only when serial writes are not being performed. Use ENABLED on disks that perform parallel writes and on nonmirrored disks. For more details, see the SERIALWRITES attribute (Default.)
FORCED	Always use full-block checkpoint for disk-write operations.

For mirrored volumes, FULLCHECKPOINTS is important only if the primary or mirror disk is not up. In that situation, FORCED or ENABLED can protect the validity of data written to the disk.

When performing full-block checkpoints, the primary process checkpoints the data to the backup process before writing to the disk. This action ensures that the data on the disk is not corrupted even if a hardware freeze or processor halt occurs during a disk write.

Use FORCED or ENABLED unless data loss is not critical.

NOTE: This attribute is only valid for 514 byte per sector disks:

- internal SCSI disks in S-series enclosures
- 45xx disks in modular disk subsystems

DP2 ignores this attribute for 512-byte-sector disks:

- disks in an FCDM disk drive enclosure
- ESS disks in an Enterprise Storage array
- SAS disks in an MSA70 or M8390-12CG disk enclosure

HALTONERROR *number*

specifies whether an internally detected, unrecoverable, disk-process error forces a halt (code %11500) in the primary processor, backup processor, or both. Forced processor halts also halt the respective disk process. Forced processor halts can provide more information at the time of a failure. To force a halt when a failure is detected, set *number* to either 2, 3, or 4.

number is one of:

1	Never halt a processor (default). No processor halts, and the primary or backup disk process that detects an unrecoverable disk-process error can go into the STOPPED state, substate DOWN.
2	Never halt the backup processor. The primary processor halts, but the backup disk process can go into the STOPPED state, substate DOWN, if it detects an unrecoverable disk-process error.
3	Never halt the primary processor. The backup processor halts, but the primary disk process can go into the STOPPED state, substate DOWN if it detects an unrecoverable disk-process error.
4	Allow both processors to halt. Both the primary and backup processors can halt if either detects an unrecoverable disk-process error.

If an unrecoverable disk-process error is detected but the processor is not halted, the disk processes perform these actions:

- If the backup disk process detects the error, the disk state remains unchanged and the backup disk process goes into a “soft down” state but does not halt. The primary disk process continues to function without an active backup.
- If the primary disk process detects the error, it gives ownership of the disk to the backup disk process. If the ownership change is successful, the primary disk process becomes a soft-down backup (the process is not available). If the ownership change is unsuccessful, the disk volume goes into the STOPPED state, substate DOWN.
- If both processes enter a soft-down state, the volume enters the STOPPED state, substate DOWN.
- To restore the backup disk process:

1. Issue a RESET DISK command to ensure that all paths start.
2. Issue a START DISK command to reactivate the backup disk process.
3. If the RESET or START command fails, you must reload the backup processor.

HIGHPIN { ON | OFF }

specifies whether the IOP can run at a high PIN (ON, the default) or only at a low PIN (OFF).

IGNOREINCONSISTENCY

specifies that a command can create an inconsistency between the system configuration database and the SMF catalogs maintained by the SMF manager process, storage pools, and virtual disks.

NOTE: This attribute is not valid for the PROFILE object.

Use this attribute only if you understand SMF architecture and know how to restore consistency between the system configuration database and the SMF catalogs. The Softdoc for the T1083 product describes several inconsistencies and the procedures for reconciling them.

KEYALGORITHM,

specifies the encryption key algorithm on a disk. Valid values are XTS-AES and CBC-AES

This attribute changes the encryption key for a disk (single drive) while that drive is DOWN. Its mirror may be UP. The disk volume will be online if the mirror is UP. The drive is initialized. The drive's volume label is left blank. If data needs to be copied from the other mirror, you must start the REVIVE manually after the INITIALIZE command.

You must specify a path.

NOTE: This attribute is not valid for the ADD DISK command and the PROFILE object.

KEYSIZE

specifies the key size for the key algorithm for an encrypted disk. Valid value is 256.

NOTE: This attribute is not valid for the ADD DISK command and the PROFILE object.

LABEL $`vol`

writes a volume label on a newly formatted disk (or on both halves of a mirrored volume) or relabels a previously labeled volume and erases all existing files

NOTE: This attribute is not valid for the PROFILE object.

Before using this attribute, carefully review all cautions and considerations in "Naming a Disk" (page 88). Refer to the procedure for the LABEL attribute, "Changing the Volume Name and Alternate Volume Name" (page 88).

LKIDLONGPOOLLEN *number*

is the memory (in 128-kilobyte units) to be allocated for lock key space. This space stores keys larger than 16 bytes when records in key-sequenced files are locked. *number* is in the range 0 through 512. A value of 0 causes DP2 to default to an internal default value.

LKTABLESPACELEN *number*

is the memory (in 128-kilobyte units) to be allocated for lock-related data structures. The memory available for these structures limits the total file and record locks allowed on the disk. *number* is in the range 0 through 512. A value of 0 causes DP2 to default to an internal default value.

MAXLOCKSPEROCB *number*

is the maximum records that can be locked outside a transaction. *number* is in the range 0 through 1,000,000. The default is 5000.

You can increase the value of this attribute while the disk is in the STARTED state. However, to decrease the value of this attribute, the disk must be in the STOPPED state. As a result, this attribute value cannot be decreased for the system disk unless you load the system from a saved system configuration database that contains a smaller value for this attribute.

Using large values for MAXLOCKSPEROCB can have a noticeable performance impact. The volume may become non-responsive for several seconds or more, based on how many locks are acquired during a transaction. The impact becomes significant when more than 100,000 locks are being released, and becomes larger in direct relationship to MAXLOCKSPEROCB. Delays may vary between 2 seconds for 150,000 locks to 20 seconds for 500,000. During this time, other applications or system processes may experience timeouts due to the extended response time of the volume. Applications should avoid obtaining 100,000 or more locks within a transaction on a regular basis.

MAXLOCKSPERTCB *number*

is the maximum records and files that a transaction can lock. *number* is in the range 0 through 1,000,000. The default is 5000.

You can increase the value of this attribute while the disk is in the STARTED state. However, to decrease the value of this attribute, the disk must be in the STOPPED state. As a result, this attribute value cannot be decreased for the system disk unless you load the system from a saved system configuration database that contains a smaller value for this attribute.

Using large values for MAXLOCKSPERTCB can have a noticeable performance impact. The volume may become non-responsive for several seconds or more, based on how many locks are acquired during a transaction. The impact becomes significant when more than 100,000 locks are being released, and becomes larger in direct relationship to MAXLOCKSPERTCB. Delays may vary between 2 seconds for 150,000 locks to 20 seconds for 500,000. During this time, other applications or system processes may experience timeouts due to the extended response time of the volume. Applications should avoid obtaining 100,000 or more locks within a transaction on a regular basis.

MBACKUPCLIM *clim-name*

is the name of the CLIM that controls the mirrored backup path to the disk.

NOTE: This attribute is not valid for the PROFILE object.

MBACKUPDEVICEID { *number* | (*shelf, bay*) }

(45*xx* and M8*xxx* Fibre Channel disks only) is the device ID of the disk accessed on the mirror backup path to the disk

NOTE: This attribute is not valid for the PROFILE object.

This attribute is optional. If specified, it must be the same device ID that you specified for theMIRRORDEVICEID attribute.

For 45*xx* disks:

number	is the unit number of the disk. *number* is in the range 1 through 7.

For M8*xxx* Fibre Channel disks:

shelf	is the FCDM's shelf number. *shelf* is in the range 1 through 4.
bay	is the number of the disk. *bay* is in the range 1 through 14.

If you configure a disk volume for use with an FCSA connected to an ESS, you cannot specify MBACKUPDEVICEID.

MBACKUPLOCATION (*group, module, slot*)

(mirrored 45*xx* and mirrored M8*xxx* disks only) is the location of the adapter that controls the mirror backup path to the disk

NOTE: This attribute is not valid for the PROFILE object.

This attribute is optional if MBACKUPSAC is given in *name* format, but required if given in *number* format.

MBACKUPPORTNAME *number*

specifies the Fibre Channel port name (WWN) used by the backup path to a mirror disk on the Enterprise Storage System (ESS)

NOTE: This attribute is not valid for the PROFILE object.

The ESS administrator must give you the WWN so that you can specify it in this attribute.

Enter *number* as a 16 character hexadecimal number, without a leading %H.

MBACKUPSAC { *number* | *name* }

(mirrored 45*xx* and mirrored M8*xxx* disks only) is the SAC that controls the mirror backup path to the disk

NOTE: This attribute is not valid for the PROFILE object.

number	Is the SAC subdevice number on the adapter. If you specify MBACKUPSAC in *number* format, you must also specify MBACKUPLOCATION.
name	Is the full name of the SAC location. For example:`SNDA.SAC-1.GRP-1.MOD-1.SLOT-53FCSA.SAC-1.GRP-11.MOD-2.SLOT-5`

MIRRORCLIM *clim-name*

is the name of the CLIM that controls the mirrored path to the disk.

NOTE: This attribute is not valid for the PROFILE object.

MIRRORDEVICEID { *number* | (*shelf,bay*) }

(mirrored 45*xx* and mirrored M8*xxx* disks only) is the device ID of the disk accessed on the mirror path to the disk

NOTE: This attribute is not valid for the PROFILE object.

For 45*xx* disks:

number	Is the unit number of the disk. *number* is in the range 1 through 7.

For M8*xxx* disks in an FCDM:

shelf	Is the FCDM's shelf number. *shelf* is in the range 1 through 4.
bay	Is the number of the disk. *bay* is in the range 1 through 14.

If you configure a disk volume for use with an FCSA connected to an ESS, you cannot specify MIRRORDEVICEID.

MIRRORPARTITION *mirror-part-num*

specifies the partition number for the mirror half of a disk volume. If this attribute is not specified, the mirror disk is not a partitioned disk.

MIRRORING { OFF | ON }

specifies whether a disk should be automatically configured to be half of a mirrored disk

NOTE: This attribute is only valid for the PROFILE object.

`ON`	Enables automatic configuration as half of a mirrored disk. A disk in an odd-numbered slot is assigned the next-higher slot number for its mirror; a disk in an even-numbered slot is assigned the next-lower slot number for its mirror. (Default)
`OFF`	Disables automatic mirroring.

MIRRORLOCATION (*group, module, slot*)

for internal disks, is the location of the mirror half of the disk volume

NOTE: This attribute is not valid for the PROFILE object.

For 45*xx*, ESS, and M8*xxx* disks, is the location of the adapter that controls the mirror path to the disk.

This attribute is optional if MIRRORSAC is given in *name* format but required if given in *number* format.

MIRRORLOCATION can be altered when all paths to the volume are in the STOPPED state.

MIRRORLOCATION can also be altered when all configured paths to the primary drive are in STARTED state and MIRRORLOCATION is not yet configured. The result of an online reconfiguration must be symmetrical. If the volume has a BACKUP path, it must also have a MBACKUP path.

You cannot change the value of this attribute by using the ALTER command online after setting it (using either the ADD or ALTER command). To change the value online, you must delete, respecify the mirror location.

MIRRORLUN *number*

specifies the logical unit number (LUN) on the Enterprise Storage System.

NOTE: This attribute is not valid for the PROFILE object.

The ESS administrator must give you the LUN so that you can specify it in this attribute.

The default value for MIRRORLUN is 0.

MIRRORPORTNAME *number*

specifies the Fibre Channel port name (WWN) used by the path to a mirror disk volume on an Enterprise Storage System (ESS)

NOTE: This attribute is not valid for the PROFILE object.

The ESS administrator must give you the WWN so to specify it in this attribute.

Enter *number* as a 16 character hexadecimal number, without a leading %H.

MIRRORSAC { *number* | *name* }

(mirrored 45*xx* and M8*xxx* disks only) is the SAC that controls the mirror path to the disk

NOTE: This attribute is not valid for the PROFILE object.

number	Is the SAC subdevice number on the adapter.
name	Is the full name of the SAC location including the SAC number. For example:`SNDA.SAC-1.GRP-1.MOD-1.SLOT-53FCSA.SAC-1.GRP-11.MOD-2.SLOT-5`

MODE { NOISY | QUIET }

specifies whether to generate additional EMS messages for debugging

NOTE: This attribute is only valid for virtual disks.

`NOISY`	Generate additional EMS messages for debugging.
`QUIET`	Generate only essential EMS messages (default).

NAMEMASK { STANDARD | *name* }

specifies a naming convention for automatically configuring disks

NOTE: This attribute is only valid for the PROFILE object.

`STANDARD`	The disk is named using the standard manufacturing naming convention (default). The name format is $D*ggss*, where *gg* is the group number and *ss* is the slot number. For example, a disk inserted

	in group 31, slot 12 is named $D3112. For M8*xxx* Fibre Channel disks, the standard name format is $FC*nn*, where *nn* is supplied by the storage subsystem. *nn* begins in the range 00 through 99 and continues with A0 through ZZ. If MIRRORING is ON, the name is created when the first disk is inserted, and the same name is used when the second disk inserted for the mirrored volume.
name	The disk is named using a nonstandard naming convention. *name* can have up to 4 characters. The name format is $*namenn*, where *nn* is supplied by the storage subsystem. *nn* begins in the range 00 through 99 and continues with A0 through ZZ. For example, NAMEMASK DATA specifies the naming convention $DATA00, $DATA01, and so on.

NEWENCRYPTKEY

sets a new encryption key on a disk.

If the disk is not encrypted, it is initialized as encrypted. If the disk is already encrypted, this attribute changes the encryption key for a disk (single drive) while that drive is DOWN. Its mirror may be UP. The disk volume will be online if the mirror is UP. The drive's volume label is left blank. If data needs to be copied from the other mirror, you must start the REVIVE manually after the INITIALIZE command.

You must specify a path.

NOTE:

- This attribute is not valid for the ADD DISK command and the PROFILE object.
- Only members of the SAFEGUARD security officer group on the local system can perform an ALTER DISK command with the NEWENCRYPTKEY attribute. Also, only security officers can initiate a revive from an encrypted disk to a non-encrypted disk.

NONAUDITEDINSERT { ON | OFF }

(physical disks) specifies whether nonaudited insert mode is enabled (ON) or disabled (OFF)

`ON`	enhances performance by buffering insertions to entry-sequenced files or relative-sequenced files that do not have the REFRESH attribute set and are open with a SYNCDEPTH greater than 0. This feature is known as nonaudited insert mode. However, the resulting reduction in the frequency of label updates potentially increases the probability of data loss in the event of a double disk failure.
`OFF`	disables nonaudited insert mode (default). This option forces write inserts, which increase EOF.

NOSTART

specifies that the mirror disk should not be started for online disk reconfiguration with the ALTER DISK command. To start the mirrored disk half, use the START DISK command.

NOTE: This attribute is not valid for the ADD DISK command and the PROFILE object.

NUMDISKPROCESSES *number*

is the number of disk processes allocated for the disk. *number* is in the range 1 through 8. The default is 4.

Specifying fewer than 4 disk processes conserves memory but could decrease system performance. Specifying more than 4 disk processes requires more memory but could improve performance. The effect on your system depends on your applications and the disk traffic they create.

You can increase the value of this attribute while the disk is in the STARTED state. However, to decrease the value of this attribute, the disk must be in the STOPPED state. As a result, this attribute value cannot be decreased for the system disk unless you load the system from a saved system configuration database that contains a smaller value for this attribute.

OSSCACHING { ON | OFF }

specifies whether caching for Open System Services (OSS) files is ON or OFF. The default is ON.

PHYSVOLSELECT { ON | OFF }

specifies whether a virtual disk process can consider this physical volume for file placement. This attribute is valid only when this physical volume is a member of a storage pool

`ON`	A virtual disk process can consider this physical volume for file placement (default).
`OFF`	A virtual disk process cannot consider this physical volume for file placement, regardless of the information supplied in a command. If PHYSVOLSELECT is OFF, the value of AUTOSELECTis ignored.

POOL *$pool*

(for virtual disks) is the name of the storage pool process associated with the virtual disk.

NOTE: This attribute is required when adding virtual disks. It has no default and it cannot be used with the ALTER DISK command.

The storage pool process must be in the STARTED state. In addition, the CATALOGLOCATION volume of that storage pool process must be in the STARTED state and enabled in TMF.

POOL { $*pool* | EXCLUDE }

(for physical disks and profiles) specifies whether a physical volume is a member of a specified storage pool

$pool	is the name of the storage pool to which the physical volume is being added.
`EXCLUDE`	removes a physical volume from a storage pool.

See "Configuring and Managing Storage Pools for Disks" (page 134).

PRIMARYCLIM *clim-name*

is the name of the CLIM that controls the primary path to the disk.

NOTE: This attribute is not valid for the PROFILE object.

PRIMARYCPU *number*

is the processor in which the primary IOP starts.

NOTE: This attribute is not valid for the PROFILE object.

The process must be in the STOPPED state or not running for you to alter this attribute.

PRIMARYDEVICEID { *number* | (*shelf*,*bay*) }

(45*xx* and M8*xxx* disks only) is the device ID of either a nonmirrored disk or of the primary disk of a mirrored volume

NOTE: This attribute is not valid for the PROFILE object.

For 45*xx* disks:

number	Is the unit number of the disk. *number* is in the range 1 through 7.

For M8*xxx* disks in an FCDM:

shelf	Is the FCDM's shelf number. *shelf* is in the range 1 through 4.
bay	Is the number of the disk. *bay* is in the range 1 through 14.

If you configure a disk volume for use with an FCSA connected to an ESS, you cannot specify PRIMARYDEVICEID.

PRIMARYLOCATION (*group*, *module*, *slot*)

(for internal disks) is the location of the slot where the primary disk of the mirrored volume is installed.

NOTE: This attribute is not valid for the PROFILE object.

(for 45*xx*, ESS, and M8*xxx* physical disks) is the location of the adapter that controls the primary path to the disk.

This attribute is optional if PRIMARYSAC is given in *name* format but required if given in *number* format.

PRIMARYLUN *number*

specifies the logical unit number (LUN) of a primary disk device.

NOTE: This attribute is not valid for the PROFILE object.

The ESS administrator must give you the LUN so that you can specify it in this attribute.

The default value for PRIMARYLUN is 0. You can configure as many as 1,000 LUNs per FCSA and as many as 125 LUNs per processor pair.

PRIMARYPARTITION *primary-part-num*

specifies the partition number for the primary half of a disk volume. If this attribute is not specified, the primary disk is not a partitioned disk.

PRIMARYPORTNAME *number*

specifies the Fibre Channel port name (WWN) used by the path to a primary disk on an Enterprise Storage System (ESS)

NOTE: This attribute is not valid for the PROFILE object.

The ESS administrator must give you the WWN so that you can specify it in this attribute.

Enter *number* as a 16 character hexadecimal number, without a leading %H.

PRIMARYSAC { *number* | *name* }

is the SAC that controls the primary path to the device

NOTE: This attribute is not valid for the PROFILE object.

number	is the SAC subdevice number on the adapter. If you specify PRIMARYSAC in *number* format, you must also specify the PRIMARYLOCATION attribute.
name	is the full name of the adapter location including the SAC number. For example:`PMF.SAC-1.GRP-1.MOD-1.SLOT-55FCSA.SAC-1.GRP-11.MOD-2.SLOT-5`Use this form to override the default selection of which adapter gets the -P path and which gets the -B path.

PROGRAM [[$*vol*.]*subvol*.]*fileid*

is the object file name of the IOP. If not specified, PROGRAM is:

```
$SYSTEM.SYSTEM.TSYSDP2 (for physical disks)
```

```
$SYSTEM.SYSTEM.OVDP (for virtual disks)
```

If you specify the program file name location as $SYSTEM.SYSTEM, the operating system first searches for the file there and on the current SYS*nn* subvolume.

PROTECTDIRECTORY { CHECKPOINT | OFF | SERIAL }

is the type of protection to use for the disk volume directory:

`CHECKPOINT`	Use a full-block checkpoint to protect directory writes.
`OFF`	Do not perform special write protection of the directory.
`SERIAL`	A write to the directory also results in a serial write to the mirror disk, if available. If the mirror disk is not available, a full-block checkpoint writes the directory. (Default)

NOTE: This attribute is only valid for 514 byte per sector disks:

- internal SCSI disks in S-series enclosures
- 45xx disks in modular disk subsystems

DP2 ignores this attribute for 512-byte-sector disks:

- disks in an FCDM disk drive enclosure
- ESS disks in an Enterprise Storage array
- SAS disks in an MSA70 or M8390-12CG disk enclosure

RECOVERYTIMEOUT *number*

specifies the number of seconds that the NonStop storage controller must wait for the device to respond to a recovery I/O. If the device does not respond within the specified time, the NonStop storage controller times out and aborts the I/O.

NOTE: This attribute only affects ESS (Enterprise Storage) disks.

Valid values are 0, 30, and 60 (seconds). A value of 0 causes DP2 to use an internal default value. For the ADD command, the default value for RECOVERYTIMEOUT is 0.

REVIVEBLOCKS *number*

is not a valid attribute for objects on systems running G06.24 or later RVUs. You can use this attribute only when you are altering objects over an Expand link and targeting a system that uses an older version of the storage subsystem.

REVIVEINTERVAL *number*

is not a valid attribute for objects on systems running G06.24 or later RVUs. You can use this attribute only when you are altering objects over an Expand link and targeting a system that uses an older version of the storage subsystem.

REVIVEPRIORITY *number*

specifies the priority of a revive task that executes in the DP2 IOP.

During the process of reviving a mirrored disk volume, DP2 must perform the time-consuming task of copying data from the STARTED half of the disk volume to the half that is in the REVIVE substate.

The REVIVEPRIORITY attribute specification can lessen the impact of this task on system resources by giving the task a lower priority. For example, if REVIVEPRIORITY is 50, the revive process defers to new requests with a priority above 50. Specifying a lower priority value ensures that the revive operation does not impact higher priority workloads, but it does lengthen the time that the revive operation takes.

You can adjust the REVIVEPRIORITY value during the revive operation to speed or slow the revive progress.

The minimum value for REVIVEPRIORITY is 0, and the maximum value is 199. The default value is 0, which causes DP2 to use its internal default of 50

REVIVERATE *number*

determines the amount of data revived between preemption checks. The higher the rate, the more the impact on higher priority workloads; the lower the rate, the less the impact on higher priority workloads. At the same time, the higher the rate, the faster the revive operation, and the lower the rate, the slower the revive operation.

The minimum value is 0, and the maximum value is 100. The default value is 0, which causes DP2 to use its internal default of 50.

This table shows the effects of specifying high and low values in the *number* field of the REVIVEPRIORITY and REVIVERATE attributes when you specify these attributes in combination. For Rate, Low is the range 0 through 50, and High is the range 51 through 100.

Priority	Rate	Effect of This Combination
Low	Low	The lowest possible impact to online activity. If there is no online activity, the revive progress is slightly throttled by a low rate.
Low	High	The revive makes progress, but it has limited impact to online activity. If there is no online activity, the revive progress will be as fast as possible.
High	High	The revive completes as quickly as possible, but there may be impact to concurrent activity. This combination is recommended only when there is a requirement to complete the revive as fast as possible in the case of high priority workloads.

DP2's internal default REVIVEPRIORITY of 50 (low) and REVIVERATE of 100 (high) is acceptable for most environments. If there is any impact to concurrent activity, lower REVIVERATE until the impact is reduced to an acceptable level.

SERIALWRITES { ENABLED | DISABLED }, ,

(mirrored volumes only) specifies whether serial writes are used when updating files

`ENABLED`	Perform serial writes if the mirror disk is available. If the primary or mirror disk is not available, and if FULLCHECKPOINTS is ENABLED or FORCED, full-block checkpoints protect the validity of the data. ENABLED minimizes potential data loss caused by certain processor error conditions. (Default)
`DISABLED`	Do not perform serial writes unless this attribute has been specified for a specific file and the mirror disk is available. DISABLED maximizes the performance of mirrored disks.

NOTE: This attribute is only valid for 514 byte per sector disks:

- internal SCSI disks in S-series enclosures
- 45xx disks in modular disk subsystems

DP2 ignores this attribute for 512-byte-sector disks:

- disks in an FCDM disk drive enclosure
- ESS disks in an Enterprise Storage array
- SAS disks in an MSA70 or M8390-12CG disk enclosure

On BladeSystems, this attribute is ignored. This attribute cannot be enabled for disk subsystems attached with CLIMs because DP2 uses parallel writes for better performance. Serial writes are not needed because the CLIM is fully buffered.

SQLMXBUFFER *number*,

(SQL/MX volumes only) is the buffer size (in megabytes) for an SQL/MX session. *number* is in the range 0 through 768. The default is 0. Do not use this attribute for TMF audit-trail volumes.

To change the SQLMXBUFFER attribute, first put the disk in the STOPPED state by using the "STOP DISK Command" (page 290):

- If you specify too large or small a value, DP2 automatically sets the needed size.
- If you specify a value of 0, DP2 automatically sets the size of the buffer to 128.
- If SQL/MX is not installed, setting this attribute to 16 allows DPS to increase DP2 cache to its maximum.

STARTSTATE { STARTED | STOPPED }

specifies whether the IOP is available to other processes (STARTED) or unavailable (STOPPED) when the system is loaded or reloaded. The default is STARTED.

SWAPMIRROR

specifies that the data path attributes (LOCATION, SAC, DEVICEID, PORTNAME, LUN) of the PRIMARY path should be swapped with the MIRROR path, and the data path attributes of the BACKUP path should be swapped with the MBACKUP path. No other data path attributes can be specified when SWAPMIRROR is used

NOTE: This attribute is not valid for the ADD DISK command and the PROFILE object.

SWAPMIRROR can be done online if all 4 paths are UP, or if 2 paths are UP in a symmetrical pattern (P and M, B and MB, P and B, or M and MB).

TYPE { MAGNETIC | VIRTUAL }

is the type of disk you are adding. The default is MAGNETIC

NOTE: This attribute is not valid for the ALTER DISK command and the PROFILE object.

VOLNAME $*vol*

is the default volume name for the disk. This name is used when the system first tries to start the volume. This attribute changes the volume name on the label of the disk but does not change the system configuration database

NOTE: This attribute is not valid for the ADD DISK command and the PROFILE object.

To use this attribute, see "Changing Either the Volume Name or Alternate Volume Name" (page 90).

WRITECACHE { ENABLED | DISABLED }

specifies whether write caching is enabled for a volume.

`ENABLED`	specifies that write caching is enabled for a volume.
`DISABLED`	specifies that write caching is disabled for a volume (default).

- Write caching can improve disk performance.
- This attribute is available only on J06.03 and subsequent J-series RVUs.
- This attribute affects only CLIM-connected SAS disk drives. It is ignored for all other disks.
- If a volume is configured with WRITECACHE enabled, HP strongly recommends that the volume be protected by an HP rack mount UPS, which will give the drives enough time to write cached data to the media after the processor stops writing in the event of power loss. See "Write Caching" (page 94) for details about why an HP rack mount UPS is necessary.
- The SUBSYS attribute UPS should be ON either before WRITECACHE is enabled or before the IOP is started. Otherwise, the IOP will run with WRITECACHE disabled.

- When adding a new disk on disk partitions, the writecache setting of the new disk must match the writecache setting of the existing disk partitions.
- Altering a writecache setting of a disk partition sets the writecache setting of all the disk partitions that are configured on the same physical disk. Changes to the other disk partitions set the writecache setting of all their mirror disks.

To use this attribute, see "Write Caching" (page 94).

WRITEVERIFY { ON | OFF }

specifies whether the disk process should enable persistent write verify.

NOTE: This attribute is not valid for the ADD DISK command and the PROFILE object.

`ON`	The SAC performs additional verification to ensure that data is written error-free.
`OFF`	No additional verification is performed (default).

When you set WRITEVERIFY ON, you give up some system performance for a higher level of data integrity. WRITEVERIFY ON is sometimes set automatically by the disk process if it detects a specific error situation that needs the extra protection.

ADD MON Command

The ADD MON command adds the $ZSMS master Storage Management Foundation (SMF) process to the system configuration database. The syntax is:

```
ADD [ / OUT file-spec / ] MON $ZSMS , SENDTO STORAGE
    [ ,  attribute-spec ]...
```

Wild-card characters are not supported.

MON $ZSMS

is the SMF master process.

SENDTO STORAGE

directs the command to the storage subsystem. This attribute is required unless you specified SENDTO STORAGE in a previous ASSUME command.

attribute-spec

is one or more "ADD MON Attributes".

ADD MON Attributes

```
[ , BACKUPCPU number ]
[ , CATALOGLOCATION $vol ]
[ , HIGHPIN { ON | OFF } ]
[ , MODE { NOISY | QUIET } ]
[ , PRIMARYCPU number ]
[ , PROGRAM [[$vol.]subvol.]fileid ]
```

BACKUPCPU *number*

is the processor in which the backup IOP starts. The default is BACKUPCPU 1.

CATALOGLOCATION *$vol*

is the physical volume where the catalog subvolume (ZSMSCAT0) for the SMF master process is located.

$vol must be in the STARTED state and enabled in TMF. If CATALOGLOCATION is not specified, *$vol* is set to $SYSTEM.

If the ZSMSCAT0 subvolume exists on *$vol*, $ZSMS assumes that the subvolume contains valid catalog information. If the subvolume does not contain valid catalog information, $ZSMS abends.

If the ZSMSCAT0 subvolume does not exist on *$vol*, $ZSMS creates the subvolume and initializes it with appropriate catalog information.

HIGHPIN { ON | OFF }

specifies whether the IOP can run at a high PIN (ON, the default) or only at a low PIN (OFF).

MODE { NOISY | QUIET }

specifies whether to generate additional EMS messages for debugging

`NOISY`	Generate additional EMS messages for debugging.
`QUIET`	Generate only essential EMS messages (default).

PRIMARYCPU *number*

is the processor in which the primary IOP starts. The default is PRIMARYCPU 0.

PROGRAM [[*$vol*.]*subvol*.]*fileid*

is the object file name of the IOP. If not specified, PROGRAM is:

`$SYSTEM.SYSTEM.OMP`

If you specify PROGRAM as $SYSTEM.SYSTEM, the operating system first searches for the file on $SYSTEM.SYSTEM and on the current SYS*nn* subvolume.

ADD MON Example

See the procedure and considerations for "Adding the SMF Master Process" (page 44).

To add the SMF master process with the catalog located on $DATA00:

```
-> ADD MON $ZSMS, SENDTO STORAGE, PRIMARYCPU 0, &
-> BACKUPCPU 1, CATALOGLOCATION $DATA00
```

ADD PARTITION Command

Effective with the H06.23/J06.12 RVUs, you can use the ADD PARTITION command to add up to four partitions to SAS HDDs (hard disk drives) and eight partitions to SAS SSDs (solid state drives) in SAS enclosures connected to CLIMs. To determine the type of HDDs that support partitioning, refer to the hardware installation guide for your particular system.

CAUTION: Be sure to save your disk configuration after partitioning. In order to repartition a disk during disk replacement, you need one of these:

- Another disk that has been partitioned in the same configuration
- An obeyform file
- A record of the partition configuration

The syntax for the ADD PARTITION command is:

```
ADD PARTITION $ZZSTO [, SENDTO STORAGE]
, PRIMARYCLIM clim-name, PRIMARYLUN lun
, BACKUPCLIM clim-name [, BACKUPLUN lun]
, [ LIKE ( clim-name, lun ) |
    STARTPARTITION partition-number,
    [ PARTITIONSIZE ( partition-size-1, … partition-size-8 ) |
      PARTITIONSIZE partition-size, PARTITIONCOUNT partition-count
    ]
  ]
[, FORCED]
```

Wild-card characters are not supported.

SENDTO STORAGE

directs the command to the storage subsystem.

PRIMARYCLIM *clim-name*

is the name of the one CLIM that is connected to the disk.

PRIMARYLUN *lun*

specifies the logical unit number (LUN) on the PRIMARYCLIM.

BACKUPCLIM *backup-clim-name*

is the name of the other CLIM that is connected to the disk. BACKUPCLIM must be specified.

BACKUPLUN *backup-lun*

specifies the logical unit number (LUN) on the BACKUPCLIM. This attribute is optional. If BACKUPLUN is not specified, the value of PRIMARYLUN is used as the BACKUPLUN.

LIKE (*clim-name*, *lun*)

identifies another physical disk whose partition configuration is copied to the physical disk specified by the PRIMARYCLIM and PRIMARYLUN. When you use the LIKE attribute, you do not use the PARTITIONSIZE, PARTITIONCOUNT and STARTPARTITION attributes.

CAUTION: The LIKE attribute will destroy all data in the physical disk specified by the PRIMARYCLIM and PRIMARYLUN.

STARTPARTITION *partition-number*

is the next available partition to be added.

PARTITIONSIZE (*partition-size-1*, ... *partition-size-8*) | PARTITIONSIZE *partition-size*

specifies a list of partition sizes.

PARTITIONCOUNT *partition-count*

the number of partitions of the same size. Only one partition size can be specified.

FORCED

if this is the first partition being created (i.e. STARTPARTITION number is 1), SCF issues a warning that any existing data on the disk will be lost and prompts for confirmation. A FORCED option displays the warning but bypasses the prompt for confirmation.

ADD PARTITION Considerations

- You must specify the physical disk location (such as primary and backup CLIM names and LUNs), along with a partition size that is a multiple of 1 GB.
- To convert an unpartitioned disk to a partitioned disk, you must first delete any NonStop disk LDEV paths that might be configured to that disk. You can then add partitions to the disk.

  NOTE: All data on an unpartitioned disk is lost when a partition is added to the disk.
- Adding the first partition to an unpartitioned disk partitions the disk and removes any existing data. If the disk was previously partitioned, an added partition is allocated at the start of the unused space, shrinking the unused space by the size of the partition that was just added. Any remaining disk space is marked as unused space.
- If the unused space cannot accommodate the partition size, the add is rejected with an error.
- Partition numbers are automatically assigned by the Storage CLIM. You can obtain the partition number by using the "INFO PARTITION Command" (page 256) to display the disk partition information.

- You can add a partition to a partitioned disk while other existing partitions on the disk are active (online). Adding the partition does not affect the data on other existing partitions.
- After adding disk partitions, each disk partition can be configured to a NonStop DISK LDEV using the ADD DISK command. The NonStop DISK LDEV configured to a disk partition must be initialized before it can be started.
- For a partitioned disk, encryption is enabled on the disk partition level instead of the whole physical disk. When a disk is replaced, encryption must be specified again during NonStop DISK LDEV initialization, which requires a security officer, as described under "Managing Encrypted Disk Drives" (page 116).

ADD PARTITION Examples

- This example uses the PARTITIONCOUNT attribute. This example adds four partitions of size 20 GB each.

```
ADD PARTITION $ZZSTO, &
     SENDTO STORAGE, &
     PRIMARYCLIM S1002533, &
     PRIMARYLUN 100, &
     STARTPARTITION 1, &
     PARTITIONSIZE 20, &
     PARTITIONCOUNT 4, &
     BACKUPCLIM S1002534
```

- This example's PARTITIONSIZE attribute takes a list of partition sizes that are also used by the INFO PARTITION, OBEYFORM command to reconstruct the partitions on a disk. This example adds four partitions of size 20 GB, 10 GB, 30 GB and 40 GB.

```
ADD PARTITION $ZZSTO, &
     SENDTO STORAGE, &
     PRIMARYCLIM S1002533, &
     PRIMARYLUN 101, &
     STARTPARTITION 1, &
     PARTITIONSIZE (20, 10, 30, 40), &
     BACKUPCLIM S1002534
```

- This example uses the LIKE attribute.

```
ADD PARTITION $ZZSTO, &
    SENDTO STORAGE, &
    PRIMARYCLIM S1002533, &
    PRIMARYLUN 100, &
    BACKUPCLIM S1002531, &
    LIKE ( S1002534, 200 )
```

ADD POOL Command

The ADD POOL command adds a storage pool to the system configuration database. The syntax is:

```
ADD [ /OUT file-spec/ ] POOL $pool , SENDTO STORAGE
   [ , LIKE object ] [ , attribute-spec ]...
```

Wild-card characters are not supported.

OUT *file-spec*

directs all SCF output to the specified file.

POOL $*pool*

is the storage pool process.

SENDTO STORAGE

directs the command to the storage subsystem. This attribute is required unless you specified SENDTO STORAGE in a previous ASSUME command.

LIKE *object*

identifies an existing object whose attribute values are copied to the object affected by this command. The object name must currently exist in the system configuration, and the object type must match that of the object affected by this command. If you specify LIKE, any attribute values that you explicitly specify override the attribute values of the LIKE object.

The location, SAC, and device ID values are not copied from the LIKE object. To assign values to these attributes, you must explicitly specify them.

attribute-spec

is one or more "ADD POOL Attributes".

ADD POOL Attributes

```
[ , AUDITED { ALLOWED | DISALLOWED | REQUIRED } ]
[ , BACKUPCPU number ]
[ , CATALOGLOCATION $vol[.subvol ] ]
[ , DISKINTERVAL number ]
[ , DISKTHRESHOLD number ]
[ , EXTENTINTERVAL number ]
[ , EXTENTTHRESHOLD number ]
[ , HIGHPIN { ON | OFF } ]
[ , MAGNETIC { ALLOWED | DISALLOWED | REQUIRED } ]
[ , MIRRORED { ALLOWED | DISALLOWED | REQUIRED } ]
[ , MODE { NOISY | QUIET } ]
[ , PRIMARYCPU number ]
[ , PROGRAM [[$vol.]subvol.]fileid ]
[ , STARTSTATE { STARTED | STOPPED } ]
[ , UPDATESTATS number ]
```

AUDITED { ALLOWED | DISALLOWED | REQUIRED }

is the type of volume you can add to the storage pool

`ALLOWED`	You can add any volume, even if it is not protected by TMF (default).
`DISALLOWED`	You can add a volume only if it is not protected by TMF.
`REQUIRED`	You can add a volume only if it is protected by TMF.

BACKUPCPU *number*

is the processor in which the backup IOP starts. The default is BACKUPCPU 1.

CATALOGLOCATION $*vol*[.*subvol*]

is the location of the catalog for the storage pool process.

$*vol* must be in the STARTED state and enabled in TMF. If CATALOGLOCATION is not specified, *$vol* is set to $SYSTEM.

If you omit *subvol*, the storage pool process selects an empty subvolume and gives it the name ZSMSP*nn*.

DISKINTERVAL *number*

is the interval (as a percentage) that, with the DISKTHRESHOLD attribute, is used by the disk process to determine when to issue a disk-full EMS message for physical volumes in the storage pool. For example, if DISKTHRESHOLD is 70 and DISKINTERVAL is 5, the disk process issues an EMS message when the disk becomes 70, 75, 80, 85, 90, and 95 percent full.

number is either -1 (the default, no disk-full EMS messages are issued) or in the range 1 through 99 percent.

This attribute has no effect if DISKTHRESHOLD is -1.

DISKTHRESHOLD *number*

is the threshold (as a percentage) that, with the DISKINTERVAL attribute, is used by the disk process to determine when to issue a disk-full EMS message for physical volumes in the storage pool. For example, if DISKTHRESHOLD is 75, the disk process generates an EMS message when the physical volume becomes 75 percent full. If you also specify DISKINTERVAL, the disk process generates an EMS message when the physical volume becomes 75, 80, 85, 90, and 95 percent full.

number is either -1 (the default, no disk-full EMS messages are issued) or in the range 1 through 100 percent.

EXTENTINTERVAL *number*

is the number of extents that, with the EXTENTTHRESHOLD attribute, is used by the disk process to determine when to issue an extents-left EMS message for physical volumes in the storage pool. For example, if EXTENTTHRESHOLD is 6 and EXTENTINTERVAL is 2, the disk process issues an EMS message when there are 6, 4, 2, and 0 extents left to be allocated for a file.

number is either -1 (the default, no extents-left EMS messages are issued) or in the range 1 through 978.

This attribute has no effect if EXTENTTHRESHOLD is -1.

EXTENTTHRESHOLD *number*

is the number of extents that, with the EXTENTINTERVAL attribute, is used by the disk process to determine when to issue an extents-left EMS message for physical volumes in the storage pool. For example, if EXTENTTHRESHOLD is 6, the disk process issues an EMS message when only six extents remain to be allocated to a file. If you also specify EXTENTINTERVAL, the disk process issues an EMS message when there are 4, 2, and 0 extents left to be allocated for a file.

number is either -1 (the default, no extents-left EMS messages are issued) or in the range 1 through 978.

HIGHPIN { ON | OFF }

specifies whether the IOP can run at a high PIN (ON, the default) or only at a low PIN (OFF).

MAGNETIC { ALLOWED | DISALLOWED | REQUIRED }

specifies whether to allow disk volumes in the storage pool

`ALLOWED`	You can add disk volumes (default).
`DISALLOWED`	You can add only nonphysical volumes.
`REQUIRED`	You can add only nonphysical volumes.

MIRRORED { ALLOWED | DISALLOWED | REQUIRED }

specifies whether to allow mirrored volumes in the storage pool

`ALLOWED`	You can add mirrored or nonmirrored volumes (default).
`DISALLOWED`	You can add only nonmirrored volumes.
`REQUIRED`	You can add only mirrored volumes.

MODE { NOISY | QUIET }

specifies whether to generate additional EMS messages for debugging

`NOISY`	Generate additional EMS messages for debugging.
`QUIET`	Generate only essential EMS messages (default).

PRIMARYCPU *number*

is the processor in which the primary IOP starts. The default is PRIMARYCPU 0.

PROGRAM [[$*vol*.]*subvol*.]*fileid*

is the object file name of the IOP. If not specified, PROGRAM is:

```
$SYSTEM.SYSTEM.OPP
```

If you specify PROGRAM as $SYSTEM.SYSTEM, the operating system first searches for the file on $SYSTEM.SYSTEM and on the current SYS*nn* subvolume.

STARTSTATE { STARTED | STOPPED }

specifies whether the IOP is available to other processes (STARTED) or unavailable (STOPPED) when the system is loaded or reloaded. The default is STARTED.

UPDATESTATS *number*

is the interval (in minutes) at which the storage pool process should collect information about the physical volumes in the storage pool. *number* is either -1 (no statistics should collected) or in the range 0 through 1440. The default is 30.

If you specify a very large *number*, statistics are gathered infrequently. Depending on how much virtual disk activity occurs, virtual disk processes could make volume-selection decisions using out-of-date information.

Specifying a small value for *number* results in statistics being gathered frequently but might affect system performance.

ADD POOL Examples

See the procedure and considerations for "Adding a Storage Pool" (page 137).

- To add the storage pool $POOL00 to the system configuration:

  ```
  -> ADD POOL $POOL00, SENDTO STORAGE
  ```

- To add the storage pool $POOL01 and update statistics every 2 minutes:

  ```
  -> ADD POOL $POOL01, SENDTO STORAGE, UPDATESTATS 2
  ```

ADD PROFILE Command

The ADD PROFILE command adds a profile to the system configuration database for use in autoconfiguration of internal disks. The syntax is:

```
ADD [ / OUT file-spec / ]
   PROFILE $ZZSTO.INTERNAL-DISK-groupnum
   , SENDTO STORAGE
   [ , LIKE object ] [ , attribute-spec ]...
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

PROFILE $ZZSTO.INTERNAL-DISK-*groupnum*

is a custom profile for enclosure *groupnum*. *groupnum* cannot have a leading zero.

SENDTO STORAGE

directs the command to the storage subsystem. This attribute is required unless you specified SENDTO STORAGE in a previous ASSUME command.

LIKE *object*

identifies an existing object whose attribute values are copied to the object affected by this command. The object name must currently exist in the system configuration, and the object type must match that of the object affected by this command. If you specify LIKE, any attribute values that you explicitly specify override the attribute values of the LIKE object.

attribute-spec

is one or more "PROFILE Attributes". Some disk attributes, like PRIMARYCPU, are not kept in a profile because they are determined by the location of the disk.

PROFILE Attributes

For a description of these attributes, see "Attribute Descriptions for Disk Commands" (page 198).

```
[ , AUDITTRAILBUFFER number ]
[ , AUTOLABEL { ON | OFF } ]
[ , AUTOREVIVE { ON | OFF } ]
[ , AUTOSELECT { ON | OFF } ]
[ , AUTOSTART { ON | OFF } ]
[ , CAPACITYMISMATCH { ON | OFF } ]
[ , CBPOOLLEN number ]
[ , FASTBULKWRITE { ON | OFF } ]
[ , FSTCACHING { ON | OFF | ENABLED } ]
[ , FULLCHECKPOINTS { DISABLED | ENABLED | FORCED } ]
[ , HALTONERROR number ]
[ , HIGHPIN { ON | OFF } ]
[ , LKIDLONGPOOLLEN number ]
[ , LKTABLESPACELEN number ]
[ , MAXLOCKSPEROCB number ]
[ , MAXLOCKSPERTCB number ]
[ , MIRRORING { OFF | ON } ]
[ , NAMEMASK { STANDARD | name } ]
[ , NONAUDITEDINSERT { ON | OFF } ]
[ , NUMDISKPROCESSES number ]
[ , OSSCACHING { ON | OFF } ]
[ , PHYSVOLSELECT { ON | OFF } ]
[ , POOL { $pool | EXCLUDE } ]
[ , PROGRAM [[$vol.]subvol.]fileid ]
[ , PROTECTDIRECTORY { CHECKPOINT | OFF | SERIAL } ]
[ , RECOVERYTIMEOUT number[ , REVIVEBLOCKS number ]
[ , REVIVEINTERVAL number ]
[ , REVIVEPRIORITY number ]
[ , REVIVERATE number ]
[ , SERIALWRITES { DISABLED | ENABLED } ]
[ , SQLMXBUFFER number ]
[ , STARTSTATE { STARTED | STOPPED } ]
[ , WRITECACHE { DISABLED | ENABLED } ]
```

ADD PROFILE Example

See the procedure for "Creating a Custom Profile" (page 75).

To make a custom profile for all new disks inserted into group 03:

```
-> ADD PROFILE $ZZSTO.INTERNAL-DISK-3, MIRRORING OFF
```

ADD SCSI Command

The ADD SCSI command adds an Open SCSI device to the system configuration. Open SCSI devices also include Open SCSI Fibre Channel devices. The syntax is:

```
ADD [ / OUT file-spec / ] SCSI $SCSI-device
    , SENDTO STORAGE [ , LIKE object ]
    [ , attribute-spec ]...
```

Wild-card characters are not supported.

OUT *file-spec*

directs all SCF output to the specified file.

SCSI $*SCSI-device*

is the name of the Open SCSI device.

SENDTO STORAGE

directs the command to the storage subsystem. This attribute is required unless you specified SENDTO STORAGE in a previous ASSUME command.

LIKE *object*

identifies an existing object whose attribute values are copied to the object affected by this command. The object name must currently exist in the system configuration, and the object type must match that of the object affected by this command. If you specify LIKE, any attribute values that you explicitly specify override the attribute values of the LIKE object.

The location, SAC, and device ID values are not copied from the LIKE object. To assign values to these attributes, you must explicitly specify them.

attribute-spec

is one or more “ADD SCSI Attributes”.

ADD SCSI Attributes

```
[ , BACKUPCPU number ]
[ , BACKUPLOCATION ( group,module,slot ) ]
[ , BACKUPPORTNAME number ]
[ , BACKUPSAC number | name ]
[ , HIGHPIN { ON | OFF } ]
[ , LUN number ]
[ , MAXOPENS number ]
[ , NUMIO number ]
[ , PRIMARYCPU number ]
[ , PRIMARYLOCATION ( group,module,slot ) ]
[ , PRIMARYPORTNAME number ]
[ , PRIMARYSAC number | name ]
[ , PROGRAM [[$vol.]subvol.]fileid ]
[ , RECSIZE number ]
[ , SCSIID number ]
[ , STARTSTATE { STARTED | STOPPED } ]
[ , STRUCTAREASIZE number ]
[ , TRACEBUFLEN number ]
[ , TRACELEVEL number ]
```

BACKUPCPU *number*

is the processor in which the backup IOP starts. If not specified, the processor is automatically chosen based on the current, primary-adapter, processor-access list. The first entry in the processor-access list becomes the primary processor and the second entry becomes the backup processor.

BACKUPLOCATION (*group,module,slot*)

is the location of the adapter that controls the backup path to the device. This attribute has no default value. If this attribute is not specified, the device is not accessible through its backup path.

BACKUPPORTNAME *number*

is the Fibre Channel port name used by the backup path to the Open SCSI device. The number should be entered as a 16 character hexadecimal number, without a leading %H.

BACKUPSAC { *number* | *name* }

is the SAC that controls the backup path to the device

number	is the SAC subdevice number on the adapter. If you specify BACKUPSAC in *number* format, you must also specify BACKUPLOCATION.
name	is the full name of the SAC location. For example:`SNDA.SAC-1.GRP-1.MOD-1.SLOT-53`

HIGHPIN { ON | OFF }

specifies whether the IOP can run at a high PIN (ON, the default) or only at a low PIN (OFF).

LUN *number*

is the logical unit number (LUN) of a device connected to the Open SCSI device. For devices connected to a SCSI controller *number* is in the range 0 through 7. For Fibre Channel devices *number* is in the range 0 through 32767.

MAXOPENS *number*

is the maximum number of concurrent opens allowed for the device. *number* is in the range 1 through 64. The default is 3.

NUMIO *number*

is the number of concurrent I/O operations that the Open SCSI IOP can have outstanding. This value determines the size of the allocated data-buffer area. *number* is in the range 1 through 15. The default is 8.

PRIMARYCPU *number*

is the processor in which the primary IOP starts. If you omit this attribute, the processor is automatically chosen based on the physical location of the device.

PRIMARYLOCATION (*group*,*module*,*slot*)

is the location of the adapter that controls the primary path to the Open SCSI device. This attribute is required.

PRIMARYPORTNAME *number*

is the fiber channel port name used by the primary path to the Open SCSI device. The number should be entered as a 16 character hexadecimal number, without a leading %H.

PRIMARYSAC { *number* | *name* }

is the SAC that controls the primary path to the device

number	is the SAC subdevice number on the adapter.
name	is full name of the adapter location including the SAC number. For example:`SNDA.SAC-1.GRP-1.MOD-1.SLOT-53`

PROGRAM [[*$vol*.]*subvol*.]*fileid*

is the object file name of the IOP. If not specified, PROGRAM is:

`$SYSTEM.SYSTEM.TDSCSI`

If you specify PROGRAM as $SYSTEM.SYSTEM, the operating system first searches for the file on $SYSTEM.SYSTEM and on the current SYS*nn* subvolume.

RECSIZE *number*

is the configured record size for the device (in bytes). This value is used by some utility programs when making requests to the IOP. *number* is in the range 1 through 57344. The default is 4096. Larger block sizes should give better performance.

SCSIID *number*

is the device ID of the Open SCSI device. Each device on the same SCSI interface must have a unique device ID. *number* is in the range 0 through 15 (6 and 7 are reserved for the adapters). The default is 2.

STARTSTATE { STARTED | STOPPED }

specifies whether the IOP is available to other processes (STARTED) or unavailable (STOPPED) when the system is loaded or reloaded. The default is STARTED.

STRUCTAREASIZE *number*

is the size (in kilobytes) of the memory area used for data structures. *number* is in the range 16 through 32. The default is 24.

TRACEBUFLEN *number*

is the size (in kilobytes) of the memory area used for internal tracing by the Open SCSI IOP. *number* is in the range 32 through 65. The default is 33.

TRACELEVEL *number*

is the level of tracing used by the Open SCSI IOP. *number* is in the range 0 through 65,535. The default is 65,535.

ADD SCSI Consideration

Before using the ADD SCSI command, verify the SCSI ID of the device is the same as the configured SCSIID value.

ADD SCSI Example

See the procedure for "Adding an Open SCSI Device" (page 174).

To add an Open SCSI device with a SCSI ID of 3:

```
-> ADD SCSI $DEV0, SENDTO STORAGE, PRIMARYLOCATION(21,1,50),&
-> SCSIID 2, LUN 0, PRIMARYCPU 0, BACKUPCPU 1
```

ADD TAPE Command

The ADD TAPE command adds a tape drive to the system configuration. The syntax is:

```
ADD [ / OUT file-spec / ] TAPE $tape , SENDTO STORAGE
    [ , LIKE object ]  [ attribute-spec ]...
```

Wild-card characters are not supported.

OUT *file-spec*

directs all SCF output to the specified file.

TAPE $*tape*

is the name of the tape device.

SENDTO STORAGE

directs the command to the storage subsystem. This attribute is required unless you specified SENDTO STORAGE in a previous ASSUME command.

LIKE $*object*

identifies an existing object whose attribute values are copied to the object affected by this command. The object name must currently exist in the system configuration, and the object type must match that of the object affected by this command. If you specify LIKE, any attribute values that you explicitly specify override the attribute values of the LIKE object.

The location, SAC, and device ID values are not copied from the LIKE object. To assign values to these attributes, you must explicitly specify them.

attribute-spec

is one or more "ADD TAPE Attributes".

ADD TAPE Attributes

```
[ , BACKUPCPU number ]

[ , COMPRESSION { ON | OFF } ]
[ , DENSITY { 1600 | 6250 } ]
[ , DEVICEID number ]
[ , HIGHPIN { ON | OFF } ]
[ , LOCATION ( group,module,slot ) ]
[ , LUN number ]
[ , MAXOPENS number ]
[ , PRIMARYCPU number ]
[ , PORTNAME number ]
[ , PROGRAM { [$vol.]subvol.]fileid } ]
[ , RECSIZE number ]
[ , SAC { number | name } ]
[ , STARTSTATE { STARTED | STOPPED } ]
```

BACKUPCPU *number*

is the processor in which the backup IOP starts. If you omit this attribute, the processor is automatically chosen based on the physical location of the device.

COMPRESSION { ON | OFF }

(cartridge tapes only) specifies whether data compression is configured

ON	The tape process compresses data using Improved Data Recovery Capability when writing to the cartridge tape (default).
OFF	The tape process does not compress data when writing to the cartridge tape.

This setting can be overridden by the MEDIACOM ALTER TAPEDRIVE MEDIADEFS command. See the discussion of date compression in the *DSM/Tape Catalog User's Guide*.

DENSITY { 1600 | 6250 }

(5170 tape drives only) is the tape density in bpi. (Other tape drives use a standard density which is displayed by the INFO TAPE command.) The default is 6250.

DEVICEID *number*

is the device ID of the tape drive.

This value must match either the SCSI ID of the tape drive (if SCSI-connected) or the device ID of the tape drive (if Fibre Channel connected).

If the tape drive is connected to a PMF CRU, an IOMF CRU, or an SNDA, *number* is in the range 0 through 5. The default is 5.

If the tape drive is connected to an F-SAC of an SNDA, *number* is 4 or 5.

HIGHPIN { ON | OFF }

specifies whether the IOP can run at a high PIN (ON, the default) or only at a low PIN (OFF).

LOCATION (*group*, *module*, *slot*)

is the location of the adapter where the tape drive is connected. This attribute is required.

LUN *number*

is the logical unit number (LUN) used by the path of the tape volume in the range 0 through 31.

MAXOPENS *number*

is the maximum concurrent opens allowed for the device. *number* is in the range 1 through 64. The default is 4.

PORTNAME *number*

is the Fibre Channel port name used by the path to a tape volume. The number should be entered as a 16 character hexadecimal number, without a leading %H.

PRIMARYCPU *number*

is the processor in which the primary IOP starts. If you omit this attribute, the processor is automatically chosen based on the physical location of the device.

PROGRAM [[*$vol*.]*subvol*.]*fileid*

is the object file name of the IOP. If not specified, PROGRAM is:

`$SYSTEM.SYSTEM.OTPPROCP`

If you specify PROGRAM as $SYSTEM.SYSTEM, the operating system first searches for the file on $SYSTEM.SYSTEM and on the current SYS*nn* subvolume.

RECSIZE *number*

is the configured record size for the device (in bytes). This value is used by some utility programs when making requests to the IOP. *number* is in the range 1 through 57344. The default is 4096. Larger block sizes should give better performance.

SAC { *number* | *name* }

is the SAC that controls the primary path to the device

number	is the SAC subdevice number on the adapter.
name	is the full name of the SAC. For example:`SNDA.SAC-1.GRP-1.MOD-1.SLOT-53` .

STARTSTATE { STARTED | STOPPED }

specifies whether the IOP is available to other processes (STARTED) or unavailable (STOPPED) when the system is loaded or reloaded. The default is STARTED.

ADD TAPE Examples

See the procedure for "Adding an Open SCSI Device" (page 174).

- To add a tape drive named $TAPE0 in group 02, slot 55:

```
-> ADD TAPE $TAPE0, SENDTO STORAGE, LOCATION (1,1,55)
```

- To add a tape drive named $TAPE1 that is connected to SAC 1 in the SNDA in group 01, slot 53:

```
-> ADD TAPE $TAPE1, SENDTO STORAGE, LOCATION (1,1,53), &
-> SAC 1, DEVICEID 4
```

ALLOWOPENS Command

The ALLOWOPENS command permits an object to accept opens again; reverses the effect of the STOPOPENS command.

ALLOWOPENS is a sensitive command.

ALLOWOPENS DISK Command

The ALLOWOPENS command permits file opens on a specified physical disk volume.

The syntax is:

```
ALLOWOPENS [ / OUT file-spec / ] DISK $disk
    [ , POOL $pool ]   [ , SEL state ]
    [ , SUB { ALL | MAGNETIC | VIRTUAL } ]
    [ , SUPERONLY ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

DISK $*disk*

is the name of the disk. This disk must be in the STARTED state.

POOL *$pool*

specifies that the command is performed only on physical disks associated with the specified storage pool.

SEL *state*

specifies that the command affects only devices in the specified state.

SUB { ALL | MAGNETIC | VIRTUAL }

specifies that the command affects only disks of the specified type. The default is ALL.

SUPERONLY

specifies that only the super ID (255,255) is allowed to open a file on the specified volume. To allow all users to open files on this volume, enter an ALLOWOPENS command without SUPERONLY.

See "Attribute Descriptions for Disk Commands" (page 198) for descriptions of all attributes for disk commands.

ALLOWOPENS DISK Examples

See "Allowing File Opens on a Disk" (page 109).

- To allow files to be opened by applications on the disk $DATA00:

```
-> ALLOWOPENS $DATA00
```

- To prevent files from being opened by anyone other than super ID (255, 255) on the disk $DATA14:

```
-> ALLOWOPENS $DATA14, SUPERONLY
```

ALTER Command

The ALTER command changes the configuration attributes of an object.

Supported objects are:

- "ALTER DISK Command" (page 228)
- "ALTER MON Command" (page 232)
- "ALTER POOL Command" (page 233)
- "ALTER PROFILE Command" (page 236)
- "ALTER SCSI Command" (page 237)
- "ALTER SUBSYS Command" (page 240)
- "ALTER TAPE Command" (page 241)

ALTER is a sensitive command.

ALTER DISK Command

The ALTER DISK command changes attribute values for an existing physical or virtual disk on your system. The syntax is:

```
ALTER [ / OUT file-spec / ] DISK $disk
[ , LIKE object ]   [ , attribute-spec ]...
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

DISK $*disk*

is the name of the disk.

LIKE *object*

identifies an existing object whose attribute values are copied to the object affected by this command. The object name must currently exist in the system configuration, and the object type must match that of the object affected by this command. If you specify LIKE, any attribute values that you explicitly specify override the attribute values of the LIKE object.

These disk attributes are not copied from the LIKE object. To assign values to these attributes, you must explicitly specify them:

- ALTNAME
- Device ID attributes:
  - BACKUPDEVICEID
  - MBACKUPDEVICEID
  - MIRRORDEVICEID
  - PRIMARYDEVICEID
- LABEL
- Location attributes:
  - BACKUPLOCATION
  - MBACKUPLOCATION
  - MIRRORLOCATION
  - PRIMARYLOCATION
- SAC attributes:
  - BACKUPSAC
  - MBACKUPSAC
  - MIRRORSAC
  - PRIMARYSAC
- PORTNAME attributes:
  - BACKUPPPORTNAME
  - MBACKUPPORTNAME
  - MIRRORPORTNAME
  - PRIMARYPORTNAME

- LUN attributes:
  - MIRRORLUN
  - PRIMARYLUN
- CLIM attributes:
  - BACKUPCLIM
  - MBACKUPCLIM
  - MIRRORCLIM
  - PRIMARYCLIM

These virtual disk attributes are not copied from the LIKE object. To assign values to these attributes, you must explicitly specify them:

- ANTLOCATION
- PENDOPSLOCATION
- POOL

attribute-spec

is one or more “Disk Attributes for the ALTER DISK Command” (page 230) or “Virtual Disk Attributes for the ALTER COMMAND” (page 231).

Disk Attributes for the ALTER DISK Command

These attributes are valid for the ALTER DISK command. For a description of these attributes, see “Attribute Descriptions for Disk Commands” (page 198)

```
[ , ALTNAME $vol ]
[ , AUDITTRAILBUFFER number ]
[ , AUTOREVIVE { ON | OFF } ]
[ , AUTOSELECT { ON | OFF } ]
[ , AUTOSTART { ON | OFF } ]
[ , BACKUPCPU number ]
[ , BACKUPCLIM clim-name ]
[ , BACKUPDEVICEID { number | ( shelf,bay ) } ]
[ , BACKUPLOCATION {group,module,slot) ]
[ , BACKUPPORTNAME number ]
[ , BACKUPSAC { number | name } ]
[ , CACHE ( block-size, num-blocks ) |
    ( ( block-size, num-blocks ),
      ( block-size, num-blocks ),... )]
[ , CAPACITYMISMATCH { ON | OFF } ]
[ , CBPOOLLEN number ]
[ , CLEARENCRYPTKEY
[ , ENCRYPTRATE rate
[ , ENCRYPTPRIORITY priority
[ , FASTBULKWRITE { ON | OFF } ]
[ , FORCED ]
[ , FSTCACHING { ON | OFF | ENABLED} ]
[ , FULLCHECKPOINTS { ENABLED | DISABLED | FORCED } ]
[ , HALTONERROR number ]
[ , HIGHPIN { ON | OFF } ]
[ , IGNOREINCONSISTENCY ]
[ , KEYALGORITHM {XTS-AES | CBC-AES }
[ , KEYSIZE 256
[ , LABEL $vol ]
[ , LKIDLONGPOOLLEN number ]
[ , LKTABLESPACELEN number ]
[ , MAXLOCKSPEROCB number ]
```

```
[ , MAXLOCKSPERTCB number ]
[ , MBACKUPCLIM clim-name ]
[ , MBACKUPDEVICEID { number | ( shelf, bay ) }  ]
[ , MBACKUPLOCATION {group,module,slot) ]
[ , MBACKUPPORTNAME number ]
[ , MBACKUPSAC { number | name } ]
[ , MIRRORCLIM clim-name ]
[ , MIRRORDEVICEID { number | ( shelf,bay ) } ]
[ , MIRRORLOCATION ( group,module,slot ) ]
[ , MIRRORPARTITION mirror-part-num ]
[ , MIRRORLUN number ]
[ , MIRRORPORTNAME number ]
[ , MIRRORSAC { number | name } ]
[ , NEWENCRYPTKEY
[ , NONAUDITEDINSERT { ON | OFF } ]
[ , NOSTART ]
[ , NUMDISKPROCESSES number ]
[ , OSSCACHING { ON | OFF } ]
[ , PHYSVOLSELECT { ON | OFF } ]
[ , POOL { $pool | EXCLUDE } ]
[ , PRIMARYCLIM clim-name ]
[ , PRIMARYCPU number ]
[ , PRIMARYDEVICEID { number | ( shelf,bay ) } ]
[ , PRIMARYLOCATION ( group,module,slot )
[ , PRIMARYPARTITION primary-part-num ]
[ , PRIMARYLUN number ]
[ , PRIMARYPORTNAME number ]
[ , PRIMARYSAC { number | name } ]
[ , PROGRAM [[$vol.]subvol.]fileid ]
[ , PROTECTDIRECTORY { CHECKPOINT | OFF | SERIAL } ]
[ , RECOVERYTIMEOUT number ]
[ , REVIVEBLOCKS number ]
[ , REVIVEINTERVAL number ]
[ , REVIVEPRIORITY number ]
[ , REVIVERATE number ]
[ , SENDTO STORAGE ]
[ , SERIALWRITES { ENABLED | DISABLED } ]
[ , SQLMXBUFFER number ]
[ , STARTSTATE { STARTED | STOPPED } ]
[ , SWAPMIRROR ]
[ , VOLNAME $vol ]
[ , WRITECACHE { DISABLED | ENABLED } ]
[ , WRITEVERIFY { ON | OFF } ]
```

Virtual Disk Attributes for the ALTER COMMAND

These virtual disk attributes are valid for the ALTER DISK command. For a description of these attributes, see “Attribute Descriptions for Disk Commands” (page 198)

```
[ , ANTCAPACITY number ]
[ , BACKUPCPU number ]
[ , CACHESIZE number  ]
[ , HIGHPIN { ON | OFF } ]
[ , MODE { NOISY | QUIET } ]
[ , PRIMARYCPU number ]
[ , PROGRAM [[$vol.]subvol.]fileid ]
[ , STARTSTATE { STARTED | STOPPED } ]
```

ALTER DISK Examples for Physical Disks

See the procedure for “Altering Disk Attribute Values” (page 84).

- To change the alternate volume name of a disk from $DATA00 to $SPARE99:

```
-> ALTER $DATA00, ALTNAME $SPARE99
```

- To change the volume name of a disk from $DATA00 to $SPARE00 and delete all files from the disk:

  ```
  -> ALTER $DATA00, LABEL $SPARE00
  ```

- To change the default volume name of a disk from $DATA00 to $SPARE98:

  ```
  -> ALTER $DATA00, VOLNAME $SPARE98
  ```

- To change the revive configuration for a disk:

  ```
  -> ALTER $DATA03, REVIVEPRIORITY 40, REVIVERATE 100.
  ```

- To alter a disk LDEV offline using the partition path information:

  ```
  ALTER DISK $SSD1 , &
           PRIMARYCLIM S1002531, &
           PRIMARYLUN 101, &
           PRIMARYPARTITION 2 &
           BACKUPCLIM S1002533, &
           MIRRORCLIM S1002533, &
           MIRRORLUN 201, &
           MIRRORPARTITION 2, &
           MBACKUPCLIM S1002531
  ```

- To alter a disk LDEV with online online disk remirroring (ODR) using the partition path information:

  ```
  ALTER DISK $SSD1, &
           MIRRORCLIM S1002533, &
           MIRRORLUN 201, &
           MIRRORPARTITION 4, &
           MBACKUPCLIM S1002531
  ```

Cache Configuration Example

See the procedure for "Configuring the Size of Disk Cache" (page 86).

- To change the cache configuration for a disk:

  ```
  -> ALTER $DATA00, CACHE ( (512,4), (2K,100), (4K,100) )
  ```

- To change the cache configuration of a disk:

  ```
  -> ALTER DISK $DATA00, CACHE ( (512,640), (1K, 800), &
  -> (2K,500), (4K,20400) )
  ```

ALTER DISK Examples for Virtual Disks

See the procedure for "Troubleshooting Problems With Adding a Virtual Disk" (page 148).

- To add a disk to the storage pool $POOL1:

  ```
  -> ALTER $DISK00, POOL $POOL1
  ```

- To allow a disk to be considered for file placement by a pool process:

  ```
  -> ALTER $DISK00, AUTOSELECT ON
  ```

- To move a disk from pool $POOL2 to pool $POOL1:

  ```
  -> ALTER DISK $DISK00, POOL EXCLUDE
  -> ALTER DISK $DISK00, POOL $POOL1
  ```

ALTER MON Command

The ALTER MON command changes the configured attribute values for the SMF master process. The syntax is:

```
ALTER [ / OUT file-spec / ] MON $ZSMS
    [ , attribute-spec ]...
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

MON $ZSMS
: is the SMF master process.

attribute-spec
: is one or more "ALTER MON Attributes" (page 233).

ALTER MON Attributes

```
[ , BACKUPCPU number ]
[ , HIGHPIN { ON | OFF } ]
[ , MODE { NOISY | QUIET } ]
[ , PRIMARYCPU number ]
[ , PROGRAM [[$vol.]subvol.]fileid ]
```

BACKUPCPU *number*
: is the processor in which the backup IOP starts. The process must be in the STOPPED state or not running for you to alter this attribute.

HIGHPIN { ON | OFF }
: specifies whether the IOP can run at a high PIN (ON, the default) or only at a low PIN (OFF).

MODE { NOISY | QUIET }
: specifies whether to generate additional EMS messages for debugging

NOISY	Generate additional EMS messages for debugging.
QUIET	Generate only essential EMS messages (default).

PRIMARYCPU *number*
: is the processor in which the primary IOP starts. The process must be in the STOPPED state or not running for you to alter this attribute.

PROGRAM [[*$vol.*]*subvol.*]*fileid*
: is the object file name of the IOP. If not specified, PROGRAM is:

  `$SYSTEM.SYSTEM.OMP`

  If you specify the program file name location as $SYSTEM.SYSTEM, the operating system first searches for the file there and on the current SYS*nn* subvolume.

ALTER MON Examples

See the procedure for "Altering the Values of the SMF Master Process Attributes" (page 45).

- To change the mode to generate all the EMS messages:

  `-> ALTER MON $ZSMS, MODE NOISY`

- To run the SMF master process in different processors:

  `-> ALTER MON $ZSMS, BACKUPCPU 0, PRIMARYCPU 1`

ALTER POOL Command

The ALTER POOL command changes the configured attribute values for a storage pool. The syntax is:

```
ALTER [ / OUT file-spec / ] POOL $pool
    [ , LIKE object ] [ , attribute-spec ]...
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

POOL $*pool*

is the storage pool process.

LIKE *object*

identifies an existing object whose attribute values are copied to the object affected by this command. The object name must currently exist in the system configuration, and the object type must match that of the object affected by this command. If you specify LIKE, any attribute values that you explicitly specify override the attribute values of the LIKE object.

attribute-spec

is one or more "ALTER POOL Attributes".

ALTER POOL Attributes

```
[ , AUDITED { ALLOWED | DISALLOWED | REQUIRED } ]
[ , BACKUPCPU number ]
[ , DISKINTERVAL number ]
[ , DISKTHRESHOLD number ]
[ , EXTENTINTERVAL number ]
[ , EXTENTTHRESHOLD number ]
[ , HIGHPIN { ON | OFF } ]
[ , MAGNETIC { ALLOWED | DISALLOWED | REQUIRED } ]
[ , MIRRORED { ALLOWED | DISALLOWED | REQUIRED } ]
[ , MODE { NOISY | QUIET } ]
[ , PRIMARYCPU number ]
[ , PROGRAM [[$vol.]subvol.]fileid ]
[ , STARTSTATE { STARTED | STOPPED } ]
[ , UPDATESTATS number ]
```

AUDITED { ALLOWED | DISALLOWED | REQUIRED }

is the type of volume you can add to the storage pool.

ALLOWED	You can add any volume, even if it is not protected by TMF (default).
DISALLOWED	You can add a volume only if it is not protected by TMF.
REQUIRED	You can add a volume only if it is protected by TMF.

BACKUPCPU *number*

is the processor in which the backup IOP starts. The process must be in the STOPPED state or not running for you to alter this attribute.

DISKINTERVAL *number*

is the interval (as a percentage) that, with the DISKTHRESHOLD attribute, is used by the disk process to determine when to issue a disk-full EMS message for physical volumes in the storage pool. For example, if DISKTHRESHOLD is 70 and DISKINTERVAL is 5, the disk process issues an EMS message when the disk becomes 70, 75, 80, 85, 90, and 95 percent full.

number is either -1 (the default, no disk-full EMS messages are issued) or in the range 1 through 99 percent.

This attribute has no effect if DISKTHRESHOLD is -1.

DISKTHRESHOLD *number*

is the threshold (as a percentage) that, with the DISKINTERVAL attribute, is used by the disk process to determine when to issue a disk-full EMS message for physical volumes in the storage pool. For example, if DISKTHRESHOLD is 75, the disk process generates an EMS message when the physical volume becomes 75 percent full. If you also specify DISKINTERVAL, the disk process generates an EMS message when the physical volume becomes 75, 80, 85, 90, and 95 percent full.

`number` is either -1 (the default, no disk-full EMS messages are issued) or in the range 1 through 100 percent.

EXTENTINTERVAL *number*

is the number of extents that, with the EXTENTTHRESHOLD attribute, is used by the disk process to determine when to issue an extents-left EMS message for physical volumes in the storage pool. For example, if EXTENTTHRESHOLD is 6 and EXTENTINTERVAL is 2, the disk process issues an EMS message when there are 6, 4, 2, and 0 extents left to be allocated for a file.

`number` is either -1 (the default, no extents-left EMS messages are issued) or in the range 1 through 978.

This attribute has no effect if EXTENTTHRESHOLD is -1.

EXTENTTHRESHOLD *number*

is the number of extents that, with the EXTENTINTERVAL attribute, is used by the disk process to determine when to issue an extents-left EMS message for physical volumes in the storage pool. For example, if EXTENTTHRESHOLD is 6, the disk process issues an EMS message when only six extents remain to be allocated to a file. If you also specify EXTENTINTERVAL, the disk process issues an EMS message when there are 4, 2, and 0 extents left to be allocated for a file.

`number` is either -1 (the default, no extents-left EMS messages are issued) or in the range 1 through 978.

HIGHPIN { ON | OFF }

specifies whether the IOP can run at a high PIN (ON, the default) or only at a low PIN (OFF).

MAGNETIC { ALLOWED | DISALLOWED | REQUIRED }

specifies whether to allow physical volumes in the storage pool

`ALLOWED`	You can add volumes (default).
`DISALLOWED`	You can add only nonphysical volumes.
`REQUIRED`	You can add only nonphysical volumes.

MIRRORED { ALLOWED | DISALLOWED | REQUIRED }

specifies whether to allow mirrored volumes in the storage pool

`ALLOWED`	You can add mirrored or nonmirrored volumes (default).
`DISALLOWED`	You can add only nonmirrored volumes.
`REQUIRED`	You can add only mirrored volumes.

MODE { NOISY | QUIET }

specifies whether to generate additional EMS messages for debugging

`NOISY`	Generate additional EMS messages for debugging.
`QUIET`	Generate only essential EMS messages (default).

PRIMARYCPU *number*

is the processor in which the primary IOP starts. The process must be in the STOPPED state or not running for you to alter this attribute.

PROGRAM [[*$vol*.]*subvol*.]*fileid*

is the object file name of the IOP. If not specified, PROGRAM is:

```
$SYSTEM.SYSTEM.OPP
```

If you specify the program file name location as $SYSTEM.SYSTEM, the operating system first searches for the file there and on the current SYS*nn* subvolume.

STARTSTATE { STARTED| STOPPED }

specifies whether the IOP is available to other processes (STARTED) or unavailable (STOPPED) when the system is loaded or reloaded. The default is STARTED.

UPDATESTATS *number*

is the interval (in minutes) at which the storage pool process should collect information about the physical volumes in the storage pool. *number* is either -1(no statistics should collected) or in the range 0 through 1440. The default is 30.

If you specify a very large *number*, statistics are gathered infrequently. Depending on how much virtual disk activity occurs, virtual disk processes could make volume-selection decisions using out-of-date information.

Specifying a small value for *number* results in statistics being gathered frequently but might affect system performance.

ALTER POOL Examples

See the procedure for "Altering Storage Pool Attribute Values" (page 139).

- To change when a pool process provides disk-full information:

  ```
  -> ALTER $POOL01, DISKINTERVAL 10, DISKTHRESHOLD 80
  ```

- To change how often a pool process collects information about physical volumes in the pool:

  ```
  -> ALTER $POOL00, UPDATESTATS 3
  ```

ALTER PROFILE Command

The ALTER PROFILE command changes a disk configuration profile. The syntax is:

```
ALTER [ / OUT file-spec / ]
   PROFILE $ZZSTO.INTERNAL-DISK[-groupnum ]
   [ , attribute-spec ]...
```

Wild-card characters are supported.

OUT *file-spec*	directs all SCF output to the specified file.
PROFILE $ZZSTO.*INTERNAL-DISK* [-*groupnum*]	is either the standard default profile, $ZZSTO.#INTERNAL-DISK, or a user-created custom profile for enclosure *groupnum*. *groupnum* cannot have a leading zero.
attribute-spec	is one or more "PROFILE Attributes".

PROFILE Attributes

For a description of these attributes, see "Attribute Descriptions for Disk Commands" (page 198).

```
[ , AUDITTRAILBUFFER number ]
[ , AUTOLABEL { ON | OFF } ]
[ , AUTOREVIVE { ON | OFF } ]
[ , AUTOSELECT { ON | OFF } ]
```

```
[ , AUTOSTART { ON | OFF } ]
[ , CAPACITYMISMATCH { ON | OFF } ]
[ , CBPOOLLEN number ]
[ , FASTBULKWRITE { ON | OFF } ]
[ , FSTCACHING { ON | OFF | ENABLED } ]
[ , FULLCHECKPOINTS { DISABLED | ENABLED | FORCED } ]
[ , HALTONERROR number ]
[ , HIGHPIN { ON | OFF } ]
[ , LKIDLONGPOOLLEN number ]
[ , LKTABLESPACELEN number ]
[ , MAXLOCKSPEROCB number ]
[ , MAXLOCKSPERTCB number ]
[ , MIRRORING { OFF | ON } ]
[ , NAMEMASK { STANDARD | name } ]
[ , NONAUDITEDINSERT { ON | OFF } ]
[ , NUMDISKPROCESSES number ]
[ , OSSCACHING { ON | OFF } ]
[ , PHYSVOLSELECT { ON | OFF } ]
[ , POOL { $pool | EXCLUDE } ]
[ , PROGRAM [[$vol.]subvol.]fileid ]
[ , PROTECTDIRECTORY { CHECKPOINT | OFF | SERIAL } ]
[ , RECOVERYTIMEOUT number ]
[ , REVIVEBLOCKS number ]
[ , REVIVEINTERVAL number ]
[ , REVIVEPRIORITY number ]
[ , REVIVERATE number ]
[ , SERIALWRITES { DISABLED | ENABLED } ]
[ , SQLMXBUFFER number ]
[ , STARTSTATE { STARTED | STOPPED } ]
[ , WRITECACHE { DISABLED | ENABLED } ]
```

ALTER PROFILE Examples

See the procedure for "Altering a Profile" (page 76).

- To modify the standard default profile for all internal disks in the system:

  ```
  -> ALTER PROFILE $ZZSTO.#INTERNAL-DISK, NAMEMASK DATA
  ```

- To change a profile for all disks inserted into group 03:

  ```
  -> ALTER PROFILE $ZZSTO.#INTERNAL-DISK-3, MIRRORING OFF
  ```

ALTER SCSI Command

The ALTER SCSI command changes the configured attribute values for the Open SCSI devices attached to your system. The syntax is:

```
ALTER [ / OUT file-spec / ] SCSI $SCSI-device
   [ , LIKE object ] [ , attribute-spec ]...
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

SCSI $*SCSI-device*

is the name of the Open SCSI device.

LIKE *object*

identifies an existing object whose attribute values are copied to the object affected by this command. The object name must currently exist in the system configuration, and the object type must match that of the object affected by this command. If you specify LIKE, any attribute values that you explicitly specify override the attribute values of the LIKE object.

The location, SAC, and device ID values are not copied from the LIKE object. To assign values to these attributes, you must explicitly specify them.

attribute-spec

is one or more "ALTER SCSI Attributes".

ALTER SCSI Attributes

```
[ , BACKUPCPU number ]
[ , BACKUPLOCATION ( group, module, slot ) ]
[ , BACKUPPORTNAME number ]
[ , BACKUPSAC number | name ]
[ , HIGHPIN { ON | OFF } ]
[ , LUN number ]
[ , MAXOPENS number ]
[ , NUMIO number ]
[ , PRIMARYCPU number ]
[ , PRIMARYLOCATION ( group,module,slot ) ]
[ , PRIMARYPORTNAME number ]
[ , PRIMARYSAC number | name ]
[ , PROGRAM [[$vol.]subvol.]fileid ]
[ , RECSIZE number ]
[ , SCSIID number ]
[ , STARTSTATE { STARTED | STOPPED } ]
[ , STRUCTAREASIZE number ]
[ , TRACEBUFLEN number ]
[ , TRACELEVEL number ]
```

BACKUPCPU *number*

is the processor in which the backup IOP starts. The process must be in the STOPPED state or not running for you to alter this attribute.

BACKUPLOCATION (*group,module,slot*)

is the location of the adapter that controls the backup path to the device. This attribute has no default value. If this attribute is not specified, the device is not accessible through its backup path.

BACKUPPORTNAME *number*

is the Fibre Channel port name used by the backup path to the Open SCSI device. The number should be entered as a 16-character hexadecimal number, without a leading %H.

BACKUPSAC { *number* | *name* }

is the SAC that controls the backup path to the device

number	is the SAC subdevice number on the adapter. If you specify BACKUPSAC in *number* format, you must also specify BACKUPLOCATION.
name	is the full name of the SAC location. For example:`SNDA.SAC-1.GRP-21.MOD-1.SLOT-53`

HIGHPIN { ON | OFF }

specifies whether the IOP can run at a high PIN (ON, the default) or only at a low PIN (OFF).

LUN *number*

is the logical unit number (LUN) of a device connected to the Open SCSI device. For devices connected to a SCSI controller *number* is in the range 0 through 7. For Fibre Channel devices *number* is in the range 0 through 32767.

MAXOPENS *number*

is the maximum concurrent opens allowed for the device. *number* is in the range 1 through 64. The default is 4.

NUMIO *number*

is the number of concurrent I/O operations that the Open SCSI IOP can have outstanding. This value determines the size of the allocated data-buffer area. *number* is in the range 1 through 15. The default is 8.

PRIMARYCPU *number*

is the processor in which the primary IOP starts. The process must be in the STOPPED state or not running for you to alter this attribute.

PRIMARYLOCATION (*group*,*module*,*slot*)

is the location of the adapter that controls the primary path to the Open SCSI device.

PRIMARYPORTNAME *number*

is the Fibre Channel port name used by the primary path to the Open SCSI device. The number should be entered as a 16-character hexadecimal number, without a leading %H.

PRIMARYSAC { *number* | *name* }

is the SAC that controls the primary path to the device.

number	is the SAC subdevice number on the adapter. If you specify PRIMARYSAC in number format, you must also specify PRIMARYLOCATION.
name	is full name of the adapter location including the SAC number. For example:`SNDA.SAC-1.GRP-1.MOD-1.SLOT-53`

PROGRAM [[$*vol*.]*subvol*.]*fileid*

is the object file name of the IOP. If not specified, PROGRAM is:

$SYSTEM.SYSTEM.TDSCSI

If you specify the program file name location as $SYSTEM.SYSTEM, the operating system first searches for the file there and on the current SYS*nn* subvolume.

RECSIZE *number*

is the configured record size for the device (in bytes). This value is used by some utility programs when making requests to the IOP. *number* is in the range 1 through 57344. The default is 4096. Larger block sizes should give better performance.

SCSIID *number*

is the device ID of the Open SCSI device. Each device on the same SCSI interface must have a unique device ID. *number* is in the range 0 through 15 (6 and 7 are reserved for the adapters). The default is 2.

STARTSTATE { STARTED | STOPPED }

specifies whether the IOP is available to other processes (STARTED) or unavailable (STOPPED) when the system is loaded or reloaded. The default is STARTED.

STRUCTAREASIZE *number*

is the size (in kilobytes) of the memory area used for data structures. *number* is in the range 16 through 32. The default is 24.

TRACEBUFLEN *number*

is the size (in kilobytes) of the memory area used for internal tracing by the Open SCSI IOP. *number* is in the range 32 through 65. The default is 33.

TRACELEVEL *number*

is the level of tracing used by the Open SCSI IOP. *number* is in the range 0 through 65,535. The default is 65,535.

ALTER SCSI Example

See the procedure for "Altering Open SCSI Attribute Values" (page 176).

To change the value of the SCSIID attribute:

```
-> ALTER $DEV1, SCSIID 5
```

ALTER SUBSYS Command

The ALTER SUBSYS command changes attributes of the storage subsystem manager. The syntax is:

```
ALTER [ / OUT file-spec / ] SUBSYS $ZZSTO
   [ , attribute-spec ]...
```

Wild card characters are not supported.

OUT *file-spec*

directs all SCF output to the specified file.

SUBSYS $ZZSTO

is the storage subsystem manager.

attribute-spec

is one or more "ALTER SUBSYS Attributes".

ALTER SUBSYS Attributes

```
[ , AUTOCONFIGURE { ON | OFF } ]
[ , AUTOREVIVE { ON | OFF } ]
[ , AUTOSTART { ON | OFF } ]
[ , BULKIO { ON | OFF } ]
[ , LABELTAPE { ON | OFF } ]
[ , UPS { ON | OFF } ]
```

AUTOCONFIGURE { ON | OFF }

specifies whether the storage subsystem performs automatic configuration of nonconfigured internal storage devices when they are discovered.

ON	Enables automatic configuration.
OFF	Disables automatic configuration (default).

For information about using this attribute, see "Automating Disk Configuration" (page 73).

AUTOREVIVE { ON | OFF }

(mirrored disks only) specifies whether to automatically start a revive operation on a mirrored volume either when a new disk is inserted or when the system is loaded and a mirrored volume is not synchronized

ON	Automatically starts a revive operation using the values for the REVIVEPRIORITY and REVIVERATE attributes.
OFF	Does not automatically start a revive operation (default).

For information about using this attribute, see "Configuring Internal Disks to Start Automatically" (page 74).

AUTOSTART { ON | OFF }

(internal and M8*xxx* disks only) specifies whether to automatically start the disk process when the disk is inserted.

ON	Enable automatic starting (default).
OFF	Disable automatic starting.

DISK AUTOSTART ON is ignored if SUBSYS AUTOSTART is OFF.

For information about using this attribute, see "Mirrored Disk Placement" (page 77).

BULKIO { ON | OFF }

specifies whether the S-series system allows direct bulk I/O operations. The default is ON.

Direct bulk I/O starts automatically and is transparent to all applications. It improves system performance during disk and tape bulk data transfers. By using direct bulk I/O, data can be transferred directly over ServerNet paths between an application data buffer (in any processor) and the storage controller under direction from the controlling IOP that is running in the same or a different processor.

Without direct bulk I/O, application data flows through an intermediate buffer in the processor of the controlling IOP. This path might take longer to transfer bulk data.

The BULKIO attribute applies only to S-series systems. You cannot set it to ON in HP Integrity NonStop systems or HP Integrity BladeSystems . If you set it to ON, the command will fail and the value in the CONFIG file will remain BULKIO OFF.

LABELTAPE { ON | OFF }

specifies whether the system allows labeled-tape processing. The default is OFF. When you enable or disable labeled-tape processing, you must:

- Manually manage the tape server process $ZSVR.
- First stop all tape drives.

See the procedure for "Enabling or Disabling Labeled-Tape Processing" (page 188).

UPS { ON | OFF }

specifies whether all disk devices are powered by uninterruptible power supplies. If ON, all disk devices must be powered by one or more uninterruptible power supplies. The default of this attribute is determined by the value of the DP2_UPSOPTION flag from the system globals.

ALTER SUBSYS Examples

- To set BULKIO on:

```
-> ALTER $ZZSTO, BULKIO ON
```

- To turn on labeled-tape processing:

```
-> ALTER $ZZSTO, LABELTAPE ON
```

- To make internal disks plug and play:

```
-> ALTER $ZZSTO, AUTOCONFIGURE ON, AUTOSTART ON, &
-> AUTOREVIVE ON
```

See "Automating Disk Configuration" (page 73).

ALTER TAPE Command

The ALTER TAPE command changes the configured attribute values for a tape device attached to your system. The syntax is:

```
ALTER [ / OUT file-spec / ] TAPE $tape
   [ , LIKE object ] [ , attribute-spec ]...
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

TAPE $*tape*

is the name of the tape device.

LIKE *object*

identifies an existing object whose attribute values are copied to the object affected by this command. The object name must currently exist in the system configuration, and the object type must match that of the object affected by this command. If you specify LIKE, any attribute values that you explicitly specify override the attribute values of the LIKE object.

The LOCATION, SAC, and DEVICEID values are not copied from the LIKE object. To assign values to these attributes, you must explicitly specify them.

attribute-spec

is one or more "ALTER TAPE Attributes".

ALTER TAPE Attributes

```
[ , BACKUPCPU number ]
[ , COMPRESSION { ON | OFF } ]
[ , DENSITY { 1600 | 6250 } ]
[ , DEVICEID number ]
[ , HIGHPIN { ON | OFF } ]
[ , KEYGENPOLICY policy[ , LOCATION (group,module,slot) ]
[ , LUN number ]
[ , MAXOPENS number ]
[ , NEWENCRYPTKEY[ , PORTNAME number ]
[ , PRIMARYCPU number ]
[ , PROGRAM [[$vol.]subvol.]fileid ]
[ , RECSIZE number ]
[ , SAC { number | name } ]
[ , STARTSTATE { STARTED | STOPPED } ]
```

BUFFERING { ON | OFF }

for tapes connected through a CLIM, allows tape processes to buffer multiple TAPE WRITE commands, which in most cases results in improved performance. The default is OFF.

BACKUPCPU *number*

is the processor in which the backup IOP starts. The process must be in the STOPPED state or not running for you to alter this attribute.

COMPRESSION { ON | OFF }

(cartridge tapes only) specifies whether data compression is configured

ON	The tape process compresses data using Improved Data Recovery Capability when writing to the cartridge tape (default).
OFF	The tape process does not compress data when writing to the cartridge tape.

This setting can be overridden by the MEDIACOM ALTER TAPEDRIVE MEDIADEFS command. See the discussion on date compression in the *DSM/Tape Catalog User's Guide*.

DENSITY { 1600 | 6250 }

(5170 tape drives only) is the tape density in bpi. (Other tape drives use a standard density which is displayed by the INFO TAPE command.) The default is 6250.

DEVICEID *number*

is the device ID of the tape drive.

This value must match either the SCSI ID of the tape drive (if SCSI-connected) or the device ID of the tape (if Fibre Channel connected).

If the tape drive is connected to a PMF CRU, an IOMF CRU, or an SNDA, *number* is in the range 0 through 5. The default is 5.

If the tape drive is connected to an F-SAC of an SNDA, *number* is 4 or 5.

HIGHPIN { ON | OFF }

specifies whether the IOP can run at a high PIN (ON, the default) or only at a low PIN (OFF).

KEYGENPOLICY { KEYPERDRIVE | KEYPERTAPE | NOENCRYPTION

specifies the key generation policy for the tape drive while it is stopped. You cannot use this attribute in the same command line as the NEWENCRYPTKEY attribute.

NOTE: Only members of the SAFEGUARD security officer group on the local system can perform an ALTER TAPE command with the KEYGENPOLICY attribute.

LOCATION (*group*, *module*, *slot*)

is the location of the adapter where the tape drive is connected.

LUN *number*

is the logical unit number (LUN) used by the path of the tape volume in the range 0 through 31.

MAXOPENS *number*

is the maximum number of concurrent opens allowed for the device. *number* is in the range 1 through 64. The default is 4.

NEWENCRYPTKEY

sets a new encryption key on a tape drive while it is stopped. The drive's KEYGENPOLICY must be set to KeyPerDrive.

NOTE: This attribute is not valid for the ADD TAPE command and the PROFILE object.

PORTNAME *number*

is the Fibre Channel port name used by the path to a tape volume. The number should be entered as a 16-character hexadecimal number, without a leading %H.

PRIMARYCPU *number*

is the processor in which the primary IOP starts. The process must be in the STOPPED state or not running for you to alter this attribute.

PROGRAM [[$*vol*.]*subvol*.]*fileid*

is the object file name of the IOP. If not specified, PROGRAM is:

`$SYSTEM.SYSTEM.OTPPROCP`

If you specify the program file name location as $SYSTEM.SYSTEM, the operating system first searches for the file there and on the current SYS*nn* subvolume.

RECSIZE *number*

is the configured record size for the device (in bytes). This value is used by some utility programs when making requests to the IOP. *number* is in the range 1 through 57344. The default is 4096. Larger block sizes should give better performance.

SAC { *number* | *name* }

is the SAC that controls the primary path to the device

number	Is the SAC subdevice number on the adapter.
name	Is the full name of the SAC; for example:`SNDA.SAC-1.GRP-1.MOD-1.SLOT-53`

STARTSTATE { STARTED | STOPPED }

specifies whether the IOP is available to other processes (STARTED) or unavailable (STOPPED) when the system is loaded or reloaded. The default is STARTED.

ALTER TAPE Examples

See the procedure for "Altering Tape Drive Attribute Values" (page 187).

- To alter the data compression attribute:

```
-> ALTER $TAPE0, COMPRESSION ON
```

CONTROL Command

CONTROL is a sensitive command.

CONTROL DISK Command

The CONTROL command allows the user to perform these tasks:

Task	Attribute
Calculate a checksum	CHECKSUM
Rebuild the free-space table	REBUILDDFS
Power a disk on or off	POWER
	POWEROFF
	POWERON
Refresh cache pages	REFRESH
Replace the bootstrap program	REPLACEBOOT
Manually spare a sector	SPARE

```
CONTROL [ /OUT file-spec/ ] DISK $disk[-P |-M ]
    [ , POOL $pool ]   [ , SEL state ]
    [ , SUB { ALL | MAGNETIC} ]
    [ , attribute-spec ]...
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

DISK $*disk*[-P |-M]

is the disk name and, optionally, the path (primary or mirror).

POOL $*pool*

specifies that the command is performed only on disks associated with the specified storage pool.

SEL *state*

specifies that the command affects only devices in the specified state.

SUB [{ ALL | MAGNETIC }]

specifies that the command affects only disks of the specified type.

attribute-spec

is one or more "CONTROL DISK Attributes".

CONTROL DISK Attributes

```
[ , CHECKSUM [ % | %H ] lsa ]
[ , POWER { ON | OFF } ]
[ , POWEROFF ]
[ , POWERON ]
[ , REBUILDDFS ]
```

```
[ , REFRESH ]
[ , REPLACEBOOT [[$vol.]subvol.]fileid ]
[ , SPARE [ % | %H ] lsa [ , FORCED ] ]
```

CHECKSUM [s% | %H] *lsa*

(physical disks only) reads the specified sector, recomputes the sector checksum value for specified portions of a disk volume, and writes the sector back to disk again. (By comparison, FUP CHECKSUM recomputes the block checksum.) CHECKSUM corrects the checksum value for a disk address, even if the address is not within a file. This attribute is normally used only by a service provider

`[ % | %H ] lsa`	Is the logical sector address of the first sector of the disk portion whose checksum is to be calculated. The logical sector address can be expressed as an octal (%), hexadecimal (%H), or decimal number.

Close all files, but leave the disk in the started state, before you issue this command. Because the checksum operation can leave data on the volume in an inconsistent state, SCF asks you to verify the request. Report the original error message (that prompted you to use the checksum operation) to your service provider.

POWER { ON | OFF }

powers a disk drive on or off. Specifying ON powers the disk drive on. Specifying OFF powers the disk drive off. All paths to the disk drive (-P and -B or -M and -MB) must be in a STOPPED state before you can use this command option. When you use this attribute, you can specify only the -P path or the-M path

NOTE: You cannot turn the power on or off for M8*xxx* disks using this attribute.

POWEROFF

powers the disk drive off. All paths to the disk drive (-P and -B or –M and -MB) must be in a STOPPED state before you can use this command option. When you use this attribute, you can specify only the -P path or the-M path

NOTE: You cannot turn the power off for M8*xxx* disks with this attribute.

POWERON

powers the disk drive on. All paths for the disk drive (-P and -B or -M and -MB) must be in a STOPPED state before you can use this command option. When you use this attribute, you can specify only the -P path or the-M path.

NOTE: You cannot turn the power on for M8*xxx* disks with this attribute.

REBUILDDFS

(physical disks only) rebuilds the disk free-space table.

REFRESH

(physical disks only) writes to disk the dirty (changed) cache pages for nonaudited files and dirty file control blocks. You should issue a CONTROL DISK, REFRESH command before:

- Shutting the system down.
- Backing up open disk files to tape (when the application must be kept running). Do a refresh when no transactions are taking place on the system. When the files are backed up, the file labels more accurately represent the state of the files that are backed up. Back up audited files with the TMF Online Dump to ensure data consistency.

The time a refresh operation takes depends on the amount of disk cache containing dirty pages in use at the time. Writing to disk can take several minutes. During this time all other disk I/O is suspended, which means that applications can time out waiting for the refresh operation to finish.

REPLACEBOOT $SYSTEM.SYS*nn*.SYSDISC

(physical disks only) replaces the disk system-load bootstrap program.

$SYSTEM.SYS*nn*.SYSDISC is the location of the disk bootstrap program. SCF creates the temporary work file ZSYSDISC.ZSCFDISC during the REPLACEBOOT operation, installs the bootstrap program in a reserved area, and when the operation is finished, purges the ZSYSDISC.ZSCFDISC file.

When you use this attribute, you cannot specify a path.

When you initiate a REPLACEBOOT operation, SCF puts one half of the mirrored volume HARDDOWN before replacing the bootstrap file. The other half must be STARTED.

If the specified bootstrap file and the processor type are incompatible, an error message is generated and the operation is not performed.

During the REPLACEBOOT operation, the specified disk might not contain enough space for the bootstrap program, or the currently allocated space might not be enough for the new bootstrap program. If so, you must either compress disk space using DCOM or delete extraneous files to provide enough space for the new bootstrap program.

If an I/O failure or system failure occurs during the REPLACEBOOT operation, the disk bootstrap program might no longer be valid and you would not be able to load the system from that disk. If no other system disk has a valid bootstrap program, contact your service provider.

To avoid this situation and aid recovery, follow the procedures described in "Replacing the Bootstrap Program" (page 107), which describes how to handle mirrored and nonmirrored disks while replacing the bootstrap program.

NOTE: The REPLACEBOOT attribute does not apply to Integrity NonStop NS-series servers. Instead, the bootstrap program is installed through firmware update into flash memory.

SPARE [% | %H] *lsa* [, FORCED]

(physical disks only) assigns an alternate sector to be used in place of a defective sector and enters the address of the defective sector into the added defect map

`[%	%H] lsa`	is the logical sector address for the spare operation. This address can be expressed as an octal (%), hexadecimal (%H), or decimal number. You can omit leading zeroes.
`FORCED`	specifies that the sector should be spared without checking the condition of the sector.

See the procedure for "Sparing a Sector Manually" (page 105). Before sparing a defective sector, verify the symptoms are not caused by a disk hardware problem.

CONTROL DISK Examples

- To correct an error for the sector that begins at logical sector address %H57342:

```
-> CONTROL $DATA00, CHECKSUM %H57342
```

- To rebuild the free-space table:

```
-> CONTROL $DATA01, REBUILDDFS
```

- To replace the system bootstrap code on the primary half of $SYSTEM:

```
-> CONTROL $SYSTEM, REPLACEBOOT $SYSTEM.SYSnn.SYSDISC
```

- To spare the sector that has logical sector address %H0000795C on the mirror half of a volume:

```
-> CONTROL $DATA01-M, SPARE %H795C
```

CONTROL SAC Command

For M8*xxx* FCDMs connected to an FCSA, the CONTROL SAC command is used to issue commands to disks connected to the SAC.

The syntax is:

```
CONTROL [ /OUT file-spec/ ]
  SAC $ZZSTO.#sac-name, DEVICEID (shelf, bay)
           [ , attribute-spec ] ...
```

OUT *file-spec*

directs all SCF output to the specified file.

#sac-name

is the name of the SAC in the form #FCSA.SAC-*sac*.GRP-*group*.MOD-*module*.SLOT-*slot*. For example:

#FCSA.SAC-2.GRP-110.MOD-2.SLOT-5

DEVICEID (*shelf, bay*)

specifies the enclosure and disk to which the command applies:

shelf	Is the FCDM's shelf number. *shelf* is in the range 1 through 4.
bay	Is the number of the disk within the enclosure. *bay* is in the range 1 through 14.

attribute-spec

is one or more "CONTROL SAC Attributes"

CONTROL SAC Attributes

```
[ , BYPASS { OFF | ON } ]
[ , BYPASSBOTH { OFF | ON } ]
[ , LED { OFF | ON } ]
[ , SPINDOWN ]
[ , SPINUP ]
```

BYPASS { OFF | ON }

causes the specified disk to be bypassed (ON) or not bypassed (OFF) on the loop connected to the SAC.

NOTE: Disk paths affected by BYPASS ON must already be in the STOPPED state, with a substate of either DOWN or HARDDOWN.

BYPASSBOTH { OFF | ON }

causes the specified disk to be bypassed (ON) or not bypassed (OFF) on both loops.

NOTE: Disk paths affected by BYPASSBOTH ON must already be in the STOPPED state, with a substate of either DOWN or HARDDOWN.

LED { OFF | ON }

turns the LED on the specified disk ON or OFF.

SPINDOWN

causes the specified disk to be spun down.

NOTE: Disk paths affected by SPINDOWN must already be in the STOPPED state, with a substate of either DOWN or HARDDOWN.

SPINUP

causes the specified disk to be spun up.

CONTROL SAC Examples

- This command turns on the LED for disk 13 in enclosure 1 connected to SAC 2:

```
-> CONTROL SAC $ZZSTO.#FCSA.SAC-2.GRP-110.MOD-2.SLOT-5, &
-> DEVICEID (1,13), LED ON
```

- This command bypasses disk 12 in enclosure 4 on the Fibre Channel loop connected to SAC 1:

```
-> CONTROL SAC $ZZSTO.#FCSA.SAC-1.GRP-110.MOD-2.SLOT-5, &
-> DEVICEID (4,12), BYPASS ON
```

DELETE Command

The DELETE command removes objects from the system configuration database.

Supported objects are:

- "DELETE ADAPTER Command" (page 248)
- "DELETE DISK Command" (page 248)
- "DELETE MON Command" (page 249)
- "DELETE PARTITION Command" (page 250)
- "DELETE POOL Command" (page 251)
- "DELETE PROFILE Command" (page 251)
- "DELETE SCSI Command" (page 252)

DELETE is a sensitive command.

DELETE ADAPTER Command

The DELETE ADAPTER command removes an adapter from the system configuration database. The syntax is:

```
DELETE [ / OUT file-spec / ]
   ADAPTER $ZZSTO.#type.GRP-g.MOD-m.SLOT-s
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

ADAPTER $*ZZSTO*.#*type*.GRP-*g*.MOD-*m*.SLOT-*s*

is a ServerNet adapter of type PMF, IOMF, or SNDA. *g* cannot have a leading zero.

DELETE ADAPTER Example

See the procedure for "Deleting an Adapter" (page 166).

To delete the adapter that is connected to the PMF CRU in group 02, slot 55:

```
-> DELETE ADAPTER $ZZSTO.#PMF.GRP-2.MOD-1.SLOT-55
```

DELETE DISK Command

The DELETE DISK command removes a physical or virtual disk from the system configuration database. The syntax is:

```
DELETE [ / OUT file-spec / ]  DISK $disk[-B |-M |-MB ]
```

```
[ , IGNOREINCONSISTENCY ] [ , POOL $pool ]
[ , SEL state ] [ , SENDTO STORAGE ]
[ , SUB { ALL | MAGNETIC | VIRTUAL }
```

Wild-card characters are not supported.

OUT *file-spec*
: directs all SCF output to the specified file.

DISK $*disk* [-B |-M |-MB]
: is the disk name and, optionally, the path (backup, mirror, or mirror backup).

IGNOREINCONSISTENCY
: specifies that a command can create an inconsistency between the system configuration database and the SMF catalogs maintained by the SMF manager process, storage pools, and virtual disks.

  Use this attribute only if you understand SMF architecture and know how to restore consistency between the system configuration database and the SMF catalogs. The Softdoc for the T1083 product describes several inconsistencies and the procedures for reconciling them.

POOL *$pool*
: specifies that the command is performed only on disks associated with the specified storage pool.

SEL *state*
: specifies that the command affects only devices in the specified state.

SENDTO STORAGE
: specifies that the command affects only disks of the specified type. The default is ALL.

SUB { ALL | MAGNETIC | VIRTUAL }
: specifies that the command affects only disks of the specified type. If not specified, `SUB ALL` is assumed.

See "Attribute Descriptions for Disk Commands" (page 198) for descriptions of all attributes for disk commands.

DELETE DISK Example

See the procedure for "Deleting a Disk" (page 93) or the procedure for "Deleting a Virtual Disk" (page 150).

To remove a volume from the system configuration database:

```
-> DELETE $DATA01
```

DELETE MON Command

The DELETE MON command removes the SMF master process from the system configuration database. The syntax is:

```
DELETE [ / OUT file-spec / ] MON $ZSMS [ , SEL state ]
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

MON $ZSMS
: is the SMF master process.

SEL *state*
: specifies that the command affects only devices in the specified state.

DELETE MON Example

See the procedure for "Deleting the SMF Master Process" (page 45).

To delete the SMF master process from the system configuration database:

```
-> DELETE MON $ZSMS
```

DELETE PARTITION Command

The DELETE PARTITION command deletes the last partition or all partitions on the specified CLIM LUN. You can delete a partition only if the NonStop DISK LDEV configured to that partition are in the STOPPED state. If there is a NonStop DISK LDEV configured to that partition, SCF will ask for confirmation.

CAUTION: When you delete a partition, all data on that partition is lost.

The syntax for the DELETE PARTITION command is:

```
DELETE PARTITION $ZZSTO [, SENDTO STORAGE],
PRIMARYCLIM clim-name,
PRIMARYLUN lun,
[LASTPARTITION partition-number | ALLPARTITIONS],
BACKUPCLIM backup-clim-name
[, BACKUPLUN backup-lun]
[, FORCED]
```

Wild-card characters are not supported.

SENDTO STORAGE
: directs the command to the storage subsystem.

PRIMARYCLIM `clim-name`
: is the name of the one CLIM that is connected to the disk.

PRIMARYLUN `lun`
: specifies the logical unit number (LUN) on the PRIMARYCLIM.

LASTPARTITION `partition-number` | ALLPARTITIONS
: specifies that the last used partition on the CLIM LUN should be deleted. `partition-number` is the number of that partition.

  ALLPARTITIONS deletes all partitions.

BACKUPCLIM `backup-clim-name`
: is the name of the other CLIM that is connected to the disk. BACKUPCLIM must be specified.

BACKUPLUN `backup-lun`
: specifies the logical unit number (LUN) on the BACKUPCLIM. This attribute is optional. If BACKUPLUN is not specified, the value of PRIMARYLUN is used as the BACKUPLUN.

FORCED
: SCF warns that any existing data on the disk will be lost and prompts for confirmation. A FORCED option displays the warning but bypasses the prompt for confirmation.

DELETE PARTITION Examples

- This is an example of the DELETE PARTITION command with the LASTPARTITION option:

  ```
  DELETE PARTITION $ZZSTO, &
      SENDTO STORAGE, &
      PRIMARYCLIM S1002533 , &
      PRIMARYLUN 100 , &
  ```

```
    LASTPARTITION 4, &
    BACKUPCLIM S1002534
```

- This is an example of the DELETE PARTITION command with the ALLPARTITIONS option:

```
DELETE PARTITION $ZZSTO, &
    SENDTO STORAGE, &
    PRIMARYCLIM S1002533 , &
    PRIMARYLUN 100 , &
    ALLPARTITIONS, &
    BACKUPCLIM S1002534
```

DELETE POOL Command

The DELETE POOL command removes a storage pool from the system configuration database. The command syntax is:

```
DELETE [ / OUT file-spec / ] POOL $pool
[ , IGNOREINCONSISTENCY ]
[ , SEL state ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

POOL $*pool*

is the storage pool process.

IGNOREINCONSISTENCY

specifies that a command can create an inconsistency between the system configuration database and the SMF catalogs maintained by the SMF manager process, storage pools, and virtual disks.

Use this attribute only if you understand SMF architecture and know how to restore consistency between the system configuration database and the SMF catalogs. The Softdoc for the T1083 product describes several inconsistencies and the procedures for reconciling them.

SEL *state*

specifies that the command affects only devices in the specified state.

DELETE POOL Example

See the procedure for "Deleting a Storage Pool" (page 140).

To delete a storage pool from the system configuration database:

```
-> DELETE $POOL00
```

DELETE PROFILE Command

The DELETE PROFILE command removes a user-configured profile from the system configuration database. The syntax is:

```
DELETE [ / OUT file-spec / ]
   PROFILE $ZZSTO.#INTERNAL-DISK-groupnum
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

PROFILE $ZZSTO.#INTERNAL-DISK-*groupnum*

is a custom profile for enclosure *groupnum*. *groupnum* cannot have a leading zero.

DELETE PROFILE Example

See "Deleting a Custom Profile" (page 76).

To delete a profile for all disks inserted into group 01:

```
-> DELETE PROFILE $ZZSTO.#INTERNAL-DISK-1
```

DELETE SCSI Command

The DELETE SCSI command removes an Open SCSI device from the system configuration database. The syntax is:

```
DELETE [ / OUT file-spec / ] SCSI $SCSI-device
     [ , SEL state ]
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

SCSI $*SCSI-device*
: is the name of the Open SCSI device.

SEL *state*
: specifies that the command affects only devices in the specified state.

DELETE SCSI Consideration

Stop the Open SCSI device before deleting it.

DELETE SCSI Examples

See "Stopping an Open SCSI Device" (page 177).

- To delete an Open SCSI device from the system configuration database:

  ```
  -> DELETE $DEV0
  ```

- You can use the SEL attribute to delete all Open SCSI devices in the STOPPED state:

  ```
  -> DELETE SCSI $*, SEL STOPPED
  ```

DELETE TAPE Command

The DELETE TAPE command removes a tape drive from the system configuration database. The syntax is:

```
DELETE  [ / OUT file-spec / ] TAPE $tape
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

TAPE $*tape*
: is the name of the tape device.

DELETE TAPE Example

See the procedure for "Deleting a Tape Drive" (page 187).

To delete a tape drive from the system configuration database:

```
-> DELETE $TAPE0
```

INFO Command

The INFO command displays system configuration information, such as the current attribute values for a specified object.

Supported objects are:

INFO ADAPTER Command

The INFO ADAPTER command displays configuration information about a ServerNet adapter. The syntax is:

```
INFO   [ / OUT file-spec / ]
   ADAPTER $ZZSTO.#type.GRP-g.MOD-m.SLOT-s [ , DETAIL ]
```

Wild-card characters are supported.

ADAPTER $ZZSTO.#*type*.GRP-*g*.MOD-*m*.SLOT-*s*

is a ServerNet adapter of *type* PMF, IOMF, or SNDA. *g* cannot have a leading zero.

DETAIL

displays detailed configuration information.

INFO ADAPTER Examples

- To display information about the PMF CRU adapter in group 01, slot 50:

  ```
  -> INFO $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50
  ```

  See the display and explanation under "Example of an INFO PMF ADAPTER Report" (page 154).

- To display detailed information about the PMF CRU adapter in group 01, slot 50:

  ```
  -> INFO $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50, DETAIL
  ```

  See the display and explanation under "Example of a Detailed INFO PMF ADAPTER Report" (page 154).

INFO CLIM Command

The INFO command displays a list of storage devices configured to use the CLIM. The syntax is:

```
INFO [ / OUT file-spec / ] CLIM $ZZSTO.clim-name, DETAIL
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

CLIM $ZZSTO.*clim-name*

is the name of the CLIM.

DETAIL

returns a list of the storage devices configured to use the CLIM, including the LUN and partition number for each path. The DETAIL option is required.

INFO CLIM Considerations

Additional information for a CLIM is available from $ZZCIP.

For more information, see the *Cluster I/O Protocols (CIP) Configuration and Management Manual*

INFO CLIM Examples

- To display a summary of information for all known CLIMs on a system:

```
-> INFO CLIM $ZZSTO.*, DETAIL
```

- To show the detailed configuration of S1002533:

```
-> INFO CLIM $ZZSTO.S1002533, DETAIL
```

- This command displays detailed information on all CLIMs in a system:

```
-> INFO CLIM $ZZSTO.S1002533,DETAIL

STORAGE - Detailed Info CLIM \NBSTS01.$ZZSTO.#S1002533

 Configured Devices:
   Type  Name     Primary CPU Backup CPU Lun  Partition
   DISK  $SSD1-P  0           1          101   1
   DISK  $SSD2-B  2           3          101   2
   DISK  $SSD3-P  1           3          101   3
   DISK  $SSD4-P  1           3          202
   DISK  $SAS1-P  2           3          208
   DISK  $SAS2-B  0           1          102
```

INFO DISK Command

The INFO DISK command displays configuration information about a physical or virtual disk. The syntax is:

```
INFO   [ / OUT file-spec / ] DISK $disk
   [ , { BAD | CACHE | CONFIG | DETAIL | LABEL | LOG |
   OBEYFORM | SECTOR [ % | %H ] lsa } ]
   [ , POOL $pool ] [ , SEL state ]
   [ , SUB { ALL | MAGNETIC | VIRTUAL } ]
```

Wild-card characters are supported.

BAD

(physical disks only) displays information about any bad sectors.

CACHE

(physical disks only) displays current cache information.

CONFIG

(physical disks only) displays current configuration information, including the LUN and partition number (if it exists) for each CLIM disk path.

DETAIL

displays detailed configuration information, including the LUN and partition number (if it exists) for each CLIM disk path.

LABEL
: (physical disks only) displays disk label information.

LOG
: (physical disks only) displays information about spared sectors.

OBEYFORM
: displays information about the object in the format used in an ADD command.

POOL *$pool*
: specifies that the command is performed only on disks associated with the specified storage pool.

SECTOR [% | %H] *lsa*
: (physical disks only) displays information about the disk sector at logical sector address *lsa*. *lsa* can be expressed as an octal (%), hexadecimal (%H), or decimal number.

SEL *state*
: specifies that the command affects only devices in the specified state.

SUB { ALL | MAGNETIC | VIRTUAL }
: specifies that the command affects only disks of the specified type. The default is ALL.

See "Attribute Descriptions for Disk Commands" (page 198) for descriptions of all attributes for disk commands.

INFO DISK Examples

- To display information about a disk:

  ```
  -> INFO $AUDIT
  ```

  See the display and explanation under "An INFO Report for a Disk" (page 49).

- To display detailed information about a disk:

  ```
  -> INFO $SYSTEM, DETAIL
  ```

  See the display and explanation under "Example 2: A Detailed INFO Report for a Disk" (page 49).

- To display information about a virtual disk:

  ```
  -> INFO $VDISK00
  ```

  See the display and explanation under "Example of an INFO DISK Summary Report" (page 145).

- To display detailed information about a virtual disk:

  ```
  -> INFO $VDISK00, DETAIL
  ```

  See the display and explanation under "Example of an INFO DISK Summary Report" (page 145).

- To display information about disk cache configuration for $SYSTEM:

  ```
  -> INFO $SYSTEM, CACHE
  ```

  See the display and explanation under "Displaying Disk Configuration Information" (page 50).

- To display information about the label for $SYSTEM:

  ```
  -> INFO $SYSTEM, LABEL
  ```

  See the display and explanation under "Displaying Disk Label Information" (page 52).

- To display information about unspared bad sectors:

  ```
  -> INFO $DATA01, BAD
  ```

  See the display and explanation under "Displaying Bad Sector Information" (page 62).

- To display information about the spared sectors:

  ```
  -> INFO $DATA01, LOG
  ```

See the display and explanation under “Displaying Profile Attributes” (page 54).

INFO MON Command

The INFO MON command displays configuration information about the SMF master process. The syntax is:

```
INFO [ / OUT file-spec / ] MON $ZSMS
   [ , { DETAIL | OBEYFORM } ]   [ , SEL state ]
```

Wild-card characters are supported.

MON $ZSMS

is the SMF master process.

DETAIL

displays detailed configuration information.

OBEYFORM

displays information about the object in the format used in an ADD command.

SEL *state*

specifies that the command affects only devices in the specified state.

INFO MON Examples

- To display information about the SMF master process:

  ```
  -> INFO $ZSMS
  ```

  See the display and explanation under “Example of an INFO MON Report” (page 42).

- To display detailed information about the SMF master process:

  ```
  -> INFO $ZSMS , DETAIL
  ```

  See the display and explanation under “Example of a Detailed INFO MON Report” (page 43).

INFO PARTITION Command

The INFO PARTITION command displays the physical disk partition information for the specified CLIM and LUN. The syntax is:

```
INFO PARTITION $ZZSTO, PRIMARYCLIM clim-name,
PRIMARYLUN lun ,
BACKUPCLIM backup-clim-name
[, BACKUPLUN backup-lun]
[, OBEYFORM ]
```

Wild-card characters are not supported.

PRIMARYCLIM *clim-name*

is the name of the one CLIM that is connected to the disk.

PRIMARYLUN *lun*

specifies the logical unit number (LUN) on the PRIMARYCLIM.

BACKUPCLIM *backup-clim-name*

is the name of the other CLIM that is connected to the disk. BACKUPCLIM must be specified.

BACKUPLUN *backup-lun*

specifies the logical unit number (LUN) on the BACKUPCLIM. This attribute is optional. If BACKUPLUN is not specified, the value of PRIMARYLUN is used as the BACKUPLUN.

OBEYFORM
: displays information about the object in the format used in an ADD command. The OBEYFORM option creates an obey file for configuring partitions on the disk. You can save the obey file and use that obey file to recreate partitions on the replacement disk. Save the obey file with a meaningful name.

INFO PARTITION Examples

- This displays the results for the basic INFO PARTITION command:

```
->INFO PARTITION $ZZSTO, PRIMARYCLIM S1002533, PRIMARYLUN 101,
BACKUPCLIM S1002531

STORAGE - LUN Partition Info

PRIMARY CLIM \NBSTS01.$ZZSTO.#S1002533

 Configured Partitions: LUN 101
   Name      Partition     Size (GB)
   $DATA00-P 1             20
   $SWAP00-M 2             10
   None       3            30
   None       4            40

BACKUP CLIM \NBSTS01.$ZZSTO.#S1002531

 Configured Partitions: LUN 101
   Name      Partition     Size (GB)
   $DATA00-B  1             20
   $SWAP00-MB 2             10
   None        3            30
   None        4            40

LUN Usage Information:
   Unpartitioned Size (GB)... 40
   Total Size (GB)........... 140
```

- This example shows the results of using the INFO PARTITION command with the OBEYFORM option, which creates an obey file for configuring partitions on the disk.

```
->INFO PARTITION $ZZSTO, PRIMARYCLIM S1002533, PRIMARYLUN 101,
BACKUPCLIM S1002531, OBEYFORM
== STORAGE - Obeyform Info PARTITION
DELETE PARTITION $ZZSTO, &
    SENDTO STORAGE, &
    PRIMARYCLIM S1002533 , &
    PRIMARYLUN 101 , &
    BACKUPCLIM S1002531 , &
    ALLPARTITIONS , &
    FORCED

ADD PARTITION $ZZSTO, &
    SENDTO STORAGE, &
    PRIMARYCLIM S1002533 , &
    PRIMARYLUN 101 , &
    BACKUPCLIM S1002531 , &
    STARTPARTITION 1, &
    PARTITIONSIZE (20, 10, 30, 40), &
    FORCED
```

INFO POOL Command

The INFO POOL command displays configuration information about a storage pool. The syntax is:

```
INFO [ / OUT file-spec / ] POOL $pool
   [ , { DETAIL | OBEYFORM } ]  [ , SEL state ]
```

Wild-card characters are supported.

POOL $*pool*

is the storage pool process.

DETAIL

displays detailed configuration information.

OBEYFORM

displays information about the object in the format used in an ADD command.

SEL *state*

specifies that the command affects only devices in the specified state.

INFO POOL Examples

- To display information about a storage pool process:

  `-> INFO $POOL1`

  See the display and explanation under "Considerations for INFO POOL" (page 134).

- To display detailed information about a storage pool process:

  `-> INFO $POOL1 , DETAIL`

  See the display and explanation under "Example of a Detailed INFO POOL Report" (page 135).

INFO PROFILE Command

The INFO PROFILE command displays configuration information about the specified profile. The syntax is:

```
INFO [ / OUT file-spec / ]
   PROFILE $ZZSTO.#INTERNAL-DISK[-groupnum ]
   [ , OBEYFORM ]
```

Wild-card characters are supported.

PROFILE $ZZSTO.#INTERNAL-DISK[-*groupnum*]

is either the standard default profile, $ZZSTO.#INTERNAL-DISK, or a user-created custom profile for enclosure groupnum. *groupnum* cannot have a leading zero.

OBEYFORM

displays information about the object in the format used in an ADD command.

INFO PROFILE Examples

- To display detailed information about the standard default internal-disk profile:

  `-> INFO $ZZSTO.#INTERNAL-DISK`

  See the display and explanation under "Example of an INFO PROFILE Report" (page 54).

- To display detailed information about the internal-disk profile for group 03 disks (assuming you have created one):

```
-> INFO $ZZSTO.#INTERNAL-DISK-3
```

- To display information about all disk profiles on the system:

  ```
  -> INFO PROFILE $ZZSTO.*
  ```

INFO SCSI Command

The INFO SCSI command displays configuration information about an Open SCSI device. The command syntax is:

```
INFO [ / OUT file-spec / ] SCSI $SCSI-device

    [ , { DETAIL | OBEYFORM } ] [ , SEL state ]
```

Wild-card characters are supported.

SCSI $*SCSI-device*
: is the name of the Open SCSI device.

DETAIL
: displays detailed configuration information.

OBEYFORM
: displays information about the object in the format used in an ADD command.

SEL *state*
: specifies that the command affects only devices in the specified state.

INFO SCSI Examples

- To display information about an Open SCSI device:

  ```
  -> INFO $DEV0
  ```

  See the display and explanation under “Example of an INFO SCSI Report” (page 171).

- To display detailed information about an Open SCSI device:

  ```
  -> INFO $DEV5, DETAIL
  ```

  See the display and explanation under“Examples of Detailed INFO SCSI Reports” (page 171)
  .

INFO SUBSYS Command

The INFO SUBSYS command displays configuration information about the $ZZSTO storage subsystem manager. The command syntax is:

```
INFO [ / OUT file-spec / ] SUBSYS $ZZSTO
```

Wild-card characters are not supported.

SUBSYS $ZZSTO
: is the storage subsystem manager.

INFO SUBSYS Consideration

The information showed by INFO SUBSYS and STATUS SUBSYS should always be the same. If the information shown in these displays differs, contact your service provider.

INFO SUBSYS Example

```
-> INFO SUBSYS $ZZSTO
```

See the display and explanation under “Using the Storage Subsystem Manager” (page 40).

INFO TAPE Command

The INFO TAPE command displays configuration information about a tape device. The syntax is:

```
INFO [ / OUT file-spec / ] TAPE $tape
   [ , { DETAIL | OBEYFORM } ] [ , SEL state ]
```

Wild-card characters are supported.

TAPE $*tape*

is the name of the tape device.

DETAIL

displays detailed configuration information.

OBEYFORM

displays information about the object in the format used in an ADD command.

SEL *state*

specifies that the command affects only devices in the specified state.

INFO TAPE Examples

- To display information about a tape drive:

  `-> INFO $TAPE0`

  See the display and explanation under "Example of an INFO TAPE Report" (page 180).

- To display detailed information about a tape drive:

  `-> INFO $TAPE0, DETAIL`

  See the display and explanation under "Examples of INFO TAPE Detailed Reports" (page 181).

- To display information about all tape drives that are not stopped:

  `-> INFO TAPE $*, SEL NOT STOPPED`

INITIALIZE Command

The INITIALIZE command prepares a disk for use on the system. This command removes the directory information, deletes the log of spared sectors, initializes the customer engineer sector and the spare-tracks table, and optionally labels the disk.

CAUTION: The INITIALIZE command removes directory information so that files are not accessible to normal software. The file data remains on the disk and could be read by specialized low level disk access utilities.

INITIALIZE is a sensitive command.

INITIALIZE DISK Command

The syntax is:

```
INITIALIZE [ / OUT file-spec / ] DISK $disk[-P |-M ]
   [ , LABEL $volume ]
   [ DISK $disk-P |-M { CLEARENCRYPTKEY | NEWENCRYPTKEY,
      KEYALGORITHM {XTS-AES | CBC-AES}, KEYSIZE 256 } ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

DISK $*disk*[-P |-M]

is the disk name and, optionally, the path (primary or mirror).

LABEL $*volume*

is the label for the disk being initialized. This attribute is ignored if you specify a disk path in this command. If you do not use this option, you must create a label using the ALTER DISK, LABEL command to give the disk a valid label.

CLEARENCRYPTKEY

clears encryption on an encrypted disk.

This attribute changes the encryption status for a disk (single drive) while that drive is DOWN. Its mirror may be UP. The disk volume will be online if the mirror is UP. The drive is initialized. The drive's volume label is left blank. If data needs to be copied from the other mirror, you must start the REVIVE manually after the INITIALIZE command.

You must specify a path.

KEYALGORITHM

specifies the encryption key algorithm on a disk. Valid values are XTS-AES and CBC-AES

This attribute changes the encryption key for a disk (single drive) while that drive is DOWN. Its mirror may be UP. The disk volume will be online if the mirror is UP. The drive is initialized. The drive's volume label is left blank. If data needs to be copied from the other mirror, you must start the REVIVE manually after the INITIALIZE command.

You must specify a path.

KEYSIZE

specifies the key size for the key algorithm for an encrypted disk. Valid value is 256.

NEWENCRYPTKEY

sets a new encryption key on a disk.

If the disk is not encrypted, it is initialized as encrypted. If the disk is already encrypted, this attribute changes the encryption key for a disk (single drive) while that drive is DOWN. Its mirror may be UP. The disk volume will be online if the mirror is UP. The drive's volume label is left blank. If data needs to be copied from the other mirror, you must start the REVIVE manually after the INITIALIZE command.

You must specify a path.

NOTE: Only members of the SAFEGUARD security officer group on the local system can perform an INITIALIZE DISK command with the NEWENCRYPTKEY attribute. Also, only security officers can initiate a revive from an encrypted disk to a non-encrypted disk.

See "Attribute Descriptions for Disk Commands" (page 198) for descriptions of all attributes for disk commands.

INITIALIZE DISK Examples

See "Swapping Processors for a Disk" (page 109).

- To initialize a disk and relabel it using the same volume name:

```
-> INITIALIZE DISK $DATA12
```

- To initialize a disk and give it another name:

```
-> INITIALIZE DISK $DATA12, LABEL $SPARE00
```

NAMES Command

The NAMES command displays object names for a specified object type and any subordinate objects

```
NAMES [ / OUT file-spec / ] [ object-spec ]
   [ , POOL $pool ]  [ , SEL state ]  [ , SUB subtype ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

object-spec

is one of these object types, as defined in "Storage Subsystem Objects" (page 30):

- ADAPTER
- DISK
- MON
- POOL
- PROFILE
- SAC
- SCSI
- SUBSYS
- TAPE

POOL *$pool*

(DISK objects only) specifies that information for disks associated only with the specified storage pool process should be displayed.

SEL *state*

specifies that the command affects only devices in the specified state.

SUB subtype

($ZZSTO or DISK objects only) specifies which subtypes of the given object-spec should be displayed.

For $ZZSTO, these subtypes are allowed (the default is ALL):

`ADAPTER  ALL  DISK  MAGNETIC  MON  POOL PROFILE  SCSI  TAPE  VIRTUAL`

For DISK objects, these subtypes are allowed (the default is ALL):

`ALL  MAGNETIC  VIRTUAL`

NAMES Examples

- To display the object names of all objects managed by the storage subsystem manager process:

```
-> NAMES $ZZSTO

STORAGE Names SUBSYS \SWEET.$ZZSTO

SUBSYS
$ZZSTO

ADAPTER
$ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-50  $ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-55
$ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50    $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-55
$ZZSTO.#PMF.GRP-2.MOD-1.SLOT-50    $ZZSTO.#PMF.GRP-2.MOD-1.SLOT-55
$ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-51  $ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-52
$ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-53

DISK
$SYSTEM   $SWEETX   $R06WHT   $R06NEV   $R05WHT   $R05NEV   $R04NEV   $OPS
$N01WHT   $G02NEV   $FUDD     $DATA00   $D1103M   $D0204    $D0118U   $D0111M
```

```
PROFILE
$ZZSTO.INTERNAL-DISK

TAPE
$TAPE4
```

- To display the names for all physical disks:

```
-> NAMES $ZZSTO, SUB MAGNETIC

DISK
$SYSTEM  $SWEETX  $R06WHT  $R06NEV  $R05WHT  $R05NEV  $R04NEV  $OPS
$N01WHT  $G02NEV  $FUDD    $DATA00  $D1103M  $D0204   $D0118U  $D0111M
$D0107M  $D0105M  $BROOK   $AUDIT
```

- To display the names for all virtual disks:

```
-> NAMES $ZZSTO, SUB VIRTUAL

STORAGE Names SUBSYS \COMM.$ZZSTO   DISK $ZIMBU
$ZERO      $WIPRO     $WEBVPT    $WEB       $WANA      $VIEWPT    $VCS2
$TIOGA     $TEMP      $TECH      $TCPIP1    $SPOOL2    $SPOCK     $SPLPRS
$SPLDEV    $SNMP3     $SNMP2     $SNMP1     $SMTPA     $SLSA1     $SLSA
$SIMS5     $SIMS4     $SIMS3     $SIMS2     $SIMS1     $SHARE1    $SFGD
$SECRET    $RTOOL2    $RPSV03    …
```

- To display the object names of all adapters:

```
-> NAMES $ZZSTO, SUB ADAPTER

STORAGE Names SUBSYS \COMM.$ZZSTO

ADAPTER $ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-50  $ZZSTO.#IOMF.GRP-11.MOD-1.SLOT-55
$ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50            $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-55
$ZZSTO.#PMF.GRP-2.MOD-1.SLOT-50            $ZZSTO.#PMF.GRP-2.MOD-1.SLOT-55
$ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-51          $ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-52
$ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-53
```

- To display the object names of all PROFILE objects managed by the storage subsystem manager process:

```
-> NAMES $ZZSTO, SUB PROFILE

STORAGE Names SUBSYS \COMM.$ZZSTO
PROFILE $ZZSTO.INTERNAL-DISK
```

- To display the names of all SCSI devices:

```
-> NAMES $ZZSTO, SUB SCSI

STORAGE Names SUBSYS \COMM.$ZZSTO

SCSI $L700C16
```

- To display the names for all TAPE objects:

```
-> NAMES $ZZSTO, SUB TAPE

STORAGE Names SUBSYS \COMM.$ZZSTO

TAPE $TAPE0  $DLT25  $DLT24  $DLT23  $DLT22
```

PRIMARY Command

The PRIMARY command swaps the primary and backup processors for a device or process. The current primary process becomes the backup process, and the current backup process becomes the primary process, but the PRIMARYCPU and BACKUPCPU values stay the same.

Supported objects are:

- “PRIMARY DISK Command” (page 264)
- “PRIMARY MON Command” (page 265)
- “PRIMARY POOL Command” (page 265)
- “PRIMARY SCSI Command” (page 266)
- “PRIMARY SUBSYS Command” (page 266)
- “PRIMARY TAPE Command” (page 267)

PRIMARY is a sensitive command.

PRIMARY DISK Command

The PRIMARY DISK command swaps the primary and backup processors for the disk process controlling the specified physical or virtual disk. The current primary process becomes the backup process, and the current backup process becomes the primary process, but the PRIMARYCPU and BACKUPCPU values stay the same. The syntax is:

```
PRIMARY [ / OUT file-spec / ]  DISK $disk
   [ , cpunumber ]   [ , FORCED ]
   [ , POOL $pool ]   [ , SEL state ]
   [ , SUB { ALL | MAGNETIC | VIRTUAL } ]
```

Wild-card characters are supported.

OUT *file-spec*	directs all SCF output to the specified file.
DISK $*disk*	is the name of the disk.
cpunumber	is the processor number of the processor that is to become the primary processor. This decimal integer must identify one of the two processors configured as primary and backup processors for the device. If you do not specify a processor number, the storage subsystem manager determines which processors are currently being used for the primary process and backup process and swaps those processors. If you specify the processor number of the current primary processor, no change occurs.
FORCED	(physical disks only) for G06.10 and earlier RVUs, specifies that all disk processes using the same controller as the specified disk must switch to the specified primary processor. Beginning with G06.11, the FORCED attribute is ignored. Because SACs can be owned by more than one processor, the PRIMARY command affects only the IOP for the disk whose primary processor is being swapped. Other disk processes are unaffected. For more details, refer to “Managing Disks” (page 96).
POOL *$pool*	specifies that the PRIMARY DISK command should be performed only on disks associated with the specified storage pool.
SEL *state*	specifies that the command affects only devices in the specified state.

SUB { ALL | MAGNETIC | VIRTUAL }
: specifies that the command affects only disks of the specified type. The default is ALL.

See "Attribute Descriptions for Disk Commands" (page 198) for descriptions of all attributes for disk commands.

PRIMARY DISK Examples

See the procedure for "Swapping Processors for a Disk" (page 109).

- To run the primary disk process in processor 3:

  ```
  -> PRIMARY $DATA00, 3
  ```

- To swap the primary and backup processors controlling a disk:

  ```
  -> PRIMARY $SYSTEM
  ```

- To switch the primary and backup processors controlling two disks:

  ```
  -> PRIMARY ($DATA01, $DATA02)
  ```

PRIMARY MON Command

The PRIMARY MON command swaps the primary and backup processors for the MON process. The current primary process becomes the backup process, and the current backup process becomes the primary process, but the PRIMARYCPU and BACKUPCPU values stay the same. The syntax is:

```
PRIMARY  [ / OUT file-spec / ]  MON $ZSMS  [ , cpunumber ]
```

Wild-card characters are supported.

MON $ZSMS

is the SMF master process.

cpunumber

is the processor number of the processor that is to become the primary processor. This decimal integer must identify one of the two processors configured as primary and backup processors for the device. If you do not specify a processor number, the storage subsystem manager determines which processors are currently being used for the primary process and backup process and swaps those processors. If you specify the processor number of the current primary processor, no change occurs.

PRIMARY MON Examples

See the procedure for "Swapping Processors for the SMF Master Process" (page 47).

- To run the primary process of $ZSMS in processor 3:

  ```
  -> PRIMARY $ZSMS, 3
  ```

- To swap the primary and backup processors controlling $ZSMS:

  ```
  -> PRIMARY $ZSMS
  ```

PRIMARY POOL Command

The PRIMARY POOL command swaps the primary and backup processors for the specified storage pool process. The current primary process becomes the backup process, and the current backup process becomes the primary process, but the PRIMARYCPU and BACKUPCPU values stay the same. The syntax is:

```
PRIMARY  [ / OUT file-spec / ] POOL $pool  [ , cpunumber ]
```

Wild-card characters are supported.

POOL $*pool*

is the storage pool process.

cpunumber

is the processor number of the processor that is to become the primary processor. This decimal integer must identify one of the two processors configured as primary and backup processors for the device. If you do not specify a processor number, the storage subsystem manager determines which processors are currently being used for the primary process and backup process and swaps those processors. If you specify the processor number of the current primary processor, no change occurs.

PRIMARY POOL Examples

See the procedure for “Swapping Processors for a Pool Process” (page 142).

- To run the primary pool process in processor 3:

```
-> PRIMARY $POOL00, 3
```

- To swap the primary and backup processors controlling a pool process:

```
-> PRIMARY $POOL01
```

PRIMARY SCSI Command

The PRIMARY SCSI command switches the primary and backup processors for the specified Open SCSI device. The current primary process becomes the backup process, and the current backup process becomes the primary process, but the PRIMARYCPU and BACKUPCPU values stay the same. The syntax is:

```
PRIMARY  [ / OUT file-spec / ] SCSI $SCSI-device

     [ , cpunumber ]
```

Wild-card characters are supported.

SCSI $*SCSI-device*

is the name of the Open SCSI device.

cpunumber

is the processor number of the processor that is to become the primary processor. This decimal integer must identify one of the two processors configured as primary and backup processors for the device. If you do not specify a processor number, the storage subsystem manager determines which processors are currently being used for the primary process and backup process and swaps those processors. If you specify the processor number of the current primary processor, no change occurs.

PRIMARY SCSI Examples

See “Swapping Processors for an Open SCSI Device” (page 177).

- To run the primary Open SCSI process in processor 3:

```
-> PRIMARY $DEV0, 3
```

- To swap the primary and backup processors:

```
-> PRIMARY $DEV0
```

PRIMARY SUBSYS Command

The PRIMARY SUBSYS command moves all storage subsystem IOPs into or away from a processor, as part of device or adapter replacement. The syntax is:

```
PRIMARY [ / OUT file-spec / ]  SUBSYS $ZZSTO
    , cpunumber , { AWAY | DEFAULT }
```

OUT *file-spec*

directs all SCF output to the specified file.

SUBSYS $ZZSTO

is the storage subsystem manager.

cpunumber

is the processor number of the processor that you want to move storage subsystem IOPS into or away from.

AWAY

specifies that all storage subsystem IOPs currently running their primary processes in *cpunumber* be moved away from that processor.

DEFAULT

specifies that all running storage subsystem IOPs that are configured with *cpunumber* as their PRIMARYCPU be moved into that processor, and that all running storage subsystem IOPs that are configured with *cpunumber* as their BACKUPCPU be moved away from that processor.

PRIMARY SUBSYS Examples

- To move all storage subsystem IOPs in the group 02 topology branch away from processor 3:

  ```
  -> PRIMARY SUBSYS $ZZSTO, 3, AWAY
  ```

- To move all storage subsystem IOPs configured with PRIMARYCPU 2 to that processor:

  ```
  -> PRIMARY SUBSYS $ZZSTO, 2, DEFAULT
  ```

PRIMARY TAPE Command

The PRIMARY TAPE command switches the primary and backup processors for the specified tape drive. The current primary process becomes the backup process, and the current backup process becomes the primary process, but the PRIMARYCPU and BACKUPCPU values stay the same. The syntax is:

```
PRIMARY  [ / OUT file-spec / ] TAPE $tape  [ , cpunumber ]
```

Wild-card characters are supported.

TAPE $*tape*

is the name of the tape device.

cpunumber

is the processor number of the processor that is to become the primary processor. This decimal integer must identify one of the two processors configured as primary and backup processors for the device. If you do not specify a processor number, the storage subsystem manager determines which processors are currently being used for the primary process and backup process and swaps those processors. If you specify the processor number of the current primary processor, no change occurs.

PRIMARY TAPE Examples

- To move the primary tape process to processor 3:

  ```
  -> PRIMARY $TAPE0, 3
  ```

- To swap the primary and backup processors:

```
-> PRIMARY $TAPE0
```

PROBE Command

The PROBE SAC command tests the Fibre Channel connection to a SAC on an FCSA

NOTE: The PROBE SAC command is available on H06.04 and subsequent H-series RVUs.

PROBE is a sensitive command.

PROBE SAC Command

The PROBE SAC command sends and receives a data packet to test the Fibre Channel connection to a SAC on an FCSA. The connection can be either an arbitrated loop or a link to a Fibre Channel switch.

To issue the PROBE SAC on an arbitrated loop, the disk paths must be in a stopped state.

The PROBE SAC command displays no output unless an error occurs.

The syntax is:

```
PROBE [ / OUT file-spec / ] SAC $ZZSTO.#sac-name
        [    , COUNT n [ , CONTINUE ] ]
        [    , DATA pattern           ]
        [    , PORTNAME wwn           ]
```

Wild-card characters are supported for the PROBE SAC command.

OUT *file-spec*

directs all SCF output to the specified file.

#sac-name

is the name of the SAC in the form *#FCSA.SAC-sac.GRP-group.MOD-module.SLOT-slot*. For example:

```
#FCSA.SAC-2.GRP-110.MOD-2.SLOT-5
```

CONTINUE

specifies that the data packet continue being sent until the count is complete, even if an error occurs.

COUNT *n*

specifies the number of times to send the data packet. Valid values are in the range 1 to 255.

DATA *pattern*

specifies a data pattern (up to 64 bytes) that can be used for diagnostic purposes. The data pattern must be entered as a hexadecimal number without a leading %H.

PORTNAME *wwn*

specifies the worldwide name of a port in the Fibre Channel link connected to the SAC.

wwn must be entered as a 16-character hexadecimal number without a leading %H.

Specify a PORTNAME to test the connection to a Fibre Channel switch.

Do not specify a PORTNAME to test the connection to an arbitrated loop.

PROBE SAC Examples

- To test the arbitrated loop connected to SAC 1 on the FCSA in slot 3, module 3 of group 111:

```
-> PROBE SAC $ZZSTO.#FCSA.SAC-1.GRP-111.MOD-3.SLOT-3
```

- To test the connection between SAC 2 on an FCSA and port 50060E8004289461:

```
-> PROBE SAC $ZZSTO.#FCSA.SAC-2.GRP-112.MOD-2.SLOT-4, &
-> PORTNAME 50060E8004289461
```

RENAME Command

The RENAME command changes the name of a disk.

RENAME is a sensitive command.

RENAME DISK Command

The RENAME command changes the default volume name and alternate volume name of a disk without destroying files on the disk. The syntax is:

```
RENAME [ / OUT file-spec / ] DISK $disk , $new-disk
```

Wild-card characters are not supported.

OUT *file-spec*

directs all SCF output to the specified file.

DISK $*disk*

is the name of the disk.

$*new-disk*

is the new volume name for the disk. See the syntax for "The DISK Object" (page 33).

RENAME DISK Example

See the considerations and procedure for "Naming a Disk" (page 88).

To rename the disk $DATA00 to $SPARE00:

```
-> RENAME $DATA00, $SPARE00
```

REPLACE Command

The REPLACE command changes the object file of an executing object.

REPLACE ADAPTER Command

The REPLACE ADAPTER command downloads new firmware to adapters and SACs.

The syntax is:

```
REPLACE [ /OUT file-spec/ ] ADAPTER $ZZSTO.#adapter-name
            [ , attribute-spec ] ...
```

OUT *file-spec*

directs all SCF output to the specified file.

ADAPTER $ZZSTO.# *adapter-name*

is a ServerNet adapter of type PMF, IOMF, SNDA or FCSA in the format *type*.GRP-*group*.MOD-*module*.SLOT-*slot*. For example:

```
#FCSA.GRP-110.MOD-2.SLOT-5
```

group cannot have a leading zero.

attribute-spec

is one of "REPLACE ADAPTER Attributes".

REPLACE ADAPTER Attributes

```
[ , ABANDON ]
[ , FIRMWARE [[$volume.]subvol.]filename [, CPU n ]
[ , FLASHBOOT [[$volume.]subvol.]filename
[ , FLASHFIRMWARE [[$volume.]subvol.]filename
```

ABANDON

specifies that the current REPLACE ADAPTER procedure for operational firmware should be aborted.

Flash firmware replacement cannot be aborted.

CPU *n*

specifies that only IOPs in the CPU indicated by *n* should download the new firmware to the SACs they are using. If you do not specify a particular CPU number, all IOPs will download the new firmware, beginning with the IOPs in the highest numbered CPU

FIRMWARE [[$*volume*.]*subvol*.]*filename*

specifies the name of the new operational firmware file. The file must exist and must have a file code of 510. The default volume and subvolume are equal to the current SYS*nn*. The adapter name must be given as $ZZSTO.#*.

FLASHBOOT [[$*volume*.]*subvol*.]*filename*

specifies the name of the new flash boot file. The file must exist and must have a file code of 510. The default volume and subvolume are equal to the current SYS*nn*. This attribute can be used only with Fibre Channel ServerNet adapters (FCSAs).

FLASHFIRMWARE [[$*volume*.]*subvol*.]*filename*

specifies the name of the new flash firmware file. The file must exist and must have a file code of 510. The default volume and subvolume are equal to the current SYS*nn*. This attribute can be used only with FCSAs.

For considerations regarding these attributes, see "Considerations for REPLACE ADAPTER" (page 168).

REPLACE SAC Command

For an M8*xxx* FCDM connected to an FCSA, the REPLACE SAC command downloads new firmware to an EMU connected to the SAC. The syntax is:

```
REPLACE [ /OUT file-spec/ ] SAC $ZZSTO.#sac-name

        [    , DEVICEID (shelf, 99) ]
        [    , FIRMWARE [[$volume.]subvol.]filename ]
```

OUT *file-spec*

directs all SCF output to the specified file.

#sac-name

is the name of the SAC in the form *#FCSA.SAC-sac*.GRP-*group*.MOD-*module*.SLOT-*slot*. For example:

```
#FCSA.SAC-2.GRP-110.MOD-2.SLOT-5
```

DEVICEID (*shelf*, 99)

specifies the shelf number of the enclosure to which the command applies. The bay number 99 designates the EMU.

FIRMWARE [[$*volume*.]*subvol*.]*filename*

specifies the name of the new EMU firmware file. The file must exist, and it must have file code 0. The default volume and subvolume is the current SYS*nn*.

RESET Command

The RESET command puts an object in a state ready for starting.

RESET is a sensitive command.

RESET DISK Command

The RESET DISK command puts a disk into the STOPPED state, substate DOWN, ready for restarting. It also can pause a disk revive operation. The syntax is:

```
RESET [ / OUT file-spec / ]  DISK $disk[-P |-B |-M |-MB ]
    [ , FORCED ]   [ , POOL $pool ]   [ , SEL state ]
    [ , SUB { ALL | MAGNETIC | VIRTUAL } ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

DISK $*disk*[-P |-B |-M |-MB]

is the disk name and, optionally, the path (primary, backup, mirror, or mirror backup).

FORCED

terminates the IOP. If you use this attribute, you cannot specify a disk path (-P, -B, -M, or -MB).

POOL *$pool*

specifies that the command is performed only on disks associated with the specified storage pool.

SEL *state*

specifies that the command affects only devices in the specified state.

SUB { ALL | MAGNETIC | VIRTUAL }

specifies that the command affects only disks of the specified type. The default is ALL.

See “Attribute Descriptions for Disk Commands” (page 198) for descriptions of all attributes for disk commands.

RESET DISK Examples

See these procedures:

- “Resetting a Disk” (page 103)
- “Resetting a Virtual Disk” (page 151)
- To reset the mirror half of a disk:

  ```
  -> RESET $DATA00-M
  ```

- To reset a virtual disk:

  ```
  -> RESET $VDISK1
  ```

RESET MON Command

The RESET MON command puts the SMF master process into the STOPPED state, substate DOWN, ready for restarting. The syntax is:

```
RESET  [ / OUT file-spec / ] MON $ZSMS
    [ , FORCED ]   [ , SEL state ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

MON $ZSMS

is the SMF master process.

FORCED

terminates the IOP.

SEL *state*

specifies that the command affects only devices in the specified state.

RESET MON Consideration

The RESET command is ignored if the process is started.

RESET MON Example

See the procedure for "Resetting the SMF Master Process" (page 47).

To terminate the SMF master process:

```
-> RESET $ZSMS, FORCED
```

RESET POOL Command

The RESET POOL command puts a storage pool process into the STOPPED state, substate DOWN, ready for restarting. The syntax is:

```
RESET [ / OUT file-spec / ] POOL $pool
   [ , FORCED ]  [ , SEL state ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

POOL $*pool*

is the storage pool process.

FORCED

terminates the IOP.

SEL *state*

specifies that the command affects only devices in the specified state.

RESET POOL Consideration

The RESET command is ignored if the process is started.

RESET POOL Example

See the procedure for "Resetting a Storage Pool" (page 142).

To terminate the storage pool process:

```
-> RESET $POOL00, FORCED
```

RESET SCSI Command

The RESET SCSI command puts an Open SCSI device into the STOPPED state, substate DOWN, ready for restarting. The syntax is:

```
RESET [ / OUT file-spec / ] SCSI $SCSI-device
   [ , FORCED ]  [ , SEL state ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

SCSI $*SCSI-device*[-P |-B]
: is the name of the Open SCSI IOP and, optionally, the path (primary or backup).

FORCED
: terminates the IOP. If you use this attribute, you cannot specify a -P or B path.

SEL *state*
: specifies that the command affects only devices in the specified state.

RESET SCSI Consideration

The RESET command is ignored if the process is started.

RESET SCSI Example

See the procedure for “Resetting an Open SCSI Device” (page 176).

To reset an Open SCSI device:

```
-> RESET $DEV0
```

RESET TAPE Command

The RESET TAPE command puts a tape drive into the STOPPED state, substate DOWN, ready for restarting. The syntax is:

```
RESET [ / OUT file-spec / ] TAPE $tape
   [ , FORCED ]   [ , SEL state ]
```

Wild-card characters are not supported.

OUT *file-spec*
: directs all SCF output to the specified file.

TAPE $*tape*
: is the name of the tape device.

FORCED
: terminates the IOP.

SEL *state*
: specifies that the command affects only devices in the specified state.

RESET TAPE Consideration

The RESET command is ignored if the process is started.

RESET TAPE Example

See the procedure for “Resetting a Tape Drive” (page 188).

To terminate a tape drive process:

```
-> RESET $TAPE0, FORCED
```

START Command

The START command initiates the operation of an object (make a stopped device accessible to user processes). Successful completion of the START command leaves the object in the STARTED state.

Supported objects are:

- “START DISK Command” (page 274)
- “START MON Command” (page 275)

START is a sensitive command.

START DISK Command

The START DISK command makes a stopped disk accessible to user processes. If needed, this command also revives the half of a mirrored volume that is in the STOPPED state, substate DOWN. The syntax is:

```
START [ / OUT file-spec / ]  DISK $disk [-P |-B |-M |-MB ]
    [ , DEBUG $terminal ]   [ , POOL $pool ]
    [ , SEL state ]   [ , SPECIAL ]
    [ , SUB { ALL | MAGNETIC | VIRTUAL } ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

DISK $*disk*[-P |-B |-M |-MB]

is the disk name and, optionally, the path (primary, backup, mirror, or mirror backup).

DEBUG *$terminal*

specifies that the process should start in debug mode on *$terminal*.

POOL *$pool*

specifies that the command is performed only on disks associated with the specified storage pool.

SEL *state*

specifies that the command affects only devices in the specified state.

SPECIAL

puts the disk into the SERVICING state, substate SPECIAL. Use this attribute to ensure that only privileged processes can access the disk. You must designate the entire primary or mirror path.

SUB { ALL | MAGNETIC | VIRTUAL }

specifies that the command affects only disks of the specified type. The default is ALL.

See "Attribute Descriptions for Disk Commands" (page 198) for descriptions of all attributes for disk commands.

START DISK Examples For Physical Disks

See the procedure for "Starting a Disk" (page 97) and the procedure for "Starting a Virtual Disk" (page 150).

- To start all disks on the system (that are in the proper state to start):

```
-> START DISK $*
```

- To start a mirrored volume:

```
-> START $DATA00-M
```

- To start a disk and put it in the SERVICING state, substate SPECIAL:

```
-> START $DATA03, SPECIAL
```

- To start all available virtual disks:

```
-> START DISK $*, SUB VIRTUAL
```

START MON Command

The START MON command makes the SMF master process available. The syntax is:

```
START   [ / OUT file-spec / ]   MON $ZSMS
    [ , DEBUG $terminal ]
    [ , SEL state ] [ , SPECIAL ]
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

MON $ZSMS
: is the SMF master process.

DEBUG *$terminal*
: specifies that the process should start in debug mode on *$terminal*.

SEL *state*
: specifies that the command affects only devices in the specified state.

SPECIAL
: specifies that the process be started in the SERVICING state, substate SPECIAL.

START MON Example

See the procedure for "Starting the SMF Master Process" (page 46).

To start the SMF master process $ZSMS in the SERVICING state:

```
-> START $ZSMS, SPECIAL
```

START POOL Command

The START POOL command makes a storage pool available to the applications on the system. The syntax is:

```
START [ / OUT file-spec / ]   POOL $pool
    [ , DEBUG $terminal ]   [ , SEL state ]   [ , SPECIAL ]
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

POOL *$pool*
: is the storage pool process.

DEBUG *$terminal*
: specifies that the process should start in debug mode on *$terminal*.

SEL *state*
: specifies that the command affects only devices in the specified state.

SPECIAL
: specifies that the process be started in the SERVICING state, substate SPECIAL.

START POOL Examples

See the procedure for "Starting a Storage Pool" (page 141).

- To start all storage pool processes available on the system:

```
-> START POOL $*
```

- To start a storage pool process in the SPECIAL substate:

```
-> START $POOL00, SPECIAL
```

START SCSI Command

The START SCSI command makes a stopped Open SCSI device or a path to an Open SCSI device accessible to user processes. The syntax is:

```
START [ / OUT file-spec / ]  SCSI $SCSI-device[-P |-B ]
   [ , DEBUG $terminal ] [ , SEL state ]
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

SCSI $*SCSI-device*[-P |-B]
: is the name of the Open SCSI IOP and, optionally, the path (primary or backup).

DEBUG *$terminal*
: specifies that the process should start in debug mode on *$terminal*.

SEL *state*
: specifies that the command affects only devices in the specified state.

START SCSI Examples

See the procedure for "Starting an Open SCSI Device" (page 177).

- To start all available Open SCSI devices on the system:

```
-> START SCSI $*
```

- To start the backup path to an Open SCSI device:

```
-> START $DEV00-B
```

START TAPE Command

The START TAPE command makes a tape drive accessible to user processes. The syntax is:

```
START [ / OUT file-spec / ] TAPE $tape
   [ , DEBUG $terminal ]  [ , SEL state ]
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

TAPE *$tape*
: is the name of the tape device.

DEBUG *$terminal*
: specifies that the process should start in debug mode on *$terminal*.

SEL *state*
: specifies that the command affects only devices in the specified state.

START TAPE Examples

See the procedure for "Starting a Tape Drive" (page 189).

- To start all tapes available on the system:

```
-> START TAPE $*
```

- To start a tape drive:

```
-> START $TAPE0
```

STATS Command

The STATS command displays cache statistics about an object and optionally resets them.

Supported objects are:

- "STATS DISK Command" (page 277)
- "STATS SAC Command" (page 278)

STATS DISK Command

The STATS DISK command displays cache statistics about physical disks. The syntax is:

```
STATS [ / OUT file-spec / ] DISK $disk
    [ , ALL ]
    [ , CACHE ]
    [ , CONTROLBLOCKS ]
    [ , IO | LONGIO ]
    [ , LOCKS ]
    [ , POOL $pool ]
    [ , RESET [ , FORCED ] ]
    [ , SEL state ]
    [ , SQLMX ]
    [ , SUB { ALL | MAGNETIC | VIRTUAL } ]
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

DISK $*disk*
: is the name of the disk.

ALL
: specifies that all statistics (that is, CACHE, CONTROLBLOCKS, IO, LOCKS, and SQLMX) be shown.

CACHE
: specifies that cache statistics be shown. This is the default if no statistics type (that is, ALL, CACHE, CONTROLBLOCKS, IO, LOCKS, or SQLMX) is specified.

CONTROLBLOCKS
: specifies that control block statistics be shown.

IO
: specifies that I/O statistics be shown

LOCKS
: specifies that locks statistics be shown.

LONGIO
: if specified, I/O path statistics with I/O times greater than one second must be shown.

POOL $*pool*
: specifies that the command is performed only on disks associated with the specified storage pool.

RESET [, FORCED]

restores statistics counters to their initial values. Only super-group users (255,*nnn*) can use this attribute..

FORCED specifies that the command be executed without any interaction with the user.

SEL *state*

specifies that the command affects only devices in the specified state.

SQLMX

specifies that SQL/MX statistics be shown.

SUB { ALL | MAGNETIC }

specifies that the command affects only disks of the specified type. The default is ALL.

See "Attribute Descriptions for Disk Commands" (page 198) for descriptions of all attributes for disk commands.

STATS DISK Examples

- To display cache information about a disk:

  ```
  -> STATS $DATA00
  ```

  The display is explained in "Displaying Disk Cache Configuration Information" (page 59). To resolve disk cache problems, see "Reconfiguring Cache to Resolve Performance Problems" (page 61).

- To reset statistical counters to their initial values:

  ```
  -> STATS $DATA00, RESET
  ```

STATS SAC Command

The STATS SAC command displays statistical information about the connections to a SAC on an FCSA. The command can display information about a port in the Fibre Channel link or the SAC itself.

NOTE: The STATS SAC command is available on H06.04 and subsequent H-series RVUs.

The syntax is:

```
STATS [ / OUT file-spec / ] SAC $ZZSTO.#sac-name
      [ , PORTNAME wwn ]
      [ , RESET [ , FORCED ]        ]
```

Wild-card characters are supported.

If PORTNAME is not specified, statistics for the SAC are displayed.

OUT *file-spec*

directs all SCF output to the specified file.

#sac-name

is the name of the SAC in the form *FCSA.SAC-sac*.GRP-*group*.MOD-*module*.SLOT-*slot*. For example:

```
#FCSA.SAC-2.GRP-110.MOD-2.SLOT-5
```

PORTNAME *wwn*

specifies the worldwide name of a port in the Fibre Channel link connected to the SAC. *wwn* must be entered as a 16-character hexadecimal number without a leading %H.

RESET [, FORCED]

restores statistics counters to their initial values. Only super-group users (255,*nnn*) can use this attribute.

FORCED specifies that the command be executed without any interaction with the user.

STATS SAC Examples

- To obtain statistics for SAC 1 on the FCSA in slot 3, module 3 of group 111:

```
-> STATS SAC $ZZSTO.#FCSA.SAC-1.GRP-111.MOD-3.SLOT-3

STORAGE - Stats SAC \IO.$ZZSTO.#FCSA.SAC-1.GRP-111.MOD-3.SLOT-3

  Command timeouts........ 0         Device Changes......... 9
  Fabric Device Changes... 0         FC Length Errors....... 0
  ISP Database Changes.... 1         ISP Downloads.......... 1
  ISP Fabric LID Available 126       ISP Firmware Errors.... 0
  ISP Hardware Errors..... 0         ISP LIP Events......... 1
  LIP Errors Reported..... 1         LIP Requests initiated. 0
  LIP Reset Requests...... 0         Loop Down.............. 0
  Loop Up................. 1         Non-specific Asyncs.... 0
  Port WWN Changes........ 0         Successful Host Logins. 9
  Unsuccessful Host Logins 0
```

- To obtain statistics for the port with worldwide name 2200000C50278562 connected to SAC 2:

```
-> STATS SAC $ZZSTO.#FCSA.SAC-2.GRP-111.MOD-2.SLOT-2, &
-> PORTNAME 2200000C50278562

STORAGE - Port Stats SAC \IO.$ZZSTO.#FCSA.SAC-2.GRP-111.MOD-2.SLOT-2

  Port Name......... 2200000C50278562

   Invalid CRC........... 1144011040 Invalid Transmissions.. 608846415
   Link Failure.......... 1359347712 Loss of Signal......... 49358809
   Loss of Sync.......... 0          Sequence Protocol Error 3336992670
```

STATUS Command

The STATUS command displays current status information about an object.

Supported objects are:

- “STATUS ADAPTER Command” (page 279)
- “STATUS CLIM Command” (page 280)
- “STATUS DISK Command” (page 282)
- “STATUS MON Command” (page 284)
- “STATUS POOL Command” (page 285)
- “STATUS SAC Command” (page 285)
- “STATUS SCSI Command” (page 288)
- “STATUS SUBSYS Command” (page 288)
- “STATUS TAPE Command” (page 289)

STATUS ADAPTER Command

The STATUS ADAPTER command displays current information about a ServerNet adapter. The syntax is:

```
STATUS [ / OUT file-spec / ]
   ADAPTER $ZZSTO.#type.GRP-g.MOD-m.SLOT-s
   [ , { DETAIL | FIRMWARE | SACS | SERVERNET | VPROCS } ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

ADAPTER $ZZSTO.#*type*.GRP-*g*.MOD-*m*.SLOT-*s*

is a ServerNet adapter of type PMF, IOMF, or SNDA. *g* cannot have a leading zero.

DETAIL

displays detailed status information.

FIRMWARE

displays only the firmware currently running in each SAC.

SACS

displays detailed status for SACs but does not list devices. See "Example of a STATUS FCSA ADAPTER, SACS" (page 162).

SERVERNET

displays only the status of the adapter's ServerNet paths. This attribute applies only to Fibre Channel ServerNet adapters (FCSAs). See "Example of a STATUS FCSA ADAPTER, SERVERNET Report" (page 164).

VPROCS

displays only the full VPROC strings for firmware. See "Example of a STATUS FCSA ADAPTER, VPROCS Report" (page 164).

STATUS ADAPTER Examples

- To display the status of the PMF CRU adapter in group 01, slot 50:

  ```
  -> STATUS ADAPTER $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50
  ```

  See the display and explanation under "Example of a STATUS PMF ADAPTER Report" (page 158).

- To display detailed status of the PMF CRU adapter in group 01, slot 50:

  ```
  -> STATUS ADAPTER $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-50, DETAIL
  ```

  See the display and explanation under "Example of a Detailed STATUS PMF ADAPTER Report" (page 158).

- To display the status of the 6760 ServerNet device adapter in group 01, slot 53:

  ```
  -> STATUS ADAPTER $ZZSTO.#SNDA.GRP-1.MOD-1.SLOT-53
  ```

  See the display and explanation under "Example of a STATUS SNDA ADAPTER Report" (page 161).

STATUS CLIM Command

The STATUS command displays current storage status information about CLIM objects. The syntax is:

```
STATUS [ / OUT file-spec / ] CLIM $ZZSTO.clim-name
 , { DETAIL | SERVERNET }
| ENCRYPTION | KEYMANAGER
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

CLIM $ZZSTO.*clim-name*

is the name of the CLIM to display.

DETAIL
: returns a list of the storage devices configured to use the CLIM, including LUN and any partition numbers.

SERVERNET
: displays the status of the CLIM's storage ServerNet paths.

ENCRYPTION
: returns a list of encrypted device paths that use the CLIM, including LUN and any partition numbers. The CLIM name can include wildcards.

KEYMANAGER
: displays the status of the connection from the CLIM to key managers. The CLIM name can include wildcards.

STATUS CLIM Considerations

Additional information for CLIMs is available through $ZZCIP. For more information, see the *Cluster I/O Protocols (CIP) Configuration and Management Manual*.

STATUS CLIM Examples

- The DETAIL or SERVERNET attribute is required:

```
3-> status clim $zzsto.s1002531
STORAGE E09033 Required attribute not specified: DETAIL or SERVERNET.
```

- To display the storage devices configured to use storage CLIM $zzsto.s1002531:

```
4-> status clim $zzsto.s1002531, detail

STORAGE - Detailed Status CLIM \BLOSM2.$ZZSTO.#S1002531

 Configured Devices:
   Name           State      Substate         Primary PID  Backup PID
   $SAS01-P      *STARTED                      0,338        1,305
   $SAS02-P      *STARTED                      0,337        1,306
   $SAS03-P      *STARTED                      0,336        1,307
   $SASJBOD-P    *STARTED                      0,356        1,327
   $SASESS-P     *STARTED                      0,357        1,326
   $SAS06-P      *STARTED                      0,335        1,308
   $SAS07-P      *STARTED                      0,334        1,309
   $SAS09-P      *STARTED                      0,354        1,310
   $SAS10-P      *STARTED                      0,353        1,312
   $SAS11-P      *STARTED                      0,352        1,313
   $SAS12-P      *STARTED                      0,351        1,314
   $SAS13-P      *STARTED                      0,350        1,315
   $SAS14-P      *STARTED                      0,349        1,316
   $SAS15-P      *STARTED                      0,348        1,317
   $SAS16-P      *STARTED                      0,347        1,318
   $SAS17-P      *STARTED                      0,346        1,319
   $SAS20-P      *STARTED                      0,345        1,320
   $SAS21-P      *STARTED                      0,362        1,321
   $SAS22-P      *STARTED                      0,361        1,322
   $SAS23-P      *STARTED                      0,360        1,323
   $SAS24-P      *STARTED                      0,359        1,324
   $SAS25-P      *STARTED                      0,358        1,325
   $SAS01-MB      STARTED                      0,338        1,305
   $SAS02-MB      STARTED                      0,337        1,306
   $SAS03-MB      STARTED                      0,336        1,307
   $SAS06-MB      STARTED                      0,335        1,308
   $SAS07-MB      STARTED                      0,334        1,309
   $SE08-M       *STARTED                      0,355        1,328
   $SAS09-MB      STARTED                      0,354        1,310
   $SAS10-MB      STARTED                      0,353        1,312
   $SAS11-MB      STARTED                      0,352        1,313
```

```
$SAS12-MB      STARTED                           0,351        1,314
$SAS13-MB      STARTED                           0,350        1,315
$SAS14-MB      STARTED                           0,349        1,316
$SAS15-MB      STARTED                           0,348        1,317
$SAS16-MB      STARTED                           0,347        1,318
$SAS17-MB      STARTED                           0,346        1,319
$SASESS-MB     STOPPED    HARDDOWN               0,357        1,326
$SAS20-MB      STARTED                           0,345        1,320
$SAS21-MB      STARTED                           0,362        1,321
$SAS22-MB      STARTED                           0,361        1,322
$SAS23-MB      STARTED                           0,360        1,323
$SAS24-MB      STARTED                           0,359        1,324
$SAS25-MB      STARTED                           0,358        1,325
$SE08-B        STOPPED    HARDDOWN               0,355        1,328
```

- To display the status of the S1002533's storage ServerNet paths.

```
-> STATUS CLIM $ZZSTO.S1002533, SERVERNET
```

- To display encrypted devices that use a CLIM.

```
-> STATUS CLIM $ZZSTO.C100271, ENCRYPTION
STORAGE - Encryption Status CLIM \JUNO1.$ZZSTO.#C100271
 Encrypted Devices:
   Name              Lun   Partition
   $DATA00-P         100     1
   $SWAP00-MB        120     2
   $G6D103-MB        203
```

- To display the status of the connection from the CLIM to key managers.

```
-> STATUS CLIM $ZZSTO.$ZZSTO.#C100271, KEYMANAGER
STORAGE - KeyManager Status CLIM $ZZSTO.#C100271
KeyManager 16.107.200.150 OK
Keymanager 16.107.200.155 unavailable
```

STATUS DISK Command

The STATUS DISK command displays current status information about physical or virtual disks. The syntax is:

```
STATUS [ / OUT file-spec / ]
   DISK $disk[-P |-B |-M |-MB |-* ]
   [ , { CONFIG | CONSISTENCY | DETAIL } ]
   [ , POOL $pool ]   [ , SEL state ]
   [ , SUB { ALL | MAGNETIC | VIRTUAL } ]
   [ , ENCRYPTION [, DETAIL]]
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

DISK $*disk*[-P |-B |-M |-MB |-*]
: is the disk name and, optionally, the path (primary, backup, mirror, mirror backup, or all paths).

CONFIG
: causes a display of the values that the running disk process is currently using as operational parameters defined by certain disk configuration attributes. These values can differ from the values entered into the system configuration database. In the display, values that differ in this way are marked with an asterisk (*).

CONSISTENCY
: verifies whether the path configuration stored in the system-configuration database is identical to the path configuration used by the disk process pair and the two SIFM processes.

DETAIL	displays detailed status information.
POOL *$pool*	specifies that the command is performed only on disks associated with the specified storage pool.
SEL *state*	specifies that the command affects only devices in the specified state.
SUB { ALL | MAGNETIC | VIRTUAL }	specifies that the command affects only disks of the specified type. The default is ALL.
ENCRYPTION	displays encryption information for an encrypted disk. You cannot enter the ENCRYPTION option with the CONFIG or CONSISTENCY options.
DETAIL	displays detailed encryption information.

See "Attribute Descriptions for Disk Commands" (page 198) for descriptions of all attributes for disk commands.

STATUS DISK Examples For Physical Disks

- To display the status of all physical disks available to the system:

  ```
  -> STATUS DISK $*, SUB MAGNETIC
  ```

- To display the status of all paths of all volumes on the system:

  ```
  -> STATUS DISK $*-*
  ```

- To display the status of a disk:

  ```
  -> STATUS $SYSTEM
  ```

  See the display and explanation under "Example 1: Displaying the Status of a Mirrored Disk" (page 64) and "Example 3: Displaying the Status of a Nonmirrored Disk" (page 65).

- To display the status of all paths to a disk:

  ```
  -> STATUS DISK $SYSTEM-*
  ```

  See the display and explanation under "Example 2: Displaying the Status and State of All Paths to a Disk" (page 64).

- To display the status of all paths to all internal disks in the group 02 topology branch:

  ```
  -> STATUS $D02*-*
  ```

  The display is shown in this example in the section "Establishing a Disk Load Balance in RVUs Prior to G06.11" (page 130).

  To display the consistency information about the disk volume $PART01:

  ```
  -> STATUS DISK $PART01, CONSISTENCY

  STORAGE - Status DISK \JUNO1.$PART01, CONSISTENCY
  Path Opinion (Grp,Mod,Slt)  Sac  Device-id/Portname  Lun    Partition
  ---------------------------------------------------------------------

  -P   CONFIG  C100271                                 1039   1

  -B   CONFIG  C100273                                 1039   1

  -M   CONFIG  C100273                                 123    2

  -MB  CONFIG  C100271                                 123    2
  ```

  See the display and explanation under "Example 4: Displaying STATUS DISK, CONSISTENCY Information" (page 65).

- To display the encryption status for a disk:

```
-> STATUS DISK $DATA05-B, ENCRYPTION
```

See the examples under “Displaying Information about Encrypted Disks” (page 58).

- To display detailed encryption status for a disk:

  ```
  -> STATUS DISK $DATA05-B, ENCRYPTION, DETAIL
  ```

  See the examples under “Displaying Information about Encrypted Disks” (page 58).

STATUS DISK Examples For Virtual Disks

- To display the status of a virtual disk:

  ```
  -> STATUS $VDISK00
  ```

  See the display and explanation under “Example of a STATUS DISK Report for Virtual Disks” (page 146).

- To display detailed status of a virtual disk:

  ```
  -> STATUS $VDISK00, DETAIL
  ```

  See the display and explanation under “Example of a Detailed STATUS DISK Report for Virtual Disks” (page 147).

Attribute Descriptions for Disk Commands

See “Attribute Descriptions for Disk Commands” (page 198) for descriptions of all attributes for disk commands.

STATUS MON Command

The STATUS MON command displays current status information about the SMF master process. The syntax is:

```
STATUS [ / OUT file-spec / ] MON $ZSMS
    [ , DETAIL ]   [ , SEL state ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

MON $ZSMS

is the SMF master process.

DETAIL

displays detailed status information.

SEL *state*

specifies that the command affects only devices in the specified state.

STATUS MON Examples

- To display the current status of the SMF master process:

  ```
  -> STATUS $ZSMS
  ```

  See the display and explanation under “Example of a STATUS MON Command” (page 43).

- To display detailed status of the SMF master process:

  ```
  -> STATUS $ZSMS, DETAIL
  ```

  See the display and explanation under “Example of a Detailed STATUS MON Report” (page 43).

STATUS POOL Command

The STATUS POOL command displays current status information about the storage pool process. The syntax is:

```
STATUS [ / OUT file-spec / ] POOL $pool
    [ , DETAIL ]   [ , SEL state ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

POOL $*pool*

is the storage pool process.

DETAIL

displays detailed status information.

SEL *state*

specifies that the command affects only devices in the specified state.

STATUS POOL Examples

- To display the status of a storage pool process:

  `-> STATUS $POOL01`

  See the display and explanation under "Example of a STATUS POOL Report" (page 136).

- To display detailed status of a storage pool process:

  `-> STATUS $POOL00, DETAIL`

  See the display and explanation under "Example of a Detailed STATUS POOL Report" (page 136).

- To display the status of all storage pool processes in the STOPPED state:

  `-> STATUS POOL $*, SEL STOPPED`

STATUS SAC Command

The STATUS SAC command displays current status information about connections to a SAC on an FCSA.

The command syntax is:

```
STATUS [ /OUT file-spec / ] SAC $ZZSTO.#sac-name
             [ , attribute-spec ]
```

OUT *file-spec*

directs all SCF output to the specified file.

#sac-name

is the name of the SAC in the form *#FCSA.SAC-sac.GRP-group.MOD-module.SLOT-slot*. For example:

`#FCSA.SAC-2.GRP-110.MOD-2.SLOT-5`

attribute-spec

is one of the STATUS SAC Attributes.

STATUS SAC Attributes

```
[ , EMU          , DEVICEID (shelf, 0)                         ]
[ , EMULIST                                                    ]
[ , LOCATION     , PORTNAME wwn ]
[ , LUNLIST      , PORTNAME wwn ]
[ , LUN number ] [ , DETAIL ] ]
[ , PATHSTATUS   , DEVICEID (shelf, 0)                         ]
[ , PORTLIST   [ , PORTNAME wwn ] [ , DETAIL ]                 ]
[ , REPLACE      , DEVICEID (shelf, 99)                        ]
```

DETAIL

displays additional information when used with LUNLIST or PORTLIST.

EMU , DEVICEID (*shelf* , 0)

displays a small subset of the inventory and status information from the environmental monitoring unit (EMU).

shelf specifies the shelf number of the enclosure to which the command applies. The bay number must be 0.

EMULIST

displays the shelf worldwide names for any EMUs visible through the SAC.

LOCATION

displays the shelf and bay of the disk with the specified PORTNAME.

LUN

specifies a Logical Unit Number (LUN) in the range 0-32767.

LUNLIST

displays the LUN numbers that are available through the specified PORTNAME. If a LUN is specified, only that LUN is shown.

PATHSTATUS , DEVICEID (*shelf*, 0)

displays the status of the path to the EMU in the specified *shelf*. The bay number must be 0.

PORTLIST

displays the port worldwide names that are visible through the SAC. If a PORTNAME is specified, only that PORTNAME is shown.

PORTNAME *wwn*

specifies the worldwide name of a port entered as a 16-character hexadecimal number without a leading %H.

REPLACE, DEVICEID (*shelf* , 99)

indicates whether a replace firmware operation is being performed for the specified enclosure. The command displays a message if the $ZFC*nn* process is busy updating firmware. Otherwise, the command produces no display.

shelf specifies the shelf number of the enclosure. The bay number must be 99.

STATUS SAC Command Examples

- This command shows the worldwide names for the EMUs in the FCDMs connected to SAC 2:

```
-> STATUS SAC $ZZSTO.#FCSA.SAC-2.GRP-111.MOD-2.SLOT-1, EMULIST

STORAGE - Emulist Status SAC \OSM2.$ZZSTO.#FCSA.SAC-2.GRP-111.MOD-2.SLOT-1
 Shelf         WWN
   1     500508B3005833C7
   2     500508B3006814C5
```

- This command shows the worldwide names for all disks in enclosure 1 connected to SAC 2:

```
-> STATUS SAC $ZZSTO.#FCSA.SAC-2.GRP-111.MOD-2.SLOT-1, &
-> EMU, DEVICEID (1,0)
```

```
STORAGE - Emu Status SAC \OSM2.$ZZSTO.#FCSA.SAC-2.GRP-111.MOD-2.SLOT-1
 Bay    Port A WWN    State      Port B WWN    State
  1  2100000C502741B6   1     2200000C502741B6   1
  2  2100000C50274285   1     2200000C50274285   1
  3  2100000C5016D461   1     2200000C5016D461   1
  4  2100000C502741E8   1     2200000C502741E8   1
```

- This command shows the worldwide names of the ports connected to SAC 1:

```
-> STATUS SAC $ZZSTO.#FCSA.SAC-1.GRP-112.MOD-2.SLOT-4, PORTLIST

STORAGE - Portlist Status SAC \IO.$ZZSTO.#FCSA.SAC-1.GRP-112.MOD-2.SLOT-4
 50060B00001CE594 170400   50060B00001CE596 170500   50060B00001CE5F8 170200
 50060B00001CE5FA 170300   50060E8004289406 120400   50060E8004289416 121700
 50060E8004289426 120500   50060E8004289436 121800
```

- This command shows the LUNs available through PORTNAME 50060E8004289416:

```
-> STATUS SAC $ZZSTO.#FCSA.SAC-1.GRP-112.MOD-2.SLOT-4, &
-> LUNLIST, PORTNAME 50060E8004289416

STORAGE - Lunlist Status SAC \IO.$ZZSTO.#FCSA.SAC-1.GRP-112.MOD-2.SLOT-4


------From-----Through  ------From-----Through  ------From-----Through
         4 -         7          12 -        19          32 -        41
```

- This command displays details for the ports connected to SAC 1:

```
-> STATUS SAC $ZZSTO.#FCSA.SAC-1.GRP-112.MOD-3.SLOT-3,&
-> PORTLIST, DETAIL

STORAGE - Detailed Portlist Status SAC \IO.$ZZSTO.#FCSA.SAC-1.GRP-112.MOD-3.SLOT-3

    PortId Port Name        Node Name        Connection  ADT State  COS
    000039 2100000C50278389 2000000C50278389 Loop        Login       00

    First 64 bytes of Symbolic Name:
    0000000000000000 0000000000000000 0000000000000000 0000000000000000
    0000000000000000 0000000000000000 0000000000000000 0000000000000000

    PortId Port Name        Node Name        Connection  ADT State  COS
    000036 2100000C50278562 2000000C50278562 Loop        Login       00

    First 64 bytes of Symbolic Name:
    0000000000000000 0000000000000000 0000000000000000 0000000000000000
    0000000000000000 0000000000000000 0000000000000000 0000000000000000

    PortId Port Name        Node Name        Connection  ADT State  COS
    000035 2100000C502785CB 2000000C502785CB Loop        Login       00

    First 64 bytes of Symbolic Name:
    0000000000000000 0000000000000000 0000000000000000 0000000000000000
    0000000000000000 0000000000000000 0000000000000000 0000000000000000

    PortId Port Name        Node Name        Connection  ADT State  COS
    00003A 2100000C50278A79 2000000C50278A79 Loop        Login       00

    First 64 bytes of Symbolic Name:
    0000000000000000 0000000000000000 0000000000000000 0000000000000000
    0000000000000000 0000000000000000 0000000000000000 0000000000000000

    PortId Port Name        Node Name        Connection  ADT State  COS
    000034 2100000C50278A7C 2000000C50278A7C Loop        Login       00

    First 64 bytes of Symbolic Name:
    0000000000000000 0000000000000000 0000000000000000 0000000000000000
    0000000000000000 0000000000000000 0000000000000000 0000000000000000

    PortId Port Name        Node Name        Connection  ADT State  COS
    000031 2100000C50278DB7 2000000C50278DB7 Loop        Login       00

    First 64 bytes of Symbolic Name:
    0000000000000000 0000000000000000 0000000000000000 0000000000000000
    0000000000000000 0000000000000000 0000000000000000 0000000000000000

    PortId Port Name        Node Name        Connection  ADT State  COS
```

```
00002E 2100000C504CFB71 2000000C504CFB71 Loop       Login       00

    First 64 bytes of Symbolic Name:
0000000000000000 0000000000000000 0000000000000000 0000000000000000
0000000000000000 0000000000000000 0000000000000000 0000000000000000

PortId Port Name        Node Name        Connection  ADT State  COS
000032 2100000C504CFCA9 2000000C504CFCA9 Loop        Login       00

First 64 bytes of Symbolic Name:
0000000000000000 0000000000000000 0000000000000000 0000000000000000
0000000000000000 0000000000000000 0000000000000000 0000000000000000

PortId Port Name        Node Name        Connection  ADT State  COS
000033 2100000C504CFD6B 2000000C504CFD6B Loop        Login       00

First 64 bytes of Symbolic Name:
0000000000000000 0000000000000000 0000000000000000 0000000000000000
0000000000000000 0000000000000000 0000000000000000 0000000000000000
```

STATUS SCSI Command

The STATUS SCSI command displays current status information about Open SCSI devices. The syntax is:

```
STATUS [ / OUT file-spec / ] SCSI $SCSI-device[-P |-B ]
    [ , DETAIL ]  [ , SEL state ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

SCSI $*SCSI-device* [-P |-B]

is the name of the Open SCSI IOP and, optionally, the path (primary or backup).

DETAIL

displays detailed status information.

SEL *state*

specifies that the command affects only devices in the specified state.

STATUS SCSI Examples

- To display the status of all Open SCSI devices on the system:

  `-> STATUS SCSI $*`

- To display the status of an Open SCSI device:

  `-> STATUS $DEV0`

  See the display and explanation under “Example of a STATUS SCSI Report” (page 173).

- To display detailed status of an Open SCSI device:

  `-> STATUS $DEV00, DETAIL`

  See the display and explanation under “Example of a STATUS SCSI Report” (page 173).

STATUS SUBSYS Command

This subsection describes the STATUS SUBSYS command. The command syntax is:

```
STATUS [ / OUT file-spec / ] SUBSYS $ZZSTO  [ , DETAIL ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

SUBSYS $ZZSTO

is the storage subsystem manager.

DETAIL

displays detailed status information.

STATUS SUBSYS Consideration

The information showed by INFO SUBSYS and STATUS SUBSYS should always be the same. Contact your service provider if the information shown in these displays differs.

STATUS SUBSYS Example

To display the summary status of storage subsystem manager:

```
-> STATUS SUBSYS $ZZSTO
```

See the display and explanation under “Using the Storage Subsystem Manager” (page 40).

STATUS TAPE Command

The STATUS TAPE command displays information about tape devices. The command syntax is:

```
STATUS [ / OUT file-spec / ] TAPE $tape  [ , DETAIL ]
[ , ENCRYPTION [, DETAIL]]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

TAPE $*tape*

is the name of the tape device.

DETAIL

displays detailed status information.

ENCRYPTION

displays encryption information for an encrypted tape drive.

DETAIL

displays detailed encryption information.

STATUS TAPE Examples

- To display the status of a tape drive:

  ```
  -> STATUS $TAPE0
  ```

  See the display and explanation under “Example of a STATUS TAPE Report” (page 182).

- To display detailed status of a tape drive:

  ```
  -> STATUS $TAPE0, DETAIL
  ```

  See the display and explanation under “Example of a Detailed STATUS TAPE Report” (page 182).

- To display encryption status of a tape drive:

  ```
  Status TAPE $TAPE07, ENCRYPTION, DETAIL
  ```

  See the display and explanation under “Example of a STATUS TAPE, ENCRYPTION Report” (page 184).

STOP Command

The STOP command terminates access to a storage device in a orderly manner. When the STOP command finishes, configured devices are left in the STOPPED state, substate DOWN. The devices remain in the system configuration database.

Supported objects are:

- "STOP DISK Command" (page 290)
- "STOP MON Command" (page 291)
- "STOP POOL Command" (page 291)
- "STOP SCSI Command" (page 292)
- "STOP TAPE Command" (page 292)

STOP is a sensitive command.

STOP DISK Command

The STOP DISK command terminates access to a disk. The command syntax is:

```
STOP [ / OUT file-spec / ]  DISK $disk[-P |-B |-M |-MB ]
   [ , FORCED ]  [ , POOL $pool ]  [ , SEL state ]
   [ , SUB { ALL | MAGNETIC | VIRTUAL } ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

DISK $*disk*[-P |-B |-M |-MB]

is the disk name and, optionally, the path (primary, backup, mirror, or mirror backup).

FORCED

specifies that the command be executed without any interaction with the user, even if the command stops the last path to the device or files are open on the device.

If you use this attribute on a physical disk, you must first stop all processes that use the disk to store object code (programs) or swap files. Otherwise, a %5113 halt could occur.

POOL *$pool*

specifies that the command is performed only on disks associated with the specified storage pool.

SEL *state*

specifies that the command affects only devices in the specified state.

SUB { ALL | MAGNETIC | VIRTUAL }

specifies that the command affects only disks of the specified type. The default is ALL.

See "Attribute Descriptions for Disk Commands" (page 198) for descriptions of all attributes for disk commands.

STOP DISK Examples

See the procedure for "Stopping a Disk" (page 101) and the procedure for "Stopping a Virtual Disk" (page 151).

- To stop access to the mirror half of a volume:

```
-> STOP $DATA01-M
```

- To stop a disk (even if there are open files) without further interaction with the operator:

```
-> STOP $DATA00, FORCED
```

- To stop all paths to a disk:

  ```
  -> STOP $DATA00
  ```

- To stop a virtual disk:

  ```
  -> STOP $VDISK00
  ```

STOP MON Command

The STOP MON command stops access to the SMF master process. The syntax is:

```
STOP [ / OUT file-spec / ] MON $ZSMS
    [ , FORCED ]   [ , SEL state ]
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

MON $ZSMS
: is the SMF master process.

FORCED
: specifies that the command be executed without any interaction with the user.

  If you use this attribute, you must first stop all processes that use the disk to store object code (programs) or swap files. Otherwise, a %5113 halt could occur.

SEL *state*
: specifies that the command affects only devices in the specified state.

STOP MON Example

See the procedure for "Stopping the SMF Master Process" (page 46).

To stop access to the SMF master process:

```
-> STOP $ZSMS
```

STOP POOL Command

The STOP POOL command stops access to the specified storage pool. The syntax is:

```
STOP [ / OUT file-spec / ] POOL $pool
    [ , FORCED ]   [ , SEL state ]
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

POOL $*pool*
: is the storage pool process.

FORCED
: specifies that the command be executed without any interaction with the user.

  If you use this attribute, you must first stop all processes that use the disk to store object code (programs) or swap files. Otherwise, a %5113 halt could occur.

SEL *state*
: specifies that the command affects only devices in the specified state.

STOP POOL Example

See “Stopping a Storage Pool” (page 141).

To stop access to a storage pool process:

```
-> STOP $POOL00
```

STOP SCSI Command

The STOP SCSI command stops access to an Open SCSI device. The syntax is:

```
STOP [ / OUT file-spec / ]  SCSI $SCSI-device[-P |-B ]
   [ , FORCED ]  [ , SEL state ]
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

SCSI $*SCSI-device* [-P |-B]
: is the name of the Open SCSI IOP and, optionally, the path (primary or backup).

FORCED
: specifies that the command be executed without any interaction with the user.

  If you use this attribute, you must first stop all processes that use the disk to store object code (programs) or swap files.Otherwise, a %5113 halt could occur.

SEL *state*
: specifies that the command affects only devices in the specified state.

STOP SCSI Examples

See “Stopping an Open SCSI Device” (page 177).

- To stop access to an Open SCSI device:

  ```
  -> STOP $DEV00
  ```

- To stop access to the backup path of an Open SCSI device:

  ```
  -> STOP $DEV1-B
  ```

STOP TAPE Command

This subsection describes the STOP TAPE command. The STOP TAPE command stops access to the specified tape drive. The command syntax is:

```
STOP [ / OUT file-spec / ] TAPE $tape
   [ , FORCED ]  [ , SEL state ]
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

TAPE $*tape*
: is the name of the tape device.

FORCED
: specifies that the command be executed without any interaction with the user, even if the command stops the last path to the device or files are open on the device.

  If you use this attribute, you must first stop all processes that use the disk to store object code (programs) or swap files. Otherwise, a %5113 halt could occur.

SEL *state*

specifies that the command affects only devices in the specified state.

STOP TAPE Examples

See the procedure for "Stopping a Tape Drive" (page 189).

To stop access to a tape drive:

```
-> STOP $TAPE3
```

- To stop access to all tape drives whose names start with $TAPE:

  ```
  -> STOP $TAPE*
  ```

STOPOPENS Command

The STOPOPENS command prevents any additional opens to an object.

STOPOPENS is a sensitive command.

STOPOPENS DISK Command

The STOPOPENS command prevents applications from opening files on the specified physical or virtual disk volume. Use the ALLOWOPENS command to let applications open files on the disk again.

CAUTION: Do not issue a STOPOPENS DISK command on the system disk. Issuing a STOPOPENS DISK command on the system disk makes it inaccessible to file-open attempts. If you do issue a STOPOPENS DISK command on the system disk, do not exit SCF. Immediately enter an ALLOWOPENS DISK, SUPERONLY command. Otherwise, the system disk becomes inaccessible when you exit SCF.

```
STOPOPENS [ / OUT file-spec / ] DISK $disk
    [ , POOL $pool ]   [ , SEL state ]
    [ , SUB { ALL | MAGNETIC | VIRTUAL } ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

DISK $*disk*

is the name of the disk.

POOL *$pool*

specifies that the command is performed only on disks associated with the specified storage pool.

SEL *state*

specifies that the command affects only devices in the specified state.

SUB { ALL | MAGNETIC | VIRTUAL }

specifies that the command affects only disks of the specified type. The default is ALL.

See "Attribute Descriptions for Disk Commands" (page 198) for descriptions of all attributes for disk commands.

STOPOPENS DISK Examples

See "Preventing File Opens on a Disk" (page 109). You cannot issue a STOPOPENS DISK command to a virtual disk. To prevent new file-system opens on a physical disk:

```
-> STOPOPENS $DATA00
```

SWITCH Command

The SWITCH command designates the active a path to a device.

Supported objects are:

- "SWITCH ADAPTER Command" (page 294)
- "SWITCH CLIM Command" (page 294)
- "SWITCH DISK Command" (page 295)
- "SWITCH SCSI Command" (page 296)

SWITCH is a sensitive command.

SWITCH ADAPTER Command

The SWITCH ADAPTER command moves all device paths from and to a SAC on an adapter, as part of adapter replacement. The syntax is:

```
SWITCH [ / OUT file-spec / ]
   ADAPTER $ZZSTO.#type.GRP-g.MOD-m.SLOT-s
    , { AWAY | DEFAULT }   [ , FORCED ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

ADAPTER $ZZSTO.#*type*.GRP-*g*.MOD-*m*.SLOT-*s*

is a ServerNet adapter of *type* PMF, IOMF, or SNDA. *g* cannot have a leading zero.

AWAY [, FORCED]

specifies that all active paths using the specified adapter become inactive, if possible.

The FORCED attribute stops all paths to the specified adapter after attempting to make them inactive.

DEFAULT [, FORCED]

specifies that all -P and -M paths using the specified adapter become active, if possible, and all -B and -MB paths become inactive if possible.

The FORCED attribute starts all paths to the specified adapter before attempting to make them active.

SWITCH ADAPTER Examples

- To stop all data paths to a PMF adapter before removing it:

```
-> SWITCH ADAPTER $ZZSTO.#PMF.GRP-1.MOD-1.SLOT-55, &
-> AWAY, FORCED
```

- To restore data paths to an adapter after inserting it:

```
-> SWITCH ADAPTER $ZZSTO.#SNDA.GRP-11.MOD-1.SLOT-54, DEFAULT
```

SWITCH CLIM Command

The SWITCH CLIM command initiates disk path switches for disks configured through the specified CLIM. The syntax is:

```
SWITCH [ / OUT file-spec / ] CLIM $ZZSTO.clim-name
        , { AWAY | DEFAULT }
          [ , FORCED ]
```

Wild-card characters are supported.

OUT *file-spec*

directs all SCF output to the specified file.

CLIM $ZZSTO.*clim-name*

is the name of the CLIM.

AWAY

specifies that all active paths using the specified CLIM be switched to inactive, if possible.

AWAY, FORCED

specifies that all paths using the specified CLIM be stopped. This may stop the last path to a device. This command is useful prior to disconnecting a CLIM.

DEFAULT

specifies that all -P and -M paths using the specified CLIM be switched to active, if possible, and all -B and -MB paths be switched to inactive, if possible.

DEFAULT, FORCED

specifies that all paths using the specified CLIM be started, if possible, before performing switches to default active paths. This will not initiate a revive of a mirrored disk. This command is useful after connecting a CLIM.

SWITCH CLIM Considerations

- The SWITCH CLIM command runs under $ZZSTO and initiates disk path switches for disks configured through the CLIM. It sometimes takes paths up or down.
- If the SWITCH CLIM command is entered correctly, the Storage Subsystem Manager generates an EMS message that reports the command, the time it was executed, the terminal from which the command was entered, and the group and user numbers of the user issuing the command.
- Using wildcards in a SWITCH CLIM, AWAY command might produce unexpected results. If the wildcard name matches CLIMs on both sides of a device, the first matched name will switch usage away from the first CLIM, but unless you also specified FORCED, the second matched name will switch usage away from the second CLIM and back onto the first CLIM. If you specify FORCED, both paths to the disk will be downed and the device will become inaccessible.
- The $ZZCIP subsystem also supports a SWITCH CLIM command. For syntax, see the *Cluster I/O Protocols (CIP) Configuration and Management Manual*.

SWITCH CLIM Examples

- Switch all paths that use S1002533 to inactive, if possible:

```
$ZZSTO 4-> SWITCH CLIM S1002533, AWAY
STORAGE E00096 Unable to perform  42 of the requested path switches:
$SAS01-B $ SAS02-B $SAS03-B $SASESS-B $SAS06-B $SAS07-B $SE08-P
$ZZSTO
5->
```

- Stop all paths that use S1002533:

```
-> SWITCH CLIM $ZZSTO.S1002533, AWAY, FORCED
```

- Switch all -P and -M paths that use S1002533 to active, and switch all -B and -MB paths to inactive:

```
-> SWITCH CLIM $ZZSTO.S1002533, DEFAULT
```

SWITCH DISK Command

The SWITCH DISK command designates the active paths to a disk, as well as the preferred SAC path for any disk accessible through dual paths. The syntax is:

```
SWITCH [ / OUT file-spec / ] DISK $disk {-P |-B |-M |-MB }
    [ , POOL $pool ]   [ , SEL state ]
    [ , SUB { ALL | MAGNETIC | VIRTUAL } ]
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

DISK $*disk*{-P |-B |-M |-MB }
: is the disk name and the path (primary, backup, mirror, or mirror backup). A single path specification is required.

POOL *$pool*
: specifies that the command is performed only on disks associated with the specified storage pool.

SEL *state*
: specifies that the command affects only devices in the specified state.

SUB { ALL | MAGNETIC | VIRTUAL }
: specifies that the command affects only disks of the specified type. The default is ALL.

See “Attribute Descriptions for Disk Commands” (page 198) for descriptions of all attributes for disk commands.

SWITCH DISK Examples

The SWITCH DISK command is not supported for virtual disks. See “Changing the Active Path for a Disk” (page 112).

- To designate the mirror path of a volume as the primary disk path:

  ```
  -> SWITCH $DATA00-M
  ```

- To designate the backup path of a volume as the primary disk path:

  ```
  -> SWITCH $DATA-B
  ```

- To combine both of the previous commands:

  ```
  -> SWITCH DISK ($DATA00-M, $DATA-B)
  ```

SWITCH SCSI Command

The SWITCH SCSI command designates the active path to an Open SCSI device and, in doing so, designates the preferred SAC path for the Open SCSI device. The syntax is:

```
SWITCH [ / OUT file-spec / ]  SCSI $SCSI-device{-P |-B }
    [ , SEL state ]
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

SCSI $*SCSI-device*{-P |-B }
: is the name of the Open SCSI IOP and the path (primary or backup).

SEL *state*
: specifies that the command affects only devices in the specified state.

SWITCH SCSI Examples

See the procedure for "Changing the Active Data Path for an Open SCSI Device" (page 177).

- To designate the backup path of an Open SCSI device as the primary access path:

```
-> SWITCH $DEV01-B
```

- To designate the primary path of an Open SCSI device as the primary access path:

```
-> SWITCH $DEV02-P
```

VERSION Command

The VERSION command displays current version information.

VERSION SUBSYS Command

The VERSION SUBSYS command displays current version information for the subsystem manager. The syntax is:

```
VERSION  [ / OUT file-spec / ] SUBSYS $ZZSTO  [ , DETAIL ]
```

Wild-card characters are supported.

OUT *file-spec*
: directs all SCF output to the specified file.

SUBSYS $ZZSTO
: is the storage subsystem manager.

DETAIL
: displays version information about the storage subsystem, Guardian, and SCF Kernel components of the system.

VERSION SUBSYS Examples

- To display version information about the storage subsystem manager process:

```
-> VERSION SUBSYS $ZZSTO

VERSION SUBSYS \ALM171.$ZZSTO: STORAGE (MGR) - T1083G05 - (01FEB01) - AAT - (12DEC00)
```

- To display detailed version information about the storage subsystem manager process:

```
-> VERSION $ZZSTO, DETAIL

Detailed VERSION SUBSYS \ALM171.$ZZSTO
  SYSTEM \ALM171
    STORAGE (MGR) - T1083G05 - (01FEB01) - AAT - (12DEC00)
    GUARDIAN - T9050 - (Q06)
    SCF KERNEL - T9082G02 - (26JUN00) (20MAR00)
    STORAGE PM - T1082G05 - (01FEB01) - AAL - (12DEC00)
```

A Storage Subsystem Error Messages

This appendix describes where to get more information about error messages.

For error messages associated with the storage subsystem, refer to the *Operator Messages Manual*. For SCF messages for all subsystems, refer to the *SCF Reference Manual for G-Series RVUs*.

If You Have to Contact Your Service Provider

If the recovery for an error message indicates to contact your service provider or the Global NonStop Support Center (GNSC), supply this following log file:

1. To collect the following displays into a single file:

   ```
   -> LOG $DATA00.LOGINFO.OCT14
   ```

2. To collect information about the product versions of the SCF components, a list of the product modules on your system, and information about any product modules running when the error occurred:

   ```
   -> LISTPM
   ```

3. To collect information about the SCF environment that was present when the error occurred:

   ```
   -> ENV
   ```

   If the error forced SCF to terminate, be sure to specify any environmental characteristics that were present when the error occurred.

4. To capture the contents of the Subsystem Programmatic Interface (SPI) command and response buffers:

   ```
   -> DETAIL CMDBUFFER ON
   -> DETAIL RSPBUFFER ON
   ```

5. Reproduce the sequence of commands that produced the SCF error.
6. Close the log file:

   ```
   -> LOG
   ```

Handling File-System Error Messages

When you encounter file-system error messages that display a number, use one of these methods to get a description of the error:

- Use online help to display it. For example, to display help text associated with file-system error 66:

  ```
  -> error 66
  ```

- Look up the error number in the *Guardian Procedure Calls Reference Manual*.

Storage Subsystem Error Messages

Following are subsystem error message:

00001

```
STORAGE 00001 Too many object names present in the command.
Object names : list
```

Cause

You specified more than 30 object names in the command.

Effect

The command is not executed.

Recovery

Reissue the command with fewer than 31 object names specified.

00002

`STORAGE 00002` *`command`* `rejected, file system error:` *`nnnnn`*

command
: is the SCF command you issued to a disk or tape device.

nnnnn
: is the file-system error number. To get more information about the error, see "Handling File-System Error Messages" (page 298).

Cause

The command was rejected by the I/O process.

Effect

The command is not executed.

Recovery

Check the returned file-system error to determine what to do next.

00003

`STORAGE 00003 An empty response was received from the Storage subsystem manager: $ZZSTO`

Cause

Response type ZSPI-VAL-ERR-AND-WARN was specified by the server but the reply buffer contained no error nor warning.

Effect

The command is not executed.

Recovery

Informational message only; no corrective action is required.

00004

`STORAGE 00004 Token conflict in requester buffer, token:` *`tokname,`* `object name:` *`objname`*

tokname
: is the name of the token issued with the command.

objname
: is the object name you specified with the command.

Cause

The specified token conflicts with other tokens in the requester buffer.

Effect

The command is not executed.

Recovery

Correct and reissue the command.

00005

`STORAGE 00005  Object type and object name mismatched, object type:` *`objtype`*`, object name:` *`objname`*

objtype
: is the object type you specified with the command.

objname
: is the object name you specified with the command.

Cause

The object name you specified does not match the object type you specified.

Effect

The command is not executed.

Recovery

Correct the command object type or object name and reissue the command.

00006

`STORAGE 00006 INTERNAL ERROR: Case value out of range.`

Cause

An invalid case value was generated with no associated case label.

Effect

The command is not executed.

Recovery

Report this problem immediately. See "If You Have to Contact Your Service Provider" (page 298)
.

00007

`STORAGE 00007 Duplicate attribute:` *`attribute`*

`attribute`
: is the attribute that you specified more than once in the command.

Cause

You entered the same attribute twice in the command.

Effect

The command is not executed.

Recovery

Remove the duplicate attribute and reissue the command.

00008

`STORAGE 00008 Invalid attribute or attribute value:` *`attribute`*

`attribute`
: is the invalid attribute you specified with the command.

Cause

You specified either:

- An invalid value for an attribute
- An invalid attribute for the object

Effect

The command is not executed.

Recovery

Correct the attribute or value and reissue the command.

00009

`STORAGE 00009 All paths to the` *`objtype devname`* `are down.`

`objtype`
: is the object type of the object you specified with the command.

`devname`
: is the name of the device you specified in the command.

Cause

All the paths to the given device can be down as a result of:

- A hardware error
- The operator stopping the device
- The device being serviced

Effect

The command is not executed.

Recovery

1. Verify the state of the device using the SCF STATUS command.
2. If needed, use the SCF RESET command to change the state of the device before retrying the operation.

00010

`STORAGE 00010 Negative response received from Storage Subsystem Manager.`

Cause

The command was rejected by the storage subsystem manager process ($ZZSTO). This situation might be caused by mismatched versions of the storage subsystem manager process and SCF.

Effect

The command is not executed.

Recovery

Check product versions with the SCF VERSION command and install the correct software, if necessary.

00011

`STORAGE 00011 Not supported by the down-version system (`*`vernum`*`).`

`vernum`

is the version number of the down-version system.

Cause

The command was rejected by the storage subsystem manager process ($ZZSTO) because the information requested by the command cannot be obtained from an older system.

Effect

The command is not executed.

Recovery

To resolve the version mismatch, contact your service provider.

00012

`STORAGE 00012 Wrong state for `*`objtype devname`*`, state: `*`state`*

`objtype`

is the object type of the object you specified with the command.

`devname`

is the name of the device you specified in the command.

`state`

is the current state of the device you specified in the command.

Cause

The device is in a state that prevents it from performing the operation requested. This error can occur when you perform online as well as offline re-configuration. You cannot perform an online reconfiguration of a mirror drive while TSM ($ZTSM) is running.

Effect

The command is not executed.

Recovery

Verify the state of the device using the STATUS command. If needed, use the RESET command to change the state of the device before retrying the operation. If online reconfiguration of a mirror drive is desired, your might need to terminate the $ZTSM process using ABORT PROCESS $ZZKRN.#TSM-SRM. Restart $ZTSM after the online reconfiguration is complete.

00013

`STORAGE 00013 Invalid object name format,` *`cmdformat`* `expected.`

`cmdformat`
: is the command format the storage subsystem was expecting.

Cause

The command was rejected by the storage subsystem manager process ($ZZSTO) because the format of the object name was not specified properly.

Effect

The command is not executed.

Recovery

Correct the command using the proper syntax.

00014

`STORAGE 00014 Internal error: Result code` *`resultcode`* `returned from call to` *`procname`*`.`

`resultcode`
: is the result code that was returned from the procedure call.

`procname`
: is the name of the procedure call that failed.

Cause

An internal error was caused by an unexpected return code from a system procedure.

Effect

The command is not executed.

Recovery

Report this problem immediately. See "If You Have to Contact Your Service Provider" (page 298)
.

00015

`STORAGE 00015 Unexpected error returned from $ZCNF, error:` *`err-num`*`, error detail:` *`err-detailnum`* `(`*`errdesc`*`).`

`err-num`
: is the error number returned from the $ZCNF process.

`err-detailnum`
: is the error detail number returned from the $ZCNF process.

`errdesc`
: is the description of the current error.

Cause

A system configuration database is corrupted.

Effect

The command is not executed.

Recovery

Verify the database record using the SCF INFO command. If needed, reload the system using a saved version of the system configuration database. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00016

`STORAGE 00016 Unexpected error returned from system function, error: `*`err-num`*`, error detail: `*`err-detailnum`*` (`*`errfunc`*`).`

err-num
: is the error number returned from a system procedure.

err-detailnum
: is the error detail number returned from a system procedure.

errfunc
: is the name of the system function that returns the unexpected error.

Cause

An unexpected error was returned from a system procedure that was called by the storage subsystem manager process ($ZZSTO).

Effect

The command is not executed.

Recovery

Report this problem immediately. See "If You Have to Contact Your Service Provider" (page 298).

00017

`STORAGE 00017 Unexpected error returned from CONFIG_GETINFO_, error: `*`err-num`*`, error detail: `*`err-detailnum`*`.`

err-num
: is the error number returned from the CONFIG_GETINFO_ system procedure.

err-detailnum
: is the error detail number returned from the CONFIG_GETINFO_ system procedure.

Cause

An unexpected error was returned from the CONFIG_GETINFO_ system procedure when it was called by the storage subsystem manager process ($ZZSTO).

Effect

The command is not executed.

Recovery

Retry the command. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00018

`STORAGE 00018 `*`objtype devname`*` does not exist: `*`text`*

objtype
: is the object type of the object you specified with the command.

devname
: is the name of the device you specified in the command.

text
: is additional information about the error.

Cause

This error can be caused by different conditions:

- The processor or the pair of processors that the I/O process (IOP) is configured for are not started.
- The IOP is not running.
- The configuration is not configured in the configuration database.

Effect

The command is not executed.

Recovery

The recovery action depends on the cause:

1. Use the SCF START command. If the processor is not down, the START command will restart the IOP process.
2. If the processors are down, reload the processors. For instructions about how to reload processors, see the *NonStop S-Series Operations Guide*.
3. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00019

```
STORAGE 00019  Invalid processor count specified in access list.
```

Cause

You specified too many processors in the command.

Effect

The command is not executed.

Recovery

Correct the number of processors specified and repeat the command.

00020

```
STORAGE 00020 Non-supported CRU type inserted in location.
```

Cause

The customer-replaceable unit (CRU) installed in the specified location is not supported by the storage subsystem.

Effect

The command is not executed.

Recovery

Move the CRU into the correct location. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00021

```
STORAGE 00021 Required attribute not specified: PRIMARYCPU
```

Cause

You did not specify the PRIMARYCPU attribute with the command.

Effect

The command is not executed.

Recovery

Reissue the command with the PRIMARYCPU attribute.

00022

```
STORAGE 00022 Required attribute not specified: BACKUPCPU
```

Cause

You did not specify the BACKUPCPU attribute with the command.

Effect

The command is not executed.

Recovery

Reissue the command with the BACKUPCPU attribute.

00023

```
STORAGE 00023 Required attribute not specified: LOCATION
```

Cause

You did not specify the LOCATION attribute with the command.

Effect

The command is not executed.

Recovery

Reissue the command with the LOCATION attribute.

00024

```
STORAGE 00024 Required attribute not specified: PRODUCT
```

Cause

You did not specify the PRODUCT attribute with the command.

Effect

The command is not executed.

Recovery

Reissue the command with the PRODUCT attribute.

00025

```
STORAGE 00025 PRIMARYLOCATION: Invalid value specified.
```

Cause

You specified an invalid value for the PRIMARYLOCATION attribute for the disk. The location must specify the group, module, and slot location of the device. For example:

```
-> ADD DISK $DATA14, SENDTO STORAGE, PRIMARYLOCATION (1,1,9)
```

Effect

The command is not executed.

Recovery

Reissue the command specifying a valid location for PRIMARYLOCATION.

00026

```
STORAGE 00026 MIRRORLOCATION: Invalid value specified.
```

Cause

You specified an invalid value for the MIRRORLOCATION attribute for the disk. Currently, the PRIMARYLOCATION and MIRRORLOCATION attributes must specify the same group. The MIRRORLOCATION must specify the group, module, and slot location of the mirror disk. For example:

```
-> ALTER DISK $DATA04, MIRRORLOCATION (1,1,8)
```

Effect

The command is not executed.

Reissue the command specifying a valid location for MIRRORLOCATION.

00027

`STORAGE 00027 PRIMARYLOCATION and MIRRORLOCATION are equal.`

Cause

You specified the same location for both the PRIMARYLOCATION and MIRRORLOCATION attributes.

Effect

The command is not executed.

Recovery

Reissue the command specifying the correct values for PRIMARYLOCATION and MIRRORLOCATION.

00028

`STORAGE 00028 Negative response from SCSI Interface Module process, error:` *`errnum`* `(`*`errdesc`*`).`

errnum
: is the error number provided by the SCSI interface module.

errdesc
: is the description of that error.

Cause

The SCSI interface module process is reporting a problem.

Effect

The command is not executed.

Recovery

Check the processing environment for the cause of the error and retry the command. For example, you should verify the processors are running and check the EMS log for errors. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00029

`STORAGE 00029` *`objtype devname`* `is not present in location (`*`grp,mod,slot`*`)`

objtype
: is the object type of the object you specified with the command.

devname
: is the name of the device you specified in the command.

grp,mod,slot
: is the group, module, and slot location of the device you specified in the command.

Cause

The device specified in a command is not physically present in the location.

Effect

The command is not executed.

Recovery

Insert the device and reissue the command. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00030

`STORAGE 00030` *`objtype devname`* `is not powered on, location (`*`grp,mod,slot`*`)`

objtype
: is the object type of the object you specified with the command.

devname

is the name of the device you specified in the command.

grp,mod,slot

is the group, module, and slot location of the device you specified in the command.

Cause

The specified device is not powered on.

Effect

The command is not executed.

Recovery

Power on the device and reissue the command. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00031

`STORAGE 00031 System configuration database ($SYSTEM.ZSYSCONF.CONFIG) is locked.`

Cause

The system configuration database is busy and therefore temporarily locked.

Effect

The command is not executed.

Recovery

Wait a short while and reissue the command. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00032

STORAGE 00032 None of the configured processors for *objtype devname* are available

objtype

is the object type of the object you specified with the command.

devname

is the name of the device you specified in the command.

Cause

The processors configured for the device are not functioning.

Effect

The command is not executed.

Recovery

Reload one or more of the configured processors and reissue the command.

00033

STORAGE 00033 *objtype devname* is not configured.

objtype

is the object type of the object you specified with the command.

devname

is the name of the device you specified in the command.

Cause

The device specified in a command was installed in the system, but it was not configured.

Effect

The command is not executed.

Recovery

Add the device to the system configuration using the SCF ADD command. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00034

`STORAGE 00034` *`command`* `failed. Mismatch between adapter ACCESSLIST. Parameter:` *`value`*`.`

command
: is the command you specified.

value
: is the value you specified in the command.

Cause

The adapters controlling the device specified in the command are not controlled by the same processors.

Effect

The command is not executed.

Recovery

Correct the values and retry the command. The adapters controlling the device must be controlled by the same processors.

00035

`STORAGE 00035` *`command`* `failed. Mismatch between adapter product numbers.`

command
: is the command you specified.

Cause

The adapters controlling the device specified in the command do not have the same product number.

Effect

The command is not executed.

Recovery

Correct the product numbers and retry the command. The adapters controlling the device must have the same product number.

00036

`STORAGE 00036` *`command`* `failed.` *`processor`* `cannot access the adapters controlling the device.`

command
: is the command you specified.

processor
: is the attribute you used in the command to specify the primary processor (PRIMARYCPU) or backup processor (BACKUPCPU).

Cause

You specified an invalid processor number in the command. The specified processor cannot access the adapters controlling the device.

Effect

The command is not executed.

Recovery

Reissue the command with the correct processor number. The adapters controlling the device must be accessible from the processors specified for both the BACKUPCPU and PRIMARYCPU.

00037

`STORAGE 00037` *`command`* `failed. Cannot find hardware in specified parameter (location, primary location, mirror location).`

command
: is the command you specified.

Cause

You specified different group values in the PRIMARYLOCATION and MIRRORLOCATION attributes.

Effect

The command is not executed.

Recovery

Reissue the command using the same group value for both the MIRRORLOCATION and PRIMARYLOCATION attributes.

00038

`STORAGE 00038 *objtype devname* is not running.`

objtype
: is the object type of the object you specified with the command.

devname
: is the name of the device you specified in the command.

Cause

A designated processor for the device is not running.

Effect

The command is not executed.

Recovery

Verify the processing environment for the device exists and retry the command. For example, you should verify the processors are running and check the EMS log for errors.

00039

`STORAGE 00039 *Object* is already being used.`

Object
: is the type of the device you specified in the command.

Cause

The device you specified in the command is already being used.

Effect

The command is not executed.

Recovery

Reissue the command and specify the correct device.

00040

`STORAGE 00040 Configuration database record not found.`

Cause

One or more configuration database records are missing. This can happen when the indicated object (or parts of the object) is not configured.

Effect

The command is not executed.

Recovery

If possible, correct the configuration and reissue the command. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00041

`STORAGE 00041 Configuration database record is in use.`

Cause

The specified object probably has other objects dependent on its existence.

Effect

The command is not executed.

Recovery

Correct the attributes and reissue the command.

00042

`STORAGE 00042 Configuration database access timeout.`

Cause

The configuration database is locked by some other process.

Effect

The command is not executed.

Recovery

Wait a few seconds and reissue the command.

00043

`STORAGE 00043 Wrong path specified: object-type name.`

Cause

You specified a command that does not allow the specified device path.

Effect

The command is not executed.

Recovery

Correct and reissue the command.

00044

`STORAGE 00044 Configuration database record already exists.`

Cause

The object is already configured. The indicated object might already be configured by another device.

Effect

The command is not executed.

Recovery

Correct the name of the object and reissue the command.

00045

`STORAGE 00045 IOP is busy.`

Cause

The object is busy and will not accept commands. This situation might occur when an IOP is unable to access its hardware. Usually a status query on the process itself will show it in a wait state of %040 (INTR).

Effect

The command is not executed.

Recovery

Correct the hardware configuration and reissue the command.

00046

STORAGE 00046 *cmd* will restart IOP.

cmd

is the command that you issued.

Cause

You issued a RESET, FORCED command, which stops and restarts the I/O process. You can use this command only in interactive mode.

Effect

The I/O process is restarted if you acknowledge the verification request.

Recovery

Informational message only; no corrective action is required.

00047

`STORAGE 00047 CPUs not running.`

Cause

You issued a START or PRIMARY command targeting a processor or set of processors that were not running.

Effect

The command is not executed.

Recovery

Reload the processors or change the configuration and reissue the command.

00048

`STORAGE 00048 Required adapter not present for location.`

Cause

You issued an ADD or ALTER command, specifying a location that does not have its corresponding adapters installed.

Effect

The command is not executed.

Recovery

Correct the command to use locations that are available.

00049

`STORAGE 00049 Command failed.`

Cause

The command failed because of an internal error. The storage subsystem manager experienced an error condition while accessing a helper server.

Effect

The command is not executed.

Recovery

Restart the storage subsystem manager and retry the command. If the problem persists, contact your service provider.

00050

STORAGE 00050 *command* failed. POOL argument error: *errnum*

command

is the command you specified.

errnum

is the POOL argument error number.

Cause

The ALTER command was issued with a storage pool attribute that is in conflict with the current configuration or was invalid or missing.

Effect

The command is not executed.

Recovery

Reissue the command using the correct storage pool attribute.

00051

`STORAGE 00051 command failed. Disk does not belong to any pool`

command

is the command you specified.

Cause

You issued an ALTER command using POOL attributes for a disk that does not belong to any storage pool.

Effect

The command is not executed.

Recovery

Reissue the command with the correct attributes.

00052

`STORAGE 00052 PRIMARYCPU and BACKUPCPU are equal.`

Cause

You specified the same value for PRIMARYCPU and BACKUPCPU.

Effect

The command is not executed.

Recovery

Reissue the command specifying unique values for PRIMARYCPU and BACKUPCPU.

00053

`STORAGE 00053 Required attribute not specified: PRIMARYLOCATION`

Cause

You did not specify the PRIMARYLOCATION attribute for a command that requires it.

Effect

The command is not executed.

Recovery

Reissue the command with the PRIMARYLOCATION attribute.

00054

`STORAGE 00054 BACKUPLOCATION: Invalid value specified.`

Cause

You issued a command using an invalid value for BACKUPLOCATION.

Effect

The command is not executed.

Recovery

Reissue the command specifying a valid location for BACKUPLOCATION.

00055

`STORAGE 00055 PRIMARYLOCATION AND BACKUPLOCATION are equal.`

Cause

You specified the same location for PRIMARYLOCATION and BACKUPLOCATION.

Effect

The command is not executed.

Recovery

Reissue the command using unique values for PRIMARYLOCATION and BACKUPLOCATION.

00056

`STORAGE 00056 SMF pool communication error.`

Cause

The command issued resulted in a error while communicating with the SMF storage pool process.

Effect

The command is not executed.

Recovery

Reissue the command after correcting the cause of the error.

00057

`STORAGE 00057 DEBUG option not valid when object is already running.`

Cause

You issued a START command using the DEBUG attribute for an object that is already running.

Effect

The command is not executed.

Recovery

Use the RESET, FORCED command to stop the object and then reissue the START command.

00058

`STORAGE 00058` *`command`* `rejected, failed to validate request with SMF.`

`command`

is the command you specified.

Cause

You issued a command that requires interaction with SMF. This interaction failed.

Effect

The command is not executed.

Recovery

Resolve the cause of the failure and reissue the command.

00059

`STORAGE 00059 Program file is not $SYSTEM.SYSTEM.`

Cause

You specified that the PROGRAM file of a storage object is located in a subvolume other than $SYSTEM.SYSTEM. You must verify you want to use the specified subvolume. No warning is given in noninteractive mode.

Effect

The command is executed if you answer Y at the prompt; it is not executed if you respond N at the prompt.

Recovery

Answer Y or N at the prompt.

00060

```
STORAGE 00060  objname is in use.
```

objname
: is the name of the object you specified in the command.

Cause

The object you issued the command to is currently in use by another requester.

Effect

In interactive mode, you are prompted to validate the request. In noninteractive mode, the command is not executed.

Recovery

In interactive mode, validate the request. In noninteractive mode, use the FORCED option to override this check.

00061

```
STORAGE 00061  Failed to obtain CRU information, error: errnum, error detail: text. CRU: cru-name
```

errnum
: is the error number associated with this failure.

text
: is additional information about the error.

cru-name
: is the name of the customer-replaceable unit (CRU) you specified in the command.

Cause

You requested information for a CRU, but that information was unavailable from the hardware inventory table.

Effect

The command is not executed.

Recovery

Verify the hardware is actually present and connected correctly.

00062

```
STORAGE 00062  Error from PROCESS_LAUNCH_, error: errnum, error detail: text.
```

errnum
: is the error number returned from the PROCESS_LAUNCH_ procedure.

text
: is additional information about the error.

Cause

The storage subsystem was unable to create the process.

Effect

The command is not executed.

Recovery

Refer to the error number and error detail for the cause of the error and retry the command. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00063

```
STORAGE 00063  cmd failed. SAC in use by incompatible devices, sacpath.
```

cmd
: is the command you specified.

sacpath
: is the path of the ServerNet addressable controller (SAC) specified in the command.

Cause

You attempted to configure a SAC for a device that the SAC does not support.

Effect

The command is not executed.

Recovery

Resolve the cause of the failure and reissue the command.

00064

`STORAGE 00064 cmd failed. No CRU info; cru-info error: errnum`

cmd
: is the command you specified.

cru-info
: is additional information about the CRU in the command you specified.

errnum
: is the error associated with this failure.

Cause

While trying to complete the requested command, communication with the CRU was interrupted.

Effect

The command is not executed.

Recovery

Verify the hardware is actually present and connected correctly. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00065

`STORAGE 00065 cmd failed. No SAC present in sacpath.`

cmd
: is the command you specified.

sacpath
: is the path of the ServerNet addressable controller (SAC) specified in the command.

Cause

You might have issued a command with an invalid SAC location. There is no SAC in the specified location.

Effect

The command is not executed.

Recovery

Verify the hardware is actually present and connected correctly. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00066

`STORAGE 00066 cmd failed. POST did not succeed for sacpath.`

cmd
: is the command you specified.

sacpath
: is the path of the ServerNet addressable controller (SAC) specified in the command.

Cause

The indicated SAC did not pass the power-on self-test (POST).

Effect

The command is not executed.

Recovery

Contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00067

`STORAGE 00067 cmd failed. Mismatched SAC type in sacpath.`

cmd
: is the command you specified.

sacpath
: is the path of the ServerNet addressable controller (SAC) specified in the command.

Cause

The type of the SAC specified in the command does not match the SAC type in the PRIMARYSAC location.

Effect

The command is not executed.

Recovery

Resolve the cause of the failure and reissue the command.

00068

`STORAGE 00068 cmd failed. Wrong CRU type for device in sacpath.`

cmd
: is the command you specified.

sacpath
: is the path of the ServerNet addressable controller (SAC) specified in the command.

Cause

The device type of the device specified in the command does not match the device type of the primary device.

Effect

The command is not executed.

Recovery

Verify the device specified in the command is the same type as the primary device. (For example, you are trying to mirror an internal disk with a 45*xx* disk.) Then, reissue the command.

00069

`STORAGE 00069 cmd failed. sacpath violates IOP placement.`

cmd
: is the command you specified.

sacpath
: is the path of the ServerNet addressable controller (SAC) specified in the command.

Cause

The indicated command specified a combination of paths and processors that is not supported by the I/O process (IOP).

Effect

The command is not executed.

Recovery

Resolve the cause of the failure and reissue the command.

00070

`STORAGE 00070` *`cmd`* `failed.` *`sacpath`* `is already in use.`

`cmd`

is the command you specified.

`sacpath`

is the path of the ServerNet addressable controller (SAC) specified in the command.

Cause

The device specified for the indicated path is already configured for another process.

Effect

The command is not executed.

Recovery

Resolve the cause of the failure and reissue the command.

00071

`STORAGE 00071  Attribute` *`attname`* `is obsolete and will not be supported in future releases.`

`attname`

is the name of the obsolete attribute you used in the command.

Cause

You specified an obsolete attribute with a command.

Effect

The command is not executed.

Recovery

Avoid using the attribute. Modify command files that contain commands using this attribute.

00072

`STORAGE 00072  The specified adapter is in use and cannot be deleted.`

Cause

The adapter you tried to delete still has devices configured to use it.

Effect

The command is not executed.

Recovery

To delete the adapter, first delete the devices configured to use it. Then retry the command.

00073

`STORAGE 00073` *`cmd`* `rejected.` *`sys-devname`* `is on wrong system.`

`cmd`

is the command you specified.

`sys-devname`

is the system and device name you specified in the command.

Cause

You issued a command containing a system name that does not match the system name of the object.

Effect

The command is not executed.

Recovery

Resolve the cause of the failure and reissue the command.

00074

`STORAGE 00074 Operation invalid, REPLACE already committed.`

Cause

You issued a REPLACE command twice.

Effect

The command is not executed.

Recovery

Use the REVERT operation to back out of the committed replace.

00075

`STORAGE 00075  REPLACE DISK` *`volname`* `failed to delete software replacement record, error` *`err-num`*`.`

`volname`

is the disk name you specified.

`err-num`

is the file-system error that occurred. For more information, refer to the *Guardian Procedure Errors and Messages Manual*.

Cause

A file-system error occurred while data was being deleted from $SYSTEM.SYS*nn*.CONFALT.

Effect

The command is not executed.

Recovery

Retry the command. If the problem persists, contact your service provider. See “If You Have to Contact Your Service Provider” (page 298).

00076

`STORAGE 00076` *`REPLACE operation already pending for device.`*

Cause

You issued a REPLACE command to a device that is already performing a replace operation.

Effect

The command is not executed.

Recovery

Follow these steps:

1. Wait for the operation to finish.
2. If it does not finish, retry the operation with the REPLACE, AGAIN command.
3. If the problem persists, contact your service provider. See “If You Have to Contact Your Service Provider” (page 298).

00077

`STORAGE 00077` *`REPLACE DISK volname software replacement record read, error: err-num.`*

`volname`

is the disk name you specified.

`err-num`

is the file-system error that occurred. For more information, refer to the *Guardian Procedure Errors and Messages Manual*.

Cause

A file-system error occurred while data was being read from $SYSTEM.SYS*nn*.CONFALT.

Effect

The command is not executed.

If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00078

`STORAGE 00078` *`REPLACE DISK volname software replacement record insert, error:err-num.`*

`volname`
: is the disk name you specified.

`err-num`
: is the file-system error that occurred. For more information, refer to the *Guardian Procedure Errors and Messages Manual*.

Cause

A file-system error occurred while data was being read from $SYSTEM.SYS*nn*.CONFALT.

Effect

The command is not executed.

If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00079

`STORAGE 00079 The` *`replace operation is not committed.`*

Cause

You have issued the REPLACE, REVERT command after a committed replace operation.

Effect

The command is not executed.

Recovery

Issue the REPLACE, ABANDON command.

00080

`STORAGE 00080 A` *`replace operation is still pending.`*

Cause

A REPLACE, PROGRAM or REPLACE, AGAIN operation has not finished.

Effect

The command is not executed.

Recovery

Follow these steps:

1. Wait for the operation to finish.
2. If it does not finish, reactivate it with the REPLACE, AGAIN command.
3. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00081

`STORAGE 00081` *`volname is not running as a process pair.`*

`volname`
: is the disk name you specified.

Cause

You issued a REPLACE command for a disk that is not running as a process pair.

Effect

The command is not executed.

Recovery

Reload the backup processor for that disk process and retry the command.

00082

`STORAGE 00082 REPLACE DISK volname software replacement record update, error: err-num.`

volname

is the disk name you specified.

err-num

is the file-system error that occurred. For more information, refer to the *Guardian Procedure Errors and Messages Manual*.

Cause

A file-system error occurred during an update of data in $SYSTEM.SYS*nn*.CONFALT.

Effect

The command fails, possibly with the replace operation half finished.

Recovery

1. Refer to the error message for the cause of the error.
2. Retry the command.
3. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00083

`STORAGE 00083 Unexpected error returned from DSC_OLSR_DEVICE_, error: err-num.`

err-num

is the file-system error that occurred. For more information, refer to the *Guardian Procedure Errors and Messages Manual*.

Cause

An unexpected error was returned from DSC_OLR_DEVICE_.

Effect

The command fails, possibly with the replace operation half finished.

Recovery

1. Refer to the error message for the cause of the error.
2. Retry the command.
3. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00084

`STORAGE 00084 DSC_OLR_DEVICE_ error from PROCESS_LAUNCH_, error: err-num, error detail: err-detail.`

err-num

is the file-system error that occurred. For more information, refer to the *Guardian Procedure Errors and Messages Manual*.

err-detail

is the error detail about the error that occurred. For more information, refer to the *Guardian Procedure Errors and Messages Manual*.

Cause

There was a process launch error returned from DSC_OLR_DEVICE.

Effect

The command fails, possibly with the replace operation half finished.

Recovery

1. Refer to the error message for the cause of the error.

2. Retry the command.
3. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00085

`STORAGE 00085` *`REPLACE DISK volname Filesystem error returned from DSC_OLR_DEVICE_, error: err-num.`*

`volname`
: is the disk name you specified.

`err-num`
: is the file-system error that occurred. For more information, refer to the *Guardian Procedure Errors and Messages Manual*.

Cause

A file-system error was returned from DSC_OLR_DEVICE_.

Effect

The command fails, possibly with the replace operation half finished.

Recovery

1. Refer to the error message for the cause of the error.
2. Retry the command.
3. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00086

`STORAGE 00086` *`DISK volname does not have a software replacement record.`*

`volname`
: is the disk name you specified.

Cause

No replace operation has been done for this object.

Effect

The command is not executed.

Recovery

This is an informational message only. No recovery action is needed.

00087

`STORAGE 00087` *`DISK volname cannot have IOP object on itself.`*

`volname`
: is the disk name you specified.

Cause

You attempted to run a disk IOP from an object file that resides on the disk controlled by that IOP.

Effect

The command is not executed.

Recovery

Run the disk IOP from an object file that is not located on the disk controlled by that IOP.

In addition, ensure that:

- There are no crosswise configurations in your system.
- Executable objects are not placed on disks that start after the target disk starts.

The exception to this rule is the system-load volume, which can have its object file on its disk.

00088

STORAGE 00088 *REPLACE DISK volname new object file error: err-num.*

volname

is the disk name you specified.

err-num

is the file-system error that occurred. For more information, refer to the *Guardian Procedure Errors and Messages Manual*.

Cause

You initiated a replace operation, and a file-system error occurred while the new object file was being verified.

Effect

The command is not executed.

Recovery

1. Ensure that the correct object is present.
2. Retry the command.

00089

STORAGE 00089 *Non-supported adapter found where devname was expected.*

devname

is the device name expected by SCF.

Cause

SCF found a nonsupported adapter where it expected to find a supported storage adapter.

Effect

The command is not executed.

Recovery

Recovery depends on the results you want.

If you want a storage adapter configured in the slot:

1. Remove whatever is in the slot.
2. Insert the storage adapter you want.
3. Retry the command.

If you do not want a storage adapter configured in the slot, delete the storage adapter configuration record for that slot.

00090

STORAGE 00090 *Vproc verification error: err-num, error detail: err-detail.*

err-num

is the PROCESS_CREATE_ error that occurred. For more information, refer to the *Guardian Procedure Errors and Messages Manual*.

err-detail

is the error detail for the PROCESS_CREATE_ error. For more information, refer to the *Guardian Procedure Errors and Messages Manual*.

Cause

You issued the REPLACE, PROGRAM command. The version procedure check for this command was not executed.

Effect

The command is not executed.

Recovery

1. Verify that the VPROC program:
   - Resides on the system disk
   - Is executable
2. Verify that the object you are using for the REPLACE, PROGRAM command is compatible with the running object.

00091

`STORAGE 00091` *`Unsupported program version.`*

Cause

You issued a REPLACE command, and one of two problems occurred:

- The program file used for this command did not contain the expected version of the procedure.
- The version of the procedure did not indicate that it supported this command.

Effect

The command is not executed.

Recovery

1. Contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).
2. Obtain a new program file from your service provider.
3. Retry the command.

00092

`STORAGE 00092` *`Unexpected error returned from CONFIG_GETINFO_, the IOP did not respond within the allocated timeout period.`*

Cause

An IOP did not respond within the timeout period, so CONFIG_INFO_ returned an unexpected error.

Effect

The command is not executed.

Recovery

1. Retry the command.
2. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00093

`STORAGE 00093` *`No attributes have been specified for this command.`*

Cause

When you entered the command, you did not enter any attributes.

Effect

The command is not executed.

Recovery

Enter the desired parameters. Retry the command.

00094

`STORAGE 00094` *`Both paths of the disk process are assigned the same fabric.`*

Cause

When you entered the command, you assigned the same fabric to access both paths to the disk.

Effect

The command is executed. This is a warning.

Recovery

Reissue the command and assign a separate path to each path.

00095

`STORAGE 00095` *`Unable to process the command due to an internal NSC error.`*

Cause

The command resulted in an internal NonStop C (NSC) language module error.

Effect

The command is not executed.

Recovery

1. Retry the command.
2. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

00096

`STORAGE E00096 Unable to perform` *`n`* `of the requested path switches:` *`process-path-list`*

Cause

A SWITCH ADAPTER command returned a count (*n*) of the number of paths that did not switch, along with a list of the unswitched paths.

Effect

The command is not executed.

Recovery

Handle those exceptions individually.

00097

`STORAGE E00097 Unable to perform` *`n`* `of the requested IOP primary changes:` *`process-list`*

Cause

A PRIMARY SUBSYS command returned a count (*n*) of the number of IOPs that did not move to the other processor, along with a list of those IOPs.

Effect

The command is not executed.

Recovery

Handle those exceptions individually.

00098

`STORAGE W00098 Action begun successfully, but not yet finished. Monitor with STATUS.`

Cause

The action was initiated successfully, but it has not yet finished.

Effect

The command is executed. This message is only a reminder.

Recovery

Use the STATUS command to monitor the progress of the action.

00099

`STORAGE E00099 The process cannot be terminated.`

Cause

The process cannot be terminated. An OSL process cannot be terminated while it has paths in STARTED state.

Effect

The command is not executed.

Recovery

Verify the state of the paths using the STATUS command. If needed, use the STOP or ABORT command to change the state of the paths before retrying the operation.

00100

```
STORAGE E00100  Command failed. Too many devices configured on a SAC
```

Cause

Too many devices configured on a SAC or too many IOPs in the same processor configured to use an adapter.

Effect

The command is not executed.

Recovery

Configure fewer devices on the SAC or configure the IOP in different processors.

00101

```
STORAGE E00101 Attribute value conflicts with device location.
```

Cause

The specified attribute value conflicts with the device location.

Effect

The command is not executed.

Recovery

Specify attribute values and device location which are consistent with each other.

00102

```
STORAGE E00102 Requested operation failed due to a file system error.
```

Cause

Requested operation failed due to a file system error.

Effect

The command is not executed.

Recovery

Check the returned file system error to determine what to do next.

00103

```
STORAGE E00103 Command rejected because there's another action pending for the specified object name.
```

Cause

An action is currently in progress for the object name you specified.

Effect

The command is rejected.

Recovery

Wait a few seconds and try the command again.

00104

STORAGE E00104 Not allowed: the specified action is invalid for the specified disk volume

Cause

You tried to perform an action that is invalid for the specified disk volume.

Effect

The command is not executed.

Recovery

To determine why the command failed, consult the accompanying error message.

00105

```
STORAGE E00105 I/O operation failed.
```

Cause

The command failed when trying to send a command to the I/O process.

Effect

The command is not executed.

Recovery

Collect the information in the error text and contact your service representative for further assistance.

00106

```
STORAGE E00106 The two paths to a device must be in different IOAM modules
```

Cause

You have specified two paths to a device through FCSA adapters in the same IOAM module. This configuration is not recommended.

Effect

The command is not executed. SCF waits for the next command

Recovery

Specify the two paths in different IOAM modules.

00107

```
STORAGE E00107 The requested configuration requires that Advanced Storage Configuration is enabled in OSM
```

Cause

The requested device path configuration requires that Advanced Storage Configuration be enabled in G-series versions of OSM. Advanced Storage Configuration is required for device configuration options which became available in T1083ABC (G06.23) including Enclosure Interleaving and Enterprise Storage (ESS) disks.

Effect

The command is not executed.

Recovery

Run the OSM Service Connection and select the action to Enable Advanced Storage Configuration.

00108

```
STORAGE E00108 Cannot configure the specified device without a license file.
```

Cause

Configuration of the device type is not allowed without a license file.

Effect

The request is not processed.

Recovery

Install the required license file. For Enterprise Storage disks, the required license file is in the T0631 product.

00109

`STORAGE E00109 Can't access fiber because no FCSMON processes are assigned.`

Cause

No FCSMON processes are running, or they have not yet been assigned to monitor the specified SAC.

Effect

The command is not executed.

Recovery

Start the FCSMON processes. If they are already running, wait 3 minutes and re-try the command.

00110

`STORAGE E00110 Alternate path SAC is not connected to the same device.`

Cause

The alternate path (BACKUP or MBACKUP) SAC is not connected to the same device as the corresponding primary path (PRIMARY or MIRROR).

Effect

The command is not executed.

Recovery

Reissue the command with corrected path attributes.

00111

`STORAGE E00111 FCSMON process returned an error.`

Cause

FCSMON process returned an error.

Effect

The command is not executed.

Recovery

Contact your service representative.

00112

`STORAGE E00112 The Pool name length along with the system name exceeds 16 chars.`

Cause

The Pool name along with the system name exceeded the 16 character maximum. The Pool name is limited to less than or equal to 6 chars when the system name is 7 characters

Effect

Alter Disk, Pool fails.

Recovery

1. Delete the Pool name and add a new Pool name with six or less characters. 2. Reissue the Alter Disk, Pool command for the desired disks.

00113

`STORAGE E00113 The two paths to a device must use different CLIMs.`

Cause

You have placed both paths to a device through the same CLIM. This is not allowed.

Effect

The command is not executed.

Recovery

Place the two paths through different CLIMs.

00115

`STORAGE E00115 Lunmgr error.`

Cause

Lunmgr returned an error.

Effect

The command is not executed.

Recovery

Correct the problem identified in the error text.

00116

`STORAGE E00116 Key change is already in progress on this drive or its mirror.`

Cause

Key change is already in progress on this drive or its mirror.

Effect

The command is not executed.

Recovery

STATUS DISK, ENCRYPTION will show where key change is in progress.

00117

`STORAGE E00117 Key change is not in progress on this path.`

Cause

Key change is not in progress on this path.

Effect

The command is not executed.

Recovery

No action is required.

00118

`STORAGE E00118 Can't mix encryption and non-encryption attributes in one command.`

Cause

Encryption and non-encryption attributes were used in the same command.

Effect

The command is not executed.

Recovery

Separate the encryption and non-encryption attributes into two commands.

00120

`STORAGE E00120 No TCP/IP stack is running.`

Cause

No TCP/IP stack is running.

Effect

The command is not executed.

Recovery

Configure and start a TCP/IP stack (e.g. $ZSSP0 and $ZTCP0).

00121

`STORAGE E00121 C I/O library error during communication with CLIM.`

Cause

C I/O library error during communication with CLIM.

Effect

The command is not executed.

Recovery

Make sure $ZZSTO can read and write in the $SYSTEM.ZZSTO subvolume.

00122

`STORAGE E00122 Can't determine IP address for CLIM.`

Cause

Can't determine IP address for CLIM.

Effect

The command is not executed.

Recovery

Make sure that STATUS CLIM $ZZCIP.*clim-name* shows the CLIM in STARTED state and that INFO CLIM $ZZCIP.*clim-name*, DETAIL shows a Maintenance Interface IP address.

00123

`STORAGE E00123 Can't run lunmgr on the CLIM.`

Cause

Can't run lunmgr on the CLIM.

Effect

The command is not executed.

Recovery

1. Make sure the CLIM is running and connected.
2. Make sure SSH can communicate with the CLIM.
3. Make sure lunmgr is installed on the CLIM.
4. Run any CLIMCMD command manually for the same CLIM, which may also help to resolve this issue. If any error is reported by CLIMCMD, correct this problem and re-issue the command.

-> CLIMCMD S1002531 pwd

00124

`STORAGE E00124 This command must target a device on the local system.`

Cause

This command must target a device on the local system.

Effect

The command is not executed.

Recovery

Log on to the same system as the targeted device.

00125

`STORAGE E00125 This command requires a member of the SECURITY-ENCRYPTION-ADMIN group.`

Cause

This command requires a member of the SECURITY-ENCRYPTION-ADMIN group.

Effect

The command is not executed.

Recovery

Log on as a member of the SECURITY-ENCRYPTION-ADMIN (65536) group.

00126

`STORAGE E00126 This command requires an encryption license.`

Cause

This command requires an encryption license.

Effect

The command is not executed.

Recovery

Install an NSVLE license file. Use STATUS SUBSYS command to verify that there is a valid license file installed on the NonStop system.

00127

`STORAGE E00127 Filesystem error during encryption command.`

Cause

File system error on the license file or SSH communication files.

Effect

The command is not executed.

Recovery

Make sure the license file, SSH and SCP are correct.

00128

`STORAGE E00128 The specified device is not CLIM-connected.`

Cause

The specified device is not CLIM-connected, so encryption commands can't be performed on it.

Effect

The command is not executed.

Recovery

No action is required.

00129

`STORAGE E00129 Old lunmgr on this CLIM does not support encryption or partition.`

Cause

lunmgr on this CLIM does not support encryption or partition.

Effect

The command is not executed.

Recovery

Install a newer version of CLIM software.

00130

`STORAGE E00130 CLIMCMD returned an error.`

Cause

To resolve an SSH error, the Storage Subsystem Manager ($ZZSTO) executed a CLIMCMD command, but it did not succeed.

Effect

The command is not executed.

Recovery

Install a newer version of lunmgr on this CLIM.

1. Make sure SSH can communicate with the CLIM.
2. Run any CLIMCMD command manually for the same CLIM, which may also help to resolve this issue. If any error is reported by CLIMCMD, correct this problem and re-issue the command.

-> CLIMCMD S1002531 pwd

00134

```
STORAGE E00134 Device WWN mismatch between PRIMARYLUN and BACKUPLUN.
```

Cause

For a PARTITION command, the device WWN of the PRIMARYLUN and the BACKUPLUN does not match.

Effect

The command is not executed.

Recovery

Make sure that correct values of CLIM and LUN are specified and reissue the command.

00135

```
STORAGE E00135 Partition count or partition entry mismatch between the PRIMARYCLIM
and the BACKUPCLIM.
```

Cause

Partition count or partition entry mismatch occurred between the PRIMARYCLIM and the BACKUPCLIM.

Effect

The command is not executed.

Recovery

Make sure that the configuration for CLIM and LUN are correct and reissue the command.

00136

```
STORAGE E00136 LUN is already being used as an unpartitioned disk volume.
```

Cause

For an ADD PARTITION command, the LUN specified is already being used as an unpartitioned disk volume.

Effect

The command is not executed.

Recovery

Delete the NonStop configuration of the unpartitioned disk and retry the operation.

00137

```
STORAGE E00137 Disk volume paths to the partition are in STARTED state.
```

Cause

For a DELETE PARTITION command, the disk volume has paths to the partition in STARTED state.

Effect

The command is not executed.

Recovery

Stop the disk paths to the partition and retry the operation.

00138

```
STORAGE W00138 Disk volume paths to the partition are in STOPPED state.
```

Cause

Warning: A configured NonStop disk partition is to be deleted.

Effect

In interactive mode, you are prompted to validate the request. In noninteractive mode, the command is not executed.

Recovery

Make sure the correct disk partition is to be deleted. In interactive mode, validate the request. In noninteractive mode, use the FORCED option to override this check.

00139

```
STORAGE E00139 A mismatch of WRITECACHE attribute value was detected
among partitions on the same device.
```

Cause

There is a mismatch of WRITECACHE attribute value among partitions on the same device.

Effect

The command is not executed.

Recovery

When adding a disk partition, make sure that the Write Cache setting for the new partition matches with the Write Cache setting of the existing partitions.

00140

```
STORAGE E00140 The device specified is not partitioned.
```

Cause

The device specified is not partitioned.

Effect

The command is not executed.

Recovery

Add partitions to the device using SCF ADD PARTITION command.

00141

```
STORAGE W00141 The DELETE PARTITION command will destroy any existing partitions
or files on the device.
```

Cause

Warning: The DELETE ALL PARTITION or the ADD PARTITION, LIKE command will destroy any existing partitions or files on the device. The DELETE LAST PARTITION command will destroy any existing files on the last partition.

Effect

In interactive mode, you are prompted to validate the request. In noninteractive mode, the command is not executed.

Recovery

Make sure the existing files can be deleted. In interactive mode, validate the request. In noninteractive mode, use the FORCED option to override this check.

00142

```
STORAGE E00142 The source device specified is same as target device.
```

Cause

For the ADD PARTITION, LIKE command, the source device specified is same as the target device.

Effect

The command is not executed.

Recovery

Specify the correct source and target devices and reissue the command.

00143

```
STORAGE E00143 The number of disks that require WRITECACHE ENABLE
or DISABLE exceeds the limit.
```

Cause

The changing of the WRITECACHE ENABLE or DISABLE affects more than the limit number of disks.

Effect

The command is not executed.

Recovery

Simplify the disk configurations so the chain of mirrored disk partitions involves fewer disks.

00145

```
STORAGE E00145 Invalid lunmgr response from the CLIM.
```

Cause

The lunmgr response from the CLIM is in an invalid format.

Effect

The command is not executed. SCF waits for the next command.

Recovery

1. Run any CLIMCMD command manually for the same CLIM. If any error is reported by CLIMCMD, correct this problem and re-issue the command.
2. If the problem persists, contact your service provider. See "Upgrade and Replacement Procedures" (page 362) for additional information.

00146

```
STORAGE E00146 Cannot enable WRITECACHE for the volume due to the disk type
in primary or mirror location.
```

Cause

Trying to enable WRITECACHE on a disk type that does not support this feature. Only CLIM SAS disks support enabling of WRITECHACE.

Effect

The request is not processed.

Recovery

This command cannot be performed on the specified disk type. If WRITECACHE was previously enabled on this disk, disable WRITECACHE using ALTER DISK, WRITECACHE command.

00147

```
STORAGE E00147 The disk process is in SOFTDOWN state which prevents
command execution.
```

Cause

Either primary and backup DP2 (the disk process) are in SOFTDOWN state or DP2 is in SOFTDOWN state.

Effect

The command is not executed. SCF waits for the next command.

Recovery

1. Take online dump for the affected DP2 processors. After the online dump is completed, use SCF START DISK command to restart the affected DP2 processes, which resets the state of the processes. Online dump action must be completed before starting the disk process.
2. If the problem persists, contact your service provider. See "Upgrade and Replacement Procedures" (page 362) for additional information.

01001

`STORAGE 01001 Must perform disk revive: *volname*`

volname
: is the name of the volume that you specified in the command.

Cause

One of the disks of a mirrored volume is not current with the other disk. A disk revive operation is required.

Effect

The disk revive operation is started.

Recovery

Informational message only; no corrective action is required. SCF initializes a disk revive operation.

01002

`STORAGE 01002 *objtype devname* is *state*. Please use RESET command.`

objtype
: is the object type of the object you specified with the command.

devname
: is the name of the device you specified in the command.

state
: is the current state of the device you specified in the command.

Cause

The device is in a state from which it cannot be started.

Effect

The device is not started.

Recovery

Issue an SCF RESET command.

01003

`STORAGE 01003 *command* rejected, a disk revive is in progress.`

command
: is the SCF command that you issued.

Cause

The operation cannot be performed because a disk revive operation is still in progress.

Effect

The command is not executed.

Recovery

Verify the state of the device by using the SCF STATUS command. Either wait until the revive operation is completed or if you need to temporarily stop the revive operation, use the SCF RESET command to suspend the revive operation before retrying the operation.

01004

`STORAGE 01004` *`command`* `rejected, disk contains active audit trail files.`

`command`
: is the SCF command that you issued.

Cause

The command you issued is not allowed for a disk that contains active audit trail files.

Effect

The command is not executed.

Recovery

1. Verify you issued the command for the correct disk.
2. If you issued the command for the correct disk, you must stop the Transaction Management Facility (TMF) subsystem and then reissue the command.

01005

`STORAGE 01005` *`command`* `rejected, disk is TMF protected.`

`command`
: is the SCF command that you issued.

Cause

The command you issued is not allowed for a disk that is protected by the TMF subsystem.

Effect

The command is not executed.

Recovery

1. Verify you issued the command for the correct disk.
2. Verify the state of the disk within the TMF subsystem. If needed, change the device's state in the TMF subsystem before retrying the operation. Refer to the *TMF Operations and Recovery Guide* for additional information.

01006

`STORAGE 01006 Invalid disk path specified:` *`diskpath`*`.`

`diskpath`
: is the disk path that you specified in the command.

Cause

You specified a disk path that does not exist.

Effect

The command is not executed.

Recovery

Determine the correct disk path and reissue the command.

01007

`STORAGE 01007 The` *`command`* `will make all paths to` *`volname`* `inaccessible.`

`command`
: is the SCF command that you issued.

`volname`
: is the name of the volume that you specified in the command.

Cause

You issued a command that would make all paths to the disk inaccessible. This message is generated by SCF only in noninteractive mode. In interactive mode, a prompt is generated asking you to verify the command.

Effect

The command is not executed.

Recovery

1. Reissue the command with the FORCED command option.
2. Execute the command from an interactive SCF prompt.

01008

`STORAGE 01008 The` *`command`* `rejected, the disk contains` *`numberof`* `open files.`

`command`
: is the SCF command that you issued.

`numberof`
: is the number of open files that exist for the device you specified in the command.

Cause

You issued a command that attempted to stop a disk that has open files.

Effect

The command is not executed.

Recovery

Either:

- Close all the open files on the disk and reissue the original command.
- Reissue the command with the FORCED command option.

01009

`STORAGE 01009 The ALTER DISK, CACHE operation might cause SEVERE performance problems on` *`volname`*`.`

`volname`
: is the name of the volume that you specified in the command to alter cache.

Cause

You issued an ALTER DISK, CACHE command in noninteractive mode. In interactive mode, you are prompted to verify the operation.

Effect

The cache counters are reset.

Recovery

Informational message only; no corrective action is required.

01010

`STORAGE 01010 The ALTER DISK, LABEL command will DESTROY any existing files on` *`volname`*`.`

`volname`
: is the name of the volume that you specified in the command to rewrite the label.

Cause

You issued the ALTER DISK, LABEL command in noninteractive mode. The ALTER DISK, LABEL command is available only in interactive mode. In interactive mode, you are prompted to verify the operation.

Effect

The label operation is not performed.

Recovery

1. Start SCF and reissue the command from the SCF prompt.
2. Remove the command from the SCF command file.

01011

`STORAGE 01011 The` *`command`* `operation could leave data in an INCONSISTENT state on` *`volname`*`.`

`command`
: is the SCF command that you issued.

`volname`
: is the name of the volume that you specified in the command.

Cause

You issued the CONTROL DISK, CHECKSUM command in noninteractive mode. The CONTROL DISK, CHECKSUM command is available only in interactive mode. In interactive mode, you are prompted to verify the operation.

Effect

The checksum operation is not performed.

Recovery

1. Start SCF and reissue the command from the SCF prompt.
2. Remove the command from the SCF command file.

01012

`STORAGE 01012 The CONTROL DISK, REPLACEBOOT operation will OVERWRITE the existing bootstrap on` *`volname`*`.`

`volname`
: is the name of the volume that you specified in the command.

Cause

You issued the CONTROL DISK, REPLACEBOOT command in noninteractive mode. The CONTROL DISK, REPLACEBOOT command is available only in interactive mode. In interactive mode, you are prompted to verify the operation.

Effect

The REPLACEBOOT operation is not performed.

Recovery

1. Start SCF and reissue the command from the SCF prompt.
2. Remove the command from the SCF command file.

01013

`STORAGE 01013 The CONTROL DISK, REPLACEBOOT operation can leave one of the disk paths in a HARDDOWN substate.`

Cause

You issued the CONTROL DISK, REPLACEBOOT command in noninteractive mode. The CONTROL DISK, REPLACEBOOT command is available only in interactive mode. In interactive mode, you are prompted to verify the operation.

Effect

The REPLACEBOOT operation is not performed.

Recovery

1. Start SCF and reissue the command from the SCF prompt.
2. Remove the command from the SCF command file.

01014

`STORAGE 01014 LSA` *`sectoraddress`* `passed controller defectiveness testing and was not spared.`

`sectoraddress`
: is the address of the sector that you attempted to spare.

Cause

You issued the CONTROL DISK, SPARE command to spare a disk sector. The disk controller has verified that the sector does not need to be spared.

Effect

The disk sector is not spared.

Recovery

1. Verify you used the correct sector address.
2. Reissue the command with the correct address, if needed.

01015

`STORAGE 001015 The STATS DISK, RESET command will reset the counters used to compute the cache performance statistics on` *`volname`*`.`

`volname`
: is the name of the volume that you have specified a statistical reset for.

Cause

You issued the STATS DISK, RESET command in noninteractive mode. The STATS DISK, RESET command is available only in interactive mode. In interactive mode, you are prompted to verify the operation.

Effect

The statistics counters are not reset.

Recovery

1. Start SCF and reissue the command from the SCF prompt.
2. Remove the command from the SCF command file.

01016

`STORAGE 01016` *`command`* `rejected, the last path to` *`volname`* `cannot be stopped!`

`command`
: is the SCF command that you issued.

`volname`
: is the name of the system disk volume that you specified in the command.

Cause

You issued a command that would stop the last path to the system disk.

Effect

The command is not executed.

Recovery

If you really want to stop the last path to the system disk, reissue the command using the FORCED option.

01017

`STORAGE 01017` *`command`*`,` *`processornum`* `failed.`

`command`
: is the SCF command that you issued.

`processornum`
: is the processor number of the processor you specified in the command.

Cause

You issued a command to change processor ownership but the storage subsystem manager process detected that it did not occur, even though no error was returned from the disk process.

Effect

The processor ownership is unchanged.

Recovery

Check the EMS log for messages pertaining to the problem.

01018

`STORAGE 01018 command failed, reason unknown.`

command
: is the SCF command that you issued.

Cause

You issued a command to make a path switch but the storage subsystem manager process detected that it did not occur, even though no error was returned from the disk process.

Effect

The path switch did not occur.

Recovery

Check the EMS log for messages pertaining to the problem.

01019

`STORAGE 01019 command rejected, disk is not demountable.`

command
: is the SCF command that you issued.

Cause

You tried to perform an operation on a $SYSTEM volume that is permitted only on a demountable disk. $SYSTEM is not a demountable disk.

Effect

The command is not executed.

Recovery

1. Verify you used the correct disk name.
2. Reissue the command, if necessary.

01020

`STORAGE 01020 ALTER DISK volname, LABEL failed, disk is in an incorrect state.`

volname
: is the name of the volume that you specified in the command.

Cause

You tried to update the label on a disk that is not in a correct state.

Effect

The label operation is not performed.

Recovery

Issue an SCF RESET DISK command and retry the operation. If the problem persists, contact your service provider. See "If You Have to Contact Your Service Provider".

01021

`STORAGE 01021 ALTER DISK volname, LABEL failed, disk is not properly formatted.`

volname
: is the name of the volume that you specified in the command.

Cause

You issued the ALTER DISK, LABEL command to a disk that is not properly formatted.

Effect

The command is not executed.

Recovery

Contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

01022

STORAGE 01022 ALTER DISK *volname*, LABEL failed, work file error: *err-num*.

volname
: is the name of the volume that you specified in the command.

err-num
: is the error number of the failure.

Cause

An error occurred in an internal work file that prevented the label operation from finishing.

Effect

The label operation is not performed.

Recovery

Contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

01023

STORAGE 01023 ALTER DISK *volname*, LABEL failed, disk error *err-num*.

volname
: is the name of the volume that you specified in the command.

err-num
: is the disk error number of the failure.

Cause

A disk error occurred that prevented the label operation from finishing.

Effect

The label operation is not performed.

Recovery

Contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

01024

STORAGE 01024 ALTER DISK *volname*, LABEL failed (internal error in the LABEL algorithm).

volname
: is the name of the volume that you specified in the command.

Cause

An internal error in the label algorithm prevented the label operation from finishing.

Effect

The label operation is not performed.

Recovery

Contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

01025

STORAGE 01025 CONTROL DISK *volname*, REPLACEBOOT failed (invalid boot section).

volname
: is the name of the volume that you specified in the command.

Cause

An invalid section in the bootstrap file prevented the REPLACEBOOT operation from finishing.

Effect

The REPLACEBOOT operation is not performed.

Recovery

Contact your service provider to obtain a new bootstrap file and retry the command. See "If You Have to Contact Your Service Provider" (page 298).

01026

```
STORAGE 01026 CONTROL DISK volname, REPLACEBOOT failed (wrong bootstrap file for processor type).
```

volname
is the name of the volume that you specified in the command.

Cause

An incompatibility between the bootstrap file and the processor type prevented the REPLACEBOOT operation from finishing.

Effect

The REPLACEBOOT operation is not performed.

Recovery

Contact your service provider to obtain a new bootstrap file and retry the command. See "If You Have to Contact Your Service Provider" (page 298).

01027

```
STORAGE 01027 CONTROL DISK volname, REPLACEBOOT failed (disk error: err-num).
```

volname
is the name of the volume that you specified in the command.

err-num
is the disk error number of the failure.

Cause

A disk error prevented the bootstrap replacement.

Effect

The REPLACEBOOT operation is not performed.

Recovery

Correct the problem indicated by the error number and reissue the command.

01028

```
STORAGE 01028 CONTROL DISK volname, REPLACEBOOT failed (I/O error err-num on bootstrap file).
```

volname
is the name of the volume that you specified in the command.

err-num
is the I/O error number of the failure.

Cause

A bootstrap file error prevented the bootstrap replacement.

Effect

The REPLACEBOOT operation is not performed.

Recovery

Correct the problem indicated by the error number and reissue the command.

01029

```
STORAGE 01029 CONTROL DISK volname, REPLACEBOOT failed (I/O error err-num on temporary work file).
```

volname
is the name of the volume that you specified in the command.

err-num
is the I/O error number of the failure.

Cause

A temporary work file error prevented the bootstrap replacement.

Effect

The REPLACEBOOT operation is not performed.

Recovery

Correct the problem indicated by the error number and reissue the command.

01030

`STORAGE 01030 CONTROL DISK` *`volname`*`, REPLACEBOOT failed (internal error was detected).`

`volname`
: is the name of the volume that you specified in the command.

Cause

An internal REPLACEBOOT algorithm error prevented the bootstrap replacement.

Effect

The REPLACEBOOT operation is not performed.

Recovery

Report this problem immediately. See "If You Have to Contact Your Service Provider" (page 298).

01031

`STORAGE 01031 CONTROL DISK` *`volname`*`, REPLACEBOOT failed (bootstrap is NO LONGER usable).`

`volname`
: is the name of the volume that you specified in the command.

Cause

A fatal error occurred during the bootstrap replacement on the disk.

Effect

The disk cannot be used to load your system.

Recovery

1. Do not use the disk to load your system.
2. Contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

01032

`STORAGE 01032 CONTROL DISK` *`volname`*`, REPLACEBOOT failed (bootstrap is usable).`

`volname`
: is the name of the volume that you specified in the command.

Cause

An error occurred during the bootstrap replacement on the disk.

Effect

The disk can be used to load your system.

Recovery

1. You can use the disk to load your system.
2. Contact your service provider. See "If You Have to Contact Your Service Provider" (page 298).

01033

`STORAGE 01033 Disk old-name changed name to new-name.`

Cause

An automatic disk rename took place. This can happen when a new disk is installed in a slot and the label has a name that does not match the name in the system configuration database. The storage subsystem detects this name change, updates the database, and issues this message.

Effect

The system configuration database is updated with the new name.

Recovery

Informational message only; no corrective action is required.

01034

`STORAGE 01034 Resuming suspended disk revive on` *`devname`*`.`

devname

is the name of the device you specified in the command.

Cause

The disk revive operation was restarted in response to an SCF START DISK command.

Effect

The revive operation continues from the point where it was suspended.

Recovery

Informational message only; no corrective action is required.

01035

`STORAGE 01035` *`cmd`* `failed. Invalid attribute value in:` *`attribute`*`.`

cmd

is the SCF command you specified.

attribute

is the attribute that you specified with an invalid value.

Cause

You specified an invalid value for the attribute in the command.

Effect

The command is not executed.

Recovery

Determine the correct value for the attribute and reissue the command.

01036

`STORAGE 01036 Disk pair is not consistent. The` *`devname`* `diskhalf not restarted.`

devname

is the name of the device you specified in the command.

Cause

The modification time stamps on the primary and mirror disks of the mirrored volume differ. A revive operation is required.

Effect

The command is executed. However, the disk half with the older time stamp is left in a STOPPED summary state, substate DOWN.

Recovery

Use the SCF DISK START command to revive the volume.

01037

`STORAGE 01037 Disk is not formatted. The revive is not started.`

Cause

A revive operation is not started because the target disk is not formatted.

Effect

The command is not executed.

Recovery

Replace the disk with a formatted disk, or format the disk before retrying the REVIVE command.

01038

```
STORAGE 01038 Name conflict, the name is already in use on the system.
```

Cause

A name conflict was detected in a RENAME or ALTER, LABEL command.

Effect

The command is not executed.

Recovery

Resolve the name conflict before retrying the command.

01039

```
STORAGE 01039 A disk label discrepancy was detected.
```

Cause

The storage subsystem manager detected a discrepancy in the disk label.

Effect

The disk is not put in a STARTED state.

Recovery

You have the choice to start the disk regardless of the label discrepancy. In same cases, you have to choose whether to start the primary or mirror half of the disk.

01040

```
STORAGE 01040 Both slots target the same SCSI chain.
```

Cause

Both locations for a mirrored disk volume target the same SCSI bus.

Effect

The command is not executed.

Recovery

Correct the command to use different SCSI buses for the primary and mirror halves of the mirrored disk volume.

01041

```
STORAGE 01041 STOPOPENS prevents files from being opened
```

Cause

The STOPOPENS command was issued.

Effect

In interactive mode, you are prompted to confirm the operation. In noninteractive mode the operation is performed.

Recovery

Informational message only; no corrective action is required.

01042

```
STORAGE 01042 STOPOPENS prevents files from being opened on $SYSTEM
```

Cause

You issued a STOPOPENS command for $SYSTEM.

Effect

In interactive mode, you are prompted to confirm the operation. In noninteractive mode the operation is performed.

Recovery

Copy the SCF program to a disk other than $SYSTEM before continuing. Do not exit SCF without having another copy on another disk.

01043

`STORAGE 01043 Operation succeeded, disk name contains num open files`

name

is the name of the disk that has open files.

num

is the number of open files on the disk.

Cause

You issued a STOPOPENS command for a disk with open files.

Effect

The command is not executed.

Recovery

Informational message only; no corrective action is required.

01044

`STORAGE 01044 ALTER DISK $DATA00, Autoselect change failed, error errnum`

errnum

is the error number provided by the file-system.

Cause

Something is preventing the Autoselect change from finishing.

Effect

The command is not executed.

Recovery

Correct the problem indicated by the file-system error code and reissue the command.

01045

`STORAGE 01045 ALTER DISK $DATA00, Physvolselect change failed, error errnum`

errnum

is the error number provided by the file-system.

Cause

Something is preventing the Physvolselect change from finishing.

Effect

The command is not executed.

Recovery

Correct the problem indicated by the file-system error code and reissue the command.

01046

`STORAGE 01046 Not allowed: The disk belongs to a storage pool.`

Cause

This command, if executed, would leave the storage pool in an inconsistent state.

Effect

The command is not executed.

Recovery

Remove the disk from the storage pool and reissue the command.

01047

`STORAGE 01047 Not allowed: The disk is audited.`

Cause

This command, if executed, would leave the TMF product in an inconsistent state.

Cause

The command is not executed.

Recovery

Disable the disk under the TMF product and reissue the command.

01048

`STORAGE 01048 Sector` *`secnum`* `is out of range. Maximum is` *`maxsecnum`*

`secnum`

is the sector number you specified with the command.

`maxsecnum`

is the largest number allowed for the sector number.

Cause

You have specified a sector number that is beyond the last sector number on the disk.

Effect

The command is not executed.

Recovery

Determine the correct sector number and reissue the command.

01049

`STORAGE 01049` *`cmd`* `rejected, disk is not mirrored.`

`cmd`

is the command that you issued.

Cause

You issued a command to a nonmirrored disk that can only be used for mirrored disks.

Effect

The command is not executed.

Recovery

Determine the correct disk, or change the command you are issuing to the nonmirrored disk.

01050

`STORAGE 01050` *`devname`* `is currently running unmirrored. REPLACEBOOT operation cannot be performed.`

`devname`

is the name of the device you specified in the command.

Cause

You issued the CONTROL DISK, REPLACEBOOT command in noninteractive mode. This command is available only in interactive mode. In interactive mode, you are prompted to verify you want to replace the disk boot code on a disk that is configured to be mirrored but is running nonmirrored.

Effect

The command is not executed.

Recovery

Either:

- Reissue the SCF command interactively.
- Remove the command from the SCF command file.

01051

`STORAGE 01051 The label is bad on disk device` *`devname`*`. The label was not altered.`

`devname`
: is the name of the device you specified in the command.

Cause

The label is bad on the indicated disk. Either the time stamp or the last mounted operating system (OS) field is null.

Effect

The command is not executed.

Recovery

Either:

- Revive the disk volume using the SCF START DISK command.
- Use the SCF INITIALIZE DISK or ALTER DISK, LABEL command to write a new label onto the disk, if required. Writing a new label deletes any existing files on the disk.

01053

`STORAGE 01053 The REPLACE DISK` *`volname`* `command is not supported in non-interactive mode.`

`volname`
: is the name of the volume that you specified in the command.

Cause

You have not used the REPLACE command from interactive mode.

Effect

The command is not executed.

Recovery

Reissue the REPLACE command from an interactive prompt.

01054

`STORAGE 01054` *`command devname`* `failed. Impossible, 3-cable configuration implied by other devices using the same SAC.`

`command`
: is the SCF command that you issued.

`devname`
: is the name of the device you specified in the command.

Cause

You attempted to add a device that requires a cable configuration that is not possible, given the other devices using the same ServerNet addressable controller (SAC).

Effect

The command is not executed.

Recovery

Reissue the command, specifying a different value for the SAC or the corresponding backup SAC.

01055

`STORAGE 01055` *`command devname`* `failed.` *`Attribute`* `cannot be decreased while` *`devname is in a STARTED state.`*

command
: is the SCF command that you issued.

devname
: is the name of the device you specified in the command.

attribute
: is the attribute that you specified with an invalid value.

Cause

You attempted to decrease the value of an attribute while the I/O process is in a STARTED state.

Effect

The command is not executed.

Recovery

Either:

- Correct the value for the indicated attribute.
- Place the I/O process in a STOPPED state and reissue the command.

01056

```
STORAGE W01056 The primary and mirror paths to a device should be on different chains.
```

Cause

You specified primary and mirror paths to a device on the same chain. This configuration is not recommended.

Effect

The command is executed. This is a warning.

Recovery

Specify the two paths on different chains.

01057

```
STORAGE W01057 All configuration changes except those related to the mirror configuration
have been performed for the disk volume.
```

Cause

You requested changes to mirror-related configuration attributes of a disk volume that is in a started state.

Effect

All configuration attributes that are not related to the mirror configuration of the disk volume has been completed.

Recovery

None. A subsequent message will inform you about the effect of the action and ask you whether to continue.

01058

```
STORAGE W01058 This action will cause the device to be inaccessible to the system.
```

Cause

You are about to power off a storage device.

Effect

The storage device will be powered off and will not be accessible for informational commands such as the STATUS command.

Recovery

Respond “Y” to the prompt requesting approval to do changes online. If you do not want to power off the storage device, enter "N" to the prompt.

01059

```
STORAGE W01059 This action will cause the mirror-related configuration of the disk to change
while the disk volume is in a started state.
```

Cause

You requested changes to mirror-related configuration attributes of a disk volume that is in a started state. This warning message is issued to inform you of the effect of the command.

Effect

You're asked whether to perform the mirror-related configuration change while the disk volume is in a started state.

Recovery

Respond “Y” to the prompt requesting approval to do changes online. If you do not want to change the mirror-related configuration attributes of the disk volume online, respond “N” to the prompt. Then you can stop the disk and reissue the ALTER DISK command.

01060

`STORAGE E01060 Configuration change failed,` *`subsystem-name`* `returned error:` *`errnum, error detail: error-detail.`*

`subsystem-name`

is the name of the subsystem returning the error message.

`errornum`

is a number identifying the error.

`error-detail`

is text that provides more information about the cause of the error.

Cause

An error occurred when you tried to change the disk configuration.

Effect

The configuration change did not take place. The *`subsystem-name`* indicates which subsystem encountered the error condition. The value in *`errnum`* identifies the nature of the problem. The *`error-detail`* text indicates the processor or processors that the error condition affected.

Recovery

To determine whether or not to retry the command, refer to the *`errnum`* and *`error-detail`* fields.

For example, if the *`error-detail`* text indicates a configuration status of “invalid in processor *`n`*,” the change you made caused the configuration information in processor *`n`* to become invalid. Such a situation might cause the loss of fault tolerance. For instance, the backup disk might have lost access to the mirror disk drive.

If you cannot resolve the problem indicated by *`errnum`* and *`error-detail`*, contact your service provider. If you determine that you can resolve the problem yourself, perform either of these two actions:

1. Reissue the command that specified the new configuration. The storage subsystem manager ($ZZSTO) then retries the configuration change.

   For example, if you previously tried to add a mirror disk drive in location (2,1,5) to a disk volume named $DATA, issue this SCF command:

   ```
   ALTER DISK $DATA, MIRRORLOCATION (2,2,5).
   ```

2. Issue a command specifying the previous configuration. The storage subsystem manager then returns the configuration to its original setting.

   For example, if you tried to add a mirror disk drive to a disk named $DATA, issue this SCF command.

   ```
   DELETE DISK $DATA-M
   ```

   If you tried to move a mirror disk drive from location (2,1,5) to location (2,1,17) for a disk named $DATA, issue these SCF commands:

   ```
   DELETE DISK $DATA-M
   ALTER DISK $DATA, MIRRORLOCATION (2,1,5)
   ```

01061

`STORAGE E01061 Configuration changed, but` *`subsystem-name`* `returned error:` *`errnum, error detail: error-detail.`*

`subsystem-name`
: is the name of the subsystem returning the error message.

`errornum`
: is a number identifying the error.

`error-detail`
: is text that provides more information about the cause of the error.

Cause

An error occurred when you changed the disk configuration.

Effect

The configuration change took place. However, *`subsystem-name`* encountered an error condition indicated by the value in *`errnum`*. The *`error-detail`* text species the processor that the error condition affected.

Recovery

To determine whether or not to retry the command, refer to the *`errnum`* and *`error-detail`* fields.

For example, if the *`error-detail`* text indicates a configuration status of "invalid in processor *`n`*," the change you made caused the configuration information in processor *`n`* to become invalid. Such a situation might cause the loss of fault tolerance. For instance, the backup disk might have lost its access to the mirror disk drive.

Depending on the error condition indicated by *`errnum`* and *`error-detail`*, perform either of these two actions:

- Reissue the command that specified the new configuration. The storage subsystem manager ($ZZSTO) then retries the configuration change,

  For example, if you previously tried to add a mirror disk drive in location (2,1,5) to a disk volume named $DATA, issue this SCF command:

  ```
  ALTER DISK $DATA, MIRRORLOCATION (2,2,5).
  ```

- Issue a command specifying the previous configuration. The storage subsystem manager then returns the configuration to its original setting.

  For example, if you tried to add a mirror disk drive to a disk named $DATA, issue this SCF command:

  ```
  DELETE DISK $DATA-M
  ```

01062

`STORAGE W01062 The checksum has been corrected, but the disk data may not be valid.`

Cause

CONTROL DISK, CHECKSUM was not able to read a good copy of the disk data.

Effect

The command is executed. The checksum is corrected. This is a warning.

Recovery

Find the data on the sector using INFO DISK, SECTOR and then verify that it is valid.

01063

```
STORAGE E01063 Command failed. Another disk using the same adapter shares one CPU but not both CPUs.
```

Cause

Another disk using the same adapter shares one CPU but not both CPUs. This configuration is not supported on G-series RVUs prior to G06.27 or on the H06.03 RVU.

Effect

The command is not executed.

Recovery

Use INFO ADAPTER, DETAIL to see other disks using the same adapter. Then reissue the command, specifying different values for the LOCATIONs or the CPUs.

01064

```
STORAGE E01064 Command failed. Another disk using the same adapter shares a CPU but not both adapters.
```

Cause

Another disk using the same adapter shares a CPU but not both adapters. This configuration is not supported on G-series RVUs prior to G06.27 or on the H06.03 RVU.

Effect

The command is not executed.

Recovery

Use INFO ADAPTER, DETAIL to see other disks using the same adapter. Then reissue the command, specifying different values for the LOCATIONs or the CPUs.

01065

```
STORAGE W01065 One or more of the specified attributes can not be immediately changed online.
```

Cause

One or more of the specified attributes can not be immediately changed to the specified value while the volume is up. For example, some attributes can be increased online but not decreased.

Effect

Any specified attributes which can be changed immediately are successfully changed immediately. All specified attributes are saved in the CONFIG file and will take effect at the next re-start of the volume.

After this warning has occurred for a volume, additional changes to other online alterable attributes will receive the same warning, until the deferred change takes effect.

Recovery

This is a warning. No action is necessary. Use STATUS DISK, CONFIG to view the values which the running disk process is currently using. To force deferred changes to take effect, stop and re-start the volume.

01066

```
STORAGE W01066 When a volume has WRITECACHE enabled, delta revive will not be allowed,
and the SUBSYS attribute UPS should be ON.
```

Cause

You enabled WRITECACHE for a volume.

Effect

Write caching will be enabled. Delta revive will not be allowed. Only full revive will be allowed on the volume.

Recovery

This is a warning. No action is required, but HP recommends that the volume be connected to an HP rack mount UPS, which will give the drives enough time to write cached data to the media after the processor stops writing in the event of power loss.

The SUBSYS attribute UPS should be ON either before WRITECACHE is enabled or before the IOP is started. Otherwise, the IOP will run with WRITECACHE disabled.

01067

`STORAGE W01067 Enabling FASTBULKWRITE can impact data integrity on a volume.`

Cause

You set FASTBULKWRITE ON for a volume.

Effect

FASTBULKWRITE is turned ON. When ON, applications and utilities using bulk writes to unstructured files may have higher throughput. Using FASTBULKWRITE ON can result in lost data in unstructured files if the CPU running the primary disk process fails. This attribute only affects disks in a disk drive enclosure or ESS disks.

Recovery

This is a warning. No action is required.

05001 Through 05999 SMF Messages

All messages with error numbers in the range 05001 through 05999 are generated by the Storage Management Foundation (SMF). These messages are documented in the *Storage Management Foundation User's Guide*.

09001 Through 09036

09001

`STORAGE 09001` *`cmd`* `rejected, SMF MON is down. File system error: errnum`

`cmd`

is the command that you issued.

`errnum`

is the error number provided by the file system.

Cause

The command you issued requires the master process to be running.

Effect

The command is not executed.

Recovery

Check the returned file-system error to determine what to do next.

09002

`STORAGE 09002` *`cmd`* `rejected, no definition with SMF MON. File system error: errnum`

`cmd`

is the command that you issued.

`errnum`

is the error number provided by the file system.

Cause

There is no SMF MON definition for the device.

Effect

The command is not executed.

Recovery

Check the returned file-system error to determine what to do next.

09003

`STORAGE 09003` *`cmd`* `rejected, SMF error:` *`errnum`*

`cmd`

is the command that you issued.

`errnum`

is the error number provided by the Storage Management Foundation (SMF).

Cause

The command you issued was rejected by SMF.

Effect

The command is not executed.

Recovery

Check the returned file-system error to determine what to do next.

09004

`STORAGE 09004` *`cmd`* `rejected, SMF MON file system error:` *`errnum.`*

`cmd`

is the command that you issued.

`errnum`

is the error number provided by the file system.

Cause

The command you issued was rejected by SMF MON.

Effect

The command is not executed.

Recovery

Check the returned file-system error to determine what to do next.

09005

`STORAGE 09005 Required attribute not specified: CATALOGLOCATION`

Cause

You did not specify the CATALOGLOCATION attribute in the command you issued.

Effect

The command is not executed.

Recovery

Reissue the command and include the CATALOGLOCATION attribute.

09006

`STORAGE 09006 Required attribute not specified: POOL`

Cause

You did not specify the POOL attribute in the command you issued.

Effect

The command is not executed.

Recovery

Reissue the command and include the POOL attribute.

09007

`STORAGE 09007 Required attribute not specified: ANTLOCATION`

Cause

You did not specify the ANTLOCATION attribute in the command you issued.

Effect

The command is not executed.

Recovery

Reissue the command and include the ANTLOCATION attribute.

09008

`STORAGE 09008 Required attribute not specified: PENDOPSLOCATION`

Cause

You did not specify the PENDOPSLOCATION attribute in the command you issued.

Effect

The command is not executed.

Recovery

Reissue the command and include the PENDOPSLOCATION attribute.

09009

`STORAGE 09009 Nonstandard name given for SMF MON process. $ZSMS is standard`

Cause

You issued a command using an invalid name for the Storage Management Foundation (SMF) process.

Effect

In interactive mode, you will be prompted to confirm the operation. In noninteractive mode, the operation is performed.

Recovery

In interactive mode, confirm the operation. In noninteractive mode, no action is required.

09010

`STORAGE 09010 ABORT is not supported for this process. STOP will be used.`

Cause

You issued the ABORT command for a process that doesn’t accept it.

Effect

The command is not executed. SCF issues the STOP command.

Recovery

Informational message only; no corrective action is required.

09011

`STORAGE 09011 Invalid format in location name.`

Cause

You issued a command with an invalid location name.

Effect

The command is not executed.

Recovery

Reissue the command using the proper format for the location name.

09012

`STORAGE 09012 Invalid subvolume name. It must begin with ZYS.`

Cause

You issued a command using a subvolume name that did not begin with ZYS.

Effect

The command is not executed.

Recovery

Reissue the command using a proper subvolume name that begins with ZYS.

09013

`STORAGE 09013 SMF dependency failure: Unable to obtain information from process` *`name`*`.`

`name`

is the name of the process that the storage subsystem is trying to get information from.

Cause

The command is waiting for information from a process before completion is possible.

Effect

The command is not executed.

Recovery

Determine the problem with the process, and reissue the command.

09014

`STORAGE 09014 SMF dependency failure: Required process` *`name`* `is not running.`

`name`

is the name of the process that is needed to complete the command.

Cause

A process is not running that is needed for the command you issued.

Effect

The command is not executed.

Recovery

Start the required process and reissue the command.

09015

`STORAGE 09015 SMF dependency failure: TMF is not running.`

Cause

The TMF product is not running.

Effect

The command is not executed.

Recovery

Start the TMF product and reissue the command.

09016

`STORAGE 09016 SMF dependency failure: Catalog disk volume` *`name`* `must be TMF audited.`

`name`

is the name of a catalog disk volume that you specified in the command.

Cause

The catalog disk volume you specified is not audited by the TMF product.

Effect

The command is not executed.

Recovery

Use the TMF product to enable the disk volume and reissue the command.

09017

`STORAGE 09017 Invalid ANTLOCATION. The volume has PHYSVOLSELECT OFF.`

Cause

You issued a command that specifies an ANTLOCATION for a disk that will not accept the file because PHYSVOLSELECT OFF disallows writing the file to that disk.

Effect

The command is not executed.

Recovery

Reissue the command to a volume that will accept the file or change the value of the PHYSVOLSELECT attribute of the disk you used in the command.

09018

`STORAGE 09018 SMF dependency failure: Catalog disk volume` *`name`* `is not in a pool.`

`name`

is the name of the catalog disk volume you used in the command you issued.

Cause

The catalog disk volume that you specified in the command does not reside in a storage pool.

Effect

The command is not executed.

Recovery

Put the catalog disk volume into a storage pool and reissue the command.

09019

`STORAGE 09019 SMF dependency failure: SMF process` *`name`* `is not in the STARTED state.`

`name`

is the name of the Storage Management Foundation (SMF) process.

Cause

The command you issued requires that the SMF process be in the STARTED state.

Effect

The command is not executed.

Recovery

Start the process and reissue the command.

09020

`STORAGE 09020 Invalid PENDOPSLOCATION. The volume has PHYSVOLSELECT OFF.`

Cause

You issued a command that specifies a PENDOPSLOCATION for a disk that will not accept the file (PHYSVOLSELECT OFF disallows writing the file to the disk).

Effect

The command is not executed.

Recovery

Reissue the command to a volume that will accept the file or change the value of the PHYSVOLSELECT attribute of the disk you used in the command.

09021

`STORAGE 09021 SMF dependency failure: Invalid syntax for filename`

Cause

The configuration database record contains an invalid file name.

Effect

The command is not executed.

Recovery

Correct the configuration database record that contains the invalid file name and reissue the command.

09022

`STORAGE 09022 SMF dependency failure: Can’t read configuration record for` *`name`*`.`

`name`
: is the name of the device.

Cause

The configuration record is not readable.

Effect

The command is not executed.

Recovery

Correct the configuration record and reissue the command.

09023

`STORAGE 09023 SMF dependency failure: Can’t obtain device information for` *`name`*`.`

`name`
: is the name of the device.

Cause

The configuration record in the database is not readable.

Effect

The command is not executed.

Recovery

Correct the configuration record and reissue the command. If the problem persists, contact your service provider and see “If You Have to Contact Your Service Provider” (page 298).

09024

`STORAGE 09024 SMF dependency failure: Ordinate process` *`name`* `is of wrong type.`

`name`
: is the name of the process.

Cause

You issued a command specifying a monitor or storage pool attribute for an object of the wrong type.

Effect

The command is not executed.

Recovery

Determine the correct object and reissue the command.

09025

`STORAGE 09025 Can't delete pool because virtual disks are associated with it.`

Cause

You are trying to delete a storage pool that still has virtual disks associated with it.

Effect

The command is not executed.

Recovery

Delete the virtual disks associated with the storage pool and then delete the storage pool. Use the SCF INFO POOL, DETAIL command to see which virtual disks are associated with the storage pool.

09026

`STORAGE 09026` *`cmd`* `failed. Mismatched attribute in` *`attname`*`.`

`cmd`
: is the command that you issued.

`attname`
: is the name of the attribute with the mismatched value.

Cause

The system configuration and Storage Management Foundation (SMF) databases are inconsistent. The virtual disk is defined in SMF but is not defined in the configuration database.

When you attempted to add an object to the configuration database, a record for the same object was found in the SMF database. All specified attributes were checked against the SMF database; if there is a mismatch, the request is rejected.

Effect

The command is not executed.

Recovery

Omit the mismatched attribute. (Omitted attributes use values from the SMF database record rather than default values.) Then try again.

09027

`STORAGE 09027 Can't put disk in pool because disk label doesn't match disk name.`

Cause

You attempted to add a disk to a storage pool whose disk name in the configuration database does not match the name on the label.

Effect

The command is not executed.

Recovery

Start the disk with or without the SPECIAL attribute. (If you need to add the disk to the configuration database first, use the SCF ADD command without the POOL attribute.)

- If you start the disk without the SPECIAL attribute, the storage subsystem automatically changes the name in the configuration database to match the disk label.
- To change the label to match the configuration database name, start the disk with the SPECIAL attribute and then use the SCF ALTER DISK command or SCF RENAME DISK command to change the disk label.
- When the name in the configuration database matches the name on the disk label, then you can use the ALTER command to add the disk to a storage pool.

09028

`STORAGE 09028 Can't delete pool because physical disks are associated with it.`

Cause

You are trying to delete a storage pool that still has disks associated with it.

Effect

The command is not executed.

Recovery

Delete the disks associated with the storage pool, then delete the storage pool. Use the SCF INFO POOL, DETAIL command to see which disks are associated with a storage pool.

09029

```
STORAGE 09029 The attribute refers to a device of an incorrect type.
```

Cause

You specified an attribute that is incompatible with the type of device specified in the command.

Effect

The command is not executed.

Recovery

Change the command so that the device and attribute match. For example, if POOL is specified, the name must be the name of a storage pool. If you specify the LIKE attribute, both object names must be of the same type.

09030

```
STORAGE 09030 This disk is already associated with another pool.
```

Cause

You issued a command to associate a physical volume with a storage pool, but the volume is already associated with another storage pool.

Effect

The command is not executed.

Recovery

Use the SCF ALTER command to delete the physical volume from the storage pool it is associated with before attempting to associate it with a different storage pool.

09031

```
STORAGE 09031 This disk is not of the same sub type as specified.
```

Cause

The specified disk type does not support the specified attribute.

Effect

The command is not executed.

Recovery

Correct the command using the proper attribute for the disk type.

09032

STORAGE 09032 *devname* is not a disk in pool *poolname*.

devname

is the name of the device you specified in the command.

poolname

is the name of the storage pool you specified in the command.

Cause

You specified an incorrect name for either the disk or the storage pool.

Effect

The command is not executed.

Recovery

Issue the command with the correct names.

09033

`STORAGE 09033` *`Required attribute not specified attribute.`*

`attribute`
: is the name of an attribute.

Cause

You issued a command without specifying a required attribute.

Effect

The command is not executed.

Recovery

Make the request again, using the required attribute.

09034

`STORAGE 09034` *`devname cannot be configured on a sacname which is on an adaptername.`*

`devname`
: is the name of the device you specified in the command.

`sacname`
: is the name of the SAC you specified in the command.

`adaptername`
: is the name of the adapter you specified in the command.

Cause

The configurations of the device, the SAC, and the adapter are incompatible.

Effect

The command is not executed.

Recovery

Make the request again, specifying a compatible configuration.

09035

`STORAGE 09035` *`Can't delete pool because a disk profile refers to the pool.`*

Cause

You attempted to delete a storage pool that is referenced by a disk profile.

Effect

The command is not executed.

Recovery

1. Issue the INFO, PROFILE command to determine which disk profiles refer to the storage pool.
2. Delete the disk profile that refers to the pool.
3. Retry the command.

09036

`STORAGE 09036` *`Invalid attribute values: Controller SCSI ids are the same for both SACs.`*

Cause

You attempted to add a SCSI device that has the same controller IDs on both its primary and backup SAC.

Effect

The command is not executed.

Recovery

Change either the primary SAC or the backup SAC so that they have different controller IDs.

The SCSI Bus IDs of the SACs on a ServerNet/DA are always 6 or 7. The ID numbers are assigned in a checkerboard fashion, so that these values will always be different:

- The IDs of the same SAC number on two adjacent boards
- The IDs of adjacent SAC numbers on the same board

B Upgrade and Replacement Procedures

This section describes:

Replacing a Tape Drive

If you physically remove a tape drive and replace it with another tape drive that is controlled by the same adapter, the tape process automatically recognizes the new tape drive. For other details about installing or removing a tape drive, see the documentation that comes with the tape drive or contact your service provider.

Stop the tape drive:

```
-> STOP $TAPE0
```

1. Verify the tape drive is in the STOPPED state:
   ```
   -> STATUS $TAPE0
   ```
2. Remove the device cable from the connector on the system adapter.
3. If you are using the same cable on the new tape drive, disconnect the device cable from the old tape drive.
4. Connect the device cable to the connector on the new tape drive.
5. Always connect the device cable to the tape drive before connecting the cable to the adapter. If the cable is not connected to the device first, the SCSI bus is unterminated and could produce SCSI bus errors.
6. Connect the device cable from the tape drive to the system connector.
7. Start the new tape drive:
   ```
   -> START $TAPE0
   ```
   The tape process automatically detects that a new tape drive has been installed.

Replacing an Open SCSI Device

To replace an Open SCSI device:

1. Stop access to the device:
   ```
   -> STOP $S11500
   ```
2. Verify the device is in the STOPPED state:
   ```
   -> STATUS $S11500
   ```
3. Physically remove the device from the SCSI chain.
4. Configure the SCSI ID and LUN on the device according to the manufacturer’s instructions.
5. Physically install the new device in the SCSI chain.
6. Depending on the manufacturer’s instructions in Step 4, alter the configuration:
   ```
   -> ALTER $S11500, SCSIID 5
   ```
7. Start the device:
   ```
   -> START $S11500
   ```
   The Open SCSI IOP automatically detects that a different model Open SCSI device has been installed.

Upgrading a Mirrored Volume Online

You can replace online each disk in a mirrored volume (one at a time). The replacement disk can be a higher capacity or faster speed. For example, you can replace 8-GB disks with 36-GB disks without bringing down the entire volume.

NOTE: You can only use a smaller replacement disk if it is able to accommodate the largest used sector address. Please note that because of fragmentation, the largest used sector address can be much larger than the actual disk space used.

1. Verify the volume is currently mirrored and that at least one path to each disk is in the STARTED state:

   ```
   -> STATUS DISK $DATA01-*
   ```

2. If the disk contains the TMF master audit-trail, either:
   - Stop TMF.
   - Use the STOP, FORCED command to allow the mirror half of the TMF MAT volume to stop while TMF is active.
3. Bring down the mirror and mirror backup paths:

   ```
   -> STOP $DATA01-M
   -> STOP $DATA01-MB
   ```

4. Physically remove the stopped mirror disk.
5. Install a higher-capacity or faster disk in the location where you removed the mirror disk.
6. If this is an internal disk, it powers on and immediately begins to revive if you have enabled AUTOSTART and AUTOREVIVE for both the specific disk and for the subsystem.
7. If you do not have AUTOSTART and AUTOREVIVE enabled, start the volume and revive the mirror disk:

   ```
   -> START $DATA01
   ```

   When the revive operation is complete on the mirror disk, the primary disk (with the lower capacity) automatically enters the STOPPED state, substate HARDDOWN.

   If the new disk has the same capacity but is faster, the primary disk does not automatically stop when the revive operation is complete. Stop the slower disk and put it into the STOPPED state, substate HARDDOWN:

   ```
   -> STOP ($DATA-P, $DATA-B)
   ```

8. Physically remove the stopped primary disk.
9. Install the second higher-capacity or faster disk in the location where you removed the disk.
10. If necessary, put the disk into the STOPPED state, substate DOWN:

    ```
    -> RESET $DATA01
    ```

11. Bring up the volume:

    ```
    -> START $DATA01
    ```

When the revive operation finishes on the primary disk, the entire mirrored volume is available for use.

Glossary

$ZCNF The name of the configuration utility process.

$ZZSTO The name of the storage subsystem manager process.

45*xx* disk A disk drive that resides in slots 0 through 7 of a modular disk subsystem outside of NonStop S-series system enclosures. The disk subsystem connects to the server through a ServerNet/DA Adapter (SNDA).

6760 ServerNet device adapter (ServerNet/DA) See ServerNet device adapter (ServerNet/DA).

ACL See automatic cartridge loader (ACL).

action An operation that can be performed on a selected resource.

adapter See ServerNet adapter.

ADAPTER object type The Subsystem Control Facility (SCF) object type for all adapters attached to your system.

alternate path A path not enabled as the preferred path. An alternate path can become a primary path when a primary path is disabled. Contrast with primary path.

assumed object The object type or object name specified by a Subsystem Control Facility (SCF) ASSUME command. If an ASSUME command has been used to establish a default object type and fully qualified default object name, and if that object type and object name together refer to a valid object, then *object-spec* can be omitted entirely from an SCF command, and the command is applied to the object known as the assumed object.

attribute For the Subsystem Control Facility (SCF), a characteristic of an entity. For example, two attributes of a process might be its program file and its user ID. An attribute is sometimes called a modifier.

automatic cartridge loader (ACL) A device that stores multiple 3480, 3490, or digital audio tape (DAT) format cartridge tapes and loads them automatically, one at a time, into a tape drive.

automatic configuration The automatic assignment of disk attributes to an internal disk drive when it is inserted into a slot. Also known as **plug and play**.

automatic disk changer See robot.

BACKUP A utility for HP NonStop servers that creates a backup copy of one or more disk files on magnetic tape. See also RESTORE.

backup processor A processor in the HP NonStop operating system that communicates with the primary processor, allowing the processors to remain independent. A component failure in one processor has no effect on any other processor. Contrast with primary processor.

block A grouping of one or more system enclosures that an HP NonStop S-series system recognizes and supports as one unit. A block can consist of either one processor enclosure, one I/O enclosure, or one processor enclosure with one or more I/O enclosures attached.

boot A synonym for load. Load is the preferred term used in this and other HP NonStop S-series system manuals. See load.

cache (cache memory) A small, fast memory holding recently accessed data designed to speed up subsequent access to the same data. Cache memory is built from faster memory chips than main memory, and it is most often used with process or main memory but also used in network data transfer to maintain a local copy of data.

CE Customer engineer. See service provider.

checksum A generic term, meaning to add together (although the definition of add need not be a normal arithmetic add) all of the data to produce a check word.

cold load A synonym for system load or load (in the case of single processor load). System load or load is the preferred term in this and other HP NonStop S-series system manuals. See system load and load.

command A demand for action by or information from a subsystem or the operation demanded by an operator or application. A command is typically conveyed as an interprocess message from an application to a subsystem.

command file An EDIT file that contains a series of commands and serves as a source of command input.

Compaq TSM Identifies a client or server software component used to manage or service HP NonStop S-series servers.

Compaq TSM package A software product for HP NonStop S-series servers that provides the information needed to perform functions such as querying resources and testing, provides notification of problems on the system, and allows local or remote access to the system for service and maintenance. The TSM package performs the same role as that of HP Tandem Maintenance and Diagnostic System (TMDS), Syshealth, and Remote Maintenance Interface (RMI) on earlier systems.

CONFIG file In G-series RVUs, the current system configuration database file, which is stored on the $SYSTEM.ZSYSCONF subvolume. See also configuration file.

configuration The definition or alteration of characteristics of an object.

configuration file In G-series RVUs, one of these files: CONFBASE, CONFIG, one or more saved configuration files named CONF*xxyy*, and CONFSAVE. See also system configuration database. In pre-G-series RVUs, the configuration file is either the OSCONFIG file used by the Configuration Utility Program (COUP) or the CONFTEXT file used by SYSGENR.

Configuration Utility Program (COUP) A utility used in D-series and earlier RVUs to make online changes to the configuration of devices and controllers. COUP is part of the Dynamic System Configuration (DSC) facility. In G-series RVUs, COUP functions are performed by the Subsystem Control Facility (SCF).

controller See ServerNet addressable controller (SAC).

COUP See Configuration Utility Program (COUP).

current configuration file See configuration file.

customer engineer (CE) See service provider.

detailed report A complete listing of status or configuration information provided by the Subsystem Control Facility (SCF) STATUS or INFO command when you use the DETAIL option. Contrast with summary report.

direct connect Connection from the FCSA to the ESS without going through an FC switch.

disk bootstrap A software entity residing on disk that is used to load the HP NonStop operating system image (OSIMAGE) into memory during a system load of a NonStop S-series server. A disk that contains the disk bootstrap is referred to as a bootable disk. The disk bootstrap is placed on the disk either as part of a tape load or as a result of the SCF CONTROL DISK, REPLACEBOOT command.

On a Integrity NonStop NS-series server, the bootstrap program is installed through firmware update into flash memory.

disk cache A temporary storage buffer into which data is read, retained, and perhaps updated before being written to disk, for more efficient processing.

disk drive A device that stores and accesses data on a disk. Random access to addressable locations on a disk is provided by disk read/write heads. See also volume.

DISK object type The Subsystem Control Facility (SCF) object type for all disk devices attached to your system.

disk volume See volume.

DSC See Dynamic System Configuration (DSC).

Dynamic System Configuration (DSC) A utility used in D-series and earlier RVUs to make online changes to the configuration of devices and controllers. Its interactive utility is called the Configuration Utility Program (COUP). In G-series RVUs, DSC functions are performed by Subsystem Control Facility (SCF).

EMS See Event Management Service (EMS).

enclosure Similar to a cabinet in HP NonStop K-series systems. An enclosure can contain components of a system or a peripheral. Base enclosures are placed on the floor and can have other enclosures stacked on top of them. Stackable enclosures can be placed on top of other enclosures. See also system enclosure and peripheral enclosure.

enclosure interleaving On HP NonStop S-series systems, configuring a mirrored disk volume to use two separate system enclosures. For internal disk drives, the two disk drives of the mirrored volume can be in separate enclosures. For 45*xx* disk drives, the adapters connected to the two disk drives of the mirrored volume can be in separate enclosures.

Enterprise Storage System (ESS) A collection of disks, their controllers, and the disk cache in a standalone cabinet or cabinets. These disks are configured by an attached console and presented to the attached server as logical volumes that can be a fraction of a physical volume or can span volumes.

error number For the Subsystem Programmatic Interface (SPI), a value that can be assigned to a return token, or to the last field of an error token, to identify an error that occurred. SPI defines a small set of error numbers, but most error numbers are defined by subsystems.

ESS disk A disk drive within an Enterprise Storage System (ESS).

Event Management Service (EMS) A Distributed Systems Management (DSM) product that provides event collection, event logging, and event distribution facilities. EMS provides different event descriptions for interactive and programmatic interfaces, lets an operator or an application select specific event-message data, and allows for flexible distribution of event messages within a system or network.

event message Text intended for a system operator that describes a change in some condition in the system or network, whether minor or serious. The change of condition is called an event. Events can be operational errors, notifications of limits exceeded, requests for actions needed, and so on. See also operator message.

extent A contiguous area on disk for allocating one file.

fabric A complex set of interconnections through which there can be multiple and (to the user) unknown paths from point to point. The term **fabric** is used to refer to the X or Y portion of the internal or external network; for example, the X fabric.

fault tolerance The ability of a HP NonStop S-series system to continue processing despite the failure of any single software or hardware component within the system.

FC switch See Fibre Channel switch (FC switch).

Fibre Channel ServerNet adapter (FCSA) A ServerNet adapter that transmits data between an HP NonStop™ server and Fibre Channel storage devices. This ServerNet adapter is installed in an IOAM enclosure.

Fibre Channel switch (FC switch) Networking hardware that can connect an Enterprise Storage System and a NonStop server. This switch allows any-to-any connectivity.

file name A unique name for a file. File names for disk files normally have at least two parts (the disk name and the file name); an example of a file name on a PC is B:MYFILE. In the Guardian environment on HP NonStop S-series systems, disk file names include a disk volume name, a subvolume name, and a file identifier. An example of a file name on a NonStop S-series system is $DISK.SUBVOL.MYFILE. For files that are network accessible, the node name precedes the volume name: \NODE.$DISK.SUBVOL.MYFILE.

generic process A process created and managed by the Kernel subsystem. Also known as a system-managed process. A common characteristic of a generic process is persistence.

gigabyte (GB) A unit of measurement equal to 1,073,741,824 bytes (1024 megabytes). See also kilobyte (KB), megabyte (MB), and terabyte (TB).

Global NonStop Support Center (GNSC) A support organization that provides telephone and remote diagnostic support for HP customers. There are GNSCs located all over the world.

group The set of all objects accessible by a pair of service processors (SPs) located in the processor multifunction (PMF) customer-replaceable unit (CRU). In an HP NonStop S-series server, there is exactly one group in a system enclosure.

Guardian An environment available for interactive or programmatic use with the HP NonStop operating system. Processes that run in the Guardian environment usually use the Guardian system procedure calls as their application program interface; interactive users of the Guardian environment usually use the HP Tandem Advanced Command Language (TACL) or another HP product’s command interpreter.

HP Integrity NonStop™ NS-series servers
The HP NonStop servers having product numbers beginning with the letters **NS**. These servers implement the ServerNet architecture and run the HP NonStop operating system.

HP NonStop™ operating system
The operating system for HP NonStop systems.

HP NonStop™ S-series servers
The set of servers in the HP NonStop range of servers having product numbers beginning with the letter **S**. These servers implement the ServerNet architecture and run the HP NonStop operating system.

HP NonStop™ Transaction Management Facility (TMF)
HP software that provides transaction protection and database consistency in demanding online transaction processing (OLTP) and decision-support environments. It gives full protection to transactions that access distributed SQL and Enscribe databases, as well as recovery capabilities for transactions, online disk volumes, and entire databases.

HP Open System Management (OSM) product
See OSM.

HP Tandem Advanced Command Language (TACL)
The user interface to the HP NonStop operating system. The TACL product is both a command interpreter and a command language. Users can write TACL programs that perform complex tasks simply or provide a consistent user interface across independently programmed applications.

I/O adapter module (IOAM)
A collection of modular components that provides I/O connectivity and can include ServerNet switch boards, Fibre Channel ServerNet adapters (FCSAs), Gigabit Ethernet 4-port ServerNet adapters (G4SAs), fans, and power supplies. In the IOAM, each module is a logical entity that represents a single service domain.

I/O cabinet
See I/O enclosure.

I/O enclosure
An HP NonStop S-series system enclosure containing exactly one module, which includes ServerNet adapters, disk drives, components related to the ServerNet fabrics, and components related to electrical power and cooling for the enclosure. An I/O enclosure is identical to a processor enclosure, except that it contains I/O multifunction (IOMF) customer-replaceable units (CRUs) instead of processor multifunction (PMF) CRUs.

I/O multifunction (IOMF) 2 CRU
A HP NonStop S-series customer-replaceable unit (CRU) that connects an I/O enclosure to a processor enclosure through a ServerNet cable and supplies power to the components within the IOMF 2 CRU as well as redundantly to the disk drives, SCSI terminators, and ServerNet adapters in that enclosure. The IOMF 2 CRU contains a power supply, a service processor (SP), a ServerNet router 2, an Ethernet controller, three configurable ServerNet ports, and three SCSI ServerNet addressable controllers (S-SACs) in a single unit. IOMF 2 CRUs are supported on G06.10 and later software RVUs.

I/O multifunction (IOMF) CRU
1. A HP NonStop S-series customer-replaceable unit (CRU) that connects an I/O enclosure to a processor enclosure through a ServerNet cable and supplies power to the components within the IOMF CRU as well as redundantly to the disk drives, SCSI terminators, and ServerNet adapters in that enclosure. The IOMF CRU contains a power supply, a service processor (SP), a ServerNet router, an Ethernet controller, an external ServerNet port, and three SCSI ServerNet addressable controllers (S-SACs) in a single unit.
2. A collective term for both IOMF CRUs and IOMF 2 CRUs when a distinction between the two types of CRUs is not required.

input/output process (IOP)
A running program (part of the HP NonStop operating system) that manages the I/O functions for one or more ServerNet addressable controllers (SACs) of the same type.

interactive mode
A mode of operation that is characterized by having the same input and output device (a terminal or a process) for the session. If a terminal is used, a person enters a command and presses Return. If a process is used, the system interface waits for the process to send a request and treats the process in the same manner as a terminal. Contrast with noninteractive mode.

Internal disk
A SCSI disk that resides in slots 1 through 18 of a NonStop S-series system enclosure.

IOMF 2 CRU
See I/O multifunction (IOMF) 2 CRU.

IOMF CRU
See I/O multifunction (IOMF) CRU.

IOP
See input/output process (IOP).

KB See kilobyte (KB).

Kernel subsystem In G-series RVUs, the subsystem for configuration and management of the Subsystem Control Facility (SCF) subsystem managers that are generic processes, some system attributes, and the ServerNet X and Y fabrics.

kilobyte (KB) A unit of measurement equal to 1024 bytes. See also gigabyte (GB), megabyte (MB), and terabyte (TB).

LDEV Logical device. The HP term for a disk in the ESS.

load **1.** To transfer the HP NonStop operating system image or a program from disk into a computer's memory so that the operating system or program can run.
2. To insert a tape into a tape drive, which prepares it for a tape operation (read or write).

logical unit number (LUN) The logical unit in the ESS that maps the LDEV onto a port. Specification of the port and LUN specifies an LDEV.

M8*xxx* disk A model M8*xxx* Fibre Channel disk drive that resides in slots 1 through 14 of a Fibre Channel disk module (FCDM) connected to FCSAs in an IOAM enclosure.

MB

See megabyte (MB).

Measure product A HP utility used to gauge system performance.

media changer See robot.

megabyte (MB) A unit of measurement equal to 1,048,576 bytes (1024 kilobytes). See also gigabyte (GB), kilobyte (KB), and terabyte (TB).

microcode Any machine code or data that can run in a microprocessor. HP produces two types of microcode for HP NonStop S-series systems: volatile and nonvolatile. Volatile microcode is loaded into the volatile random-access memory (RAM) of some types of printed wiring assemblies (PWAs) and is not retained in a host PWA when power to the PWA is interrupted.

mirrored volume A pair of identical disk drives that are used together as a single logical volume. One drive is considered primary and the other is called the mirror. Each byte of data written to the primary drive is also written to the mirror drive; if the primary drive fails, the mirror drive can continue operations. See also volume.

modular ServerNet expansion board (MSEB) A ServerNet expansion board (SEB) that uses plug-in cards (PICs) to provide a choice of connection media for routing ServerNet packets. See also ServerNet expansion board (SEB).

module A set of components sharing a common interconnection, such as a backplane. A module is a subset of a group, and it is usually contained in an enclosure. In a HP NonStop S-series server, there is exactly one module in a group.

MON object type The Subsystem Control Facility (SCF) object type for the Storage Management Foundation (SMF) Master Process. See also Storage Management Foundation (SMF).

MSEB See modular ServerNet expansion board (MSEB).

noninteractive mode A mode of operation that usually involves a command file (an EDIT file that contains a series of commands). Contrast with interactive mode.

nonsensitive command A command that can be issued by any user or program that is allowed access to a subsystem—that is, a command on which the subsystem imposes no further security restrictions. For SCF, nonsensitive commands are those that cannot change the state or configuration of objects; most of them are information commands. Contrast with sensitive command.

NonStop™ NS-series servers See HP Integrity NonStop™ NS-series servers.

NonStop™ operating system See HP NonStop™ operating system.

NonStop™ S-series servers See HP NonStop™ S-series servers.

OBEY file See command file.

object One or more of the devices, lines, processes, and files in a subsystem; any entity subject to independent reference or control by one or more subsystems. In the Subsystem Control Facility (SCF), each object has an object type and object name.

object name A unique name for an SCF object within a subsystem.

object type The category of Subsystem Control Facility (SCF) objects to which a specific SCF object belongs; for example, a specific disk has the object type DISK and a specific terminal may have the object type SU. Each subsystem has a set of object types for the objects it manages.

object-name template. A name that stands for more than one Subsystem Control Facility (SCF) object. Such a name includes one or more wild-card characters, such as * (asterisk) and ? (question mark). See also wild-card character.

operator message A message, intended for an operator, that describes a significant event on a HP NonStop S-series system. An operator message is the displayed-text form of an Event Management Service (EMS) event message. See also event message.

OSM Stands for HP Open System Management (OSM) Interface. Replaces TSM as the system management tool of choice for NonStop servers. Provides the same functionality as TSM while overcoming limitations of TSM. OSM is required for support of new functionality released in G06.21 and later.

peripheral enclosure An enclosure that contains components related to one or more peripherals. The 519x tape subsystem is an example of a peripheral enclosure. Peripheral enclosures are not part of the set of system enclosures. Contrast with system enclosure.

Peripheral Utility Program (PUP) A utility used in D-series and earlier RVUs to manage disks and other peripheral devices. In G-series RVUs, PUP functions are performed by the Subsystem Control Facility (SCF).

PIN See process identification number (PIN).

PMF 2 CRU See processor multifunction (PMF) 2 CRU.

PMF CRU See processor multifunction (PMF) CRU.

POOL object type The Subsystem Control Facility (SCF) object type for Storage Management Foundation (SMF) storage pools. See also Storage Management Foundation (SMF).

primary path A path enabled as the preferred path. When a primary path is disabled, an alternate path becomes the primary path. Contrast with alternate path.

primary processor The processor that is designated as owning the ServerNet addressable controller (SAC) connected to separate processors running the HP NonStop operating system. The primary processor is the processor that has direct control over the SAC. Contrast with backup processor.

process A program that has been submitted to the operating system for execution, or a program that is currently running in the computer.

process ID A number that uniquely identifies a process. It consists of the processor (CPU) number and the process identification number (PIN).

process identification number (PIN) A number that uniquely identifies a process running in a processor. The same number can exist in other processors in the same system. See also process ID.

processor **1.** A functional unit of a computer that reads program instructions, moves data between processor memory and the input/output controllers, and performs arithmetic operations. Processors are sometimes referred to as central processing units (CPUs), but a HP NonStop system has multiple cooperating processors rather than a single CPU.
2. One or more computer chips, typically mounted on a logic board, that are designed to perform data processing or to manage a particular aspect of computer operations.

processor enclosure A HP NonStop S-series system enclosure containing exactly one group, which includes processors, ServerNet adapters, disk drives, components related to the ServerNet fabrics, and components related to electrical power and cooling for the enclosure.

processor multifunction (PMF) 2 CRU A HP NonStop S-series customer-replaceable unit (CRU) that contains a power supply, service processor (SP), ServerNet router 2, Ethernet controller, three ServerNet addressable controllers (SACs), and a processor and memory system in a single unit. The PMF 2 CRU consists of three subassemblies: the processor and memory board (PMB), the multifunction I/O board (MFIOB), and the power supply subassembly.

processor multifunction (PMF) CRU — **1.** A HP NonStop S-series customer-replaceable unit (CRU) that contains a power supply, service processor (SP), ServerNet router 1, Ethernet controller, three ServerNet addressable controllers (SACs), and a processor and memory system in a single unit. The PMF CRU consists of three subassemblies: the processor and memory board (PMB), the multifunction I/O board (MFIOB), and the power supply subassembly.
2. A collective term for both PMF CRUs and PMF 2 CRUs when a distinction between the two types of CRUs is not required.

PROFILE object type — The Subsystem Control Facility (SCF) object type for the storage subsystem configuration profile.

PUP — See Peripheral Utility Program (PUP).

RESTORE — A utility for the HP NonStop range of servers that copies files from a backup tape to disk. See also BACKUP.

robot — A media-changer device that transfers an tape cartridge from a storage cell to a drive for use, then returns the cartridge to the storage cell.

SAC — See ServerNet addressable controller (SAC).

SAN — System area network. The preferred term is fabrics (see fabric).

SCF — See Subsystem Control Facility (SCF).

SCSI — See small computer system interface (SCSI).

SCSI object type — The Subsystem Control Facility (SCF) object type for the Open SCSI object.

sensitive command — A Subsystem Control Facility (SCF) command that can be issued only by a user with super-group access, by the owner of the subsystem, or by a member of the group of the owner of the subsystem. The owner of a subsystem is the user who started that subsystem (or any user whose application ID is the same as the server ID—the result of a PROGID option that requires super-group access). Contrast with nonsensitive command.

ServerNet adapter — A component that connects peripheral devices to the rest of the system through a ServerNet bus interface (SBI). A ServerNet adapter is similar in function to an I/O controller logic board (LB) and backplane interconnect card (BIC) in HP NonStop K-series servers.

ServerNet addressable controller (SAC) — An I/O controller that is uniquely addressable by a ServerNet ID in the ServerNet fabrics. A SAC is typically implemented on some portion of a processor multifunction (PMF) customer-replaceable unit (CRU), an I/O multifunction (IOMF) CRU, or a ServerNet adapter.

ServerNet device adapter (ServerNet/DA) — A ServerNet adapter that controls 45*xx* disk or tape drives installed in a modular storage subsystem. The 6760 ServerNet/DA contains up to four ServerNet addressable controllers (SACs), each of which can control either disk drives or tape drives.

ServerNet expansion board (SEB) — **1.** A connector board that plugs in to the backplane to allow one or more ServerNet cables to exit the rear of the enclosure. The SEBs and ServerNet cables allow processors in one group to communicate with processors in another group. Each SEB provides either the ServerNet X fabric or the ServerNet Y fabric for a group.
2. A collective term for both SEBs and modular SEBs (MSEBs) when a distinction between the two types of SEBs is not required.

ServerNet/DA — See ServerNet device adapter (ServerNet/DA).

service provider — **1.** A person trained and qualified to service field-replaceable units (FRUs).
2. An organization, such as the Global NonStop Support Center (GNSC), that helps you resolve problems with your HP NonStop S-series server. The HP TSM package allows you to use the help of a service provider by configuring TSM to support remote notification and remote access.

slot — A physical, labeled space for a customer-replaceable unit (CRU) or field-replaceable unit (FRU) that is part of a module. A module contains one or more slots.

slot location — A three-number identifier for a particular slot on a system that consists of the group number, module number, and slot number; for example, 02,01,08 (group 02, module 01, slot 08).

small computer system interface (SCSI) — An ANSI-standard protocol used by a controller to access a device.

SMF — See Storage Management Foundation (SMF).

SNDA See ServerNet device adapter (ServerNet/DA).

state In SCF, one of the generally defined possible conditions of an object with respect to the management of that object. Examples of states are DEFINED, STARTED, and STOPPED.

Storage Management Foundation (SMF) A subsystem used by the storage subsystem that facilitates automation of storage management tasks by providing location-independent naming, storage pools, and virtual disks on HP NonStop S-series systems.

storage subsystem A subsystem of the HP NonStop operating system that handles configuration and management of disk and tape devices in G-series RVUs.

storage subsystem manager process The generic process that starts and manages disk and tape drives. The $ZZSTO storage subsystem manager process is started and managed by the $ZZKRN Kernel subsystem manager process through the $ZPM persistence manager process.

SUB option In some Subsystem Control Facility (SCF) subsystems, the designation that the object name given in a command stands not just for itself, but for the names of all objects at the next-lower level in the hierarchy. The given object name can stand both for itself and for the subordinate objects, or it can stand only for the subordinate objects, depending on the value of the SUB option.

subordinate objects In Subsystem Control Facility (SCF), objects that are logically subordinate to other objects. Some subsystems are structured hierarchically, with objects of one type logically subordinate to (that is, controlled by) an object of another type. For example, a number of subdevices can be configured on a single line. Some SCF commands include a SUB option that refers to subordinate objects.

substate Further information about the state of a device. The state and substate together provide information about the current condition of a device or path to a device.

SUBSYS object type The Subsystem Control Facility (SCF) object type for most subsystems that use SCF as the user interface.

subsystem A program or set of processes that manages a cohesive set of Subsystem Control Facility (SCF) objects. Each subsystem has a manager through which applications can request services by issuing commands defined by that subsystem. See also subsystem manager.

Subsystem Control Facility (SCF) An interactive interface for configuring, controlling, and collecting information from a subsystem and its objects. SCF enables you to configure and reconfigure devices, processes, and some system variables while your HP NonStop S-series server is online.

Subsystem Control Point (SCP) The message router for SCF. There can be several instances of this process. Using the Subsystem Programmatic Interface (SPI), applications send each command for a subsystem to an instance of the SCP process, which in turn sends the command to the manager process of the target subsystem. SCP also processes a few commands itself. It provides security features, version compatibility, support for tracing, and support for applications implemented as process pairs.

subsystem manager A process that performs configuration and management functions for a Subsystem Control Facility (SCF) subsystem.

subvolume A group of related files stored on a disk; all the files have the same volume and subvolume name, but each file has a unique file identifier.

summary report A brief informational listing of status or configuration information provided by the Subsystem Control Facility (SCF) STATUS or INFO command. Contrast with detailed report.

super group The group of user IDs that have 255 as the group number. This group has special privileges; many HP utilities have commands or functions that can be executed only by a member of the super group.

super-group user A user who can read, write, execute, and purge most files on the system. Super-group users have user IDs that have 255 as the group number.

SYS*nn* subvolume A subvolume on the $SYSTEM volume where the new version of the HP NonStop operating system image is located. Also located on the SYS*nn* subvolume is system-dependent and RVU-dependent software. *nn* is an octal number in the range %00 through %77.

system A node. All the processors, controllers, firmware, peripheral devices, software, and related components that are directly connected together to form an entity that is managed by one HP NonStop operating system image and operated as one computer.

system configuration database
The database file on the $SYSTEM.ZSYSCONF subvolume that contains configuration information for all system objects that can be configured by the Subsystem Control Facility (SCF). Configuration information for all system objects that can be configured by SYSGENR is contained on the $SYSTEM.SYS*nn* subvolume. See also configuration file and SYS*nn* subvolume.

system enclosure
An enclosure for NonStop S-series system components. Processor enclosures and I/O enclosures are both system enclosures. Contrast with peripheral enclosure.

system engineer (SE)
See service provider.

system load
1. To start the system; to load the HP NonStop operating system image into the memory of a processor.
2. The process of loading the operating system. A system load changes a system from an inactive to an active (or operational) state by loading software that establishes communication between the operating system and configured system peripherals.

system-managed process
Another name for a generic process. See generic process.

TACL
See HP Tandem Advanced Command Language (TACL).

tape drive
A device that moves magnetic tape past magnetic read/write heads, which read data from or write data to the tape.

TAPE object type
The Subsystem Control Facility (SCF) object type for all tape drives attached to your system.

TB
See terabyte (TB).

terabyte (TB)
A unit of measurement equal to 1,099,511,627,776 bytes (1024 gigabytes). See also gigabyte (GB), kilobyte (KB), and megabyte (MB).

TMF
See HP NonStop™ Transaction Management Facility (TMF).

topology
The physical layout of components that define a local area network (LAN), wide area network (WAN), or ServerNet fabric.

topology branch
A NonStop S-series processor enclosure and the I/O enclosures attached to it.

TSM
See Compaq TSM.

TSM package
See Compaq TSM package.

uninterruptable power source (UPS)
The equipment used to provide an uninterruptable source of power to connected equipment if a main power outage occurs. The basic components of any UPS system are a rectifier/charger that converts alternating-current (ac) power to direct-current (dc) power, batteries that store the dc power, and an inverter that converts the dc power back into ac power for distribution to the load.

UPS
See uninterruptable power source (UPS).

volume
A logical disk drive, which can be one or two disk drives. In HP NonStop S-series systems, volumes have names that begin with a dollar sign ($), such as $DATA. See also mirrored volume.

wild-card character
A character that stands for any possible character or characters in a search string or in a name applying to multiple objects. In Subsystem Control Facility (SCF) object-name templates, two wild-card characters can appear:? (question mark) for a single character and * (asterisk) for zero or more consecutive characters. See also object-name template.

world-wide name (WWN)
A unique, 64-bit number assigned to hardware ports.

WORM
Write-Once Read-Many times. A media storage class in which data, once written, cannot be erased or overwritten.

WWN
See world-wide name (WWN).

WWN zone
Similar to Ethernet virtual LANs (VLANs), WWN zones organize the cabling and interactions between components in a storage and server system.

X fabric
The X side of the internal or external ServerNet fabrics. See also fabric.

Y fabric
The Y side of the internal or external ServerNet fabrics. See also fabric.

Index

Symbols

A

B

C

M

U

V

W

X

Y