Origin™ FibreVault and Fibre Channel RAID Administrator's Guide

Origin™ FibreVault and Fibre Channel RAID Administrator's Guide
Document Number 007-3715-001

Contents

List of Figures

List of Tables

About This Guide

This guide explains the Fibre Channel administrative software, including the command-line interfaces (CLIs) and the Fibre Channel RAID (Redundant Array of Independent Disks) graphical user interface (GUI). These interfaces to the Silicon Graphics Fibre Channel storage systems enable you to administer, monitor, and adjust FibreVault (non-RAID) and Fibre Channel RAID disk arrays.

The Fibre Channel storage systems interface with Silicon Graphics Origin2000, Origin200, Onyx2, and OCTANE systems. The software described in this guide runs on a console or terminal attached to one of these workstations, which is running IRIX level 6.4 or later.

Audience

This guide is for the system administrator responsible for configuring and administering Silicon Graphics fibre channel options. This person should have an understanding of the Silicon Graphics Fibre Channel RAID and non-RAID (also known as JBOD, or "just a bunch of disks") storage-system hardware and its administration, RAID types, and the hardware and software components of all array configurations that the software is to administer.

For information on RAID levels supported for IRIS FailSafe, ask your Silicon Graphics Sales Engineer.

Structure of This Guide

This guide is organized as follows:

- Chapter 1, "Fibre Channel RAID Graphical User Interface Features," introduces the RAID GUI and the agent *ssmagent*, and explains RAID features and types (levels).

- Chapter 2, "Using the RAID GUI to Configure Arrays," explains how to configure an array with the RAID GUI.

- Chapter 3, "Reconfiguring and Fine-Tuning," explains how to reconfigure a LUN by changing its parameters or storage-processor (SP) ownership and how to change caching parameters.

- Chapter 4, "Monitoring Arrays and Displaying System Statistics," explains how to display status and statistics information for arrays and their SPs, LUNs, disk modules, and caching. It also explains the roles of polling and statistics logging.

- Chapter 5, "Identifying and Correcting Failures," describes how the RAID GUI displays information about array health, and outlines procedures you can use to detect and correct problems.

- Chapter 6, "The Fibre Channel RAID Command-Line Interface," explains how to use the *ssmcli* command for communicating with RAID fibre channel storage.

- Chapter 7, "The Fibre Channel Non-RAID Command-Line Interface," explains how to use the *fccli* command for communicating with non-RAID (JBOD) fibre channel storage.

- Appendix A, "Error Recovery," gives instructions on restarting the agent, re-establishing communication with an array, and recovering from an incorrect bind, and explains the error codes that appear in the SP event log.

- Appendix B, "Server Serial Connector Pinouts," lists pinouts for Silicon Graphics platforms that can be used with Fibre Channel RAID disk arrays.

An index completes this guide.

Other Documentation

Besides this manual, have handy these owner's guides:

- *Origin FibreVault and Fibre Channel RAID Owner's Guide*

- owner's guide(s) for the server(s) in your configuration

If you don't have these guides handy, the information is also online in the following locations:

- IRIS InSight Library: from the Toolchest, choose Help > Online Books >
  SGI EndUser or SGI Admin, and select the applicable owner's or hardware guide.

- Technical Publications Library: if you have access to the Internet, enter the
  following URL in your Web browser location window:
  http://techpubs.sgi.com/library/

Conventions

In command syntax descriptions and examples, square brackets ([ ]) or angle brackets
(< >) surrounding an argument indicate an optional argument. Variable parameters are
in *italics*. Replace these variables with the appropriate string or value.

Command-line flags and switches are in **boldface regular type**; these are preceded with
dashes or with a plus or minus, for example, **-e**.

Commands, IRIX filenames, and document titles are in *italics*.

Messages and prompts that appear onscreen are shown in `fixed-width type`. Entries
that are to be typed exactly as shown are in **`boldface fixed-width type`**.

Names of buttons in the RAID GUI are shown in *italics*. Menu selections are indicated
with this convention: Array > Write Cache State > Disable means choose Write Cache
State in the Array menu, and then choose Disable in the popup menu that appears.

xvii

Fibre Channel RAID Graphical User Interface Features

The Silicon Graphics Fibre Channel administrative software provides a graphical user interface, *ssmgui*, that lets you configure and manage Silicon Graphics Fibre Channel RAID disk-array storage systems from a physical component viewpoint. Using *ssmgui*, you can group a Fibre Channel RAID storage system's physical disks into logical units (LUNs[1]), monitor the status on the physical disks and other components that make up a storage system, and perform other administrative tasks.

Note: You can also communicate with RAID storage with a command-line interface, explained in Chapter 6, "The Fibre Channel RAID Command-Line Interface." For communication with non-RAID (JBOD) storage, use a different command-line interface, which is explained in Chapter 7, "The Fibre Channel Non-RAID Command-Line Interface." No GUI is provided for communicating with non-RAID storage.

This chapter contains these sections:

- "ssmgui Architecture" on page 2
- "Starting a RAID GUI Session" on page 3
- "Exiting a RAID GUI Session" on page 8
- "Array Configuration and Management" on page 8
- "Data Availability and Performance" on page 9
- "RAID Types" on page 12
- "RAID Hot Spare" on page 20

---

[1] The physical disk unit number is also known as the logical unit number, or LUN. The LUN is a logical concept, but is recognized as a physical disk unit by the operating system; hence the seemingly contradictory names.

ssmgui Architecture

The *ssmgui* graphical user interface program and the *ssmcli* command-line interface communicate with a dedicated agent, *ssmagent*, on the same server or on other servers on the network. The management station and the remote agents communicate with each other over a TCP/IP network. The *ssmagent* communicates with the licensed internal code running in an array's storage processors (SPs). Figure 1-1 diagrams Fibre Channel administrative software architecture with a management station that is also an array server.
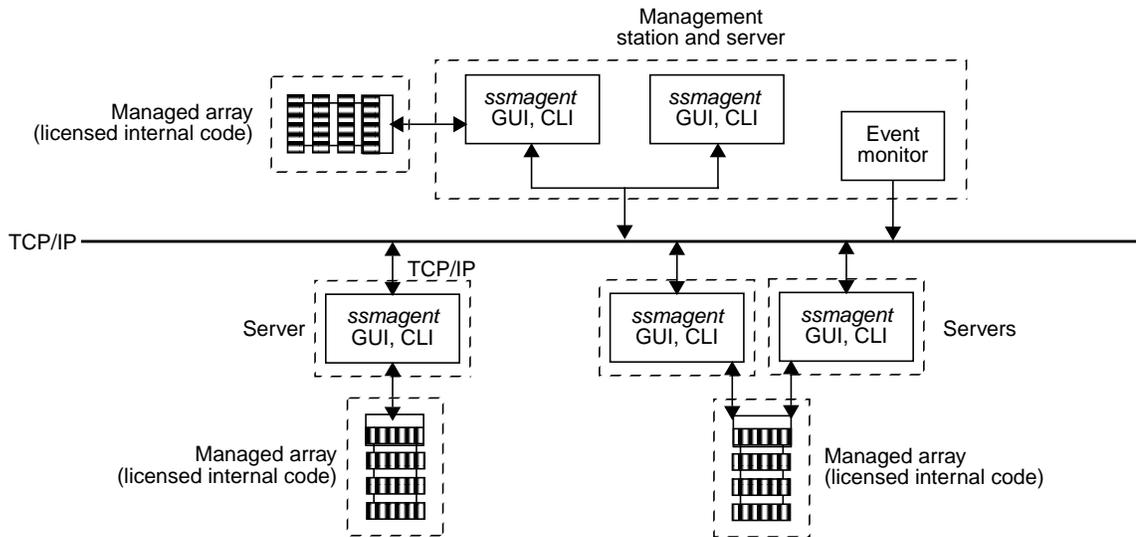
Figure 1-1   ssmgui Architecture: Management Station That Is Also an Array Server

Note: In the chapters describing the RAID graphical user interface, *ssmgui* and RAID GUI are synonymous, although *ssmgui* generally denotes the program as a whole and RAID GUI denotes the graphical user interface itself.

The licensed internal code includes background verify and sniff processes and reporting. This array-resident software process runs continuously in the background, scans all data, and repairs soft error s and data inconsistencies before these anomalies can become compounded and unrecoverable. This software is transparent to user applications and affects performance by less than 1%. All corrective actions are logged in the SP event log (described in "Displaying an SP Event Log" on page 113 in Chapter 4).

Starting a RAID GUI Session

This section contains the following topics:

- "Checking ssmagent Installation" on page 3
- "ssmagent Users and Sessions" on page 3
- "Configuring the Path to ssm Executables in .cshrc" on page 4
- "Starting the Fibre Channel RAID Agent ssmagent" on page 4
- "Starting the RAID GUI" on page 5
- "Storage System Manager Window" on page 6

Checking ssmagent Installation

To confirm that the Fibre Channel RAID *ssmagent* is installed on the host system, enter the following at the command line:

```
chkconfig -f ssm on
```

If the software is not available on the host, install it using the information in the release notes and the CD included with the Fibre Channel RAID option.

ssmagent Users and Sessions

Any user can run a RAID GUI session from any management station on which it is installed, and can use it to monitor the storage system. Only authorized users can use this program to configure or reconfigure a storage system. A user is authorized if the file */etc/config/ssmagent.config* on the server contains an entry for the user. For information, consult the man page (reference page) ssmagent(7M).

ssmagent allows more than one RAID GUI session to access the same storage system at the same time. Make sure that two authorized users do not use the RAID GUI to configure or reconfigure the same array enclosure at the same time. To determine whether another user is using *ssmgui*, enter

```
ps -ef | grep ssmgui
```

3

The output does not inform you if the person is trying to manage or configure a specific array.

Configuring the Path to ssm Executables in .cshrc

The following steps configure the host system's *.cshrc* file to execute *ssmgui* commands from any directory on the host. If this path is not set, you must be in the */usr/ssm/lib* directory and enter the complete path */usr/ssm/bin/ssmgui* each time you want to launch *ssmgui*.

Use the following information to set the proper path on the host system:

1. As superuser, use your favorite line editor to open the *.cshrc* file.

2. Enter the following in the *.cshrc* file:

    ```
    set path = ($path /usr/ssm/bin /usr/ssm/lib)
    ```

3. Write and quit the *.cshrc* file.

4. Enter the following command sequence to reinitialize the *.cshrc* file:

    ```
    source .cshrc
    ```

Starting the Fibre Channel RAID Agent ssmagent

Follow these steps:

1. Check for the presence of the agent, as explained in "Checking ssmagent Installation" on page 3.

2. Make sure your username is entered in */etc/config/ssmagent.config*, as explained in "ssmagent Users and Sessions" on page 3.

3. If desired, configure the host system's *.cshrc* file for convenience in using *ssmgui*, as explained in "Configuring the Path to ssm Executables in .cshrc" on page 4.

4. Make sure that the host system on which the agent is to run has the correct date and time; enter

    ```
    date
    ```

    If the date and time are not correct, reset them; see this command's man page, date(1M).

5. Start *ssmagent* by entering

    /etc/init.d/ssm start

Depending on the number of RAID arrays you administer, it takes the agent from 30 seconds to several minutes to finish launching arrays and become ready to administer them. You can check whether the agent has started up by looking in */var/adm/SYSLOG* for this string:

```
Agent has started up
```

This message indicates that the agent has made connection to all necessary drivers.

Note: To stop the Fibre Channel RAID agent, enter

/etc/init.d/ssm stop

Starting the RAID GUI

The agent *ssmagent* must be operating before you start the RAID GUI; see "Starting the Fibre Channel RAID Agent ssmagent" on page 4.

To start the RAID GUI, enter

ssmgui &

If you have not configured a path to *ssm* executables as explained in "Configuring the Path to ssm Executables in .cshrc" on page 4, use

/usr/ssm/bin/ssmgui &

After displaying a preliminary window, the RAID GUI looks first in your home directory for a *ssmhosts* file that contains the hostnames of servers with arrays to manage. If such a file exists, *ssmgui* tries to extract hostnames from this file. If it succeeds, it opens the Storage System Manager window, as shown in Figure 1-2.

If *ssmgui* cannot extract the hostnames, it opens the Host Administration dialog, in which you can enter the hostnames of servers whose arrays you want to manage using the RAID GUI. This process is explained in "Using the Host Administration Window to Add Servers" on page 22 in Chapter 2.

Storage System Manager Window

The main RAID GUI window is the Storage System Manager window. Figure 1-2 points out the main features of this window.
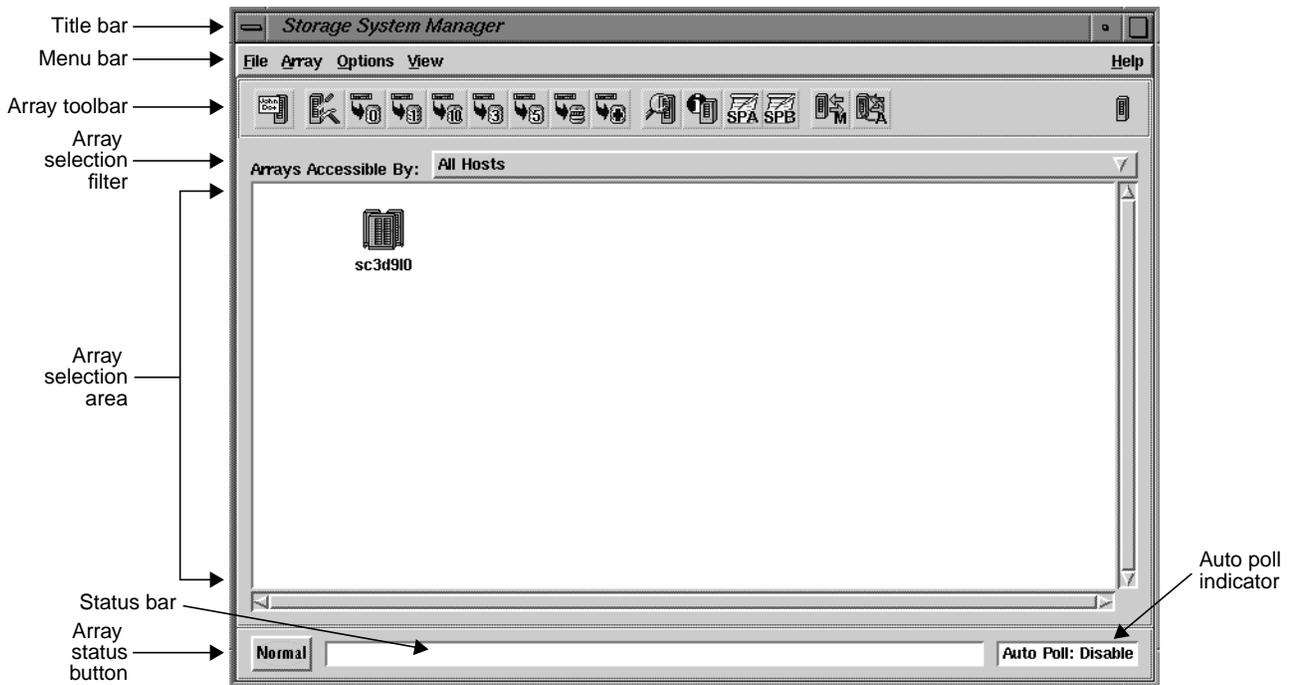
Figure 1-2   Storage System Manager Window

From this window, you can specify arrays to manage, bind standard LUNs, display other information windows on arrays, and display the SP event log. These functions are explained in subsequent chapters of this guide.

In the array selection area, the software displays an icon for each array that it finds connected to each server whose hostname appears in the *ssmhosts* file. It creates only one icon for each array, regardless of how many servers are connected to the array.

Main features of this window are as follows:

- The array selection area shows icons for the arrays specified by the array selection filter. "Interpreting Array Icons" on page 27 in Chapter 2 explains how to interpret icons in this area.

- The array selection filter (Arrays Accessible By field) near the top of the Storage System Manager window allows you to show arrays for all hosts, or specific arrays. Its use is explained in "Using the Array Selection Filter (Managing and Unmanaging Arrays)" on page 25 in Chapter 2.

- The auto poll indicator field at the lower right indicates whether automatic polling is disabled (default) or enabled. Polling is explained in "Using ssmagent Polling" on page 84 in Chapter 4.

- The array status button at the lower left indicates whether there is any fault in the systems being managed. Chapter 4, "Monitoring Arrays and Displaying System Statistics," and Chapter 5, "Identifying and Correcting Failures," explain how to display system statistics and how to find and correct error conditions.

The Storage System Manager window has many standard GUI features, such as

- status bar, which displays GUI and system messages

- button descriptions that you display by positioning the cursor on a button for a few seconds without clicking the button (the description is also displayed in the status bar)

- Help menu for displaying online help and information about *ssmgui*

  Note: To print information from a Help browser window, create a document in an ASCII text editor (such as *vi*), copy the information, and print the document.

- View menu for hiding or displaying toolbars, and for other viewing options

- horizontal and vertical scroll bars for various windows and menus

7

Exiting a RAID GUI Session

In the File menu of the Storage System manager window (and certain other windows), click *Exit* to exit the *ssmgui* session. In the confirmation window that opens, click *OK*. All *ssmgui* windows close.

Note: When you exit, all changes you set in all windows are saved to *$HOME/.vgalaxy.1.vr.*

Array Configuration and Management

Note the following terms:

- A storage system managed by the administrative software is a *managed array.*

- A host with a managed array is an *array server* or a *server.*

- The host running the administrative software is a *management station* (this host can also be a server with managed arrays.

To configure an array using the RAID GUI, you follow these basic steps:

1. Determine the arrays you want to manage.

2. Make sure the RAID GUI agent *ssmagent* is running on all servers connected to arrays you want to manage, as explained in "Checking ssmagent Installation" on page 3 in this chapter.

3. Start the RAID GUI (*ssmgui*), as explained in "Starting a RAID GUI Session" on page 3 in this chapter.

4. In the RAID GUI, set up array memory and create LUNs, as explained in Chapter 2, "Using the RAID GUI to Configure Arrays."

5. Make the LUNs available to the server's operating system, as explained in "Making LUNs Available to the Server Operating System" on page 60 in Chapter 2.

To manage the array, you follow these basic steps:

1. Monitor the health of the array, as explained in Chapter 4, "Monitoring Arrays and Displaying System Statistics."

2. Display system status and statistics, as explained in Chapter 4, "Monitoring Arrays and Displaying System Statistics."

3. Reconfigure the array, if necessary, as explained in Chapter 3, "Reconfiguring and Fine-Tuning."

4. Deal with failed components if necessary, as explained in Chapter 5, "Identifying and Correcting Failures."

Data Availability and Performance

The RAID storage system hardware implements data availability and performance enhancements. This section discusses the following topics:

- "Data Redundancy" on page 9
- "Enhanced Performance: Disk Striping" on page 10
- "Enhanced Performance: Storage System Caching" on page 10
- "Data Reconstruction and Rebuilding After Disk Module Failure" on page 11

For information on RAID levels supported for IRIS FailSafe, ask your Silicon Graphics Sales Engineer.

Data Redundancy

RAID technology provides redundant disk resources in disk-array and disk-mirror configurations that make the storage system more highly available. Data redundancy varies for the different RAID types (levels) supported by Silicon Graphics Fibre Channel RAID storage systems: RAID 0, RAID 1, RAID 1/0, RAID 3, and RAID 5.

A RAID 3 and RAID 5 group maintain parity data that lets the disk group survive a disk module failure without losing data. In addition, the group can survive a single fibre loop failure.

A RAID 1 mirrored pair, or a RAID 1/0 group, which uses RAID 1 technology, duplicates data on two groups of disk modules. If one disk module fails, the other module provides continuing access to stored information. Similarly, a RAID 1 mirrored pair or RAID 1/0 group can survive a single fibre loop failure.

Enhanced Performance: Disk Striping

In disk striping, the SP lays out data records, usually large data records or a number of small records for the same application, across multiple disks. For most applications, these disks can be written to or read from simultaneously and independently. Because multiple sets of read/write heads work on the same task at once, disk striping can enhance performance.

The amount of information read from or written to each module makes up the stripe element size (for example, 128 sectors). The stripe size is the number of data disks in a group multiplied by the stripe element size. For example, assume a stripe element size of 128 sectors (the default). If the RAID 5 group has five disks (four data disks and one parity disk), multiply by 4 the stripe element size of 128 to yield a stripe size of 512 sectors.

Enhanced Performance: Storage System Caching

Caching is available for Fibre Channel RAID storage systems that have two SPs, each with at least 16 MB of read cache memory, 16 MB of write cache memory, two power supplies in the Fibre Channel RAID enclosure (DPE), and a fully charged standby power supply (SPS), and disk modules in slots 00 through 08. With storage system caching enabled, each SP temporarily stores requested information in its memory.

Caching can save time in two ways:

- For a read request, if data is sought after the request is already in the read cache, the storage system avoids accessing the disk group to retrieve the data.

- For a write request, if the information in the write cache is modified by the request and thus must be written to disk, the SP can keep the modified data in the cache and write it back to disk at the most expedient time instead of immediately. Write caching, in particular, can enhance storage system performance by reducing write time response.

To ensure data integrity, each SP maintains a mirror image of the other SP's write caches If one SP fails, the data in its write cache is available from the other SP.

As explained in Chapter 2, "Using the RAID GUI to Configure Arrays," and Chapter 3, "Reconfiguring and Fine-Tuning," you can enable storage system caching and specify basic cache parameters, and enable or disable read and write caches for individual disk units.

Note: The SPS must be fully functional in the Fibre Channel RAID chassis for systems using write cache to ensure that data is committed to disk in the event of an AC input power failure.

Data Reconstruction and Rebuilding After Disk Module Failure

All RAID types except RAID 0 provide data redundancy: the storage system reads and writes data from and to more than one disk at a time. Also, the system software writes parity information that lets the array continue operating if a disk module fails. When a disk module in one of these RAID types fails, the data is still available because the SP can *reconstruct* it from the surviving disk(s) in the array.

Data *rebuilding* occurs under these conditions:

- A hot spare (dedicated replacement disk module) is available.

- The failed disk module is replaced with a new disk module.

If a disk module has been configured (bound) as a hot spare, it is available as a replacement for a failed disk module. (See "RAID Hot Spare" on page 20.) When a disk module in any RAID type except RAID 0 fails, the SP automatically writes to the hot spare and rebuilds the group using the information stored on the surviving disks. Performance is degraded while the SP rebuilds the data and parity on the new module. However, the storage system continues to function, giving users access to all data, including data stored on the failed module.

Similarly, when a new disk module is inserted to replace a failed one, the SP automatically writes to it and rebuilds the group using the information stored on the surviving disks. As for the hot spare, performance is degraded during rebuilding, but data is accessible.

The length of the rebuild period, during which the SP recreates the second image after a failure, can be specified when RAID types are set and disks are bound into RAID units. These processes are explained in Chapter 2, "Using the RAID GUI to Configure Arrays.".

RAID Types

The Fibre Channel RAID system supports these levels of RAID:

- RAID 0 group: nonredundant array
- RAID 1 group: mirrored pair
- RAID 1/0 group: mirrored RAID 0 group
- RAID 3 group: parallel access array
- RAID 5 group: individual access array
- individual disk unit

Caution: Use only Fibre Channel RAID disk modules to replace failed disk modules. These disk modules contain proprietary firmware that the storage system requires for correct functioning. Using any other disks, including those from other Silicon Graphics systems, can cause failure of the storage system. Swapping disk modules within a Fibre Channel RAID storage system is also not recommended, particularly disk modules in slots 00, 01, and 02, which contain the licensed internal code. or slots 00 through 08, which serve as the storage system cache vault.

Chapter 2 provides detailed instructions on configuring all RAID types.

RAID 0 Group: Nonredundant Array

Three to sixteen disk modules can be bound as a RAID 0 group. A RAID 0 group uses striping; see "Enhanced Performance: Disk Striping," earlier in this chapter. You might choose a RAID 0 group configuration when fast access is more important than high availability. You can software-mirror the RAID 0 group to provide high availability; see the section on using *xlv make* to create volume objects in *Getting Started With XFS Filesystems*.

Caution: The hardware does not maintain parity information on any disk module for RAID 0 the way it does for other RAID types. Failure of a disk module in this RAID type results in loss of data.

RAID 1/0 Group: Mirrored RAID 0 Group

A RAID 1/0 configuration mirrors a RAID 0 group, creating a primary RAID 0 image and a secondary RAID 0 image for user data. This arrangement consists of four (minimum), six, eight, ten, twelve, fourteen, or sixteen disk modules. These disk modules make up two mirror images, with each image including two to eight disk modules. A RAID 1/0 group uses striping and combines the speed advantage of RAID 0 with the redundancy advantage of mirroring.

Figure 1-3 illustrates the distribution of user data with the default stripe element size of 128 sectors (65,536 bytes) in a six-module RAID 1/0 group. Notice that the disk block addresses in the stripe proceed sequentially from the first mirrored disk modules to the second mirrored disk modules, to the third mirrored disk modules, then from the first mirrored disk modules, and so on.

A RAID 1/0 group can survive the failure of multiple disk modules, providing that one disk module in each image pair survives. For example, the RAID 1/0 group shown in Figure 1-3 has three disk modules in each image of the pair.
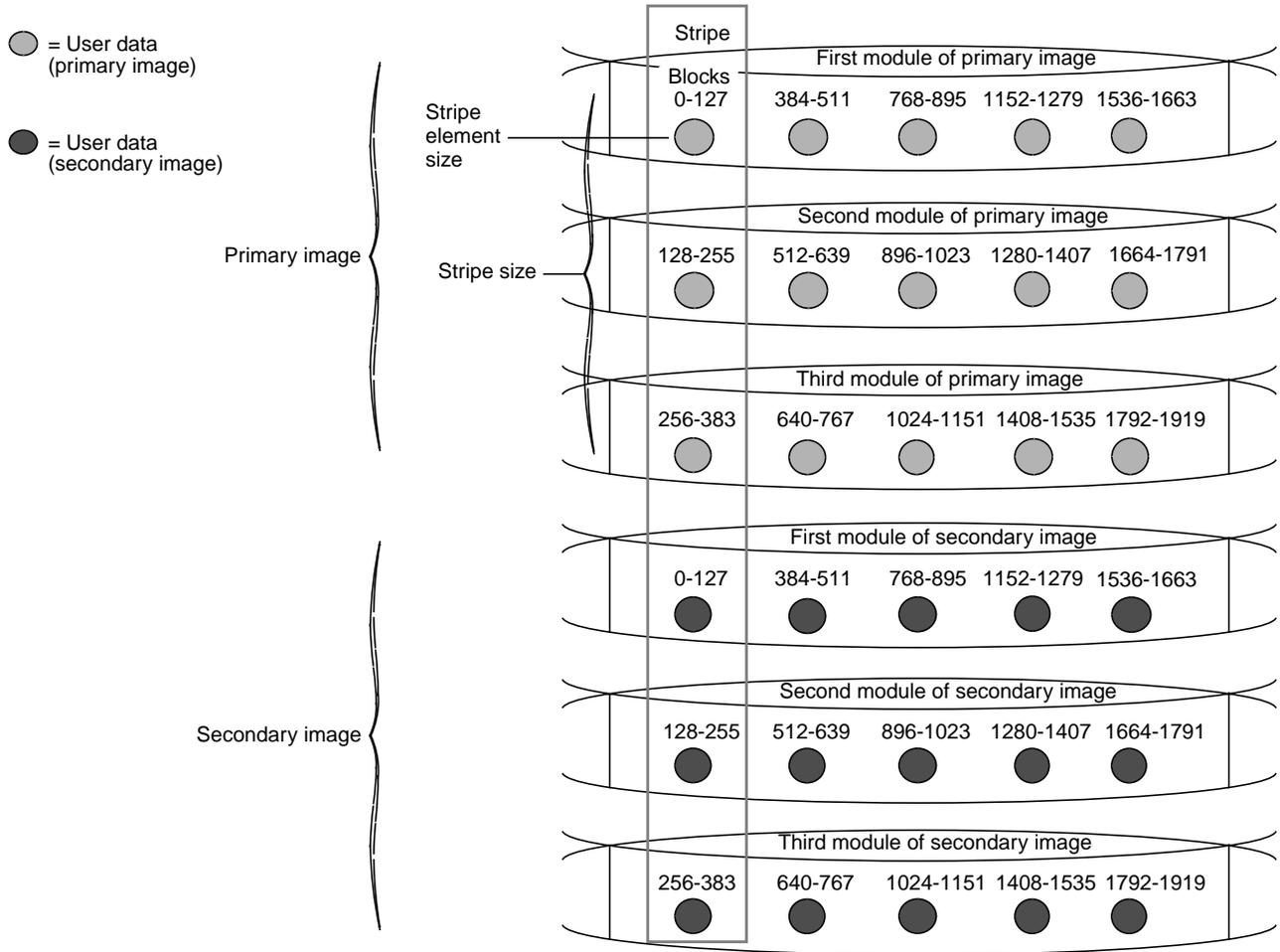
Figure 1-3   Distribution of User Data in a RAID 1/0 Group

RAID 1: Mirrored Pair

In the RAID 1 configuration, two disk modules can be bound as a mirrored pair. In this disk configuration, the SP duplicates (mirrors) the data records and stores them separately on each disk module in the pair. The disks in a RAID 1 pair cannot be split into individual units (as can a software mirror composed of two individual disk units).

Features of this RAID type include

- fault tolerance

- automatic mirroring: no commands are required to initiate it

- physical separation of images

- faster write operation than RAID 5

With a RAID 1 mirrored pair, the storage system writes the same data to both disk modules in the mirror, as shown in Figure 1-4.
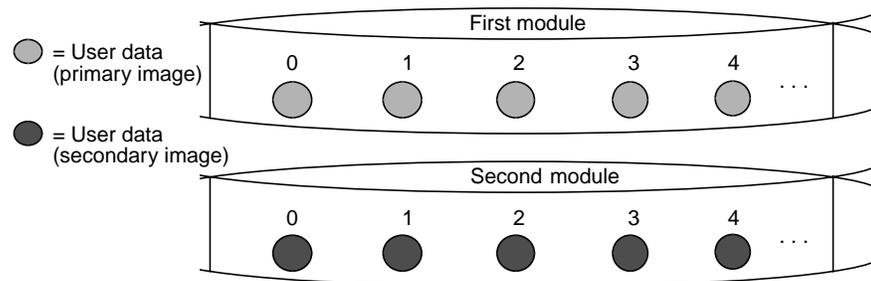
Figure 1-4   RAID 1 Mirrored Pair (Hardware-Mirrored Pair)

RAID 3: Parallel Access Array

A RAID 3 configuration always consists of five (four data, one parity: 4 + 1) or nine (eight data, one parity: 8 + 1) disk modules bound as a RAID 3 group. In a RAID 3 group, the hardware always reads from or writes to all its disk modules. A RAID 3 group uses disk striping; see "Enhanced Performance: Disk Striping," earlier in this chapter for an explanation of this feature. RAID 3 striping has a fixed stripe size of one sector.

The Fibre Channel RAID storage system writes parity information that lets the group continue operating if one of the disk modules fails. When you replace the failed module, the SP can rebuild the group using the information stored on the working disks. Performance is degraded while the SP rebuilds the data or parity on the new module. However, the storage system continues to function and gives users access to all data, including data that had been stored on the failed module.

RAID 3 differs from RAID 5 in several important ways:

- In a RAID 3 group, the hardware processes disk requests serially, whereas in a RAID 5 group the hardware can interleave disk requests.

- In a RAID 3 group, the parity information is stored on one disk module; in a RAID 5 group, it is stored on all disks.

A RAID 3 group works well for single-task applications that use I/Os of one or more 2-KB blocks, aligned to start at disk addresses that are multiples of 2 KB from the beginning of the logical disk.

Figure 1-5 illustrates user and parity data with a data block size of 2 KB within a RAID 3 group. Notice that the byte addresses proceed from the first module to the second, third, and fourth, then to the first, and so on.
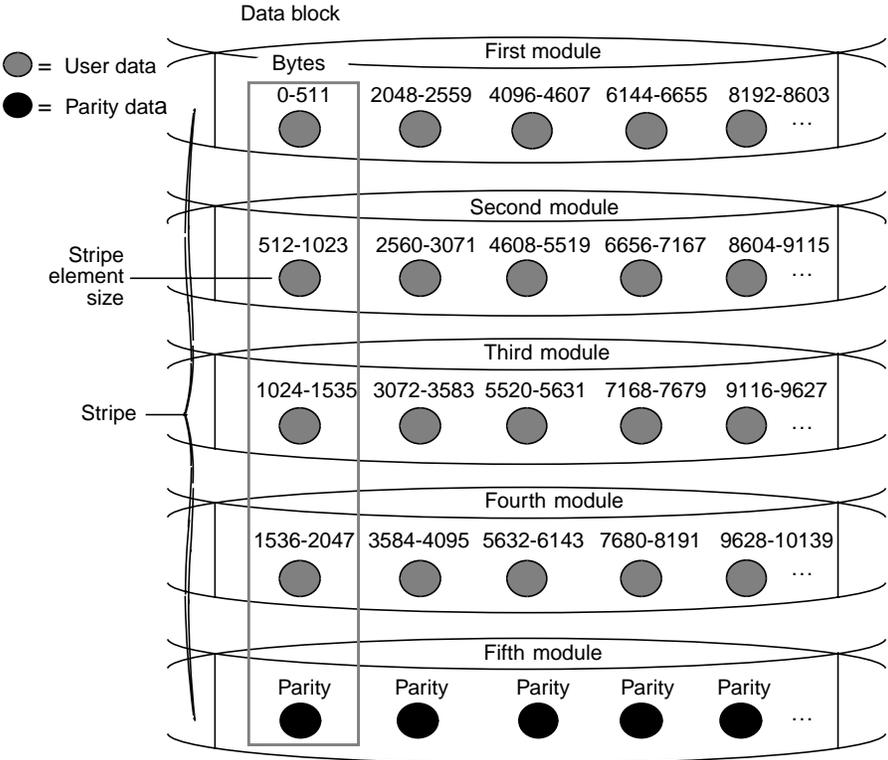
Figure 1-5   Distribution of User and Parity Data in a RAID 3 Group

The storage system performs more steps writing data to a RAID 3 group than to all the disks in a RAID 1 mirrored pair or a RAID 0 group, or to an individual disk unit. For each correctly aligned 2 KB write operation to a RAID 3 group, the storage system performs the following steps:

1. Calculates the parity data.
2. Writes the new user and parity data.

If the write is not a multiple of 2 KB or the starting disk address of the I/O does not begin at an even 2 KB boundary from the beginning of the logical disk, the storage system performs the following steps:

1. Reads data from the sectors being written and parity data for those sectors.

2. Recalculates the parity data.

3. Writes the new user and parity data.

RAID 5: Individual Access Array

This configuration usually consists of five disk modules (but can have three to sixteen) bound as a RAID 5 group. An array of five disk modules (or fewer) provides the greatest level of data redundancy. A RAID 5 group maintains parity data that lets the disk group survive a disk module failure without losing data.

With RAID 5 technology, the hardware writes parity information to each module in the array. If a module fails, the SP can reconstruct all user data from the user data and parity information on the other disk modules. After you replace a failed disk module, the SP automatically rebuilds the disk array using the information stored on the remaining modules. The rebuilt disk array contains a replica of the information it would have contained had the disk module never failed.

A RAID 5 group uses disk striping; see "Enhanced Performance: Disk Striping," earlier in this chapter for an explanation of this feature. Figure 1-6 illustrates user and parity data with the default stripe element size of 128 sectors (65,536 bytes) in a five-module RAID 5 group. The stripe size comprises all stripe elements. Notice that the disk block addresses in the stripe proceed sequentially from the first module to the second, third, fourth, and fifth, then back to the first, and so on.

For each write operation to a RAID 5 group, the RAID storage system must perform the following steps:

1. Read data from the sectors being written and parity data for those sectors.

2. Recalculate the parity data.

3. Write the new user and parity data.
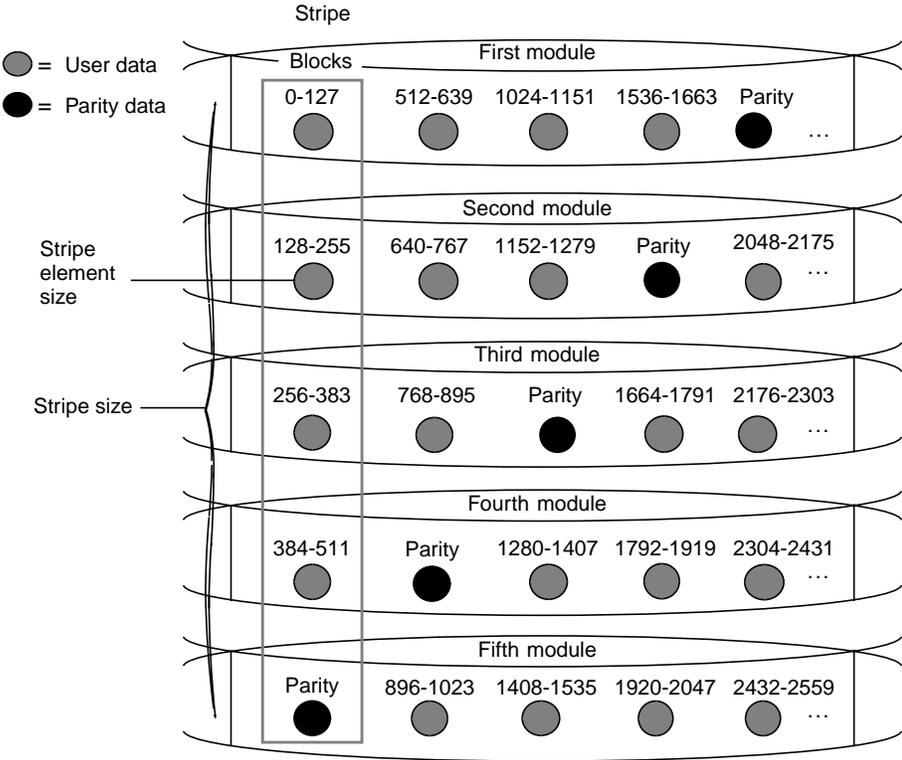
Figure 1-6   Distribution of User and Parity Data in a RAID 5 Group

Individual Disk Unit

An individual disk unit is a disk module bound to be independent of any other disk module in the cabinet. An individual unit has no inherent high availability, but you can make it highly available by software-mirroring it with another individual unit.

RAID Hot Spare

A hot spare is a dedicated replacement disk unit on which users cannot store information. The capacity of a disk module that you bind as a hot spare must be at least as great as the capacity of the largest disk module it might replace.

Note: The hot spare is not available for RAID 0, because this RAID type does not provide data redundancy. RAID 3 LUNs use only one hot spare, even though additional ones are available.

If any disk module in a RAID 5, RAID 3, or RAID 1/0 group or in a RAID 1 mirrored pair fails, the SP automatically begins rebuilding the failed disk module's structure on the hot spare. When the SP finishes rebuilding, the disk group functions as usual, using the hot spare instead of the failed disk. When you replace the failed disk, the SP starts copying the data from the former hot spare onto the replacement disk. When the copy is done, the disk group consists of disk modules in the original slots, and the SP automatically frees the hot spare to serve as a hot spare again.

Note: The SP finishes rebuilding the disk module onto the hot spare before it begins copying data to the new installed disk, even if you replace the failed disk during the rebuild process.

A hot spare is most useful when you need the highest data availability. It eliminates the time and effort needed for someone to notice that a module has failed, find a suitable replacement module, and insert it.

You can have one or more hot spares per storage system. Any module in the storage system can be configured as a hot spare except for disk modules in slots 00 through 08, which serve other purposes.

For example, assume that the disk modules in slots 10-14 are a RAID 5 group, those in slots 15 and 16 are a RAID 1 mirrored pair, and the module in 17 is a hot spare. If module 13 fails, the SP immediately begins rebuilding the RAID 5 group using the hot spare. When it finishes, the RAID 5 group consists of disk modules 10, 11, 12, 17, and 14.

When you replace the failed module in 13, the SP starts copying the structure on 17 to 13. When it finishes, the RAID 5 group once again consists of modules 10-14 and the hot spare becomes available for use if any other module fails. A similar sequence would occur if, for example, module 15 in the mirrored pair failed.

Using the RAID GUI to Configure Arrays

This chapter explains how to use the RAID GUI to configure an array. It contains the following sections:

- "Using the Host Administration Window to Add Servers" on page 22
- "Removing Servers" on page 24
- "Selecting an Array to Configure" on page 24
- "Setting Up Array Memory for Caching or RAID 3 LUNs" on page 32
- "Planning the Bind" on page 44
- "Binding Disk Modules" on page 49
- "Making LUNs Available to the Server Operating System" on page 60
- "Enabling Command-Tagged Queuing" on page 60

Using the Host Administration Window to Add Servers

When you start a RAID GUI session and no server hostnames are in the *ssmhosts* file, the list of host identifiers in the window is empty. The Host Administration window appears, as shown in Figure 2-1.
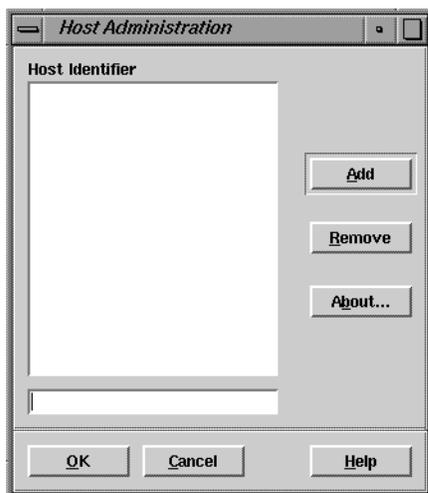
Figure 2-1   Host Administration Window With No Server Hostnames

(You can open the Host Administration window during an *ssmgui* session from the Storage System Manager window by choosing Select Hosts in the Storage Management window's File menu. In this situation, the list contains the hostnames of the servers with currently managed arrays. You can add or remove server hostnames from this list.)

To add a server to the list, follow these steps:

1. In the Host Identifier field, type the server hostname.

2. Click *Add*.

3. Continue to add server hostnames as desired.

    Servers on this list are called *managed servers*.

    Note: If you include the same server under two names (for example, as *machinename* and as *machinename.domain.com*), the GUI opens each as a separate host, which thus takes twice as long as necessary.

4. When the list contains all the server hostnames you want, click *OK*. A window opens asking if you want to save the new host list in the *ssmhosts* file in the home directory.

5. Register your choice:

    •   Click *Yes* if the list contains only servers with storage systems that you want to manage in most future *ssmgui* sessions.

    •   Click *No* if you added servers that you want to manage in the current session only.

    ssmgui creates an icon for each array that it finds connected to each server whose hostname you added to the list (unless the icon already exists). *ssmgui* displays these icons in the array selection area of the Storage System Manager window (see Figure 2-3). If an array is connected to two servers, only one icon appears.

Figure 2-2 shows a Host Administration window with a server hostname.

Figure 2-2　　Host Administration Window With Server Hostname

If *ssmgui* cannot communicate with a server you added to the list, it displays an error message.

Removing Servers

To remove a server hostname, select it in the Host Administration window and click *Remove*. The RAID GUI asks for confirmation.

Save the list as explained in "Using the Host Administration Window to Add Servers" on page 22. *ssmgui* removes the icon for that array unless the array is connected to another server on the list. It also removes the hostname from the Host Administration window.

Selecting an Array to Configure

This section explains how to use RAID GUI features to select an array to configure:

- "Using the Array Selection Filter (Managing and Unmanaging Arrays)" on page 25
- "Interpreting Array Icons" on page 27
- "Using Storage System Manager Toolbar Buttons" on page 29
- "Selecting an Array" on page 30
- "Using the Array Configuration Window to Set Up the Array" on page 31

Using the Array Selection Filter (Managing and Unmanaging Arrays)

Use the array selection filter (Arrays Accessible By field) near the top of the Storage System Manager window to display only those arrays on a server. Figure 2-3 points out this portion of the Storage System Manager window, and other features.
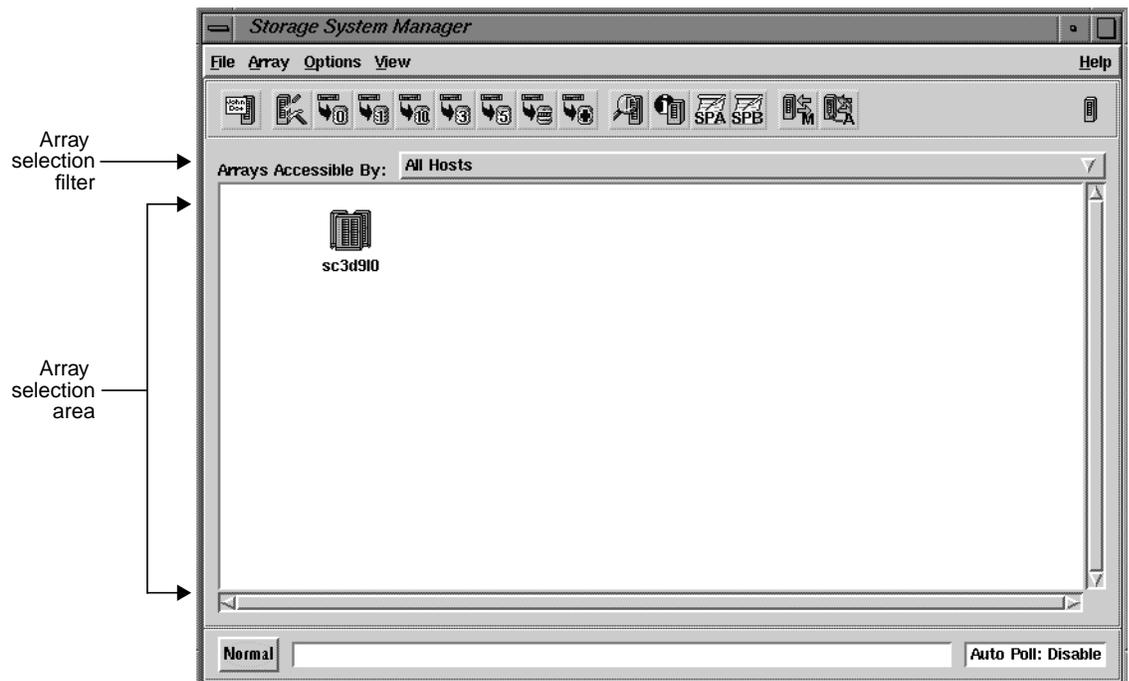
Figure 2-3   Storage System Manager Window: Array Selection Filter and Array Selection Area

Other ways to specify arrays are as follows:

- To display the icon for a specific array for all managed servers, follow these steps:

  1. In the array selection area, select the icon for each array that you do not want to manage.

  2. From the Array menu, choose Unmanage. *ssmgui* removes the icons for the selected arrays from the array selection area.

  Note: This display does not affect the list of managed servers and their arrays as listed in the Host Administration menu.

- To display icons for all arrays for one managed server:

    – If the Arrays Accessible By field (see Figure 2-3) shows the hostname of the server whose arrays you want represented in the array selection area, do nothing.

    – If this field does not show the server hostname, pull down the selection list and select the desired server hostname. In the array selection area, *ssmgui* shows icons for arrays that it finds connected to this server.

- To display icons for specific arrays for one managed server only, follow these steps:

    1. If necessary, select the desired server hostname in the Arrays Accessible By field (see Figure 2-3). In the array selection area, *ssmgui* shows icons for arrays that it finds connected to this server.

    2. In the array selection area, select the icon for each array that you do not want to manage.

    3. From the Array menu, choose Unmanage. *ssmgui* removes the icons for the selected arrays from the array selection area.

To convert unmanaged arrays to managed arrays, follow these steps:

1. Choose Manage Arrays from the File menu. The Manage Arrays window opens, as shown in Figure 2-4.
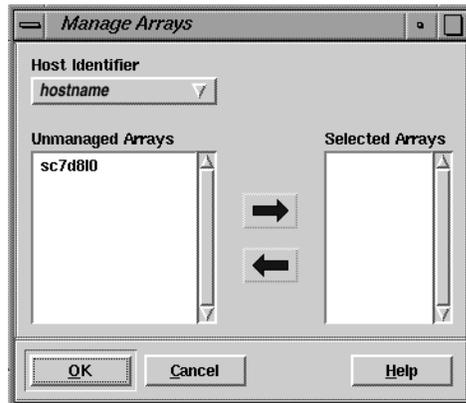
Figure 2-4   Manage Arrays Window

2. If the Host Identifier field does not contain the name of the server with the unmanaged arrays that you want to start managing, pull down its selection list and select the server's hostname (or type it in the Host Identifier field). The names of the server's unmanaged arrays appear in the Unmanaged Arrays list on the left, as shown in Figure 2-4.

3. For each array you want to start managing:

    ■   Select the array name in the Unmanaged Arrays list.

    ■   Click the right arrow button. The array's name moves to the Selected Arrays list, on the right.

4. When the Selected Arrays list contains the names of only those unmanaged arrays that you want to start managing, click *OK*. An array icon for each selected array appears in the array selection area of the Storage System Manager window.

Interpreting Array Icons

The array selection area (see Figure 2-3) contains an icon for each array specified by the array selection filter. The icon for an array consists of the array name and a graphic.

Array Name

The default name for an array is determined by the host and device entries in the *ssmagent.config* file on the server connected to the array. You can change the array's default name by choosing Name in the Array menu or by using the name array button at the left end of the toolbar. Changing the name does not affect the configuration file.

Enclosure Health and Accessibility

The color of an icon for an array indicates its health:

•   Gray indicates that no failure is detected.

•   Amber indicates a fault in some part of the array.

The graphic for an array icon indicates the status of the array:

- Accessible: *ssmgui* can communicate with the array.

- Inaccessible: *ssmgui* cannot communicate with the array because its name is wrong in the *ssmagent* configuration file on its server, or because the *ssmagent* on its server was started by a user who was not logged in as root.

  The icon for an inaccessible array has a shield over it:

- Unsupported: The device entry in the *ssmagent* configuration file on its server is for a device that *ssmgui* does not support (for example, an internal disk on the server).

  The icon for an unsupported array has a question mark over it:

- Accessible, but faulted: *ssmgui* can communicate with the array, but one or more components is faulty. Chapter 5, "Identifying and Correcting Failures," explains how to investigate and correct faulted components.

  The icon for a faulted array is amber and contains an F:

Note: The external Fibre Channel Hub is transparent to *ssmgui* and the command-line interfaces.

Icons Connected to More Than One Server

Only one icon for each array appears in the array selection area even it if is connected to more than one server.

To determine if an array is connected to more than one server, use the Array Information button or choose Information from the Array menu. This process is explained in "Using the Array Information Window" on page 92 in Chapter 4.

Using Storage System Manager Toolbar Buttons

You can use the toolbar in the Storage System Manager window for many steps explained in this guide. Figure 2-5 shows toolbar buttons and their functions.
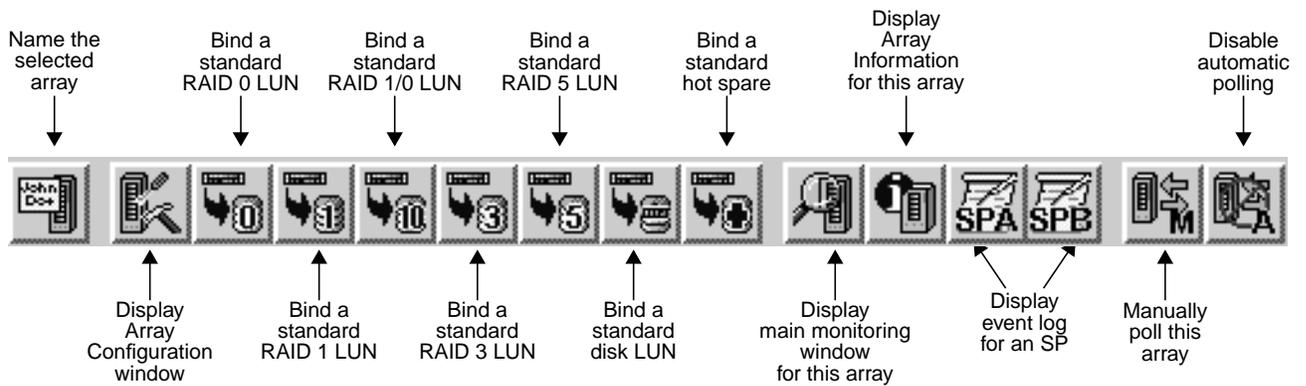
Figure 2-5   Storage System Manager Toolbar

These buttons duplicate functions available in the Storage System Manager window's Array menu.

Tip: When you select an array, click the right mouse button in the array selection area to bring up a menu with the same functions as the toolbar buttons.

Selecting an Array

In the array selection area, click the icon for the array. A black box appears around the icon, as shown in Figure 2-6.
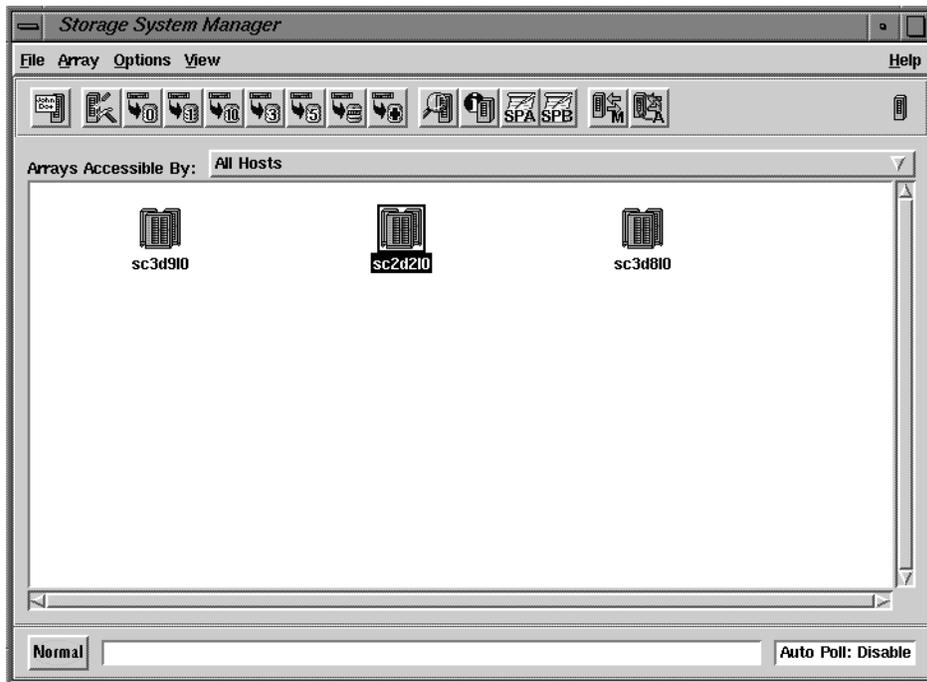
Figure 2-6   Selecting an Array

To display the Array menu when you select an array, use the right mouse button and hold the button down. The black box appears around the icon and the Array menu drops down to the right of the icon.

To select more than one array at a time:

• Click the icon for one array, and hold down the Shift key and click the icon for each of the other arrays you want to select.

or

• Drag the cursor to create a box around the icons for the arrays you want to select.

The black box appears around each icon you select.

Using the Array Configuration Window to Set Up the Array

In the Storage System Manager window, choose Configure from the Array menu. The
Array Configuration window appears; Figure 2-7 shows important features.
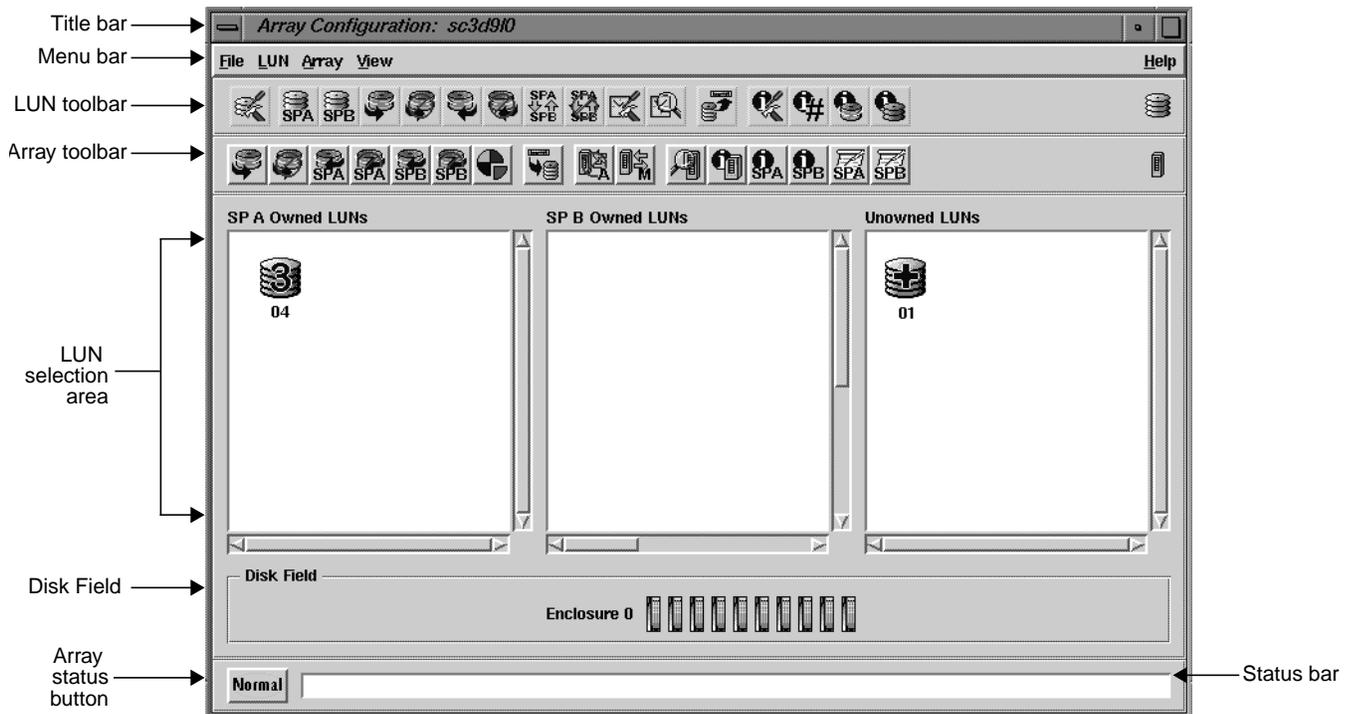
Figure 2-7   Array Configuration Window

The title bar identifies the array that is represented in the window.

Using Array Toolbar Buttons

The Array Configuration window has two toolbars, the array toolbar and the LUN toolbar. Figure 2-8 shows array toolbar buttons and their functions. (LUN toolbar buttons are shown in Figure 3-1 in Chapter 3.)
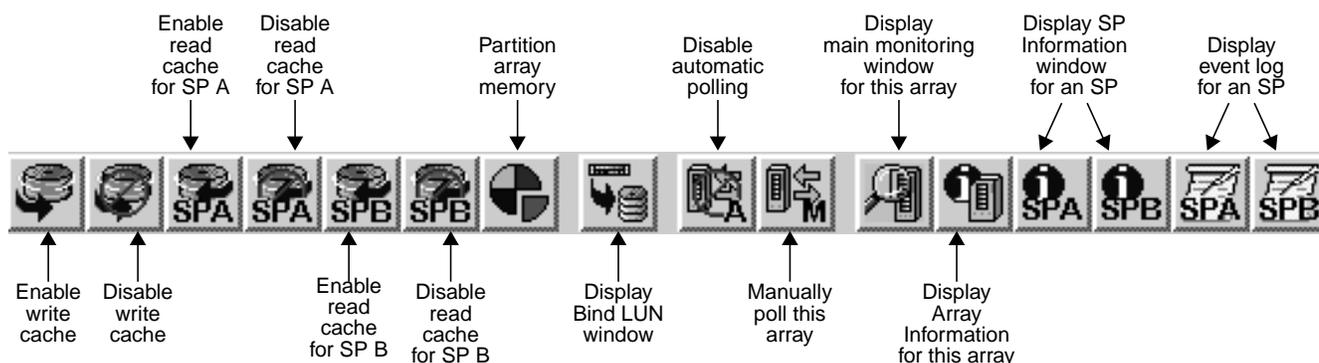
Figure 2-8   Array Toolbar in Array Configuration Window

These buttons duplicate functions available in the Array Configuration window's Array menu.

Note: You can also use this window to reconfigure various parameters or to display information on the array or its components. These tasks are explained in Chapter 3, "Reconfiguring and Fine-Tuning," and Chapter 4, "Monitoring Arrays and Displaying System Statistics," respectively.

Setting Up Array Memory for Caching or RAID 3 LUNs

To use read or write caching or bind RAID 3 LUNs, you must have the required hardware, and specify the array memory partitions. This section explains

- "Caching" on page 33
- "Hardware Requirements for Caching" on page 33
- "Hardware Requirements for RAID 3 LUNs" on page 35
- "Memory Partitioning and SP Memory" on page 35
- "Partitioning Array Memory" on page 37

Caching

The read and write caches are dedicated DIMMs on the storage processor (SP); DIMMs are available in 8-MB and 128-MB sizes. With the maximum of four 128-MB DIMMs installed, the maximum for each type of cache is 512 MB.

You can enable or disable read cache for either SP in an array. The read cache memory that you allocate when you partition memory is shared by all the SP's LUNs for which the read cache is enabled. Read cache is independent on each SP.

Unlike read cache, write cache is always the same (mirrored) on both SPs in an array. You can enable and disable an SP's write cache or change its size; all such changes happen automatically for the array's other SP as well. The write cache memory is shared by all LUNs for which the write cache is enabled.

Besides the array caches, you can assign a specific cache to a particular LUN. However, LUN caching is not enabled until array caching is enabled.

Hardware Requirements for Caching

Hardware requirements for caching are as follows:

*   Read caching requires an SP with at least 16 MB of read cache memory.

*   Write caching requires a Fibre Channel RAID enclosure with

    –   a link controller card (LCC) for each SP in the Fibre Channel RAID enclosure, as well as in any FibreVault attached to it

    –   two functional power supplies in the Fibre Channel RAID enclosure

    –   disk modules in slots 00 through 08 (no failed disk modules)

    –   a fully functional (charged, plugged in, and connected) standby power supply with a fully charged battery

    –   two SPs, each with at least 16 MB of write memory

        Each SP must have the same number of dual in-line memory modules (DIMMs); all DIMMs for the read and write memory module locations must be the same size.

        Note: Read cache DIMMs are physically separate from write cache DIMMs.

33

The amount of write cache you can allocate depends on the SP part number; see Table 2-1 and Table 2-2.

Table 2-1   Cache Capability for SP 9470198

Cache Page Size	User-Configurable Write Cache	User-Configurable Read Cache	Cache Configuration
2 KB	14 MB	0	100% write cache
	0	21 MB	100% read cache
	7 MB	7 MB	50% write/50% read
4 KB	28 MB	0	100% write cache
	0	43 MB	100% read cache
	14 MB	14 MB	50% write/50% read
8 KB	56 MB	0	100% write cache
	0	86 MB	100% read cache
	28 MB	28 MB	50% write/50% read
16 KB	113 MB	0	100% write cache
	0	120 MB	100% read cache
	56 MB	56 MB	50% write/50% read

Table 2-2   Cache Capability for SP 9470207

Cache Page Size	User-Configurable Write Cache	User-Configurable Read Cache	Cache Configuration
2 KB	194 MB	0	100% write cache
	0	248 MB	100% read cache
	106 MB	106 MB	50% write/50% read
4 KB	388 MB	0	100% write cache
	0	497 MB	100% read cache
	213 MB	213 MB	50% write/50% read
8 KB	504 MB	0	100% write cache
	0	504 MB	100% read cache
	426 MB	426 MB	50% write/50% read
16 KB	504 MB	0	100% write cache
	0	504 MB	100% read cache
	504 MB	504 MB	50% write/50% read

Necessary hardware is installed by a qualified Silicon Graphics System Support Engineer; contact your authorized service provider.

Hardware Requirements for RAID 3 LUNs

For RAID 3 LUNs, each SP must have at least as much memory as the sum of the following:

- 16 MB each for read and write memory locations; total 32 MB per SP

and

- if possible, 15 MB of memory (8 MB minimum) allocated for each RAID 3 LUN you want to bind

Note: Binding other types of LUNs with RAID 3 LUNs in the same array is not supported.

Memory Partitioning and SP Memory

SP memory is divided into three memory banks:

- front-end memory: read and write memory banks on the SP board (at least 16 MB in each)
- back-end memory: available memory in the disk enclosures

  Note: For a Fibre Channel storage system, the front end consists of the host, the SPs, and the communication between them, and the back end consists of the disk modules.

- CPU or control RAM: the licensed internal code (LIC) running on the SPs uses 1 MB (mixed mode disabled) or 8 MB (mixed mode enabled) from the read memory and 1 or 8 MB from the write memory on the SP board, plus a dedicated memory bank on the board. (Mixed mode is explained later in this chapter, in "Enabling and Disabling Mixed Mode" on page 45.)

The LIC allocates the read cache partition to the back-end memory, the write cache partition to front-end memory, and the RAID 3 partition across both. The LIC uses the control RAM to manage the read and write caches and the RAID 3 partition (for example, it stores the cache page tables in the control RAM).

Memory that you can allocate to the read and write caches and the RAID 3 partition is the unallocated memory in the front-end and back-end memory; it is known as user free memory.

Note the following points about allocating memory partitions to specific memory areas:

- Changing the RAID 3 partition size causes *ssmgui to* reboot the array. Rebooting restarts the SPs in the array, terminating all outstanding I/O to the array. If you plan to change RAID 3 partition size, make sure that no users are conducting I/O with any filesystems or partitions on the array, and unmount these filesystems or partitions. After the process is completed, remount the filesystems or partitions and notify the users.

- The minimum memory required for accessing a RAID 3 LUN is 4 MB on each SP (2 MB each per read cache and write cache).

- The default amount of memory in read cache and write cache is 15 MB each, with mixed mode disabled as shipped. With mixed mode enabled, it is 8 MB per read cache and write cache.

- The LIC uses 8 MB from read cache and 8 MB from write cache with mixed mode enabled. With mixed mode disabled (the default as shipped), it uses 1 MB from read cache and 1 MB from write cache.

- You might not be able to allocate all of the user free memory to a specific partition.

  For example, because the read cache partition is only in the back-end memory (disk storage), you cannot allocate user free memory in the front end to the read cache partition. Similarly, because the write cache partition is only in the front end, you cannot allocate any user free memory in the back end to the write cache partition.

- You might not be able to increase the size of one partition without decreasing the size of another partition.

  When you increase the size of a partition, *ssmgui* tries to assign user free memory to the partition you are increasing. When *ssmgui* allocates all the user free memory that is available for the partition, it attempts to take memory from another partition to accommodate your request. As a result, when you increase the size of one partition, the size of another partition might decrease. This size change is shown graphically in the Memory Partition window, as discussed on page 41 in "Partitioning Array Memory."

  For example, the size of the RAID 3 partition influences the size of the read cache partition, so that when you increase the size of the read cache partition, the size of the RAID 3 partition might decrease.

Note that when you decrease the size of a partition, the amount of deallocated memory returns to the user free partition, and the size of the other partitions do not increase.

• You might not be able to allocate all user free memory to a partition if the page tables for the new partition sizes would not leave enough free control RAM for the LIC.

The size of the page tables for the read or write cache partition are affected by the size of the cache page size in addition to the size of the partition. A small cache page size results in more cache pages, and thus a larger page table and less free control RAM for the LIC. Conversely, a large cache page size results in less cache pages, and therefore smaller cache page tables and more free control RAM for the LIC. (You can set the cache page size using the Set Page Size option in the Array menu of the Array Configuration window.)

Partitioning Array Memory

You partition memory for each SP using the Memory Partition window, which depicts the enclosure's two SPs and the amount of memory allocated to various functions.

Specifying memory partitions consists of these basic steps:

• "Disabling Caching for Memory Partitioning" on page 38

• "Setting Up Memory Partitions" on page 41

• "Enabling Caching" on page 43

Caution: If you change the RAID 3 partition size, *ssmgui* reboots the array. Rebooting restarts the SPs in the array, terminating all outstanding I/O to the array. If you plan to change RAID 3 partition size, make sure no users are conducting I/O with any filesystems or partitions on the array, and unmount these filesystems or partitions. After the process is completed, remount the filesystems or partitions and notify the users.

Disabling Caching for Memory Partitioning

Before you allocate memory, you must disable write and read caching for the SPs in the array. To disable caching, follow these steps:

1. In the Storage System Manager window, select the arrays whose SP memory you want to partition: click the configure array button near the left end of the array toolbar, or choose Configure from the Array menu.

    To specify memory for one array, double-click the array icon to open the Array Configuration window.

2. In the Array Configuration window (see Figure 2-7), disable array write and read caching:

    *   To disable write caching for both SPs, click the write cache disable button (second from left in the array toolbar, as shown in Figure 2-8), or choose Array > Write Cache State > Disable.

    *   To disable read caching for SP A, click the SP A disable read cache button on the array toolbar, or choose Array > Read Cache State > SP A > Disable.

    *   To disable read caching for SP B, click the SP B disable read cache button on the array toolbar, or choose Array > Read Cache State > SP B > Disable.

3. Because disabling write caching may take a while if the SPs need to write data in the write cache to disks, you must determine when array write and read caching is disabled before proceeding with allocating memory.

    Click the SP information button for an SP in the array (near the right end of the array toolbar). The SP Information window for the SP opens; Figure 2-9 shows an example.
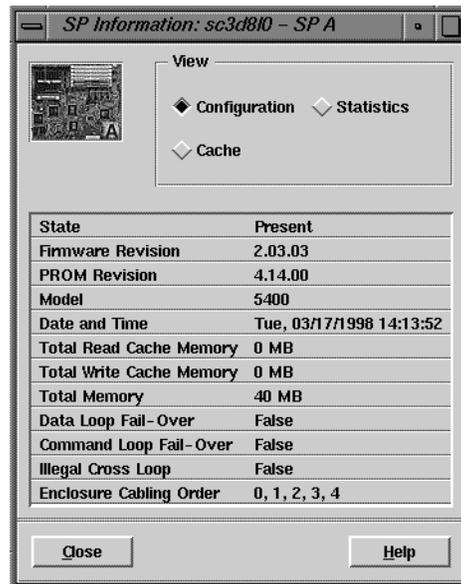
Figure 2-9   Example SP Information Window

4.   In the SP Information window, click *Cache*. The SP Cache Information window
      opens; Figure 2-10 shows an example.
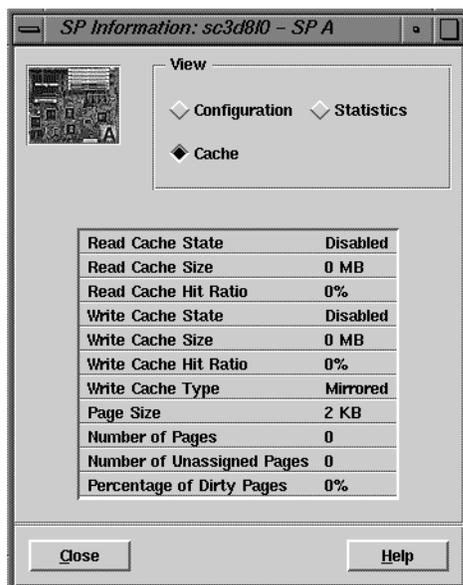
Figure 2-10   Example SP Cache Information Window

5. Check the Read Cache State and Write Cache State entries; they should be
   `Disabled`.

   Disabling write caching may take a while if the SPs need to write data in the write
   cache to disks. If the entry is still `Enabled`, poll the array every few seconds to get
   the latest status: click the manual poll button (toward the right in the array toolbar,
   as shown in Figure 2-8) or choose Poll from the Array menu.

6. When the Read Cache State and Write Cache State entries are `Disabled`, click *Close*
   to close the SP Information window.

7. Repeat these steps for the other SP in the array.

Setting Up Memory Partitions

To allocate memory, follow these steps:

1. In the Array Configuration window, click the partition memory button, or choose Partition Memory from the Array menu. The Memory Partition window for the array opens, as shown in Figure 2-11.
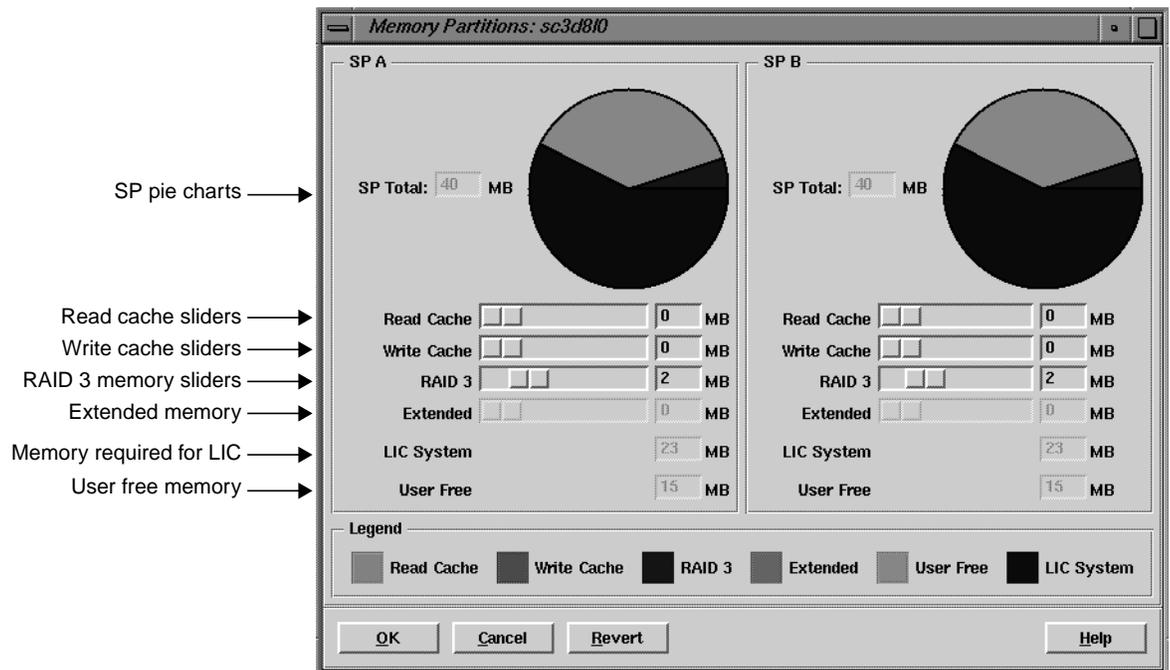
Figure 2-11   Memory Partition Window

Below the pie charts, the fields—Read Cache, Write Cache, RAID 3, Extended, LIC System, and User Free—show the amount of SP memory allocated to those partitions of SP memory.

User free memory is the amount of memory not already allocated to the other memory partitions; its default size is equal to the total memory minus that required for the LIC. Thus, the User Free field shows the amount of memory you can allocate to the other partitions. See "Memory Partitioning and SP Memory" on page 35 for an explanation of the interaction and requirements of memory partitions.

Note: Extended memory is always 0 and cannot be changed.

2. If desired, and if the array has the necessary hardware as explained in "Hardware Requirements for Caching" on page 33, set SP A read cache partition size by dragging the slider until the desired number of megabytes appears in the field to the right of the slider, or by typing the size directly in the field. As you move the slider, the pie chart shows the amount of memory allocated to read cache. (Memory partitions are color-coded; the legend near the bottom of the window explains color coding.)

    Note: Array read caching for SP A is disabled if the SP's read cache partition is 0 MB. It stays disabled you until allocate memory to the partition, and enable caching as explained in "Enabling Caching" on page 43.

    Follow the same procedure to set SP B read cache partition size. Read caches for the two SPs are independent of each other.

3. If desired, and if the array has the necessary hardware as explained in "Hardware Requirements for Caching" on page 33, set SP A write cache partition size by dragging the slider until the desired number of megabytes appears in the field to the right of the slider, or by typing the size directly in the field.

    As you move the slider, the pie charts for both SPs change to show the portion of memory allocated to each SP's write cache partition, because the write cache partition must be the same size on each SP (unlike the read cache partition).

    Note: Array write caching for SP A is disabled if the SP's write cache partition is 0 MB. It stays disabled you until allocate memory to the partition. Use a minimum of 2 MB for the write cache partition on each SP.

4. If desired, and if the array has the necessary hardware as explained in "Hardware Requirements for RAID 3 LUNs" on page 35, set RAID 3 partition size.

    Caution: If you change the RAID 3 partition size, the software reboots the array. Rebooting restarts the SPs in the array, terminating all outstanding I/O to the array. Before you change RAID 3 partition size, make sure no users are conducting I/O with any filesystems or partitions on the array, and unmount these filesystems or partitions.

    Drag the slider until the desired number of MB appears in the field to the right of the slider, or type the size directly in the field.

    As you move the slider, the pie charts for both SPs change to show the portion of memory allocated to each SP's RAID 3 partition, because it must be the same size on each SP.

    Note: If valid RAID 3 LUNs have been bound and you reduce RAID 3 memory to 0 MB, the RAID 3 LUNs are unowned and are no longer accessible after a reboot.

5. To put the memory allocation changes into effect, click *OK* at the lower left of the Memory Partition window. In the configuration window that appears, click *Yes* to confirm the new sizes. The reboot starts.

6. After the reboot, remount any filesystems or partitions you unmounted.

Enabling Caching

If you have allocated memory to the Read Cache or Write Cache partition, use the Array Configuration window to enable read caching for SP A or SP B or array write caching:

* To enable array write caching for both SPs, click the write cache enable button (at the far left of the array toolbar), or choose Array > Write Cache State > Enable.

* To enable read caching for SP A, click the SP A enable read cache button or choose Array > Read Cache State >SP A > Enable.

* To enable read caching for SP B, click the SP B enable read cache button or choose Array > Read Cache State >SP B > Enable.

You have set up the array(s) to perform read caching, write caching, or both. The array uses the default values for caching parameters. To change these settings, see Chapter 3, "Reconfiguring and Fine-Tuning." Otherwise, proceed to the next section, "Planning the Bind."

Note: After you enable read or write caching, the software informs you that enabling was successful, regardless of whether the array has the necessary hardware for such caching.

You may have to restart the agent on the server connected to the array whose memory you just partitioned; see next section, "Planning the Bind."

Planning the Bind

You must bind disk modules into LUNs so that the server's operating system can recognize them. Consider the following before you bind LUNs:

- number of disk modules and their capacities

- parameters for the LUN, such as SP ownership, caching, and so on

- possible effect of bind operation on operations in progress (whether rebooting is required)

Table 2-3 summarizes the number of disk modules you can have in LUNs, and other important factors for binding.

Table 2-3   Binding Disk Modules

LUN Type	Disk Modules	Notes
Any LUN	Unbound disk modules only; all must have the same capacity	For LUNs except RAID 3, you can deallocate RAID 3 memory (see "Setting Up Array Memory for Caching or RAID 3 LUNs" on page 32). Binding RAID 3 LUNs with other LUN types in the same array is not supported.
RAID 5	3 minimum (5 recommended minimum), 16 maximum	You can bind one less module per LUN that you eventually use by selecting an empty slot icon. However, the LUN operates in a degraded mode until a disk module is installed in the empty slot and the array integrates it into the LUN. You can select the modules in any order.
RAID 3	5 or 9	You can bind one less module per LUN than you eventually use by selecting an empty slot icon. However, the LUN operates in a degraded mode until a disk module is installed in the empty slot and the array integrates it into the LUN. You can select the modules in any order.
		Check RAID 3 memory before binding, as explained in "Partitioning Array Memory" on page 37. If you do not allocate adequate memory to the RAID 3 partition, the LUN is unowned.
		Binding other LUN types in the same array is not supported.
RAID 1	2	
RAID 1/0	Even number: 4 minimum, 16 maximum	Disk modules are paired into mirrored images in the order in which you select them: see Figure 2-14 on page 53.
RAID 0	3 minimum, 16 maximum	You can select the modules in any order.

Table 2-3 (continued)   Binding Disk Modules

LUN Type	Disk Modules	Notes
Individual disk	1	
Hot spare	1	You cannot bind as hot spares disk modules 0:0 through 0:8.
		The capacity of a hot spare must be at least as great as the capacity of the largest disk module that it might replace.
		For a RAID 3 LUN, only one hot spare is used in case of disk failure; a second hot spare is not used in case of a second disk failure.

This section contains the following topics:

- "Enabling and Disabling Mixed Mode" on page 45
- "Bind Parameter Defaults" on page 46
- "Caching" on page 33

Note: Binding takes a while; how long varies with the type of SP and size of the disk modules. A RAID 3 or RAID 5 LUN might take more than four hours.

Enabling and Disabling Mixed Mode

Using a menu choice in the Array menu of the Array Configuration window, you must enable *mixed mode* for the LUNs in an array or leave it disabled (the default), depending on LUN type; see Table 2-4.

Table 2-4   Mixed Mode

If Menu Item Is...	Mixed Mode State Is...	RAID 5, 1, 1/0, 0	RAID 3 and Hot Spares
Disable Mixed Mode	Enabled	These LUNs can be bound and accessed.	These LUNs can be bound and accessed.
Enable Mixed Mode	Disabled	These LUNs can be bound, but not accessed until mixed mode is enabled.	These LUNs can be bound and accessed; RAID 3 operation is optimized.

Note the following:

- Mixed mode is disabled by default, which optimizes performance for the array's RAID 3 transfers. With mixed mode disabled (the menu item reads Enable Mixed Mode), you can bind other RAID types, but you cannot access them.

- If you are binding or accessing RAID 5, 1, 1/0, or 0 LUNs, enable mixed mode (the menu item must read Disable Mixed Mode).

- If RAID 5, 1, 1/0, 0 LUNs are bound when mixed mode is disabled, they are unowned; if you then enable mixed mode, they become owned.

- If mixed mode is enabled, disabling it reboots the array's SPs and restarts the LIC. Before beginning this process (clicking Disable Mixed Mode), make sure that no users are conducting I/O with any filesystems or partitions on the array.

- RAID 3 and hot spares work whether mixed mode is enabled or disabled, but RAID 3 is optimized if it is disabled.

Steps for these processes are included in the instructions for binding LUNs later in this chapter.

Bind Parameter Defaults

Arrays set up according to instructions in "Setting Up Array Memory for Caching or RAID 3 LUNs" on page 32 use default values for low and high watermarks and for read caching prefetch parameters. To change these settings, follow instructions in "Binding Disk Modules" on page 49.

When you set up a LUN, it uses certain *bind parameters*:

Rebuild time   Time that the array allots to reconstruct the data on either a hot spare or a new disk module that replaces a failed disk module in a LUN. The rebuild time applies to all RAID LUNs except RAID 0, individual disk LUNs, and hot spares. Generally, the rebuild takes as long as is required; the default is 4 hours.

Verify time   Time that the array allots to checking parity. If an SP detects parity inconsistencies, it starts a background process to check all the parity sectors in the LUN. The time you specify determines the amount of resource the SP devotes to verifying instead of to normal I/O activity. The default is 4 hours.

Element size (stripe element size)

    Number of disk sectors that the array can read or write to a single disk module without requiring access to another disk module (assuming that the transfer starts at the first sector in the stripe). The stripe element size can affect the performance of a RAID 5 or RAID 1/0 LUN. A RAID 3 LUN has a fixed stripe element size of one sector. The smaller the stripe element size, the more efficient the distribution of data read or written. However, if the stripe size is too small for a single I/O operation, the operation requires access to two stripes, which causes the hardware to read and/or write from two disk modules instead of one.

Default SP   The SP that assumes ownership of the LUN after the array's power is turned off and then on again. If the array has two SPs, you can choose to bind some LUNs using one SP as the default, and the rest using the other SP as the default. This tactic balances the load across the SPs or, in a dual-server configuration, establishes the primary route for the LUN. The primary route to a LUN is the route through the default SP; the secondary route is through the other SP.

Read cache   You can enable or disable read cache for either LUN. The read cache memory that you allocate when you partition memory is shared by all the SP's LUNs for which the read cache is enabled. Read cache is independent on each SP.

Write cache   You can enable or disable write cache for either LUN. The write cache memory that you allocated when you partitioned memory is shared by all LUNs for which the write cache is enabled.

Auto assignment state

    Auto assignment state controls the ownership of the LUN when one SP fails in a array with two SPs. With auto assign enabled, if the SP that owns a LUN fails and the server tries to access that LUN through the second SP, the second SP assumes ownership of the LUN so the access can occur. The second SP continues to own the LUN until the SP's power is turned off and on again, at which point ownership of each LUN returns to its default SP. If auto assign is disabled in an array with two SPs, the other SP does not assume ownership of the LUN, so the access to the LUN does not occur. It is recommended that auto assign be left disabled, which is the default for all LUNs (except hot spares, to which it does not apply).

Minimal latency reads

For RAID 3 LUNs only, minimal latency reads provide a more constant bandwidth between the SP and the server when the bandwidth between the disk modules in a RAID 3 LUN and the SP decreases because of a slow response from a disk module. Minimal latency reads change how the SP responds to a request from the server to read a RAID 3 LUN.

When minimal latency reads are disabled, the SP responds to a read request to a RAID 3 LUN with n disk modules by issuing a read request to the n-1 disk modules in the RAID 3 LUN that contain data. It performs a checksum on the data as it receives the data from the n-1 disk modules. If the checksum is correct, it sends the data to the server; if the checksum is incorrect, it reads the parity from the remaining disk module and uses it to reconstruct the data before sending the data to the server.

When minimal latency reads are enabled, the SP responds to a read request to a RAID 3 LUN with n disk modules by issuing a read request to all n disk modules in the LUN. It performs a checksum on the data as it receives it from the first n-1 drives to respond. If the checksum is correct, it sends the data to the server; if the checksum is incorrect, it uses the parity it has received to reconstruct the data before sending the data to the server.

Because minimal latency reads always require access to all the disk modules in the RAID 3 LUN, they nominally decrease the bandwidth between the disk modules and the SPs by about 25%; for a RAID 3 LUN with nine disk modules, it decreases by about 12%.

Not all of these parameters apply to all RAID types, as summarized in Table 2-5.

Table 2-5   RAID Types and Bind Parameters

RAID Type	Default SP	Rebuild Time	Stripe (Element) Size	Verify Time	Auto Assign	Caching
RAID 5	Required	Required	Required	Yes	Disable	Read and write
RAID 3	Required	Required	Not required	Yes	Disable	Not required
RAID 1	Required	Required	Not required	Yes	Disable	Read and write
RAID 1/0	Required	Required	Required	Yes	Disable	Read and write
RAID 0	Required	Not required	Required	No	Disable	Read, write, or both

Table 2-5 (continued)　　　RAID Types and Bind Parameters

RAID Type	Default SP	Rebuild Time	Stripe (Element) Size	Verify Time	Auto Assign	Caching
Individual disk	Required	Not required	Not required	No	Disable	Read, write, or both
Hot spare	Not required	Not required	Not required	No	Not required	Not required

The LUN number must be specified for all RAID types in Table 2-5. RAID 3 also requires minimal latency reads.

Table 2-6 summarizes the values that *ssmgui* sets for standard LUN parameters.

Table 2-6　　　Default LUN Parameters

RAID Type	Rebuild Time	Verify Time	Element Size	Disk Modules	Read and Write Caching	Auto Assign	Minimal Latency Reads
RAID 5	4 hours	4 hours	128	5	Enabled	Disabled	N/A
RAID 3	4 hours	4 hours	N/A	5	N/A	Disabled	Disabled
RAID 1	4 hours	4 hours	N/A	2	Enabled	Disabled	N/A
RAID 1/0	4 hours	4 hours	128	6	Enabled	Disabled	N/A
RAID 0	N/A	N/A	128	5	Enabled	Disabled	N/A
Individual disk	N/A	N/A	N/A	1	Enabled	Disabled	N/A

For all LUN types, the default SP is determined by load balancing; if only one SP is connected, that SP is the default.

Binding Disk Modules

The bind procedure varies depending on the type of LUN you are binding. This section contains the following topics:

- "Setting Bind Parameters: RAID 5, 1, 1/0, or 0" on page 50
- "Setting Bind Parameters: RAID 3 and Hot Spares" on page 56

Note: Using LUN buttons in the Storage System Manager toolbar (see Figure 2-5) to bind LUNs is not recommended. Use them only under the following conditions:

- You are not changing the mixed mode setting; see "Enabling and Disabling Mixed Mode" on page 45.

- You are not changing any bind parameters from the defaults; see Table 2-6.

- There are no performance considerations; disk modules are bound in order of availability and cannot be selected or specified.

- You are using the standard number of disk modules for the LUN; see Table 2-6.

Setting Bind Parameters: RAID 5, 1, 1/0, or 0

To set bind parameters for RAID 5, 1, 1/0, or 0, follow these steps:

1. If you plan to use caching, allocate memory as necessary; see "Setting Up Array Memory for Caching or RAID 3 LUNs" on page 32. You can deallocate any memory allocated to RAID 3 memory (binding RAID 3 memory and other types of LUNs in the same array is not supported). This process may involve rebooting.

2. To speed up binding, turn off automatic polling if it is on (see the field at the lower right of the Storage System Manager window): click the button at the far right of the Storage System Manager toolbar (as shown in Figure 2-8).

    To see status information during binding, poll for it manually by clicking the manual poll button next to the automatic poll button in the toolbar, or choose Poll from the Array menu. To see the polling information, double-click the LUN icon in the Array Configuration window to open the LUN Information window (see "LUN Configuration Information" on page 95 in Chapter 4 for an explanation of fields in the window).

3. In the Storage System Manager window, select the array whose disk modules you want to bind. You can select more than one array if you want to bind all their disk modules into the same type of LUN.

4. Double-click on the array to open its Array Configuration window.

5. In the Array menu of the Array Configuration window, make sure mixed mode is enabled. If you see the menu choice Disable Mixed Mode, then mixed mode is currently enabled; proceed to step 6.

    If you see the menu choice Enable Mixed Mode, then mixed mode is currently disabled, and you must enable it.

6. Open the Bind LUNs dialog for the array(s): click the button near the middle of the LUN (lower) toolbar, or choose Bind LUN from the Array menu. Figure 2-12 shows an example of the Bind LUNs dialog.
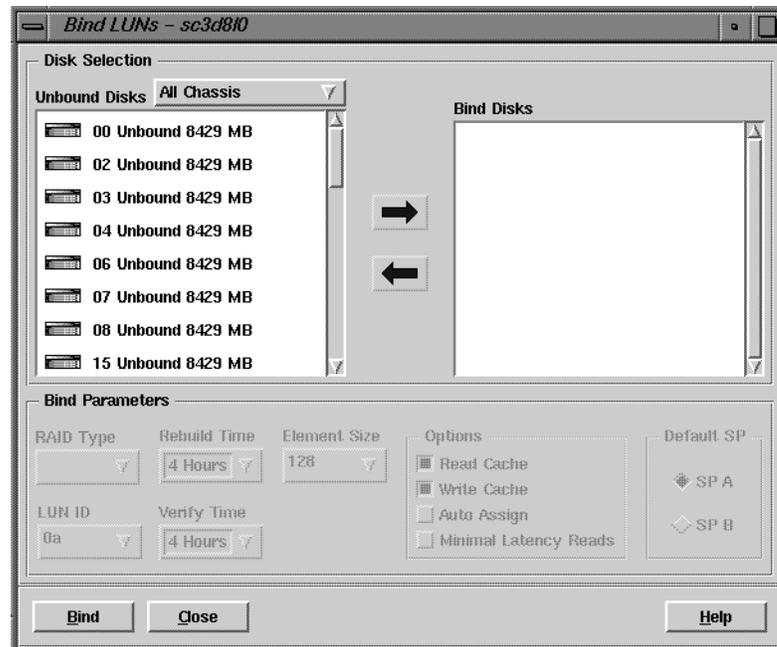
Figure 2-12   Bind LUNS Dialog

7. Select disk modules for the LUN. If the array has disk modules in more than one enclosure, select the enclosures containing the disk modules you want to bind:

- To select disk modules from all enclosures in the array: If the Unbound Disks field contains `All Chassis`, continue to step 8. If it contains the name of an enclosure, pull down its selection list and choose All Chassis.

- To select disk modules from one enclosure in the array: If the Unbound Disks field shows the name of the enclosure you want, proceed to step 8. Otherwise, choose the enclosure name from the pull-down menu.

In the Unbound Disks area, select the disk modules that you want to bind into a LUN, and click the right arrow button. (Alternatively, you can use both mouse buttons to drag and drop the disk modules from the Unbound Disk area to the Bind Disks area.) Figure 2-13 shows disk modules selected for binding.
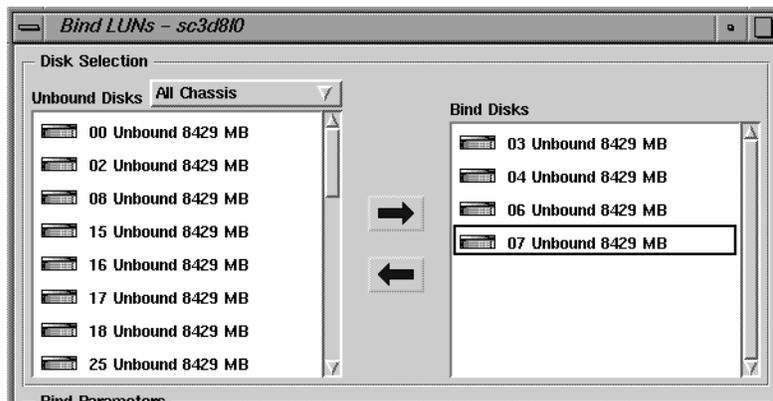
Figure 2-13   Example Bind LUNs Dialog With Disk Modules Selected for Binding

See Table 2-3 for the number of disks in various types of LUNs.

Tip: To select multiple consecutive disk modules, click the first disk module icon and drag the cursor over the other disk modules. (Alternatively, you can hold down the Ctrl key and click disk modules.)

If you move a wrong disk module to the Bind Disks area, select it and click the left arrow button to move it back to the Unbound Disks area. (Or drag and drop it into the Unbound Disks area.

If you are binding a RAID 1/0 LUN, the order in which you select modules is important; Figure 2-14 diagrams this order.
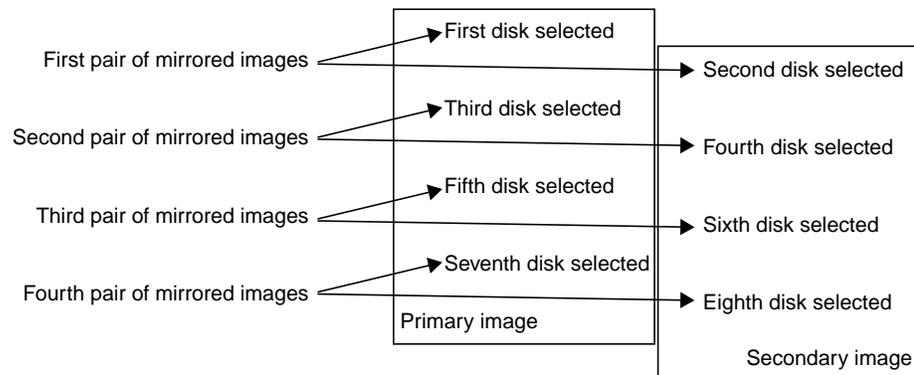
Figure 2-14   Selecting Modules for Mirror Images for a RAID 1/0 LUN

8. In the RAID Type field, select the RAID type. The list displays only RAID types that are available for the number of disk modules you selected.

9. Make sure Auto Assign in the Options section of the Bind LUNs dialog is unselected (the default), as shown in Figure 2-15. For more information on auto assign, see page 47.
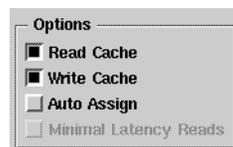
Figure 2-15   Options Display: Auto Assign Unselected

10. To change other bind parameters in the Bind LUNs dialog:

  • LUN ID: Select another LUN hexadecimal identifier (ID) from the LUN ID field.

    The default LUN ID is the next hex number available, starting with 0 and ending with 1f. The list displays only numbers that are available. The default number is 0 for the first LUN that you bind, regardless of the number of SPs or servers attached to the array. The default number for the second LUN you bind is 1; for the third LUN, it is 2; for the fourth LUN, it is 3, and so on. You can

specify a nondefault number if desired. After you bind a LUN with a nondefault number, the default number for the next LUN is the lowest number you skipped. The maximum number of LUNs is 32.

For hot spares, assign LUN numbers starting with the highest number available and continue downwards.

- Rebuild Time: The default time of 4 hours is adequate for most situations. Generally, the rebuild takes as long as is required. For more information on rebuild time, see page 46.

  If the LUN is not a RAID 0 LUN, individual disk, or hot spare, and you want a different rebuild time from that in the Rebuild Time field, enter the number of hours in the field or click the field list button and select the desired number of hours from the list that opens.

  Actual rebuild time can differ significantly from the time you specify, especially for a RAID 1/0 LUN. Since a RAID 1/0 with n disk modules can continue functioning with up to as many as $n/2$ failed disk modules and only one disk module at a time is rebuilt, the actual rebuild time for such a LUN is the time you specify multiplied by the number of failed disk modules.

  Note: For a rebuild time of greater than 4 hours, you must change the rebuild time after the LUN is bound, as explained in "Changing LUN Bind Parameters That Do Not Require Unbinding" on page 65 in Chapter 3.

- Verify Time: If the LUN is not a RAID 0 LUN, individual disk, or hot spare, and you want a different verify time from that in the Verify Time field, enter the number of hours in the field or click the field list button and select the desired number of hours from the list that opens. For more information on verify time, see page 46.

  Note: For a verify time of greater than 4 hours, you must change the verify time after the LUN is bound, as explained in "Changing LUN Bind Parameters That Do Not Require Unbinding" on page 65 in Chapter 3.

- Element Size: For a RAID 0, RAID 1/0, or RAID 5 LUN, if you want the LUN to have a stripe element size with a different number of sectors from the number in the Element Size field, select the desired number of sectors from the field list.

  Generally, use the smallest stripe element size that rarely forces access to another stripe. The default stripe element size for RAID 5 is 128 sectors. Any size you choose should be an even multiple of 16 sectors; supported values are 4, 8, 16, 32, 64, and 128. For more information on element size, see page 47.

- Read Cache: For LUNs other than RAID 3 or hot spares, to set the state of the read or write cache for the LUN, click the SP's read cache button near the left end of the array toolbar. The display indicates whether read or write cache for the default SP is enabled or disabled for the LUN, as shown in Figure 2-16.
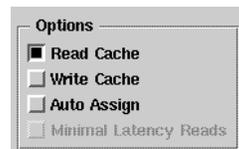
Figure 2-16   Options Display: Read Cache Enabled

Enabling read cache for any type of LUN (except a RAID 3 LUN or hot spare) is recommended.

If you enable the default SP read cache for a LUN, caching occurs only when the array read is enabled for the default SP. You enable the array read cache for SP A from the Array Configuration window using the appropriate button (such as the SP A enable read cache button) on the array toolbar or by choosing, for example, Array > Read Cache State > SP A > Enable.

The read cache memory that you allocated when you partitioned memory is shared by all LUNs for which the read cache is enabled.

- Write Cache: For LUNs other than RAID 3 or hot spares, to set the state of the read or write cache for the LUN, click the SP's read cache button near the left end of the array toolbar. The display indicates whether write cache for the default SP is enabled or disabled for the LUN.

Enabling write cache for a RAID 5 LUN is highly recommended; enabling it for other LUN types for which it is possible is also recommended. Write cache is always mirrored and thus requires two SPs.

If you enable the default SP write cache for a LUN, caching occurs only when the array write cache is enabled for the default SP. You enable the array read or write cache for SP A from the Array Configuration window using the appropriate button (such as the SP A enable write cache button) on the array toolbar or by selecting, for example, Array > Write Cache State > SP A > Enable.

The write cache memory that you allocated when you partitioned memory is shared by all LUNs for which the write cache is enabled.

- Default SP: If the LUN is not a hot spare, click the button for the other SP to change the LUN's default owner to that SP. This option is available only for arrays with two SPs. For more information on default SP, see page 47.

11. When all bind parameters for the LUN are the way you want them, click *Bind* at the lower left in the Bind LUNs dialog.

12. In the confirmation window that opens, click *Yes* to start the bind operation. A window opens stating that the bind operation was successfully initiated; click *OK*.

    A blue icon for the LUN appears in the Unowned LUNs area in the Array Configuration window. A small letter T in the icon indicates its transitional state.

    Binding takes a while; how long varies with the type of SP and size of the disk modules. (A RAID 5 LUN might take more than four hours.) When polling determines that the bind operation is completed, the LUN icon moves to the selection area for its default SP and becomes gray.

    Note: To stop a bind that is in progress, use the RAID CLI *rebootsp* subcommand; see "rebootSP" on page 177 in Chapter 6. Alternatively, you can use the GUI to remove all drives in the bind.

13. Once the LUN is bound, change the mixed mode setting from Disabled to Enabled: choose Enable Mixed Mode from the Array menu. (This message indicates that mixed mode is currently disabled.)

14. In the confirmation window that opens, click *Yes* to enable mixed mode.

Setting Bind Parameters: RAID 3 and Hot Spares

To set bind parameters for RAID 3 and hot spares, follow these steps:

1. Make sure the memory allocated for RAID 3 is as you want it; see "Setting Up Array Memory for Caching or RAID 3 LUNs" on page 32. This process may involve rebooting.

2. To speed up binding, turn off automatic polling if it is on; see step 2 on page 50.

3. In the Storage System Manager window, select the array whose disk modules you want to bind. You can select more than one array if you want to bind all their disk modules into the same type of LUN.

4. Double-click on the array to open its Array Configuration window.

5. In the Array menu of the Array Configuration window, look for the mixed mode menu choice:

- If you see Enable Mixed Mode, then mixed mode is currently disabled; proceed to step 6.

- If you see Disable Mixed Mode, then mixed mode is currently enabled; you must disable it.

If you must disable mixed mode, follow these steps:

Caution: Disabling mixed mode reboots the array's SPs and restarts the LIC.

- Make sure that no users are conducting I/O with any filesystems or partitions on the array.

- Click Disable Mixed Mode in the Array menu of the Array Configuration window.

- In the warning window that appears, click *Yes*.

6. Select all the disk modules for the LUN; for RAID 3, select either 5 or 9. See instructions at step 6 on page 51. In binding hot spares, you can select as many disks in as many enclosures as you like, and bind them all at the same time.

   Note: Binding other types of LUNs with RAID 3 LUNs in the same array is not supported (except hot spares).

7. In the RAID type field, select RAID 3 or Hot Spare Only RAID types that are available for the number of disk modules you selected appear in the list.

8. For RAID 3 LUNs, make sure Auto Assign in the Options section of the Bind LUNs dialog is unselected (the default), as shown in Figure 2-16. For more information on auto assign, see page 47.

9. Change other bind parameters:

   - LUN ID: see page 53.

     For hot spares, assign LUN numbers starting with the highest number available and continue downwards.

   - Verify Time: For a RAID 3 LUN, if you want a different verify time from that in the Verify Time field, enter the number of hours in the field or select the desired number of hours from the list that opens. For more information on verify time, see page 46.

     Note: For a verify time of greater than 4 hours, you must change the verify time after the LUN is bound, as explained in "Changing LUN Bind Parameters That Do Not Require Unbinding" on page 65 in Chapter 3.

- Read Cache, Write Cache: Enabling these is not supported for RAID 3 LUNs or hot spares.

- Minimal Latency Reads: To change this option, which is available only for RAID 3 LUNs, click *Minimal Latency Reads* in the Options section. For more information on minimal latency reads, see page 48.

- Default SP: If the LUN is not a hot spare, click the button for the other SP to change the LUN's default owner to that SP. This option is available only for arrays with two SPs. For more information on default SP, see page 47.

10. If your application that is writing to the RAID 3 LUN is single-threaded and performance is more important than data integrity, you can increase the performance of these applications by enabling RAID 3 buffering:

   ■ Display the Array Configuration window for the array whose RAID 3 write buffering you want to enable.

   ■ Determine whether the array's RAID 3 write buffering is enabled: click the array information window button near the right end of the array toolbar in the Array Configuration window, or choose Array Information from the Array menu. A window opens; Figure 2-17 shows an example.
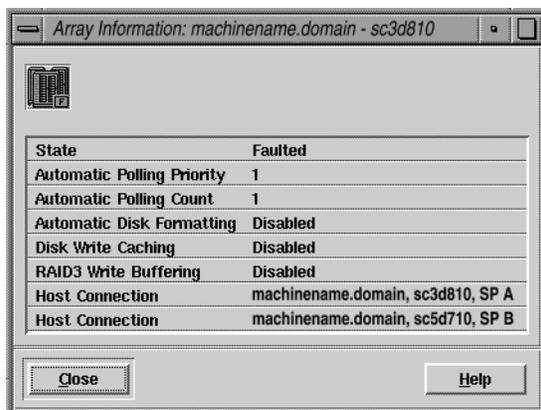
Figure 2-17   Array Information Window

   ■ If the array's RAID 3 write buffering is not already enabled, choose Enable RAID3 Write Buffering from the Array menu.

   ■ In the confirmation window that opens, click *Yes*.

(To disable RAID 3 write buffering, choose Disable RAID3 Write Buffering from the Array menu, and click *Yes* in the confirmation window.)

11. When all bind parameters for the LUN are the way you want them, click *Bind* at the lower left in the Bind LUNs dialog.

12. In the confirmation window that opens, click *Yes* to start the bind operation. A window opens stating that the bind operation was successfully initiated; click *OK*.

    A blue icon for the LUN appears in the Unowned LUNs area in the Array Configuration window. A small letter T in the icon indicates its transitional state.

    Binding takes a while; how long varies with the type of SP and size of the disk modules. (A RAID 3 LUN might take more than four hours.) When polling has determined that the bind operation is completed, the LUN icon moves to the selection area for its default SP and becomes gray.

    Note: To stop a bind that is in progress, use the RAID CLI *rebootsp* subcommand; see "rebootSP" on page 177 in Chapter 6. Alternatively, you can use the GUI to remove all drives in the bind.

13. Once the LUN is bound, change the mixed mode setting from Enabled to Disabled:

    Caution: Disabling mixed mode reboots the array's SPs and restarts the LIC.

    ■ Make sure that no users are conducting I/O with any filesystems or partitions on the array.

    ■ Choose Disable Mixed Mode from the Array menu of the Array Configuration window.

    ■ In the confirmation window that appears, click *Yes*.

Note: A RAID 3 LUN can use only one hot spare in case of disk failure. If a second hot spare is available and a second disk module in the RAID 3 LUN fails, the LUN does not use the second hot spare.

Making LUNs Available to the Server Operating System

When all LUNs are assigned, make the LUNs available to the server's operation system:

- Reboot the system.

or

- Enter

  scsiha -lp *controllernumber***; ioconfig -f /hw**

  For example:

  ```
  scsiha -lp 3; ioconfig -f /hw
  ```

Enabling Command-Tagged Queuing

Command-tagged queueing (CTQ) allows multiple outstanding commands to a single SCSI target (that is, a LUN in a storage system), resulting in increased I/O performance.

The SP supports SCSI-2 queuing of requests for its LUNs. The requests are first-come, first-served in that all requests can be sent to one LUN from one initiator and cause a Queue Full status (unexpected SCSI status byte 0x28) to be returned for all other I_T_L SCSI selections. This condition continues until one of the outstanding requests completes and thus frees queue space. The SP can handle up to 250 CTQs.

If Queue Full status is returned for a given I/O request, that request is retried. If the requests cannot be sent to the SP after four retries, the request is aborted, which, in the case of a write request, can have unfortunate consequences.

When CTQ is enabled (with *fx*) for a given LUN, the default CTQ depth for the LUN is 2; this value is stored in the LUN's volume header. You must use *fx* to change this value. Table 2-7 defines the maximum CTQ depth values per LUN for single-hosted SPs and dual-hosted SPs.

Table 2-7   Maximum CTQ Depths per LUN

Number of LUNs	Single-Hosted SPs	Dual-Hosted SPs
1	250	126
2	125	62

Table 2-7 (continued)   Maximum CTQ Depths per LUN

Number of LUNs	Single-Hosted SPs	Dual-Hosted SPs
3	62	31
4	31	15

For optimum system performance, enable command-tagged queuing. Table 2-8 shows performance benefits of CTQ.

Table 2-8   CTQ Performance Benefits for 2 KB Random Read, 16 Threads

CTQ	avque	r+w/s	blks/s	w/s	wblks/s	avwait (μs)	avserv (ms)
Disabled	16.0	69	137	0	0	217.6	14.5
Enabled	16.0	304	607	0	0	48.8	3.3

The *fx* program syntax is as follows.

```
fx -x "controllertype(controller_number,drive_number,lun_number)"
```

Follow these steps:

1. Enter the *fx* command with appropriate parameters; for example:

   ```
   fx -x "dksc(6,2,2)"
   ```

   Output such as the following appears:

   ```
   fx version 6.4, Aug 3, 1998
   ...opening dksc(6,2,2)
   ...controller test...OK
   Scsi drive type == SGI RAID 5 0757
   fx: Warning: bad sgilabel on disk

   creating new sgilabel

   ----- please choose one (? for help, .. to quit this menu)-----
   [exi]t            [d]ebug/          [l]abel/          [a]uto
   [b]adblock/       [exe]rcise/       [r]epartition/    [f]ormat
   ```

2. Update parameters; at the `fx>` prompt, enter

   fx> /label/set/param

   Output such as the following appears:

   ```
   fx/label/set/parameters: Error correction = (enabled)
   fx/label/set/parameters: Data transfer on error = (enabled)
   fx/label/set/parameters: Report recovered errors = (enabled)
   fx/label/set/parameters: Delay for error recovery = (enabled)
   fx/label/set/parameters: Err retry count = (0)
   fx/label/set/parameters: Transfer of bad data blocks = (enabled)
   fx/label/set/parameters: Auto bad block reallocation (write) =
   (enabled)
   fx/label/set/parameters: Auto bad block reallocation (read) =
   (enabled)
   fx/label/set/parameters: Read ahead caching = (disabled)
   ```

3. At the `Enable CTQ` prompt, enter **enable**:

   ```
   fx/label/set/parameters: Enable CTQ = (disabled) enable
   ```

4. At the `CTQ depth` prompt, enter **10**:

   ```
   fx/label/set/parameters: CTQ depth = (2) 10
   ```

   Output such as the following appears.

   ```
   fx/label/set/parameters: Read buffer ratio = (0/256)
   fx/label/set/parameters: Write buffer ratio = (0/256)
    * * * * * W A R N I N G * * * * *
   ```

5. At the following prompt in the last line above, enter **yes**:

   ```
   about to modify drive parameters on disk dksc(6,2,2)! ok? yes
   ```

   The following output appears:

   ```
   ----- please choose one (? for help, .. to quit this menu)-----
   [exi]t            [d]ebug/           [l]abel/            [a]uto
   [b]adblock/       [exe]rcise/        [r]epartition/      [f]ormat
   ```

6. Type **exit** to exit *fx*. The following message appears:

   ```
   label info has changed for disk dksc(6,2,2). write out changes?
   (yes)
   ```

7. Type **y** to write the changes to disk.

Reconfiguring and Fine-Tuning

You configured a LUN initially when you created it as described in Chapter 2. After you create a LUN, you can change the LUN bind parameters, except for its RAID type, without unbinding the disk modules that constitute the LUN. Also without unbinding, you can also change the LUN read caching (prefetch) parameters, which the array automatically set to their default values when you created the LUN.

However, if you want to change the LUN RAID type or the number or capacity of the disk modules in the LUN, you must unbind the disk modules that constitute it, and then rebind the desired disk modules into another LUN.

After you configure a LUN, you might have to adjust parameters such as cache size or read-ahead caching parameters. This chapter has guidelines for this kind of fine-tuning.

This chapter contains the following sections:

Note: For optimum system performance, enable command-tagged queuing as explained in "Enabling Command-Tagged Queuing" on page 60 in Chapter 2.

You can use the LUN toolbar in the Array Configuration window for many steps explained in this chapter. Figure 3-1 shows LUN toolbar buttons and their functions.
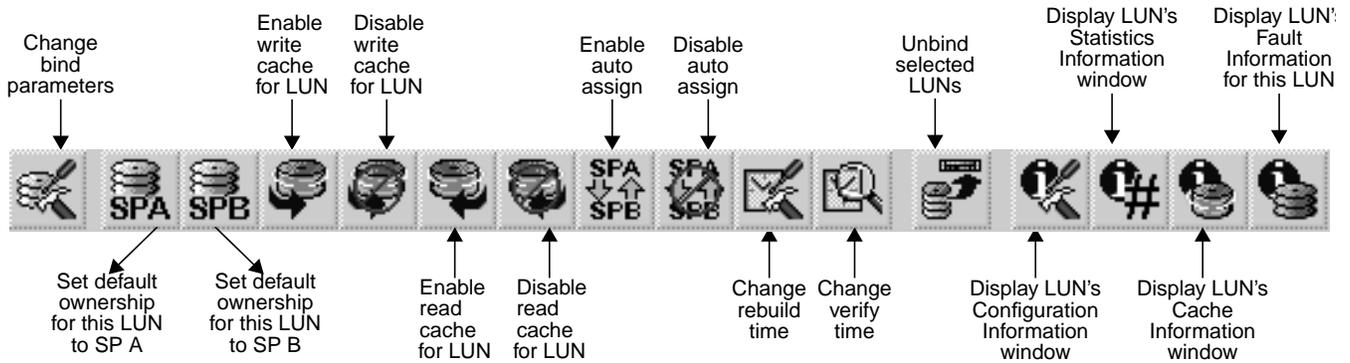
Figure 3-1   LUN Toolbar in Array Configuration Window

Moving Disk Modules

You may need to move disk modules. Generally, modules should not be moved from one slot to another; but if moving one is absolutely necessary, the system operator or service person can move it with the following requirements:

- Do not move disk modules in slots 00, 01, and 02, which contain the licensed internal code.

- The disk module must be unbound. Moving a module that is part of a LUN to another slot makes all information on the LUN inaccessible. "Unbinding a LUN" on page 70 explains how to unbind LUNs.

- You must remove and install the disk module while the array is powered on. Use the procedure explained in the *Origin FibreVault and Fibre Channel RAID Owner's Guide.*

In a Fibre Channel RAID enclosure, you can choose any disk modules for a LUN of any RAID type without affecting the performance or high availability of the LUN. For simplicity, however, choose consecutive disk modules.

Changing LUN Bind Parameters That Do Not Require Unbinding

You can change these LUN parameters for any LUN except a hot spare without unbinding the LUN, provided that it is available to the operating system:

- default SP (any LUN)

- read and write cache state (any LUN except RAID 3)

- rebuild time (RAID 5, 3, 1, or 1/0 LUNs only)

- verify time (RAID 5, 3, 1, or 1/0 LUNs only)

- auto assignment state (any LUN)

    Note: It is recommended that auto assign be disabled (unselected) for all LUNs except hot spares.

- minimum latency reads state (RAID 3 LUNs only)

- prefetch (read-caching) parameters (any LUN)

Note: Changing LUN bind parameters does not affect the data stored on the LUN. You can determine LUN parameters except for prefetch by looking at the LUN Information window; see "Using the LUN Information Windows" on page 94 in Chapter 4.

This section explains how to change each of these parameters except default SP and prefetch. For information on changing the default SP, see "Transferring Control of a LUN (Manual Trespass)" on page 68; for information on changing prefetch parameters, see "Changing Prefetch (Read-Ahead Caching) Parameters" on page 77.

For details on the meaning of these settings, see "Planning the Bind" on page 44 in Chapter 2.

You can change a specific LUN parameter from the Array Configuration window using options in the LUN menu, or you can use the Change Bind Parameters window to change several parameters at once.

Changing LUN Parameters in the Array Configuration Window

Follow these steps:

1. In the Storage System Manager window, select the array with the LUN whose rebuild time, verify time, auto assignment state, minimum latency reads, read cache state, or write cache state you want to change.

2. Click the configure button at the left end of the toolbar, or choose Configure from the Array menu to open the Array Configuration window for the selected array. Figure 3-2 shows this window.
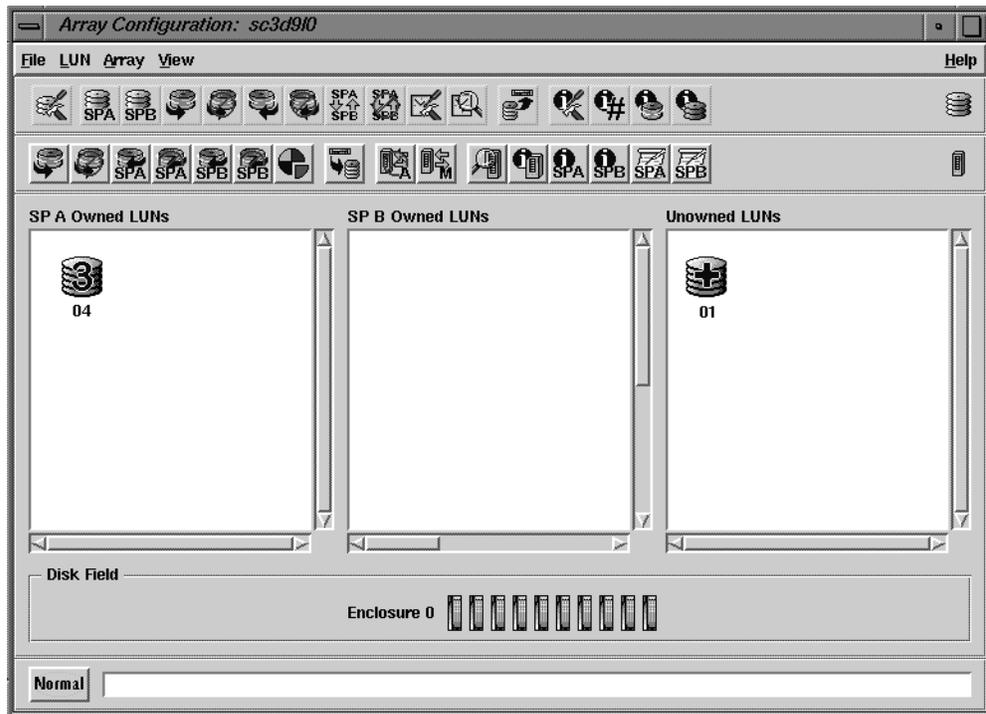
Figure 3-2   Array Configuration Window

3. In the Array Configuration window, select the LUN whose parameters you want to change.

   Tip: When you select an LUN, clicking the right mouse button in its LUN selection area brings up a menu with the same functions as the LUN toolbar buttons.

4. Change the desired parameters: choose LUN and then the option, such as Change Rebuild Time. Select the desired option from the list that opens. Any change to the rebuild time takes effect immediately.

   Choices that are not available for the type of the LUN you selected are grayed out; for example, minimum latency reads is not available for any LUN other than RAID 3.

   Note: The LUN can use read or write caching only if the read or write cache, respectively, is enabled for its default SP and certain hardware requirements are met. See "Enabling or Disabling Array Caching" on page 73.

Changing LUN Parameters in the Change Bind Parameters Window

Follow these steps:

1. In the Storage System Manager window, select the array with the LUN whose rebuild time, verify time, auto assignment state, minimum latency reads, read cache state, or write cache state you want to change. Open the Array Configuration window (see step 2 on page 66).

2. In the Array Configuration window, select the LUN whose parameters you want to change.

3. Choose Change Bind Parameters from the LUN menu. The Change Bind Parameters window for the selected LUN opens; Figure 3-3 shows an example.
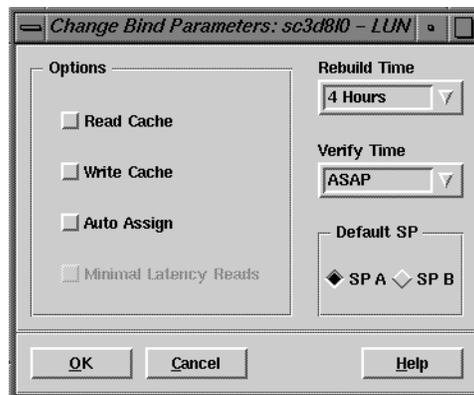
Figure 3-3   Change Bind Parameters Window

Caution: Although you can use the Change Bind Parameters window to change the default SP, read the information in "Transferring Control of a LUN (Manual Trespass)" on page 68 before doing so. The process requires powering the storage system off and then on again for the change to take effect.

4. Change the desired parameter by clicking the appropriate button and selecting a value from the list.

   Choices that are not available for the type of the LUN you selected are grayed out; for example, minimum latency reads is not available for any LUN other than RAID 3.

   Note: The LUN can use read or write caching only if the read or write cache, respectively, is enabled for its default SP. See "Enabling or Disabling Array Caching" on page 73.

5. When the bind parameters for the LUN are set as you want, click *OK*.

   Note: Any change to the rebuild time takes effect immediately.

Transferring Control of a LUN (Manual Trespass)

This section explains how to transfer control of a LUN from the SP that owns it (the primary route to the LUN) to the other SP (the secondary route to the LUN). Transferring control of a LUN is also know as changing SP ownership of a LUN. This procedure is also known as manual trespass.

Use the procedure in this section when the system operator or service person has installed a second SP and you want to assign some of the LUNs to the new SP. Another use is to balance LUNs between two SPs.

Depending on the type of server, you might want to use the procedure if any of the following failure situations occurs:

• A failed SP has been replaced, and you want to transfer control to the working SP.

• In an array connected to two host bus adapters, one adapter or the connection to it fails, and you want the working adapter to access the LUNs owned by the failed adapter.

• One server in a dual-server configuration fails, and you want the working server to access the failed server's LUNs.

Note: For some dual-server applications, special failover software and hardware is available, such as IRIS FailSafe. If you have IRIS FailSafe, use it to handle the failure situations listed above, instead of the procedure described in this section.

XLV volumes and the auto-assignment parameter also transfer control of a LUN from one SP to another. For information on auto assignment, see page 47 in Chapter 2; for information on XLV volumes, see *Getting Started with XFS Filesystems*.

Transferring control of a LUN from one SP to the other can affect how the operating system accesses the LUN. However, any change you make in ownership does not take effect until the SP is rebooted.

To transfer control of a LUN (manual trespass), you can use these commands:

- *trespass*: See "trespass" on page 183 in Chapter 6. This procedure does not require rebooting

- *scsifo*: See its man page, scsifo<1M). This procedure does not require rebooting.

- *ssmgui*: see the following steps. This procedure requires rebooting.

To use *ssmgui* to transfer control of a LUN, follow these steps:

1. Because this procedure requires rebooting the SP (either by powering the array off and then on again, or by using the *rebootSP* command), make sure all users are off the system before you begin and that no I/O operations are in progress.

2. In the Storage System Manager window, select the array with the LUN for which you want to change the default SP. Open the Array Configuration window (see step 2 on page 66).

3. In the Array Configuration window that opens, choose Change Default SP from the LUN menu.

4. In the list that appears, select the SP that you want to control the LUN.

5. In the confirmation window that opens, click *Yes*.

6. Unmount all filesystems associated with the LUN.

7. Power the array off and then on again, following procedures explained in the *Origin FibreVault and Fibre Channel RAID Owner's Guide*. Alternatively, you can use the *rebootSP* command; see "rebootSP" on page 177 in Chapter 6.

8. Make the LUNs available to the server's operating system:

scsiha -lp *controllernumber***; ioconfig -f /hw**

For example:

```
scsiha -lp 3; ioconfig -f /hw
```

You can also change LUN ownership in the Change Bind Parameters window; see "Changing LUN Parameters in the Change Bind Parameters Window" on page 67.

Unbinding a LUN

Typically, you unbind a LUN only if you want to add disk modules to it or use its disk modules in a different LUN. In either of these situations, make sure that the LUN contains the disk modules you want. To determine the disk modules that make up a LUN, follow these steps:

1. In the Storage System Manager window, select the array with the LUN you want to unbind. Open the Array Configuration window (see step 2 on page 66).

2. In the LUN selection area, select the LUN whose disk modules you want to identify.

   In the Disk Field, each disk module in the LUN is surrounded by a box with the same color as the box surrounding the selected LUN.

   You can also determine the disk modules in a LUN by using the LUN IDs. If the LUN IDs do not appear in any disk modules in the Disk field, choose Show LUN IDs from the View menu.

Caution: Unbinding a LUN destroys any data on it. Observe these precautions:

- Back up any data you want to retain from the LUN before beginning the unbind procedure.

- If you plan to unbind a LUN, make sure no users are conducting I/O with any filesystems or partitions on the array.

- Unbind only one LUN at a time.

- Unbinding the last LUN may prevent you from accessing the storage system.

- If you unbind all of the LUNs and lose access to the storage system, see "Re-Establishing Communication With an Array" on page 202 in Appendix A.

To unbind a LUN, follow these steps:

1. Make sure that no I/O operations to this LUN are in progress. If appropriate, back up data from the LUN.

2. In the Storage System Manager window, select the array with the LUN for which you want to change the default SP. Open the Array Configuration window (see step 2 on page 66).

3. In the LUN selection area, select the LUN whose disk modules you want to unbind.

4. Click the unbind LUN button near the middle of the LUN toolbar, or choose Unbind from the LUN menu. A window displays a warning message and asks you to confirm the unbind operation.

5. If you are sure, click *Yes* in the warning window to confirm the operation.

    A window opens telling you that the LUNs were unbound. The LUN icon disappears from the LUN selection area of the Storage System Manager window.

Changing LUN RAID Type

If you want to change the LUN RAID type or the number or capacity of its disk modules, you must change the bound configuration of a LUN, which means that you destroy the LUN and recreate it.

To change the number of disk modules in a LUN, you must follow the same guidelines as you do to create a new configuration; see "Binding Disk Modules" in Chapter 2.

All disk modules in a LUN must have the same capacity to utilize disk space fully. Binding other types of LUNs with RAID 3 LUNs in the same array is not supported.

Caution: Before you change LUN type, you must unbind the LUN, which destroys any data on it. Observe these precautions:

• Back up any data you want to retain from the LUN before beginning the unbind procedure.

• If you plan to unbind a LUN, make sure no users are conducting I/O with any filesystems or partitions on the array.

- Unbind only one LUN at a time.

- Unbinding the last LUN may prevent you from accessing the storage system.

  If you unbind all of the LUNs and lose access to the storage system, see "Re-Establishing Communication With an Array" on page 202 in Appendix A.

To change the LUN RAID type, follow these steps:

1. Make sure that no I/O operations to this LUN are in progress.

2. If appropriate, back up data from the LUN.

3. Unmount all affected filesystems associated with the LUN.

4. Unbind the LUN following instructions in "Unbinding a LUN" on page 70.

    If you are adding new disk modules for the LUNs, you do not need to unbind them.

5. If you are moving data from RAID 3 LUNs to RAID 5 LUNs, enable mixed mode.

6. Move disk modules or install new disk modules if needed; follow instructions in the *Origin FibreVault and Fibre Channel RAID Owner's Guide.*

7. If you are using newly installed disk modules for the LUNs, copy the data from the old disk modules in the LUN onto them.

8. Bind the disk modules into the desired LUNs, following instructions in "Binding Disk Modules" in Chapter 2.

9. Make the newly created LUNs available to the operating system as described in "Making LUNs Available to the Server Operating System" on page 60 in Chapter 2.

Upgrading an Array With Higher Capacity Disk Modules

You can replace any disk modules in an array with higher capacity modules, as long as you do not replace all three disk modules that contain the licensed internal code database at the same time. These disk modules (also known as database disk modules) have disk IDs 00, 01, and 02 in the first enclosure (DPE) in a chain, or the only enclosure in the array. At least two are required for DPE operation. If all are removed at the same time, contact with the array is lost.

Caution: Do not power off the array during this procedure.

To install higher capacity disk modules, follow these steps:

1. Unbind the LUNs whose disk modules you want to replace, following instructions in "Unbinding a LUN" on page 70.

2. Replace all disk modules you want to replace.

   If you are replacing the first three disk modules in a DPE (first or only enclosure in a chain), replace all disk modules except for the one in slot 00; leave the disk module in slot 00 in place.

3. Bind the disk modules into the desired LUNs, following instructions in "Binding Disk Modules" in Chapter 2.

   Tip: Use a minibind in this case: *ssmcli bind* with the **-z** option. For information on the **-z** option, see page 146 in Chapter 6.

   When you bind the LUNs, the SP copies the licensed internal code from the 00 disk module onto the other two database disk modules (01 and 02).

Changing Caching Parameters

This section contains the following topics:

- "Enabling or Disabling Array Caching" on page 73
- "Changing Cache Page Size" on page 75
- "Changing Write Cache Watermarks" on page 76

Enabling or Disabling Array Caching

At power-on, an array enables the read and write caches on each SP if the required hardware is working and the read and write cache partitions have nonzero sizes. Array read and write caching is enabled when the SP read and write caches are enabled. (For more information on caching, see "Caching" on page 33 in Chapter 2.)

You can disable or enable array read or write caching without affecting the information stored on the LUNs:

- Enable the array read or write caching for the array to use caching for LUNs with their read or write caches enabled.

- Disable the array read and write caches before changing the memory partitions and most of the LUN caching parameters.

- Disable the array write caching before an SP is replaced.

You disable or enable array write caching by setting the state of each SP's write cache to Disable or Enable, respectively.

Note: Write cache state on one SP is independent of that on the other SP, so you must set the state of each cache separately.

Disabling or Enabling Array Write Caching

To disable or enable array write caching, open the Array Configuration window for the arrays whose write caching you want to disable or enable. (Figure 3-3 shows this window.)

Use the Array Configuration window as follows:

- To disable write caching, click the write cache disable button near the left end of the array toolbar, or choose Array > Write Cache State > Disable.

- To enable write caching, click the write cache enable button, or choose Array > Write Cache State > Enable.

To determine whether write caching is enabled or disabled, look at the Write Cache State field in the SP's Cache Information window; see "SP Cache Information" on page 109 in Chapter 4.

It may take the array a while to disable write caching if the SPs need to write data in the write cache to disk. As a result, the Write Cache State field in the SP's Cache Information window may continue to show `Enabled`. Poll the array every few seconds to get the latest status: click the manual poll button (toward the right in the array toolbar, as shown in Figure 2-8) or choose Poll from the Array menu.

Disabling and Enabling Read Caching for an SP

To disable read caching for an SP, such as SP A, click the SP A disable read cache button near the left end of the array toolbar, or choose Array > Read Cache State > SP A > Disable.

To disable read caching for an SP, such as SP A, click the SP A enable read cache button, or choose Array > Read Cache State > SP A > Enable.

To determine whether write caching is enabled or disabled, look at the Read Cache State field in the SP's Cache Information window; see "SP Cache Information" on page 109 in Chapter 4. If desired, poll the array every few seconds to get the latest status: click the manual poll button (toward the right in the array toolbar, as shown in Figure 2-8) or choose Poll from the Array menu.

Changing Cache Page Size

The cache is managed by pages instead of sectors. Cache page size specifies the number of KB storage in one cache page. Available page size values are 2, 4, 8, or 16 KB; the default is 2 KB.

The larger the page size, the more continuous sectors the cache stores in a single page. Generally, page size should be 8 KB for general fileserver applications and 2 or 4 KB for database applications.

You can determine page cache size by looking at the either SP's Cache Information window; see "SP Cache Information" on page 109 in Chapter 4.

To change cache page size, follow these steps:

1. Select the arrays whose cache page size you want to set or change.

2. Disable array write caching for the SPs in the array, as explained in "Disabling or Enabling Array Write Caching" on page 74.

3. Disable read caching for each SP in the array, as explained in "Disabling and Enabling Read Caching for an SP" on page 75.

4. In the Array Configuration window, choose Set Page Size from the Array menu.

5. In the menu that appears, select the value for the page size.

6. Re-enable array write and read caching.

Changing Write Cache Watermarks

The write cache high and low watermarks determine when the SPs start and stop flushing their write caches, respectively. When an SP flushes its write cache, it writes its dirty pages to disk. A dirty page is a write cache page with modified data that has not been written to disk.

The high watermark is the percentage of dirty pages in the write cache. When this percentage is reached, the SPs begin flushing their write cache. The default value is 96 percent. If you specify a lower value, the SPs start flushing the write cache sooner.

The low watermark is the percentage of dirty pages in the write cache. This value determines when write cache flushing stops. The default value is 80 percent. When the low watermark is reached during a flush operation, the SPs stop flushing the write cache.

Note the following:

* Available watermark values are 0% through 100%; default values are 96% for high watermark and 80% for low watermark.

* The high watermark cannot be a smaller percentage than the low watermark.

* To turn off watermark processing, set both the low and high watermarks to 100%.

To change the watermarks, follow these steps:

1. Check that all read and write caching for the array is disabled. See "Enabling or Disabling Array Caching" on page 73.

2. To change the high watermark:

    ■   In the Array Configuration menu, choose Array > Set Watermark > High Watermark.

    ■   In the window that opens, enter the high watermark value you want.

3. To change the low watermark:

    ■   In the Array Configuration menu, choose Array > Set Watermark > Low Watermark.

    ■   In the window that opens, enter the high watermark value you want.

Changing Prefetch (Read-Ahead Caching) Parameters

Prefetching (read-ahead caching) allows the SP to anticipate the data that an application will request so that the SP can read the data into its read cache before the data is needed. The SP monitors I/O requests to each LUN for sequential reads; if it finds any, it prefetches data automatically from the LUN. You can define a specific type of prefetch operation by specifying the values of the prefetch parameters.

You set prefetch parameters in the Prefetch window; see Figure 3-4.
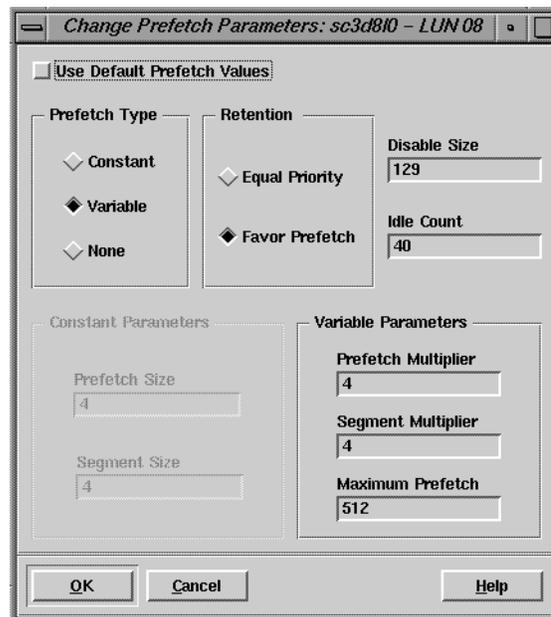
Figure 3-4   Prefetch Window

Prefetch parameters are as follows:

Prefetch type   Determines whether to prefetch data of a variable length (Variable, the default) or a constant length (Constant), or to disable prefetching (None). Your selection determines whether the Constant Parameters, Variable Parameters, or no parameters are accessible.

77

Prefetching works best when the host issues many large sequential read requests to a physical unit. The SP monitors I/O for sequential reads; if it finds any, it prefetches the data automatically.If the amount of data to be read varies from request to request, specify variable prefetch type (the default). If the amount of data read is fairly constant, specify Constant.

Retention   Determines whether prefetched data has equal or favored priority over host-requested data when the read cache becomes full. Specify Favor Prefetch for most applications.

Disable size   Determines when a read request is so large that prefetching data would not be beneficial; for example, if the amount of requested data is equal to or greater than the size of the read cache. Recommended sizes: 129 (the default), 257, 513, 1025, 2049.

A prefetch operation is not performed for a read request if the amount of data requested is equal to or greater than the disable size.

Idle count   Amount of time in 100-ms units that a LUN must be below the idle threshold in order to be considered idle. For example, 40 equals 4000 ms. Once a LUN is considered idle, any dirty pages in the cache can begin idle time flushing. Determine the idle threshold with *ssmcli getcontrol* (see "getcontrol" on page 159 in Chapter 6.

Leave this parameter set to its default value of 40 (or set it higher).

Constant Parameters:
Prefetch Size: the number of blocks to prefetch from the LUN during a single read operation.

Segment Size: number of blocks of data to prefetch for each read request. An SP reads one segment at a time from the LUN because smaller prefetch requests interfere less with other host requests.

Variable Parameters:
Prefetch Multiplier: the amount of data to prefetch relative to the amount of data requested. For example, if the prefetch size is 8, the amount of data to prefetch is 8 times the amount of data requested.

Segment Multiplier: determines the amount of data to retrieve from the LUN in a single prefetch operation. Specify a factor equal to or less than the prefetch multiplier. If the segment multiple is equal to the prefetch multiplier, prefetch operations are not divided into segments. The

amount of data prefetched for each read request equals or is less than the number of blocks of data requested multiplied by the prefetch multiplier.

Maximum Prefetch: maximum number of blocks to prefetch for variable-length prefetching.

Table 3-1 summarizes values and ranges for prefetch parameters.

Table 3-1   Prefetch Parameter Values and Ranges

Parameter Type	Parameters	Valid Values
General	Prefetch Type	Constant Variable (default) None
	Retention	Equal Priority Favor Prefetch (default)
	Disable Size	0 to 4097 sectors; recommended sizes: 129 (default), 257, 513, 1025, 2049
	Idle Count	1 to 100; recommended: 40 (default) or higher
Constant	Prefetch Size	0 to 2048 blocks (default: 4)
	Segment Size	0 to 254 blocks (default: 4)
Variable	Prefetch Multiplier	0 to 32 (default: 4)
	Segment Multiplier	0 to 32 (default: 4): must be equal to or less than the prefetch multiplier
	Maximum Prefetch	1 to 1024 blocks (default: 512)

Use the default values unless you are certain that the applications that access the LUN will benefit from different values. To change prefetch parameters, follow these steps:

1. Display the Array Configuration window for the array with the LUN whose parameters you want to set.

2. In the LUN selection area of the Array Configuration window, select the LUN whose parameters you want to change.

3. Select LUN > Prefetch > Change Prefetch Parameters. The Change Prefetch Parameters window appears; Figure 3-4 shows an example.

4. Change prefetch values according to the needs of the applications that access the LUN. Change them from the defaults only if you are certain that the applications will benefit from different values.

   Do not change the idle count; leave it at the default of 40 (or set it higher).

5. When the parameters are the way you want them, click *OK*. In the confirmation window that appears, click *Yes*.

You can also change individual prefetch parameters using options in the Prefetch submenu of the LUN menu.

To reset all prefetch parameters to the defaults, choose LUN > Prefetch > Set Default Prefetch Values, or click *Use Default Prefetch Values* at the upper left corner (see Figure 3-4).

Note: The LUN cannot use read caching with the prefetch parameters you specify until you enable the SP's read cache, as explained in "Disabling and Enabling Read Caching for an SP" on page 75.

Using Storage System Statistics for Cache Optimizing

You can use a LUN's cache information window to determine how caching and prefetching are affecting LUN performance. The LUN Cache Information window provides the following information:

- the hit ratios for the read and write caches, which can tell you if caching is helping LUN performance

- number of prefetched blocks and unused prefetched blocks, which tells you if prefetching is efficient for this LUN

- number of forced flushes, which suspend other I/O

To display the LUN Cache Information window, see "Using the LUN Information Windows" on page 94 in Chapter 4. To interpret the caching information, see "LUN Cache Information" on page 98 in Chapter 4.

Upgrading an Array for Caching

To upgrade an array to support caching, necessary hardware must be installed and you must set up array caching. Hardware requirements for caching are as follows:

- Read caching requires a storage processor (SP) with at least 16 MB of read cache memory.

- Write caching requires a Fibre Channel RAID enclosure with

  - a link controller card (LCC) for each SP in the Fibre Channel RAID enclosure

  - two functional power supplies in the Fibre Channel RAID enclosure

  - disk modules in slots 00 through 08 (no failed disk modules)

  - a fully functional (charged, plugged in, and connected standby power supply with a fully charged battery

  - two SPs, each with at least 16 MB of write memory

    Each SP must have the same number of DIMMs; all DIMMs for the read and write memory module locations must be the same size.

    Note: Read cache DIMMs are physically separate from write cache DIMMs.

    The amount of write cache you can allocate depends on the SP part number; see Table 3-2 and Table 3-3.

Table 3-2   Cache Capability for SP 9470198

Cache Page Size	User-Configurable Write Cache	User-Configurable Read Cache	Cache Configuration
2 KB	14 MB 0 7 MB	0 21 MB 7 MB	100% write cache 100% read cache 50% write/50% read
4 KB	28 MB 0 14 MB	0 43 MB 14 MB	100% write cache 100% read cache 50% write/50% read
8 KB	56 MB 0 28 MB	0 86 MB 28 MB	100% write cache 100% read cache 50% write/50% read
16 KB	113 MB 0 56 MB	0 120 MB 56 MB	100% write cache 100% read cache 50% write/50% read

Table 3-3   Cache Capability for SP 9470207

Cache Page Size	User-Configurable Write Cache	User-Configurable Read Cache	Cache Configuration
2 KB	194 MB 0 106 MB	0 248 MB 106 MB	100% write cache 100% read cache 50% write/50% read
4 KB	388 MB 0 213 MB	0 497 MB 213 MB	100% write cache 100% read cache 50% write/50% read
8 KB	504 MB 0 426 MB	0 504 MB 426 MB	100% write cache 100% read cache 50% write/50% read
16 KB	504 MB 0 504 MB	0 504 MB 504 MB	100% write cache 100% read cache 50% write/50% read

Necessary hardware is installed by a qualified Silicon Graphics System Support Engineer; contact your authorized service provider.

To set up caching after the required hardware is installed, enable array write and read caching for the desired SPs, as explained in "Changing LUN Bind Parameters That Do Not Require Unbinding" on page 65. Enable write caching for the LUNs that you want to use write caching; see page 55 in "Binding Disk Modules" in Chapter 2.

Monitoring Arrays and Displaying System Statistics

This chapter describes how to display information windows that provide status and statistics information about an array, SP, LUN, disk module, or caching. It also explains the role of polling and statistics logging and describes how to turn on and off the statistics log that supplies the information for these windows and how to display and save the event log for an SP.

This chapter consists of the following sections:

Using ssmagent Polling

Before you use the RAID GUI to monitor the health of managed arrays, you need to make sure that it has up-to-date information on them. *ssmgui* updates its information for each array by polling the *ssmagent* on the managed servers for changes in the state of the disks, SPs, LUNs, and read and write cache.

You can automatically poll one or more arrays at specified intervals that can you can set for each array, or you can manually poll selected arrays whenever you want. Automatic and manual polling are completely independent procedures; that is, one does not affect the other.

This section contains the following topics:

- "ssmagent Polling Interval and Polling Requests" on page 84
- "Automatic Polling" on page 85
- "Disabling and Enabling Automatic Polling" on page 85
- "Changing the Polling Interval" on page 86

ssmagent Polling Interval and Polling Requests

The *ssmagent* has a polling interval that you can specify in its configuration file (see its man page, ssmagent(7). This polling interval is the minimum interval at which the *ssmagent* polls an array on the server on which the *ssmagent* is running. Each time *ssmagent* polls an array, it updates its information for the array.

Regardless of how frequently a client application, such as *ssmgui*, polls the agent for information about an array, the *ssmagent* actually polls the array only in response to the first client poll of the array after the polling interval has elapsed. As a result, the *ssmagent* polling interval can prevent client applications from overwhelming the *ssmagent* with excessive poll requests.

Because the *ssmagent* does not always poll an array every time a client application requests a poll, the client application represents only the information that the *ssmagent* currently has for the array. If the information for an array has changed, but the polling interval did not elapse at the time of the poll request, the *ssmagent* does not poll the array. As a result, it cannot notify the client application of the change.

For example, suppose the *ssmagent* polling interval is 60 seconds. If a client application sends the *ssmagent* a request to poll an array at 6:00:00. The *ssmagent* polls the array, and notifies the client application of any change in the array. In this situation, the client application reflects the current state of the array after the poll request. The *ssmagent* does not poll the array again until at least 6:01:00. If a client application requests a poll of the array between 6:00:00 and 6:01:00, the application reflects only the state of the array at 6:00:00. If a disk module in the array fails at 6:00:25, no client applications that request a poll of the array between 6:00:25 and 6:01:00 are notified of the disk module failure. In response to the first client application to poll the array after 6:00:10, the *ssmagent* polls the array, updates its information on the array, and notifies the request from a client application of the disk module failure.

Automatic Polling

You can automatically poll an array at a specified intervals, or you can manually poll selected arrays whenever you want. Automatic and manual polling are completely independent and do not affect each other.

By default, automatic polling is disabled for the *ssmgui* session and for each array because the polling process uses array resources, thus affecting performance. You enable automatic polling for a session by specifying an automatic polling interval, and for an array, by specifying an automatic polling priority for the array.

When automatic polling is enabled for the *ssmgui* session and a selected array, the frequency at which the *ssmagent* automatically polls the array equals the automatic polling interval multiplied by the automatic polling priority for the array. For example, if the automatic polling interval is 5 minutes and the automatic polling priority for the array is 3, *ssmgui* polls *ssmagent* for array information every 15 minutes.

Disabling and Enabling Automatic Polling

Follow these instructions to enable or disable polling:

• To enable automatic polling for an *ssmgui* session, choose Automatic Polling Interval from the Options menu of the Storage System Manager window. Choose Enable from the submenu that opens.

85

- To enable automatic polling for one or more arrays, select the arrays in the Storage System Manager window. Choose Automatic Polling Interval from the Options menu of the Storage System Manager window. Choose Enable from the submenu that opens.

- To disable automatic polling for a session, choose Automatic Polling Interval from the Options menu of the Storage System Manager window. Choose Disable from the submenu that opens.

- To disable automatic polling for one or more arrays, select the arrays in the Storage System Manager window. Choose Automatic Polling Interval from the Options menu of the Storage System Manager window. Choose Disable from the submenu that opens.

Changing the Polling Interval

Follow these instructions to change the polling interval:

- To change the automatic polling interval, choose Automatic Polling Interval from the Options menu of the Storage System Manager window. In the submenu that opens, select the desired number of seconds or minutes, or choose Other to specify a number of minutes that is not listed.

- To specify the automatic polling priority for one or more arrays, select the arrays whose priorities you want to change. In the Storage System Manager window, choose Automatic Polling Interval from the Options menu; select the desired priority from the submenu that opens.

Because *ssmgui* gets changes in the status of any array from *ssmagent* running on the array's server, *ssmgui* cannot get this information any faster than *ssmagent*. Thus, it is useless to set the polling interval for *ssmgui* to less than that for *ssmagent*. The default polling interval for *ssmagent* is 60 seconds.

Using Statistics Logging

Many information windows explained in this chapter display statistics information. Each SP maintains a log of statistics for the LUNs, disk modules, and array caching. Normally, statistics information is not logged, since logging it may affect performance. For meaningful statistics information, you must enable statistics logging as explained in this chapter.

The statistics log uses a 32-bit counter to maintain the statistics numbers. When the counter is full, the statistics numbers restarts at zero. As a result, you see a sudden decrease in a statistics number if you view it shortly before the counter is full and shortly after the counter restarts at zero. If you want to keep the log turned on for more than two weeks, reset the log about every two weeks, so you know when the numbers start at zero.

Follow these steps to turn statistics logging on or off:

1. In the Storage Management window, select the arrays whose logging state you might want to change.

2. Click the configure button near the left end of the array toolbar, or choose Configure from the Array menu.

   An Array Configuration window opens for each array selected.

3. In the Array Configuration window, choose Statistics Logging State from the Array menu.

4. Select the SP whose statistics logging state you want to change.

5. In the confirmation window that opens, click *Yes* to change the setting or *No* to leave it unchanged.

Getting LUN and Disk Information in the Array Configuration Window

The Array Configuration window shows the LUNs in an array and their ownership: each LUN appears in the field of the SP that owns it, as shown in Figure 4-1.

You can display the location of disks in an array enclosure. Click the LUN to select it; a colored outline surrounds the LUN. In the Disk Field below the LUN selection area, the disks in the LUN are highlighted in the same color in the enclosure in which they are located. Each LUN is outlined in a different color in the display (seeFigure 4-1).
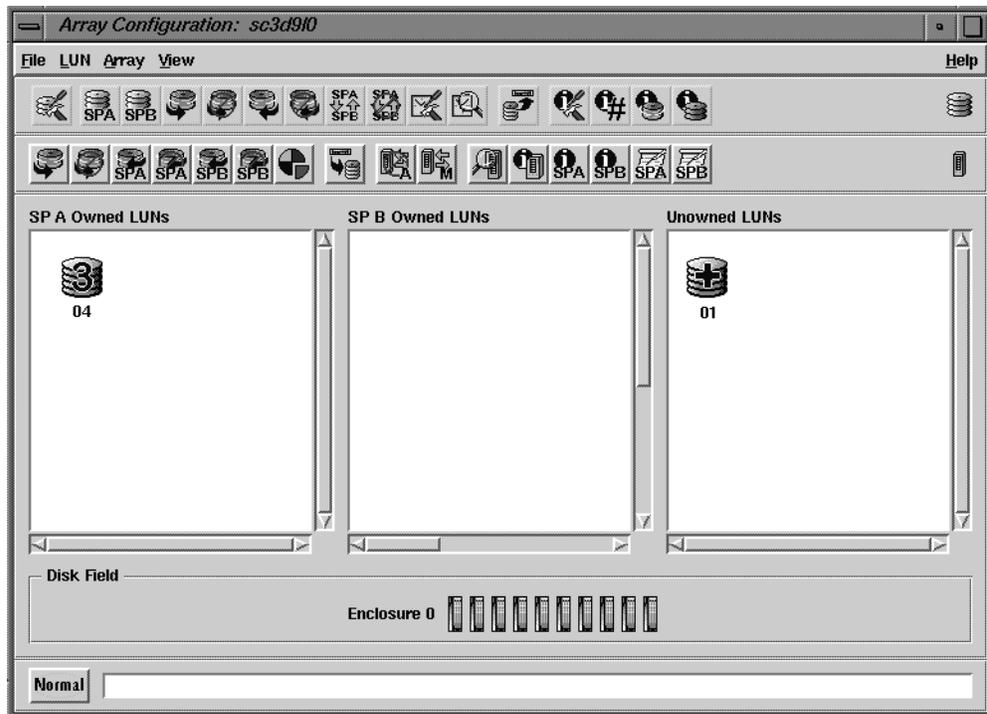
Figure 4-1   Array Configuration Window: LUN and Disk Location Information

This section contains the following topics:

- "LUN Icons and IDs" on page 89
- "Disk Field" on page 90
- "LUN and Disk Module Icon Color" on page 92

LUN Icons and IDs

LUN icons indicate the LUN type, as summarized in Table 4-1.

Table 4-1   LUN Icons and Types

LUN Icon	LUN Type
	RAID 5
	RAID 3
	RAID 1
	RAID 1/0
	RAID 0
	Individual disk
	Hot spare
	Faulted LUN

LUN IDs are two-digit numbers. To display them, choose Show LUN IDs from the
View menu.

Disk Field

The disk field near the bottom of the Array Configuration window shows the disk modules in the array enclosures. It can also show LUN ownership of the disks, and disk module IDs. A letter E in the disk slot indicates an empty slot.

LUN Ownership of Disk Modules

To determine the location of disk modules in a LUN, click the LUN to select it; a colored outline surrounds the LUN.

In the disk field, the disks in the LUN are highlighted in the same color in the enclosure in which they are located. Each LUN is outlined in a different color in the display; Figure 4-2 shows an example.
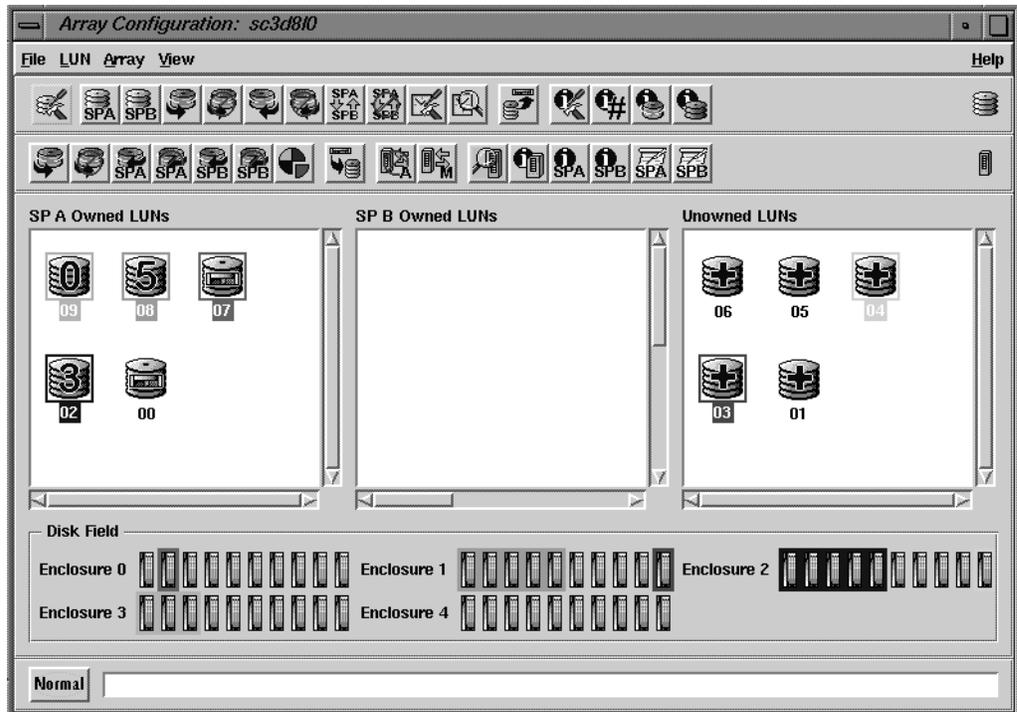
Figure 4-2   Array Configuration Window: LUN Ownership of Disk Modules

Disk Module IDs

To display disk module IDs, in the Array Configuration window, choose Show Disk IDs from the View menu. Figure 4-3 shows an example.
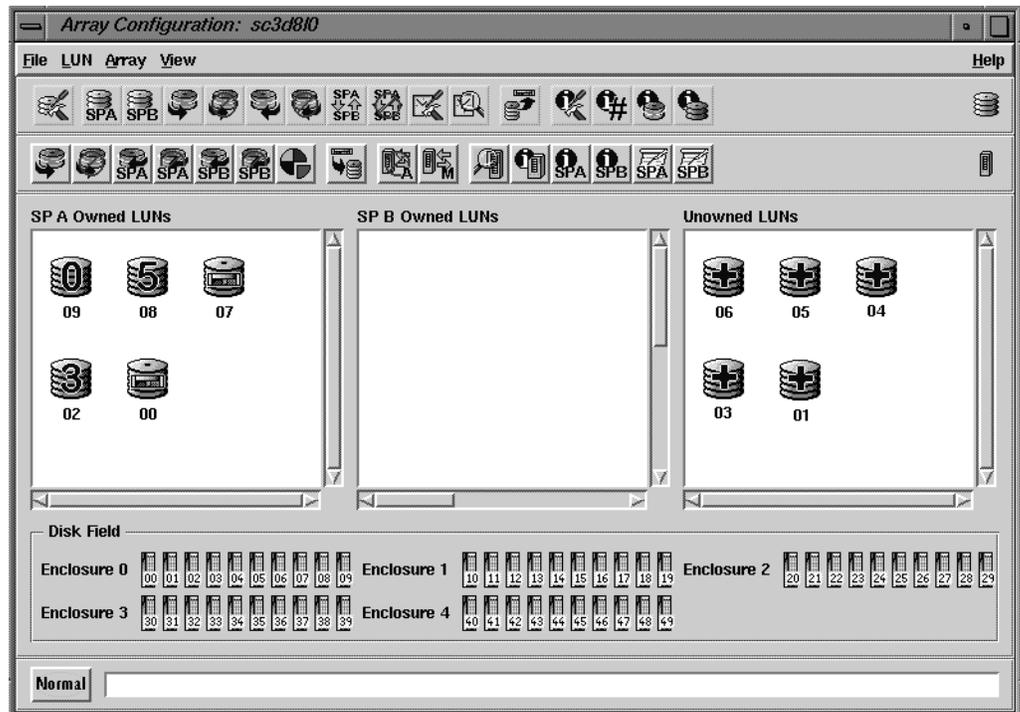
Figure 4-3   Array Configuration Window: Disk Module IDs

The disk module ID is a two-digit decimal number: the first digit is the enclosure number multiplied by 10, and the second digit is the number of the slot containing the disk module. For example, the disk module ID for the disk module in slot 3 of enclosure 2 is 23.

LUN and Disk Module Icon Color

LUN and disk module icon color indicates its status:

Gray indicates that the LUN or disk is operating normally.

- Blue indicates a transition state, such as being bound or incorporating a hot spare (LUNs) or being bound or powered on (disk modules). A small letter T in the icon also indicates a transitional state.

- Amber indicates a fault condition, such as a failed array component or a breakdown in communication (LUN) or a failure (disk module). A small letter F in the icon also indicates a fault condition.

To get more information on arrays, LUNs, disks, and SPs, you can launch other windows, as explained in the rest of this chapter.

Using the Array Information Window

The Array Information window displays information on the selected array; Figure 4-4 shows an example.

State	Faulted
Automatic Polling Priority	1
Automatic Polling Count	1
Automatic Disk Formatting	Disabled
Disk Write Caching	Disabled
RAID3 Write Buffering	Disabled
Host Connection	machinename.domain, sc3d810, SP A
Host Connection	machinename.domain, sc5d710, SP B
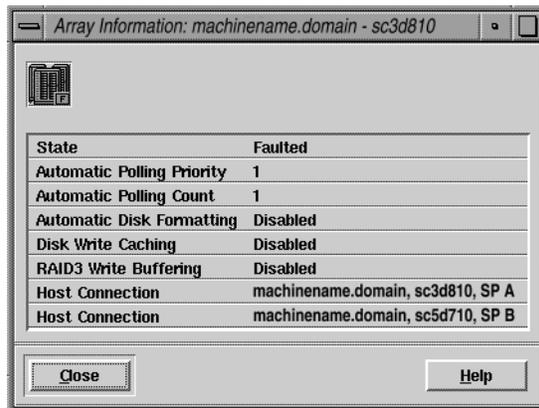
Figure 4-4   Array Information Window

To display the Array Information window, follow these steps:

1. In the Storage System Manager window, select the arrays whose status you want to view.

2. Click the array information button near the middle of the array toolbar, or choose Information from the Array menu.

Use the fields in the Array Information window as follows:

*   State: Faulted or Normal.

*   Automatic Polling Priority: When automatic polling is enabled, the automatic polling priority and polling interval determine how often the software automatically polls the array. If the automatic polling priority is 1, *ssmgui* polls the selected array each time the polling interval elapses. If the priority is 2, *ssmgui* polls the selected arrays after two intervals elapse; if the priority is 3, *ssmgui* polls the selected array after three intervals elapse; and so on. The automatic polling interval has no effect on manual polling.

    When *ssmgui* polls the selected arrays for status, it retrieves information about array status changes, and updates the status information for the polled arrays in any open window. An automatic poll of an array functions the same way as a manual poll of the array.

*   Automatic Polling Count: Number of priority cycles that elapsed since this array was polled. For example, if the priority is 3, a count of 2 indicates that 1 priority cycle has elapsed; on the next (third) cycle the array will be automatically polled.

*   Automatic Disk Formatting: Indicates whether the array firmware automatically formats any disk module that has an unrecognizable format. This setting cannot be changed.

*   Disk Write Caching, RAID3 Write Buffering: Indicates whether these are disabled or enabled.

*   Host Connection: The hostname followed by the operating system device entry and the SP that communicates with *ssmgui*. The device entry is from the *ssmagent* configuration file on the array's server. The SP is the default owner of the LUN. With two SPs, two host connections are required to provide information on each SP.

Using the LUN Information Windows

The LUN Information window displays either configuration, statistics, or cache information. When you open the window, it displays LUN configuration information, as shown in Figure 4-5.
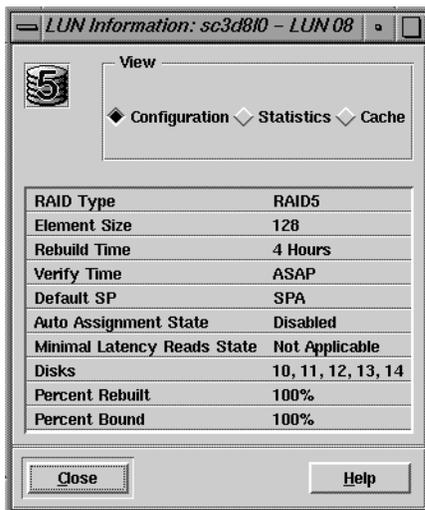
Figure 4-5   LUN Information Window: Configuration Information

To display the LUN Information window, follow these steps:

1.   In the Storage System Manager window, select the arrays whose LUN status you want to view.

2.   Click the configure button near the left end of the array toolbar, or choose Configuration from the Array menu. An Array Configuration window opens for each selected array.

3.   In the Array Configuration window, double-click the LUN you want to examine. Alternatively, select multiple LUNs and choose LUN > Display Information > Configuration. The LUN Information window opens, displaying LUN configuration information, as shown in Figure 4-5.

This section contains the following topics:

LUN Configuration Information

To display LUN configuration information, click the leftmost radio button in the View area of the LUN Information window, if necessary. This information is displayed when you first open the LUN Information window; see Figure 4-5 for an example.

The fields have the following meanings:

- RAID Type: Indicates the LUN 's RAID type, as summarized in Table 4-2.

Table 4-2   LUN Configuration Information: RAID Types

RAID Type	Meaning
RAID 5	Individual access array
RAID 3	Parallel access array
RAID 1	Mirrored pair
RAID 1/0	Mirrored RAID 0 group
RAID 0	Nonredundant individual access array
DISK	Individual disk module
HOT SPARE	Hot spare

RAID type is set when the LUN is configured. To change it, see "Changing LUN RAID Type" on page 71 in Chapter 3.

- Element Size: Number of disk sectors (stripe element size) that the array can read or write to a single disk module without requiring access to another disk module (assuming that the transfer starts at the first sector in the stripe). The stripe element size can affect the performance of a RAID 5 or RAID 1/0 LUN. A RAID 3 LUN has a fixed stripe element size of one sector. The smaller the stripe element size, the more efficient the distribution of data read or written. However, if the stripe size is too small for a single I/O operation, the operation requires access to two stripes, which causes the hardware to read and/or write from two disk modules instead of one.

  This number is set when the LUN is configured.

- Rebuild Time: Time that the array allots to reconstruct the data on either a hot spare or a new disk module that replaces a failed disk module in a LUN. The rebuild time applies to all RAID LUNs except RAID 0. The time you specify determines the amount of resource the SP devotes to rebuilding instead of to normal I/O activity.

  This number is set when the LUN is configured; to change it, see "Changing LUN Bind Parameters That Do Not Require Unbinding" on page 65 in Chapter 3.

- Verify Time: Time that the array allots to checking parity. If an SP detects parity inconsistencies, it starts a background process to check all the parity sectors in the LUN. The time you specify determines the amount of resource the SP devotes to verifying instead of to normal I/O activity.

  This number is set when the LUN is configured. To change it, see "Changing LUN Bind Parameters That Do Not Require Unbinding" on page 65 in Chapter 3.

- Default SP: The SP that owned the LUN when the array was powered on. To change the default SP, see "Changing LUN Bind Parameters That Do Not Require Unbinding" on page 65 in Chapter 3.

- Auto Assignment State: Enabled or Disabled; the original default is Disabled. Auto assign controls the ownership of the LUN when one SP fails in a array with two SPs. With auto assign enabled, if the SP that owns a LUN fails and the server tries to access that LUN through the second SP, the second SP assumes ownership of the LUN so the access can occur. The second SP continues to own the LUN until the SP's power is turned off and on again, at which point ownership of each LUN returns to its default SP. If auto assign is disabled in an array with two SPs, the other SP does not assume ownership of the LUN, so the access to the LUN does not occur.

  Auto assign is set when the LUN is configured. To change it, see "Changing LUN Bind Parameters That Do Not Require Unbinding" on page 65 in Chapter 3.

- Minimal Latency Reads State: Disabled or Enabled; applies to RAID 3 LUNs only. For information, see page 48 in Chapter 2. To change the state, see "Changing LUN Bind Parameters That Do Not Require Unbinding" on page 65 in Chapter 3.

- Disks: The disk modules that make up the LUN. For example, `00 - 04` means disk modules in slots 0 through 4 in enclosure 0.

- Percent Rebuilt: If nonzero, this number indicates that a disk in the LUN is being rebuilt; the number shows the percentage of the rebuild completed.

- Percent Bound: If nonzero, this number indicates that a disk in the LUN is being bound; the number shows the percentage of the bind completed.

LUN Statistics Information

To display LUN statistics information, click its radio button in the View area of the LUN Information window. Figure 4-6 shows an example.
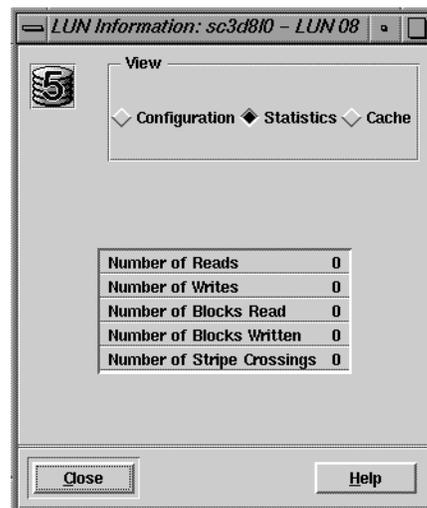
Figure 4-6   LUN Information Window: Statistics Information

Note: Statistics logging must be enabled for entries in this window to be viewed.

The fields have the following meanings:

- Number of Reads, Number of Writes: Total number of read and write requests made to the LUN.

- Number of Blocks Read, Number of Blocks Written: Total number of data blocks read from and written to the LUN.

- Number of Stripe Crossings: Number of times that a read or write crosses a stripe boundary on any disk module in a RAID 5, RAID 1/0, or RAID 0 LUN. Generally, stripe crossings are undesirable because each one requires an additional I/O. The ideal stripe element size is the smallest size that does not cause an additional I/O to another disk module. From the number of crossings, you can determine the percentage of I/Os that required a stripe boundary crossing: add the number of reads to the number of writes, and divide the number of crossings by this sum. A relatively low percentage indicates a relatively efficient stripe element size.

LUN Cache Information

To display LUN cache information, click its radio button in the View area of the LUN Information window. Figure 4-7 shows an example.
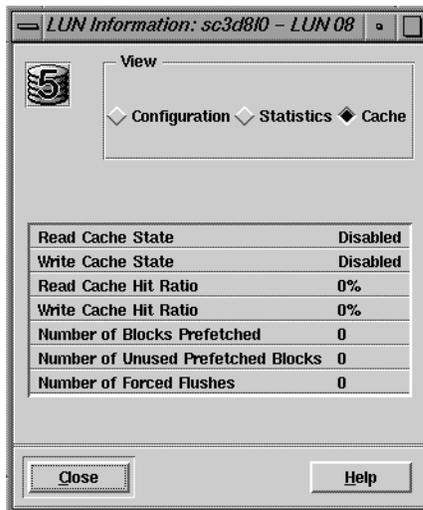
Figure 4-7   LUN Information Window: Cache Information

The fields have the following meanings:

- Read Cache State, Write Cache State: Enabled or Disabled for this LUN. For caching to occur with a LUN, it must be enabled for the array and for the LUN. For information on how to enable an SP's read cache, see "Disabling or Enabling Array Write Caching" on page 74 in Chapter 3. For information on how to enable an array's read cache, see "Enabling or Disabling Array Caching" on page 73 in Chapter 3.

- Read Cache Hit Ratio, Write Cache Hit Ratio: The percentage of read cache and write cache hits for the LUN:

    - A read cache hit occurs when the SP finds a sought page in read cache memory, and thus does not need to read the page from disk.

    - A write cache hit occurs when the SP finds and modifies data in cache memory, which usually saves a write operation. For example, with a RAID 5 LUN, a write hit eliminates the need to read, modify, and write the data.

    High hit ratios are desirable because each hit indicates at least one disk access that was not needed. You may want to compare the read and write hit ratios for the LUN with the read and write hit ratio for the entire array in an SP Cache window (see "SP Cache Information" on page 109).

    For a LUN to have the best performance, the hit ratios should be higher than those for the array. A very low read or write hit ratio for a busy LUN may mean that caching is not helping the LUN's performance.

- Number of Blocks Prefetched: Number of disk blocks (512 bytes each) prefetched for the LUN. This entry and the next indicate how well (how often) prefetching is working for this LUN.

- Number of Unused Prefetched Blocks: Number of disk blocks that were prefetched but not used. A higher number might mean that prefetching is not an efficient choice for this LUN.

- Number of Forced Flushes: Number of times the write cache was flushed because the cache filled to its high water mark. A high number is not desirable because a forced flush suspends all other I/O.

Using the Disk Information Windows

The Disk Information window displays either configuration, statistics, or error information. When you open the window, it displays disk module configuration information, as shown in Figure 4-8.
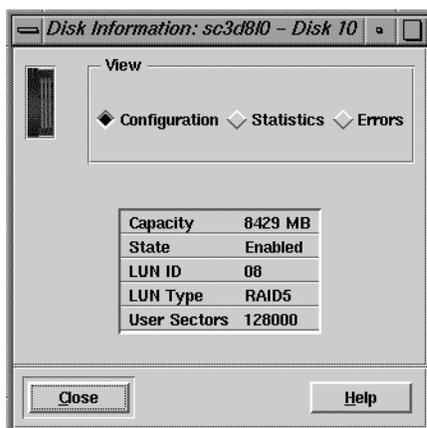
Figure 4-8   Disk Information Window: Configuration Information

To display the Disk Information window, follow these steps:

1. In the Storage System Manager window, select the arrays whose disk module status you want to view.

2. Click the configure button near the left end of the array toolbar, or choose Configuration from the Array menu. An Array Configuration window opens for each selected array.

3. In the disk field near the bottom of the window, double-click the disk module you want to examine. The Disk Information window opens, displaying disk module configuration information, as shown in Figure 4-8.

This section contains the following topics:

*   "Disk Module Configuration Information" on page 101

*   "Disk Module Statistics Information" on page 103

*   "Disk Module Error Information" on page 104

Disk Module Configuration Information

To display disk module configuration information, click the leftmost radio button in the View area of the Disk Information window, if necessary. This information is displayed when you first open the Disk Information window; see Figure 4-8 for an example.

The fields have the following meanings:

• Capacity: Storage capacity of the disk module in MB.

• State: Operational state of the disk module, as summarized in Table 4-3.

Table 4-3   Disk Module Configuration Information: State

Disk Module State	Meaning
Binding	Being bound into a LUN.
Enabled	Either a hot spare on standby or part of a bound LUN that is assigned to (owned by) the SP you are using as the communication channel to the enclosure. If the array has another SP, this module's status is Ready when you use the other SP as the communication channel to the array.
Equalizing	Data from a hot spare is being copied onto a replacement disk module.
Failed	Powered off or inaccessible.
Not Present	No disk module is in the slot.
Off	Powered off by the SP, which can happen if a disk module of the wrong capacity is inserted.
Powering Up	Power is being applied to the disk module.
Ready	Disk module is part of a broken LUN or a LUN that is bound and unassigned. This state can mean that the disk module is part of a LUN that is not owned by the SP that you are using as the communication channel to the enclosure. If the disk module is part of a LUN assigned to an SP other than the one you are using as the communication enclosure, the module's status is either Enabled or Ready. It is Enabled when you use the other SP as the communication channel to the array.

Table 4-3 (continued)   Disk Module Configuration Information: State

Disk Module State	Meaning
Rebuilding	Disk module is either a hot spare or a replacement disk module to replace a failed module in a LUN. The data is being rebuilt on a hot spare or a replacement disk module.
Removed	Removed from the enclosure; applies only to a disk module that is part of a LUN.
Standby	Disk module is either a hot spare or a replacement disk module to replace a failed module in a LUN. The data is being rebuilt on a hot spare or a replacement disk module.
Synchronizing	The write cache on the disk module is being synchronized (during a rebuild).
Unbound	Ready to be bound into a LUN.

- LUN ID: Hexadecimal number identifying the LUN to which the disk module is bound. This number is specified when the LUN is bound; see page 53 in Chapter 2 for more information on the LUN ID.

- LUN Type: Indicates the LUN 's RAID type, as summarized in Table 4-2 on page 95.

- User Sectors: Number of user-accessible sectors (512-byte blocks) on the disk module.

Disk Module Statistics Information

To display disk module statistics information, click its radio button in the View area of the Disk Information window. Figure 4-9 shows an example.
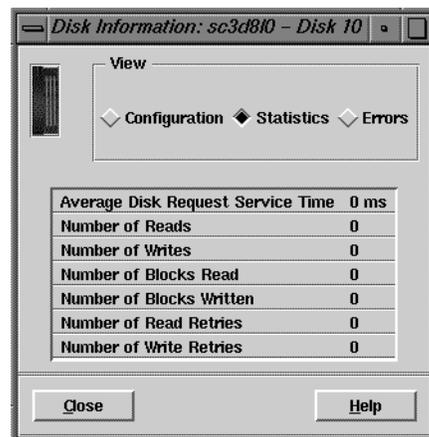
Figure 4-9   Disk Information Window: Statistics Information

Note: Statistics logging must be enabled for entries in this window to be viewed.

The fields have the following meanings:

- Average Disk Request Service Time: Average time in milliseconds that the disk module required to execute an I/O request after the request reached the top of the queue.

- Number of Reads, Number of Writes: Total number of read and write requests made to the disk module. The LUN read or write information displayed in the LUN Statistics window ("LUN Statistics Information" on page 97) can be more useful because it is for the entire LUN, and not just for one of the disk modules in the LUN.

- Number of Blocks Read, Number of Blocks Written: Number of data blocks read from and written to the disk module.

- Number of Read Retries, Number of Write Retries: Number of times read and write requests to the disk module were retried because of soft or hard errors. See "Disk Module Error Information."

Disk Module Error Information

To display disk module error information, click its radio button in the View area of the Disk Information window. Figure 4-10 shows an example.
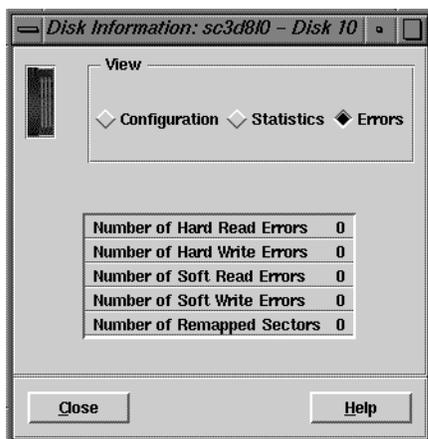
Figure 4-10   Disk Information Window: Error Information

The fields have the following meanings:

- Number of Hard Read Errors, Number of Hard Write Errors: Number of read or write errors for all the disk modules in the LUN that persisted through all the retries. An increasing number of hard errors can mean that one or more of the LUN's disk modules is nearing the end of its useful life.

- Number of Soft Read Errors, Number of Soft Write Errors: Number of read or write errors for all the disk modules in the LUN that disappeared before all the retries. An increasing number of soft errors can indicate that one of the LUN's disk modules is nearing the end of its useful life.

- Remapped Sectors: Number of disk sectors on all the disk modules in the LUN that were faulty when written to, and thus were remapped to different parts of the disk modules.

Using the SP Information Windows

The SP Information window displays either configuration, statistics, or cache information. When you open the window, it displays SP configuration information, as shown in Figure 4-11.
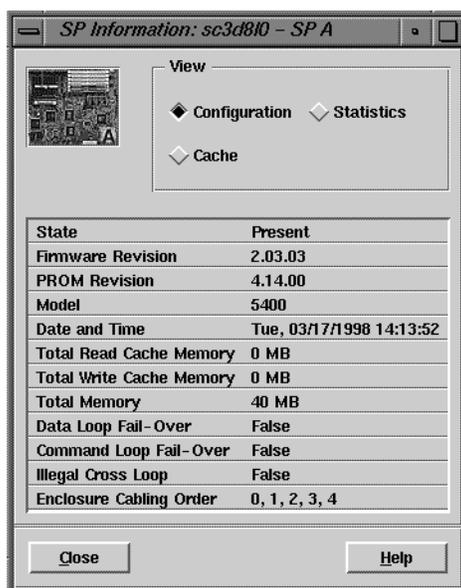
Figure 4-11   SP Information Window: Configuration Information

To display the SP Information window, follow these steps:

1. In the Storage System Manager window, select the arrays whose SP status you want to view.

2. Click the Configure button near the left of the array toolbar, or choose Configuration from the Array menu. An Array Configuration window opens for each selected array.

3. In the Array Configuration window, click the SP A or SP B information button in the array toolbar. Alternatively, choose SP Information from the Array menu, and select the SP for which you want information. The SP Information window opens and displays SP configuration information, as shown in Figure 4-11.

This section contains the following topics:

- "SP Configuration Information" on page 106
- "SP Statistics Information" on page 108
- "SP Cache Information" on page 109

SP Configuration Information

To display SP configuration information, click the top radio button in the View area (upper right) of the SP Information window, if necessary. This information is displayed when you first open the SP Information window; see Figure 4-11 for an example.

The fields have the following meanings:

- State: summarizes possible SP states as summarized in Table 4-4.

Table 4-4   SP Configuration Information: State

SP State	Meaning
Present	This SP is the communication channel you are using to communicate with the array.
Not Connected	The agent cannot talk to the SP because a communication channel specifying the SP is not in the *ssmagent*'s configuration file for the selected host. For example, the SP is connected to a different host from that for the SP in the communications channel for the array.
Not Present	The SP was not present in the enclosure when the *ssmagent* started.
SP Removed	The SP was present in the enclosure when the *ssmagent* started and has since been removed.

- Firmware Revision: Revision level of Licensed Internal Code that the SP is running. Each SP in the array runs the same LIC revision level.

  If the firmware revision or the PROM revision entries are zero or blank, the host connection is missing or invalid.

- PROM Revision: Revision level of SP's PROM code. Each SP in the array runs the same revision level. PROM code is updated automatically when an SP's LIC is updated.

- Model: Product model number for the SP.

- Total Read Cache Memory: Number of megabytes allocated for read caching. This figure can be different from that for a second SP in the enclosure.

- Total Write Cache Memory: Number of megabytes allocated for write caching. This figure is the same as that for a second SP in the enclosure.

- Total Memory: Number of megabytes of the SP memory, which includes memory needed for cache buffers, RAID 3 usage, or the LIC). Each SP in the array must have the same amount of memory to make full use of the memory.

  Note: Total Memory is not the same as user free memory; see "Setting Up Memory Partitions" on page 41 in Chapter 2 for more information.

- Data Loop Fail-Over, Command Loop Fail-Over, Illegal Cross Loop: Displays whether any of these conditions exist.

Enclosure Cabling Order: Displays the order in which the enclosures are cabled.

SP Statistics Information

To display SP statistics information, click its radio button in the View area (upper right) of the SP Information window. Figure 4-12 shows an example.
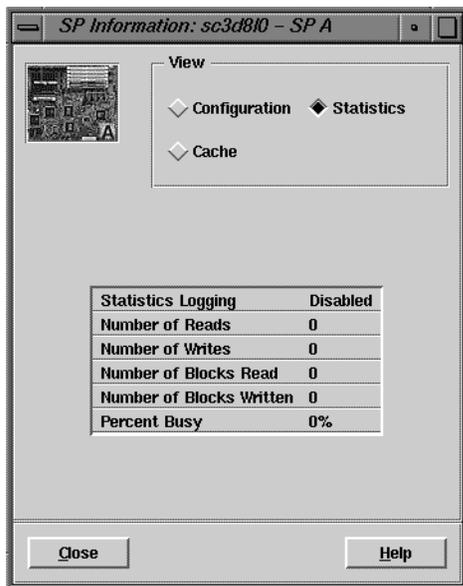
Figure 4-12   SP Information Window: Statistics Information

Note: Statistics logging must be enabled for values in this window to be viewed.

The fields have the following meanings:

- Statistics Logging: Enabled or Disabled, depending on the setting in the Array Configuration window, as explained on page 87 in "Using Statistics Logging."

- Number of Reads, Number of Writes: Total number of read and write requests made to the SP.

- Number of Blocks Read, Number of Blocks Written: Total number of data blocks read from and written to the SP.

- Percent Busy: Percentage of time that the SP was busy processing requests, and not idle. The number shows the relative load on the SP.

SP Cache Information

To display SP cache information, click its radio button in the View area (upper right) of the SP Information window. Figure 4-12 shows an example.
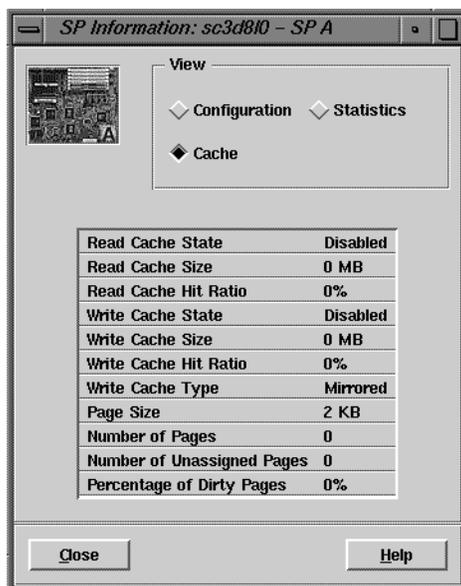
Figure 4-13   SP Information Window: Cache Information

The fields have the following meanings:

- Read Cache State, Write Cache State: Current state of the SP's read or write cache: Read cache states are Enabled, Disabling, and Disabled. Write cache states are Enabled or Disabled, and several transition states, such as Initializing, Enabling, Disabling, Dumping, and Frozen.

- Read Cache Size, Write Cache Size: Number of MB allocated to this SP's read and write caches. Some memory is required for system buffers and other system operations as explained in "Setting Up Array Memory for Caching or RAID 3 LUNs" on page 32 in Chapter 2.

- Read Cache Hit Ratio: The percentage of read cache hits for the SP. A read cache hit occurs when the SP finds a sought page in read cache memory, and thus does not need to read the page from disk. The ratio is meaningful only if the SP's read cache is enabled. For information on how to enable an SP's read cache, see "Disabling or Enabling Array Write Caching" on page 74 in Chapter 3.

- Write Cache Hit Ratio: The percentage of write cache hits for the SP's write cache. A write cache hit occurs when the SP finds and modifies data in the write cache memory, which usually saves a write operation. For example, a hit that occurs when a page is sought in a RAID 5 LUN eliminates the need to read, modify, and write the data. High ratios are desirable because each hit indicates at least one disk access that was not needed.

- Write Cache Type: Write caching requires two SPs and is always mirrored.

- Page Size: Number of KB in the cache page: 2, 4, 8, or 16. The default is 2 KB.

- Number of Pages: The total number of pages in the SP. Each page has the cache page size you selected when setting up array caching. This number equals the cache size divided by the cache page size, minus space for checksum tables. If the array has two SPs and both are working, they divide the total number of pages between them. If an SP is idle for a long period or fails, the active SP can increase its share of pages.

- Number of Unassigned Pages: Percentage of unassigned dirty pages in both SPs' write caches. Unassigned dirty pages are dirty pages belonging to a LUN that is not enabled for either SP, that is, not accessible from either SP. Unassigned pages would result, for example, if an SP fails and its write cache contains dirty pages. In such a case, the dirty pages belonging to the failed SP become unassigned pages.

  If the LUNs owned by the failed SP are transferred to the working SP, any unassigned pages for those LUNs transfer automatically to the working SP. The working SP writes these unassigned pages to the LUNs and the Unassigned Dirty Pages value returns to 0%. If the LUN that owns the dirty pages has an irrecoverable failure, you can clear the unassigned pages by unbinding the LUN, as explained in "Unbinding a LUN" on page 70 in Chapter 3.

- Percentage of Dirty Pages: Percentage of pages that have been modified in the SP's write cache, but that have not yet been written to disk. A high percentage of dirty pages means the cache is handling many write requests.

Getting Information on Other Array Components

To display information on the components of an array, select it in the Storage System Manager window; click the monitor array button near the middle of the toolbar or choose Monitor from the Array menu. The Components For Array window opens Figure 4-14 shows an example.
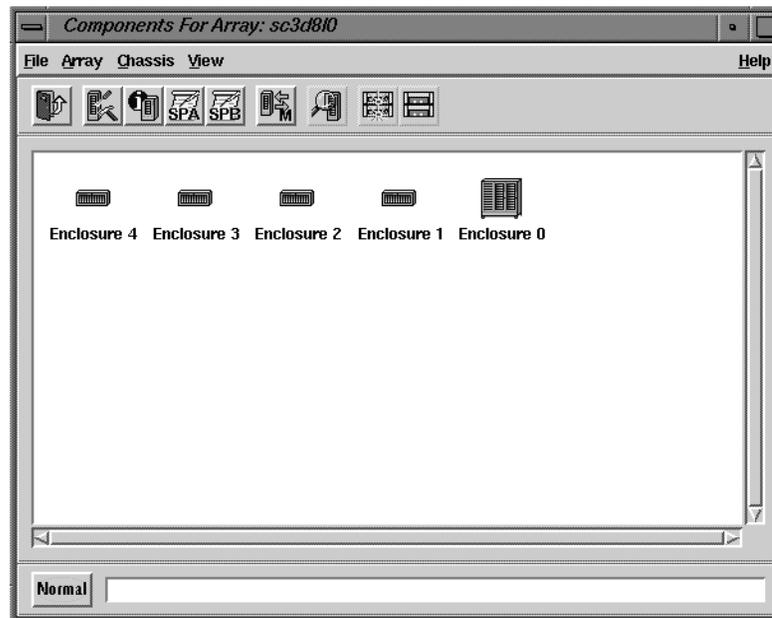
Figure 4-14   Example Equipment View

The icons indicate the type of the enclosure:

- deskside Fibre Channel RAID enclosure (DPE):

- rackmount Fibre Channel RAID enclosure (DPE):

- deskside FibreVault with RAID disk modules (DAE):

- rackmount FibreVault with RAID disk modules (DAE):

111

For information on the array components, double-click the enclosure icon in the
Components For Array window. The Equipment View window opens, as shown in
Figure 4-15.
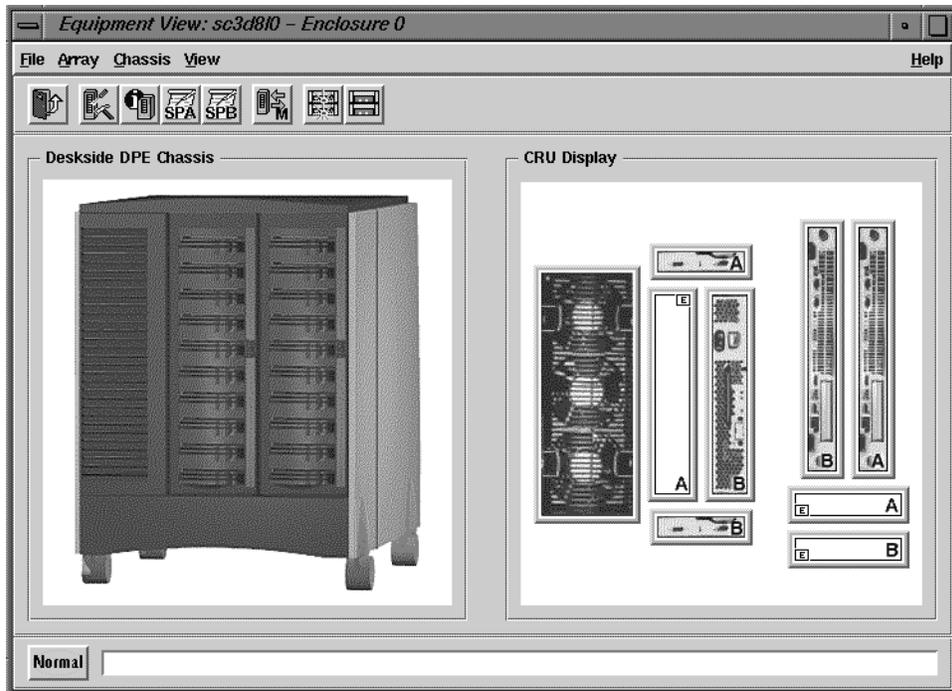
Figure 4-15   Example Equipment View Window

Double-clicking the icon for a specific component displays a window with more
information on the component. "Checking a Faulted Array" on page 122 in Chapter 5
explains this information.

Displaying an SP Event Log

Each SP maintains a log of event messages. These events include hard errors, startups, and shutdowns involving disk modules, fans, SPs, LCCs, power supplies, and the SPS. The messages appear in the order in which the events occurred, so the most recent messages are at the end of the log. Periodically, the SP writes this log to disk to maintain it when SP power is off. The log can hold 16,800 bytes.

You can display all the events in the log or only the events that occurred from a date and time you specify up to the current time. You can also save the contents of the log to a file you specify.

Follow these steps to open an SP event log:

1. In the Storage System Manager window, select the array whose SP event log you want to view.

2. Click the History SP A or History SP B button near the right end of the array toolbar, or choose History from the Array menu, and then choose SP A or SP B.

The Event Log window opens. Scroll down to see the most recent events; Figure 4-16 shows an example.

Figure 4-16   Example Event Log

In this window, the columns headed `Sense Key` and `Ext Code` (extended code) are for use by service personnel.

The hexadecimal codes fall into four series:

- informational codes (600h series) that do not indicate an error condition and require no action, but can help in establishing history

- soft codes (800h series): most of these codes require no operator action unless they occur frequently

- error codes (900h series) that indicate a serious error condition

- fatal error codes (A00h series) that indicate a fatal error condition

For explanations of the error codes, see the release notes.

To display only events from a certain date, follow these steps:

1. At the Display Entries As Of: field near the top of the Event Log window, select a time interval from the pull-down menu (such as the last 24 hours), or type in a date and time.

2. Click *Refresh* at the bottom of the window.

To clear the list of displayed events, click *Reset*. The Event Log window shows only events added to the SP log as of the time you clicked *Reset*.

To clear the log (destroy all contents) and reset the log's counters to zero, click *Clear Log*; in the confirmation window, click *Yes*.

To display all events in the log, click *Show All*.

To save the contents of the log to a file, click *Save*. In the window that opens, specify filename, file type, and path.

Using the Event Monitor

You can use the event monitor to receive email notice of error messages within a range that you specify. This software is installed with the agents and their user interfaces, although it is separate from them; see Figure 1-1 in Chapter 1.

The event monitor polls the SP event log at a specified interval (default 60 seconds). If the SP logs an event within the range you specify, the event monitor logs the event as a message, and saves it in */usr/ssm/etc/messages*. A launch application sends the message to the email addresses you specify. The following is an example message:

```
machine,company.com = host, sc3d8l0(SP A) = device
Poll failed, check agent status on host machine,company.com AGENT NOT RUNNING
Event Code Number = c02
```

To use the event monitor, follow these steps:

1. Determine the range of error messages you want to know: see the information on hexadecimal codes on page 114 in "Displaying an SP Event Log." View the messages in the release notes if necessary.

2. In the */usr/ssm/etc* directory, copy the template event monitor file *ssmevent.config.proto* to *ssmevent.config*.

3. In a text editor, modify *ssmevent.config* for your needs:

- At `error`, enter the range of error messages you want emailed to you, such as `0x900 0xaff` (serious and fatal error codes).

- At `launch`, enter the email address or IP address that is to receive the information.

- At the three `format` lines, change the defaults if desired:

  The first `format` line displays the hostname and device name.

  The second `format` line displays the event error message.

  The third `format` line displays the event code. This line is optional; the other two are necessary to display meaningful error information.

- At `poll`, enter the interval (number of seconds) at which you want the SP log polled; the default is 60.

  The default poll and refresh values keep network activity to a minimum, but allow fairly quick notification of any events that occur. Polling faster does not bring much benefit. If the management station is monitoring many large systems on several hosts so that the aggregate time to complete a poll of all systems might take a minute, set a slower poll rate (such as every 3000 to 6000 seconds (five to ten minutes). Use a slower rate also for a system using only the tty line.

- At `refresh`, enter the interval (number of seconds) at which you want the event monitor to poll the internet connection with the host to determine whether contact with the host is still valid; the default is 15.

4. In a text editor, open */usr/ssm/etc/.hosts* and enter the hosts to be monitored.

5. To start the *ssmevent* daemon and initiate the notification process, enter

   /etc/init.d/ssm_monitor start

116

The following sample *ssmevent.config* file sends mail to user@company.com for SP log events 0x600 to 0xCFF (all error messages).

```
file /usr/ssm/etc/.hosts
monitor default
error 0x600 0xcff
launch /usr/lib/sendmail user@company.com <
format %H = host, %D = device
format %E
format Event Code Number = %N

poll 60

refresh 15
```

When the event monitor starts, it checks the syntax of the statements in the configuration file and reports any errors in *per/ssm/log/ssmevent.log*. If it finds no serious syntax errors, the event monitor tries to connect with the storage systems defined in the file, reporting the results to *per/ssm/log/ssmevent.log*. Each time the event monitor refreshes, it tries to connect to any server or storage system that it could not connect with before; if it is successful, it reports the connection to *per/ssm/log/ssmevent.log*.

As long as the *ssmevent* daemon is running, messages that are older than 48 hours are removed from *per/ssm/etc/messages*.

Note: To turn off the notification process, enter

/etc/init.d/ssm_monitor stop

Identifying and Correcting Failures

This chapter describes how the Storage System Manager window displays information about array health and outlines procedures you can use to detect and correct problems that may arise with managed arrays. This chapter is organized as follows:

Array Health Indicators in the Storage System Manager Window

The Storage System Manager window displays fault information several ways; Figure 5-1 points out these features.
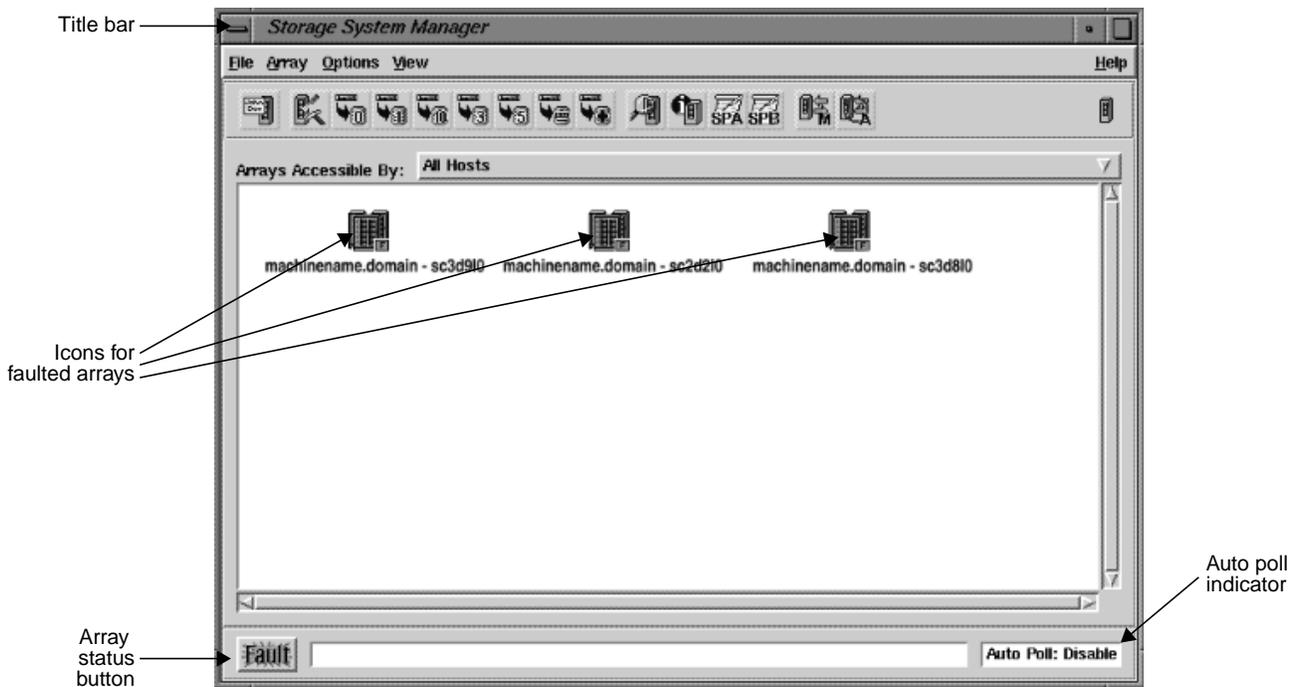
Figure 5-1   Array Health Indicators in the Storage System Manager Window

Note: When the Storage System Manager window is minimized to an icon, a slanted line appears across the array icon graphic to indicate that one or more managed arrays are faulted, as shown in Figure 5-2.

Figure 5-2   Storage System Manager Icon With Faulted Array(s)

Fault indication features are explained in separate sections below.

Array Status Button

The array status button at the lower left of the window indicates array health:

- Normal (button is gray) indicates that *ssmgui* detects no failures in the managed arrays

- Fault (button is amber) indicates that *ssmgui* detects a failure in one or more arrays, or that one or more arrays is inaccessible

Clicking this button displays information on system faults; Figure 5-3 shows an example.
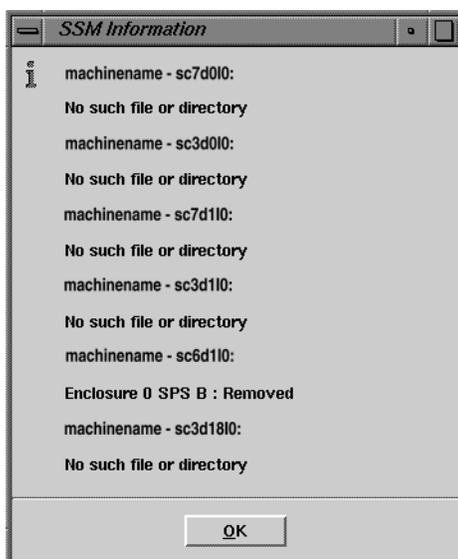
Figure 5-3   SSM Information Window (Array Fault Button)

Array Icons in the Array Selection Area

The color of an icon for an array indicates its health:

- gray indicates that no failure is detected

- amber indicates a fault in some part of the array; a small F appears at the lower right

The graphic for an array icon indicates the status of the array; see "Enclosure Health and Accessibility" on page 27 in Chapter 2.

If you are managing many arrays, checking the Array Status button is more convenient than clicking the array icons. This button also appears in the Array Configuration window.

Auto Poll Indicator

The Auto Poll indicator at the lower right of the window shows the status of automatic polling. You can make sure that you are getting the latest status information on the arrays in two ways:

- Enable automatic polling for the managed arrays and for *ssmgui*, as explained in "ssmagent Polling Interval and Polling Requests" on page 84 in Chapter 4.

- Manually poll the managed arrays periodically by clicking the Manual Poll button at the right end of the array toolbar; this button has a letter M at its lower right.

If any array that you want to monitor does not appear in the Storage System Manager window, see "Using the Host Administration Window to Add Servers" on page 22 in Chapter 2.

Checking a Faulted Array

To determine the state of each component in a faulted array, follow these steps:

1. In the Storage System Manager window, select the faulted array for which you want information.

2. Open the Components For Array or Equipment View windows for the selected arrays either by clicking the monitor array button near the middle of the array toolbar or by choosing Monitor from the Array menu.

For an array with multiple enclosures, the Components For Array window opens; for an array with only one enclosure, the Equipment View window opens.

3. If the Components For Array window opens:

   ■   Select the faulted enclosures (amber icons).

   ■   Open the Equipment View window for the selected enclosures by choosing Monitor... in the Chassis menu, or by double-clicking the icon for the enclosure.

   Note: The icon type in the Components For Array window indicates the kind of enclosure; see "Interpreting Array Icons" on page 27 in Chapter 2.

4. In the Equipment View window for each enclosure, look for amber icons. Figure 5-4 shows an example.
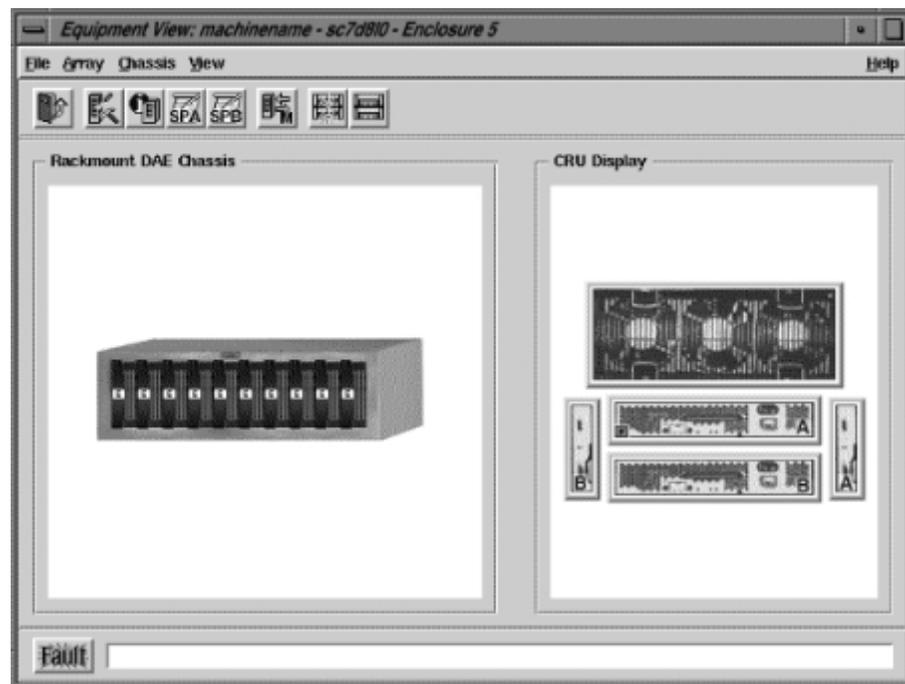
Figure 5-4   Equipment View Window With Faulted Power Supply

The right field in this window displays system components, also known as CRUs: customer-replaceable units, although end users replace only disk modules. The rest of this chapter and the *Origin FibreVault and Fibre Channel RAID Owner's Guide* have information on replacing components.

A white icon, which also displays the letter E ({empty}), indicates a missing component.

If you do not know the name of the CRU represented by an amber icon, you can display it by positioning the cursor over the CRU for a few seconds without moving it. The RAID GUI displays a description of the CRU near the icon, as well as in the status bar.

The external Fibre Channel Hub is transparent to *ssmgui* and the command-line interfaces.

Note: For configurations with multiple disk enclosures, you can cause the LEDs on specific disk modules to flash as an aid to locating them. In the Components window or Equipment View toolbar, use the second button from the right to start flashing the disk module LEDs; use the button at the far right to stop LED flashing.

Faulted LUN

If a LUN is faulted, its icon is amber. Figure 5-5 shows an example with the faulted LUN selected.
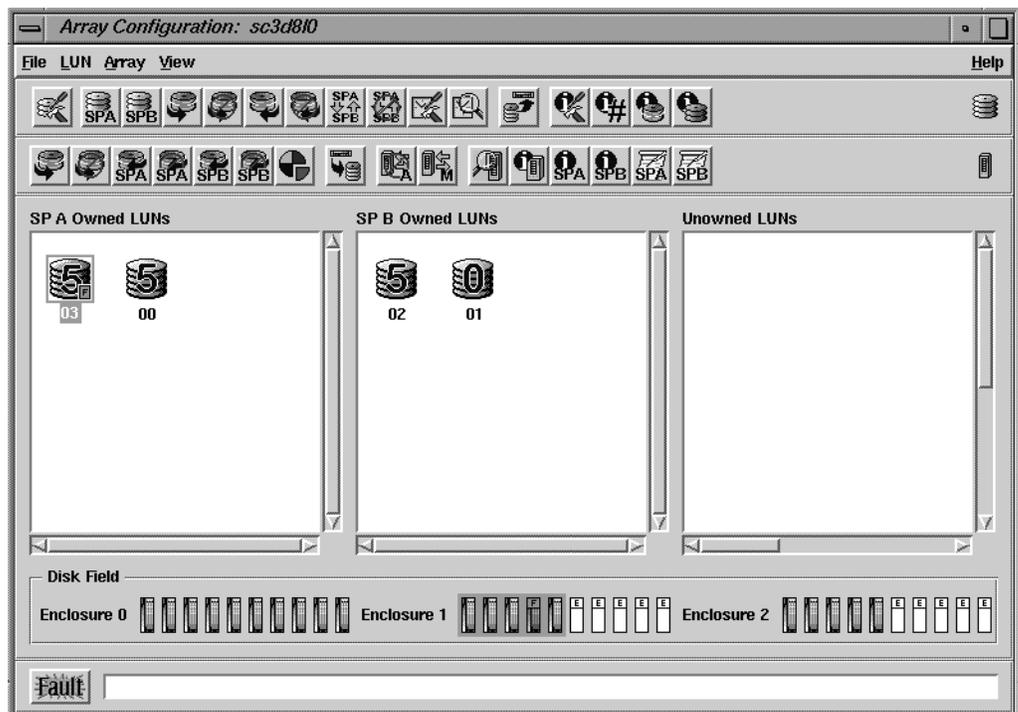
Figure 5-5   Faulted LUNs

If a LUN is inaccessible, its icon appears in the Unowned LUNs field in the Array Configuration window.

Click the button at the far right of the LUN (upper) toolbar (or choose Describe Fault Indication from the LUN menu) to display fault information. Figure 5-6 shows an example.
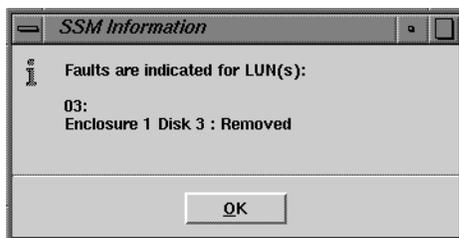
Figure 5-6   LUN Fault Information

For complete information on the LUN, display its windows as explained in "Using the LUN Information Windows" on page 94 in Chapter 4.

Note: If a DPE has two LCCs (A and B), each DAE cabled to it must also have two LCCs. If a DAE has no LCC B, binding LUNs on SP B is restricted in some circumstances: If SP A uses disk modules 00 through 09 (that is, all the disk modules in the DPE) in LUNs, you cannot bind any LUNs on SP B, because the connection to the disk modules in the DAE is missing. There is no path through LCC B (absent) and no path through the disk modules on the DPE (all are bound on SP A).

Faulted Disk Module

An amber disk module icon indicates that the disk module it represents is in one of these states:

- failed: either powered on but inaccessible, or powered off by the SP, which can happen when a disk module has been replaced with one of the wrong capacity

- removed from the enclosure after the *ssmagent* started running

This section contains the following topics:

- "Fibre Channel Disk Module Types" on page 127
- "LUN Integrity and Disk Module Failure" on page 128
- "Replacing a Disk Module" on page 129

- "Replacing a Database or Cache Vault Disk Module" on page 131
- "Rebuilding a RAID 5, 3, 1, or 1/0 LUN" on page 131
- "Rebuilding a RAID 0 LUN" on page 133

Fibre Channel Disk Module Types

Replace a failed disk module only with another Silicon Graphics FC-AL disk module; see disk replacement information in the *Origin FibreVault and Fibre Channel RAID Owner's Guide*. Follow disk-replacement procedures explained in that manual.

Special rules apply when you replace database or cache vault disk modules:

- The first three disk modules in the first enclosure (DPE) in a chain, or in the only enclosure in the array, contain licensed internal code (LIC). These disk modules, with disk IDs 00, 01, and 02, are known as *database disk modules*. At least two are required for DPE operation.

  If all three are removed at the same time when the array is powered on, contact with the array is lost. (The SP fault LED illuminates.) If you remove these three disk modules with the array powered off, label these as you remove them, because they must be reinstalled in their original positions.

- If the array uses write caching, the disk modules that the array uses for its cache vault are 00 through 08.

  If a cache vault disk module fails, the array dumps its write cache image to the remaining modules in the vault. Then it writes all dirty (modified) pages to disk and disables write caching. Write caching remains disabled until a replacement disk module is inserted and the array rebuilds the LUN with the replacement module in it. The Write Cache State field in the Cache section of the SP Information window indicates whether array write caching is enabled or disabled (see "SP Cache Information" on page 109 in Chapter 4).

"Replacing a Database or Cache Vault Disk Module" on page 131 has guidelines for replacing these disk modules.

Disk Module Failure and Powering Off, Unbinding, or Taking Disks Offline

The system operator can replace or reinsert any failed disk module without powering off the array or interrupting user applications.

If the failed disk module is part of a RAID 5, 3, 1, or 1/0 LUN, the LUN can continue functioning without interrupting user applications.

Unbinding a RAID disk module is equivalent to taking it offline. Unbinding is required only in these situations:

- The failed disk module is an individual disk: see "Replacing a Disk Module" on page 129.

- More than one disk module in a LUN fails; see "LUN Integrity and Disk Module Failure" on page 128.

- The extremely unlikely event that all three database modules fail; in this case, contact your support provider, who must send a pair of replacement database modules.

LUN Integrity and Disk Module Failure

If a failed disk module is part of a RAID 5, 3, 1, or 1/0 LUN, the LUN, you can replace the disk module without powering off the array or interrupting applications. If the array contains a hot spare on standby, the SP automatically rebuilds the failed module on the hot spare. When you replace a disk module in a RAID 5, 3, 1, or 1/0 LUN, the SP equalizes the new module, and then begins to reconstruct the data. While rebuilding occurs, users have uninterrupted access to information on the LUN. (For more information on how these LUNs are rebuilt, see "Rebuilding a RAID 5, 3, 1, or 1/0 LUN" on page 131 and "Rebuilding a RAID 0 LUN" on page 133.)

RAID disk modules do not have to be unbound to be replaced unless two or more in a LUN fail so that the LUN's data redundancy is compromised:

- RAID 0: a single disk module fails
- RAID 1/0: both modules in a pair
- RAID 1: both modules in a pair

- RAID 3: two modules in a LUN (RAID 3 does not use a second hot spare)

- RAID 5: two modules in a LUN

In these cases, the LUN becomes unowned (not accessible by either SP). After you unbind the affected LUN(s) and replace the disk modules (one at a time), you rebind the affected LUN(s). If the data on the failed disks was backed up, restore it to the new disks.

A hot spare is unowned until it becomes part of a LUN when one of the LUN's disk modules fails. The failure of an unowned hot spare does not make any LUN inaccessible.

If an individual disk LUN fails, you must unbind it, replace the disk, and rebind the LUN.

If you want to move disk modules from one array to another, back up the data, unbind the LUN(s), move each disk module one at a time to its new location, rebind, and restore the backed-up data.

Replacing a Disk Module

Removing the wrong disk module can introduce an additional fault that shuts down the LUN containing the failed module.

Note: For configurations with multiple disk enclosures, you can cause the LEDs on specific disk modules to flash as an aid to locating them. In the Components window or Equipment View toolbar, use the second button from the right to start flashing the disk module LEDs; use the button at the far right to stop LED flashing.

Before removing a disk module, follow these steps:

1. Check module's amber check or fault LED. If it is illuminated, or if both LEDs are off, the host is bypassing it, indicating a failure.

2. Double-click the disk icon to read fault information (see "Disk Module Error Information" on page 104 in Chapter 4).

3. Read the event log for the SP that owns the LUN containing the faulty disk module for a message about the disk module; see "Displaying an SP Event Log" on page 113 in Chapter 4.

4. Check for any other messages that indicate a related failure, such as a failure of a SCSI bus or a general shutdown of an enclosure. Such a message could mean the disk module itself has not failed.

5. A message about the disk module contains its module ID. If the disk module button does not, in the Array Configuration window, choose Show Disk IDs from the View menu to see the location of the disk module. Figure 5-7 shows an example.
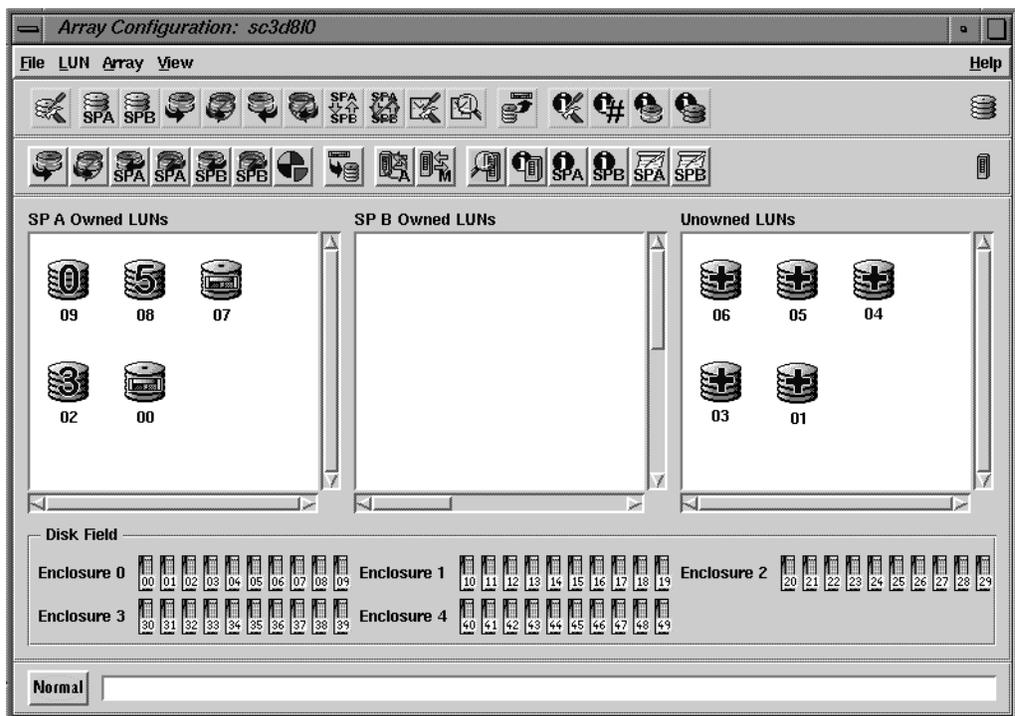
Figure 5-7   Array Configuration Window: Disk Module IDs

6. If the disk is an individual disk LUN, you must unbind the LUN before you replace the disk module, and rebind the LUN afterward. Unbinding and rebinding is not necessary for other LUN types.

7. After you confirm the failure of a disk module, replace it following instructions in the *Origin FibreVault and Fibre Channel RAID Owner's Guide*. Be sure to use only Silicon Graphics FC RAID disk modules (part number 9470192).

Replacing a Database or Cache Vault Disk Module

Special rules apply when you replace a database or cache vault disk module; see "Fibre Channel Disk Module Types" on page 127

If you must replace all database disk modules, follow these steps:

1. Unbind the LUN(s) containing the disk modules you want to replace, following instructions in "Unbinding a LUN" on page 70 in Chapter 3.

2. One by one, replace all failed disk modules except for the one in slot 00; leave the disk module in slot 00 in place. (See instructions in the *Origin FibreVault and Fibre Channel RAID Owner's Guide*.)

3. Bind the disk modules into the desired LUNs, following instructions in "Binding Disk Modules" in Chapter 2.

   Tip: Use a minibind in this case: *ssmcli bind* with the **-z** option. For information on the **-z** option, see page 146 in Chapter 6.

   When you bind the LUNs, the SP copies the licensed internal code from the 00 disk module onto the other two database disk modules (01 and 02).

4. Unbind the LUN containing the 00 disk module.

5. Replace this disk module and rebind the LUN.

Rebuilding a RAID 5, 3, 1, or 1/0 LUN

Although you can remove a disk module within a RAID 5, 3, 1, or 1/0 LUN without damaging the data on the LUN, do so only when the disk module has actually failed.

If the array contains a hot spare on standby, the SP automatically rebuilds the failed module on the hot spare. When you replace a disk module in a RAID 5, 3, 1, or 1/0 LUN, the SP equalizes the new module, and then begins to reconstruct the data. While rebuilding occurs, you have uninterrupted access to information on the LUN.

You can use the Disk Information window (see Figure 5-8) to follow the status of the new disk module during the rebuilding process. To open this window, choose Configuration from the Array menu; in the Array Configuration window, double-click the disk module you are rebuilding.
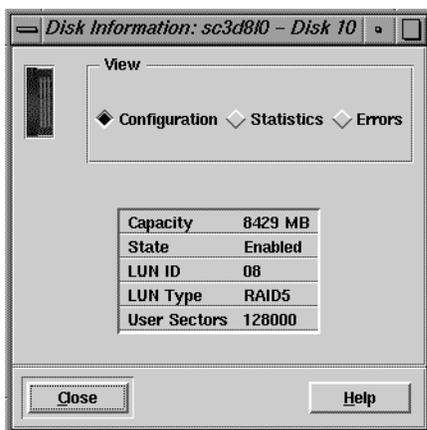
Figure 5-8   Disk Information Window: Configuration Information

The Status field shows these states:

1. Powering Up: The hot spare or replacement disk module is being powered on.

2. Rebuilding: The SP rebuilds the data on the hot spare.

3. Equalizing: Data from a hot spare is being copied onto a replacement disk module.

4. Enabled: The hot spare is fully integrated into the LUN, or the failed disk module has been replaced with a new module and the SP copies the data from the hot spare onto the new module.

5. Ready: The copy is complete. The LUN consists of the disk modules in the original slots and the hot spare is on standby.

(For full information on this window, see "Disk Module Configuration Information" on page 101 in Chapter 4.)

Rebuilding occurs at the same time as user I/O. The rebuild time that is specified when the LUN is bound determines the duration of the rebuild process and the amount of SP resources dedicated to rebuilding. A short rebuild time consumes many resources and may significantly degrade performance. A long period consumes fewer resources with

less effect on performance. You can determine the rebuild period by looking at the configuration section of the LUN Information window (see "LUN Configuration Information" on page 95 in Chapter 4).

Rebuilding a RAID 0 LUN

If one of the disk modules in a RAID 0 LUN fails, the LUN changes state from owned to unowned. The state does not change back to owned after you replace the failed disk module.

Follow these steps to rebuild the RAID 0 LUN:

1. Replace the failed disk module following instructions in the *Origin FibreVault and Fibre Channel RAID Owner's Guide*.

2. Make sure the RAID 0 LUN is not in use.

3. Reset the LUN's ownership state to owned in one of the following ways:

   - Determine the SP that owns the RAID 0 LUN and reboot the SP with the *ssmcli rebootSP* command; see "rebootSP" on page 177 in Chapter 6.

   or

   - Use *hinv* to determine the LUN ID and enter the following command:

     scsicontrol -i *lunid*

     For example:

     ```
     scsicontrol -i sc2d0l4
     ```

   or

   - Use *ssmcli trespass* to change LUN ownership:

     ssmcli -d *devicename* **lun** *lun-number*

     For example:

     ```
     ssmcli -i sc2d0l4 lun 4
     ```

     This operation does not require a reboot. For more information, see "trespass" on page 183 in Chapter 6.

Faulted SP

An amber SP icon in the Equipment View window indicates that the SP has failed. When an SP fails:

- One or more LUNs may become inaccessible and the array's performance can degrade if array read or write caching was enabled.

- The SP's check or service light turns on, along with the check or service light on the front of the array.

- If the array has a second SP and a number of LUNs are inaccessible, you may want to transfer control of the LUNs to the working SP. Follow instructions in "Transferring Control of a LUN (Manual Trespass)" on page 68 in Chapter 3.

To display more information on the fault, double-click the amber SP icon. The SP Information window appears; see "Using the SP Information Windows" on page 105 in Chapter 4.

An authorized Silicon Graphics System Support Engineer can replace the SP under power, without interrupting applications. Call your service provider.

Caution: If array write caching is enabled, it must be disabled before an SP is replaced. See "Enabling or Disabling Array Caching" on page 73 in Chapter 3 for instructions.

Faulted LCC

An amber link control card (LCC) icon in the Equipment View window indicates that the LCC has failed. In addition, the LCC's fault light turns on, along with the service light on the front of the array.

When an LCC fails, the SP it is connected to loses access to its LUNs, and the array's performance can degrade. If the array has a second LCC and a number of LUNs are inaccessible, you may want to transfer control of the LUNs to the SP that is connected to the working LCC. Follow instructions in "Transferring Control of a LUN (Manual Trespass)" on page 68 in Chapter 3.

Double-click the amber icon to display more information on the fault. See the *Origin FibreVault and Fibre Channel RAID Owner's Guide* for information on replacing the LCC.

Faulted Fan

An amber fan pack icon in the Equipment View window indicates that the fan group it represents is in one of the following states:

- Faulted: The fan group has failed.

- Removed: The fan group was removed after the *ssmagent* started running.

- Not Present: The fan group failed or a fan pack was removed before the agent started running.

If one fan in a pack fails, the other fans speed up to compensate so that the array can continue operating. If a second fan fails and the temperature rises, the array shuts down after about two minutes to prevent damage to the disk modules from overheating.

Double-click the amber icon to display more information on the fault. *ssmgui* refers to the drive fan pack as the rear fan (FAN A in the command-line interface), and to the SP fan pack as the front fan (FAN B in the CLI). A DPE has both types of fan pack; a DAE has only the rear fan pack.

If you see an amber fan icon, replace the entire fan pack as soon as possible; call your service provider. The replacement fan pack must be on hand for this process. If the fan pack is removed for more than two minutes, the SPs and the disk modules power off; they power on when a functional replacement fan pack is installed. See the *Origin FibreVault and Fibre Channel RAID Owner's Guide* for information on replacing the fan pack.

If the fan pack is removed or disabled (more than one fan fails), the disk modules spin down. Note that the green disk module LEDs and the green (active) SP LEDs remain illuminated; though the disk modules and SP(s) are shut down, they are not faulted. The disk modules and SPs automatically power back on when a functional fan pack is installed.

Faulted Power Supply

An amber power supply icon in the Equipment View window indicates that the power supply it represents is in one of the following states:

- Down: Failed or removed after the *ssmagent* started running.

- Not Present: Failed or removed before the agent started running.

A Fibre Channel RAID enclosure has a power supply A; it can also have an optional power supply B. An array with two power supplies can recover from the failure of one power supply and provide uninterrupted service while the defective power supply is replaced. If a second power supply fails or is removed, the entire enclosure shuts down immediately.

In enclosures with no SPS, failure of the AC distribution system (line cord, utility power, and so on) also immediately shuts down the entire enclosure.

Double-click the amber icon to display more information on the fault. See the *Origin FibreVault and Fibre Channel RAID Owner's Guide* for information on replacing the power supply.

Note: When a DPE shuts down, the operating system loses contact with the LUNs. When the DPE powers on again, you might need to reboot the server to re-establish operating system access to the LUNs; in this case, you must also restart *ssmagent*, as explained in "Restarting ssmagent" on page 201 in Appendix A.

Faulted SPS

An amber SPS icon in the Equipment View window indicates a problem with the standby power supply (SPS), also known as the battery backup unit (BBU). Double-click the amber icon to display more information on the fault.

Also, check the status lights on the SPS. These indicate when the SPS has an internal fault, when it is recharging, and when its battery pack needs replacing. (See the *Origin FibreVault and Fibre Channel RAID Owner's Guide* for a full description of this and other system component hardware.)

If the SPS status lights indicate an internal fault, it might still be able to run, but the SP disables write caching. The array can use write caching only when a fully charged, working SPS is present. After the faulted SPS is replaced with a functional SPS, the SP automatically re-enables write caching when it detects the presence of the functional SPS. (If a second fully charged, working SPS is present, write caching continues.)

If the SPS status lights indicate that a battery pack needs replacing, or indicate any other failure, contact your service provider. In an array with a second SPS that is healthy, an authorized Silicon Graphics System Support Engineer can replace the faulted SPS while the DPE is powered on, without interrupting applications.

The Fibre Channel RAID Command-Line Interface

The Fibre Channel RAID command-line interface, *ssmcli*, communicates with the SP (storage processor) in the Fibre Channel RAID enclosure. The *ssmcli* program communicates directly with the Fibre Channel RAID agent *ssmagent*. It provides an alternative to the RAID GUI (*ssmgui*). The CLI is accessible using a serial-port terminal, or remote login.

This chapter introduces the *ssmcli*, explains the *ssmcli* command, and describes each subcommand for the *ssmcli* command in these sections:

- "Using the Fibre Channel CLI" on page 139
- "ssmcli" on page 140
- "ssmcli Subcommands" on page 142
- "ssmcli Error Values" on page 184

Using the Fibre Channel CLI

The RAID CLI is typically run on a management station, such as a console or terminal attached to an Origin2000 workstation, although it can also run on an array server. Like the RAID GUI, the RAID CLI can run locally or remotely (communicating with the agent *ssmagent* over the network).

You can use the CLI to configure, control, and retrieve status from the Fibre Channel RAID enclosure. You can also use the RAID CLI to automate disk-array management functions by writing shell scripts or batch files that incorporate the *ssmcli* command.

You run the RAID CLI in an IRIX shell. Each *ssmcli* command consists of the *ssmcli* command (and switches) together with another subcommand (and its switches). The *ssmcli* command does not perform any function when executed by itself. However, when used in combination with the other commands described in this chapter, you can configure and manage the Fibre Channel RAID storage systems.

ssmcli

The *ssmcli* command sends RAID storage device management and configuration requests to an API (application programming interface) on a local or remote server. The RAID storage device consists of disk drives, power supplies, fans, a standby power supply (SPS), and a second RAID storage processor (SP); *ssmcli* subcommands monitor and control these components.

Note: The agent *ssmagent* must be running before *ssmcli* commands are accepted. See "Starting the Fibre Channel RAID Agent ssmagent" on page 4 in Chapter 1.

This section contains the following topics;

- "ssmcli Syntax" on page 140
- "Environment Variables RaidAgentHost and RaidAgentDevice" on page 141

For more information on *ssmcli*, see its man page ssmcli(7M).

ssmcli Syntax

The *ssmcli* command syntax is

ssmcli [**-help**] [**-d** *device*] [**-p**] [**-t** *timeout*] **-**[**v** | **q**] [**-h** *hostname*] *command <optional switches>*

Note: You must be logged in as root or as the system administrator for this command.

Optional switches are as follows:

-help	Display the help screen and do not start the *ssmcli* process.
command	One of a set of subcommands that are used with the *ssmcli* command to configure and manage a storage system, such as *bind* or *getdisk*. See "ssmcli Subcommands" on page 142.
-d *device*	The target RAID device. If you do not specify the **-d** switch, the environment variable RaidAgentDevice is the default value for the device. (RaidAgentDevice is explained later in this section.) If RaidAgentDevice is not set and the **-d** switch is not present, an error is generated on all commands that need device information. Unless the environment variable indicates otherwise, every instance of the *ssmcli*

command must have the **-d** switch. The **-d** switch overrides the RaidAgentDevice environment variable. You can get a list of RAID devices present using the *getagent* command with *ssmcli*.

-h *hostname*   The network name of the destination server. The environment variable RaidAgentHost is the default value for the server if the **-h** switch is not present. The **-h** switch overrides the RaidAgentHost setting. If RaidAgentHost is not set and the **-h** switch is not present, the local hostname is used.

-p   Parses the entered *command* without making a network call to the application programming interface (API). If the string does not parse correctly, an error message prints to *stderr*; otherwise a message verifying that the string parsed correctly prints to *stderr*.

-q   Suppresses error messages. This switch can be useful in a script.

-t *timeout*   Sets the timeout value in seconds. Default is 25 seconds.

-v   Enables verbose error descriptions. This is the default unless **-q** is specified.

command <optional switches>

A subcommand and its switches. These subcommands are described in "ssmcli Subcommands"; Table 6-1 on page 142 summarizes the subcommands.

Environment Variables RaidAgentHost and RaidAgentDevice

The environment variables RaidAgentHost and RaidAgentDevice are default values for the server and the device if **-h** and **-d** switches are not supplied with the *ssmcli* command. (The **-h** and **-d** switches are explained above.)

If RaidAgentDevice is not set and the **-d** switch is not present, an error is generated on all commands that need device information.

The following examples illustrate environment command settings:

```
setenv RaidAgentDevice sc1d0l0
setenv RaidAgentHost panther
```

In these examples, `sc1d0l0` is a device name and `panther` is a hostname.

ssmcli Subcommands

This section explains all *ssmcli* subcommands. Table 6-1 summarizes the subcommands.

Table 6-1   ssmcli Subcommand Summary

Subcommand	Page	Purpose
bind	page 143	Binds (groups) physical disks into RAID types
chglun	page 150	Changes parameters for a currently bound group
clearlog	page 153	Destroys the contents of the SP error log and resets its counters to zero
clearstats	page 153	Resets statistics logging on the SP, setting all log counters to 0
failback	page 153	Returns control of the fibre loop to the specified SP
getagent	page 154	Gets names and descriptions of devices controlled by the destination API
getcache	page 155	Gets information about the storage-system caching environment
getcontrol	page 159	Gets information about the SP configuration
getcrus	page 161	Displays status information on all system components, such as the fan module, SPS, and so on
getdisk	page 164	Displays status information on disk modules in the system
getlog	page 167	Displays the SP log, in which errors are stored, or displays the oldest or newest entries
getloop	page 168	Returns information about the fibre loop.
getlun	page 169	Displays information about a disk group
getsniffer	page 175	Displays background and sniff information for a LUN
r3wrbuff	page 177	Sets RAID 3 write buffering
rebootSP	page 177	Reboots one or both SPs in the system
setcache	page 178	Changes information about the FC RAID controller's caching environment
setraid5	page 180	Sets the state of mixed-mode RAID 5
setsniffer	page 181	Sets sniffer and background verify parameters

142

Table 6-1 (continued)   ssmcli Subcommand Summary

Subcommand	Page	Purpose
setstats	page 182	Set statistics logging or returns its state
trespass	page 183	Sets the trespass value of a LUN
unbind	page 183	Deconfigures physical disks from their current logical configuration, destroying all data on LUN

Note: You must be logged in as root or as the system administrator to use *ssmcli*.

bind

The *bind* command groups physical disks into logical RAID types (levels). This section contains the following topics:

- "bind Syntax" on page 143

- "LUN Binding Order and Restrictions" on page 147

- "Selecting Modules for Mirror Images for a RAID 1/0 LUN" on page 149

bind Syntax

bind is used with *ssmcli* as follows:

ssmcli bind raid-type lun-number disk-names <optional switches>

Required parameters are as follows:

raid-type   Specifies the RAID type for the array:

- *r5* = RAID type 5

- *r3* = RAID type 3

- *r1* = RAID type 1

- *r1_0* = RAID type 1/0

- *r0* = RAID type 0

- *id* = individual disk

- *hs* = hot spare

lun-number   Logical unit number (LUN) to assign the unit (valid range: 0-31).

disk-names   Physical disks to bind, in the format *e_d*, where *e* is enclosure number (in hex) and *d* is the disk number on that bus.

                  For example, 2_3 represents enclosure 2, disk 3. Valid enclosure values are the hexadecimal numbers 0 through 9, A (the hex representation of 10), and B (the hex representation for 11). Valid disk numbers are 0 through 9.

Optional switches are as follows. For specific RAID types, some of these parameters are required; see Table 6-3.

-c *cache-flags*   Enables either read or write caching, both read and write caching, or neither type of caching for any type of LUN except for a RAID 3 LUN or a hot spare. If you enable any type of caching for the LUN, caching occurs only when storage-system caching is enabled; see "setcache" on page 178. Write caching has special hardware requirements; see "Hardware Requirements for Caching" on page 33 in Chapter 2.

                  Set the type of caching used as follows:

- *none* = no caching
- *read* = read caching
- *write* = write caching
- *rw* = read and write caching

                  The default is *none*. Whether you should use caching for a specific LUN depends on the LUN's RAID type; see Table 6-2.

Table 6-2   Binding Restrictions for RAID Types and Caching

RAID Type	Write Caching	Read Caching
RAID 5	Highly recommended	Recommended
RAID 3	Not acceptable	Not acceptable
RAID 1	Acceptable	Recommended
RAID 1/0	Acceptable	Recommended
RAID 0	Acceptable	Recommended
Individual disk	Acceptable	Recommended
Hot spare	Not acceptable	Not acceptable

144

-n *min_latency_read*

For RAID 3 only, enables minimal latency reads functionality. A setting of 0 (the default) disables minimal latency reads; a setting of 1 enables it.

-r *rebuild-time*

Maximum time in hours allocated to reconstruct data on either a hot spare or a new disk module that replaces a failed disk module in a LUN. This parameter applies to all RAID LUNs except RAID 0 and hot spares. Legal values are 0 through 254. The default value is 4, which is adequate for most situations.

The time you specify determines the amount of resource that the SP devotes to rebuilding instead of to normal I/O activity. A rebuild time of 2 hours rebuilds the disk more quickly, but degrades response time slightly. A rebuild time of 0 hours rebuilds the disk module as quickly as possible, but degrades response time significantly. If your site requires fast response time and you want to minimize degradation to normal I/O activity, you can set the rebuild to a longer period of time, such as 24 hours.

Note: You cannot change the rebuild time for a LUN while the LUN is rebuilding. Any change to the rebuild time takes effect as soon as the operation completes.

The actual rebuild time can differ significantly from the time you specify, especially for a RAID 1/0 LUN, or a LUN containing 4- GB disk modules. Because a RAID 1/0 with *n* disk modules can continue functioning with up to as many as *n*/2 failed drive modules and only one drive at a time is rebuilt, the actual rebuild time for such a LUN is the time you specify multiplied by the number of failed drives. For 4-GB disk modules, the minimum rebuild time should be 2 hours, because a rebuild time of less than 2 hours actually takes about 2 to 3 hours. If you specify any other time, actual disk module rebuilding takes about twice as long.

-s *stripe-size*

Number of blocks per physical disk within a LUN. The default value is 128 and should be an even multiple of 16 sectors; supported values are 4, 8, 16, 32, 64, 128, and 256.

The stripe element size can affect the performance of a RAID 5 or RAID 1/0 LUN. This value is ignored for RAID 1 LUNs, individual disks, and hot spares. For RAID 3 binds, the only legal value is 1.

145

The smaller the stripe size, the more efficient the distribution of data read or written. However, if the stripe size is too small for a single I/O operation, the operation requires access to two stripes, which causes the hardware to read to or write from two disk modules instead of one. Generally, you should use the smallest stripe size that rarely forces access to another stripe.

-v *max_verify_time*

The maximum amount of time to spend rebuilding a LUN if an SP fails and is replaced during I/O operations. A longer value rebuilds the LUN more slowly and has less impact on overall system performance. The allowable range is 0-254 hours; the default value is 4 hours.

-z *stripe-count* The number of stripes in a LUN. For example, when you bind a RAID 5 with a stripe count of 2, partition the LUN into 2 stripes, thus preventing access to the remainder of the available space. This switch is useful for short bind operations (minibinds). Legal values are any number greater than or equal to zero. The default value is 0, which binds the maximum number of stripes available.

Table 6-3 identifies bind parameters that you must configure for each RAID type.

Table 6-3   RAID Types and Bind Parameters

RAID Type	Default SP	Rebuild Time	Stripe (Element) Size	Verify Time	Caching
RAID 5	Required	Required	Required	Yes	Read and write
RAID 3	Required	Required	Not required	Yes	Not required
RAID 1	Required	Required	Not required	Yes	Read and write
RAID 1/0	Required	Required	Required	Yes	Read and write
RAID 0	Required	Not required	Required	No	Read, write, or both
Individual disk	Required	Not required	Not required	No	Read, write, or both
Hot spare	Not required	N/A	Not required	No	Not required

LUN number must be specified for all RAID types in Table 6-3. RAID 3 also requires minimal latency reads.

The following example binds LUN 3 as disks 1, 2, 3, 4, and 5 from enclosure 1 into a RAID 5, with a 4-hour maximum rebuild time, a 128-block stripe size per physical disk, and with read cache enabled.

```
ssmcli -d sc2d0l0 bind r5 3 1_1 1_2 1_3 1_4 1_5 -r 4 -s 128 -c read
```

This subcommand has no output. Errors are printed to *stderr*.

LUN Binding Order and Restrictions

For greatest flexibility, bind LUNs in the following order in the array:

1. RAID 1/0

2. RAID 5, RAID 3, and RAID 0

3. RAID 1

4. individual units and hot spares

Before you start binding disk modules, read the restrictions and recommendations in Table 6-4.

Table 6-4   Binding Restrictions for RAID Types

LUN Type	Disk Modules	Notes
Any LUN	Unbound disk modules only; all must have the same capacity	For LUNs except RAID 3, you can deallocate RAID 3 memory. Binding RAID 3 LUNs with other LUN types in the same array is not supported.
RAID 5	3 minimum (5 recommended minimum), 16 maximum	You can bind one less module per LUN that you eventually use by selecting an empty slot icon. However, the LUN operates in a degraded mode until a disk module is installed in the empty slot and the array integrates it into the LUN. You can select the modules in any order.

Table 6-4 (continued)   Binding Restrictions for RAID Types

LUN Type	Disk Modules	Notes
RAID 3	5 or 9	You can bind one less module per LUN than you eventually use by selecting an empty slot icon. However, the LUN operates in a degraded mode until a disk module is installed in the empty slot and the array integrates it into the LUN. You can select the modules in any order.
		Check RAID 3 memory before binding. If you do not allocate adequate memory to the RAID 3 partition, the LUN is unowned.
		Binding other LUN types in the same array is not supported.
RAID 1	2	
RAID 1/0	Even number: 4 minimum, 16 maximum	Disk modules are paired into mirrored images in the order in which you select them: see Figure 6-1 on page 149.
RAID 0	3 minimum, 16 maximum	You can select the modules in any order.
Individual disk	1	
Hot spare	1	You cannot bind as hot spares disk modules 0:0 through 0:8.
		The capacity of a hot spare must be at least as great as the capacity of the largest disk module that it might replace.
		For a RAID 3 LUN, only one hot spare is used in case of disk failure; a second hot spare is not used in case of a second disk failure.

Selecting Modules for Mirror Images for a RAID 1/0 LUN

If you are binding a RAID 1/0 LUN, select disk modules in alternating order for the primary and secondary images. Figure 6-1 diagrams this order.
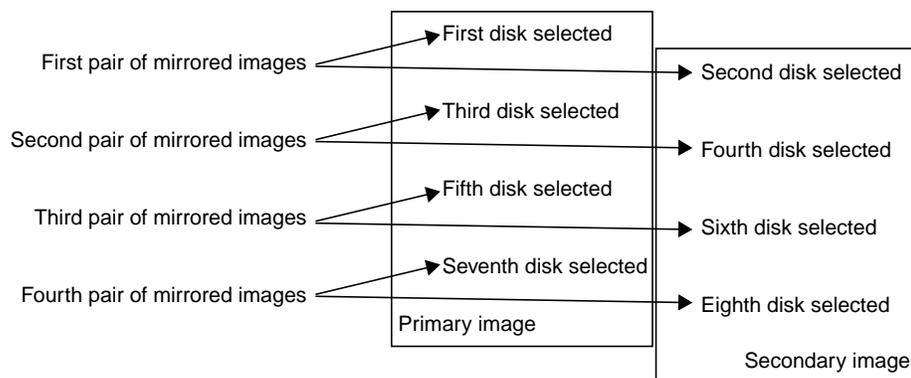
Figure 6-1   Selecting Modules for Mirror Images for a RAID 1/0 LUN

The first and second modules selected for a RAID 1/0 LUN are a pair of peer image parts, the third and fourth modules are another pair of peer image parts, and so on.

When a LUN is assigned to an SP, its state becomes Assigned. When you use the SP that owns the LUN as the communications path to the chassis, the state of its disk modules becomes Enabled.

If you need to change the *raid-type* or **-s** *stripe-size* parameters, you must unbind the LUN and then rebind it with the correct parameters. (The *unbind* subcommand is explained on page 183.) If you need to change the maximum rebuild time, read cache state, or write cache state, use the *chglun* command (see page 150).

chglun

The *chglun* subcommand changes parameters on a currently bound LUN. You must use the LUN number **-l** *lun* switch with this subcommand; you must also use at least one other switch with it.

chglun is used with *ssmcli* as follows:

ssmcli chglun **-l** *lun <optional switches>*

where **-l** *lun* specifies which logical unit to change. This switch is mandatory.

Optional switches are as follows:

-a *auto*   Automatically reassigns ownership of a LUN to an active SP if an SP failure occurs. By default this feature, known as auto assign, is disabled for all RAID types except hot spares, to which it does not apply. The values are

- *0* = disable auto assignment for the specified LUN (default)

- *1* = enable auto assignment for the specified LUN

-c *cache-flags*   See the description of this parameter on page 144 in "bind."

-d *default-owner*

Change the default owner of the LUN.

- *0* = change ownership of the LUN to SP A

- *1* = change ownership of the LUN to SP B

-i *idle-thresh*   Maximum number of I/Os that can be outstanding to a LUN for the LUN still to be considered idle. Used to determine cache flush start time. Valid values are 0 through 254; the default is 0.

-t *idle-delay-time*

Amount of time in 100-ms units that a LUN must be below the *idle-thresh* in order to be considered idle. For example, 5 equals 500 ms. Once a LUN is considered idle, any dirty pages in the cache can begin idle time flushing. The default is 20.

-pc *maxIO*   Maximum number of I/Os which the LUN can have outstanding and still perform prefetching. The default is 40; valid values are 1 through 100.

-v *verify_time*   Maximum number of hours to spend rebuilding a LUN if an SP fails and is replaced during I/O operations. A longer value rebuilds the LUN more slowly and has less impact on overall system performance. The range is 0-254 hours; the default value is 0 hours.

-pd *min_size_disable*

Sets the prefetch disable, which is the size in blocks of the smallest read request for which prefetching is disabled. Prefetching is disabled for all read requests whose size is equal or larger than the Prefetch Disable size. The range is 0 through 65534; the default value is 129 (blocks).

-n *min_latency_reads*

Enables minimal latency reads functionality. A setting of *0* disables; a setting of *1* enables. This parameter applies to RAID 3 only.

-pf *prefetch_type*

Determines whether to prefetch data of a variable or constant length or to disable prefetching.

- *0* = disabled

- *1* = constant length

- *2* = variable length (default)

-pm *prefetch_size* / *mult*

Determines the amount of data prefetched for one host read request.

- For constant-length prefetching, the prefetch size is the number of blocks to prefetch.

- For variable-length prefetching, the prefetch multiplier is the amount of data to prefetch relative to the amount of data requested.

For example, if the prefetch size is 8, the amount of data to prefetch is 8 times the amount of data requested. The range is 1 through 32; the default value is 4.

-mp *maxlength*   For variable-length prefetching, gets the absolute maximum number of disk blocks to prefetch in response to a read request. This value is ignored for constant length.

-sm *segment-size/mult*

> For constant-length prefetching, sets the number of blocks of data to prefetch for each read request. An SP reads one segment at a time from the LUN because smaller prefetch requests interfere less with other host requests.
>
> For variable-length prefetching, determines the amount of data to retrieve from the LUN in a single prefetch operation. Specify a factor equal to or less than the prefetch multiplier. If the segment multiple is equal to the prefetch multiplier, prefetch operations are not divided into segments. The amount of data prefetched for each read request must equal or be less than the prefetch multiplier.

-r *rebuild-time*   See the description of this parameter on page 145 in "bind."

> **Note:** You cannot change the rebuild time for a LUN while the LUN is rebuilding. Any change to the rebuild time takes effect as soon as the operation completes.

-rt *retain_prefetch_data*

> Enables retaining prefetch data. A setting of *0* disables; a setting of *1* enables. The range is 16 through 65534; the default is 2048 (blocks).

-w *write-aside*   Sets the smallest write request size, in blocks, that bypasses the cache and goes directly to the disks.

Note: RAID 3 LUNs do not use read or write caching.

If you are binding a RAID 1/0 LUN, the order in which you select modules is important; see "Selecting Modules for Mirror Images for a RAID 1/0 LUN" on page 149.

Note: A LUN cannot use write caching until you enable the storage-system write cache. Write caching requires two SPs with equal amounts of memory. The LUN cannot use read caching until you enable the read cache for the SP that owns the LUN (see "setcache" on page 178).

This example changes LUN 3 to perform write caching and to rebuild in 4 hours:

```
ssmcli -d sc2d0l0 chglun -l 3 -c write -r 4
```

This subcommand has no output. Errors are printed to *stderr*.

clearlog

The *clearlog* command destroys the contents of the RAID SP's unsolicited error log. All log counters are reset to zero. *clearlog* is used with *ssmcli* as follows:

ssmcli clearlog

For example:

```
ssmcli -d sc2d0l0 clearlog
```

Example output is as follows:

```
SP unsolicited error log cleared.
```

clearstats

The *clearstats* command resets statistics logging on the RAID SP. If statistics logging is on, it is turned off and then back on. If statistics logging is off, it is turned on and then off. *clearstats* is used with *ssmcli* as follows:

ssmcli clearstats

For example:

```
ssmcli -d sc2d0l0 clearstats
```

Example output is as follows:

```
Statistics successfully cleared, statistics logging disabled.
```

failback

The *failback* command returns control of the fibre loop to the specified SP. *failback* is used with *ssmcli* as follows:

ssmcli failback

For example:

```
ssmcli -d sc2d0l0 failback
```

This subcommand has no output.

153

getagent

The *getagent* command returns the names of devices controlled by the destination API along with descriptions about each device that the API is configured to communicate with. If you use the *getagent* command without switches, all of the values are displayed. If you enter the *getagent* command with switches, only the values for those switches are displayed.

This subcommand is used with *ssmcli* as follows:

ssmcli getagent <optional switches>

Optional switches are as follows:

-cabinet	Cabinet type.
-desc	ASCII description of the device.
-mem	Size in MB of the SP's memory. Each SP in the storage system must have the same amount of memory to make full use of the memory.
-model	Model number of the device.
-name	Name of the device.
-node	Location in file system of device.
-peersig	Signature of peer SP.
-prom	Revision level of the SP's PROM code. Each SP in the storage system runs the same revision of PROM code. PROM code is updated automatically when an SP's LIC is updated.
-rev	Microcode revision number: the revision of the Licensed Internal Code (LIC) that the SP is running. Each SP in the storage system runs the same revision of LIC.
-scsiid	SCSI ID number for the SP host connection, which is determined by switch settings on the SP. For information on these switch settings, see the *Origin FibreVault and Fibre Channel RAID Owner's Guide*
-serial	Enclosure serial number.
-sig	Unique SP signature.
-type	Model type (deskside, rackmount).
-ver	Agent revision number: the version of the agent that is running.

For example:

```
ssmcli -d sc2d0l0 getagent
```

Following is example output for one device:

```
Agent Rev: 1.50
Name: ttyb
Desc: sandshark
Node: ttyb
Signature: 0xd454201
Peer Signature: 0x0
Revision 8.15.13
SCSI Id: 0
Model: 7310
Model type: Deskside
Prom Rev: 2.79.0
SP Memory: 32
Serial No: 94-2754-252
Cabinet: 5
```

getcache

The *getcache* subcommand returns information about the RAID SP's caching environment. This subcommand is used with *ssmcli* as follows:

ssmcli getcache <optional switches>

Optional switches are as follows:

-bbutd   Day of the week that BBU self-test is performed.

-bbutt   Time of day that BBU self-test is performed. Format: *hhmm*.

-freebackend Free memory available in the back-end memory bank.

-freecontrol Free memory available in the control memory bank.

-freefrontend Free memory available in the front-end memory bank.

-high   Write cache high watermark. The percentage of dirty pages that, when reached, causes the SP to begin flushing the cache. The default is 100%. A lower value causes the SP to start flushing the cache sooner. The high watermark cannot be less than the low watermark. Watermark processing is turned off when both the low and high watermark values are 100.

-low	Write cache low watermark. The percentage of cache dirty pages that determines when cache flushing stops. When the write cache low watermark is reached during a flush operation, the SP stops flushing the cache. The default value is 50%. Do not set the low watermark higher than the high watermark. Watermark processing is turned off when both the low and high watermark values are 100.
-mirror	Write cache mirrored state.
-page	Size of a cache page in KB: 2, 4, 8, or 16 KB. The default size is 2 KB.
-pdp	Percent dirty pages currently in cache. The percentage of pages that have been modified in the SP's write cache, but that have not yet been written to disk. A high percentage of dirty pages means the cache is handling many write requests.
-pmbackend	Physical memory size for the back-end memory bank.
-pmfrontend	Physical memory size for the front-end memory bank.
-ppo	Percent pages owned by SP.
-rh	Read hit ratio. The percentage of cache read hits for the SP. A read hit occurs when the SP finds a sought page in cache memory, and thus does not need to read the page from disk. The ratio is meaningful only if the SP's read cache is enabled. High hit ratios are desirable because each hit indicates at least one disk access that was not needed. You may want to compare the read and write hit ratios for the LUN with the read and write hit ratio for the entire storage system. For a LUN to have the best performance, the hit ratios should be higher than those for the storage system. A very low read or write hit rate for a busy LUN may mean that caching is not helping the LUN's performance.
-rm3a	Amount of memory, in MB, reserved for the optimized RAID 3 data buffers on SP A.
-rm3b	The amount of memory, in MB, reserved for the optimized RAID 3 data buffers on SP B.
-rsta	Current internal state of the read cache on SP A.
-rstb	Current internal state of the read cache on SP B.
-rsza	Amount of read cache memory SP A is allowed to use (in MB).
-rszb	Amount of read cache memory SP B is allowed to use (in MB).

156

-spa	Number of pages owned by SP A. The total number of pages in the SP, each page with the cache page size you selected when setting up storage-system caching. This number equals the cache size divided by the cache page size, minus space for checksum tables. If the storage system has two SPs and both are working, they divide the total number of pages between them. If an SP is idle for a long period or fails, the active SP may increase its share of pages.
-spb	Number of pages owned by SP B. The total number of pages in the SP, each page with the cache page size you selected when setting up storage-system caching. This number equals the cache size divided by the cache page size, minus space for checksum tables. If the storage system has two SPs and both are working, they divide the total number of pages between them. If an SP is idle for a long period or fails, the active SP may increase its share of pages.
-state	Current state of the SP's read or write cache. The possible read cache states are Enabled, Disabling, and Disabled. An SP's read cache is automatically enabled on power-on if the cache size is a valid number and the SP has at least 8 MB of memory. The possible write cache states are Enabled or Disabled, and several transition states, such as Initializing, Enabling, Disabling, Dumping, and Frozen.
-ucp	Unassigned cache pages (that is, owned by neither SP A nor B).
-wh	Write hit ratio: the percentage of cache write hits for the SP's write cache. A write hit occurs when the SP finds and modifies data in cache memory, which usually saves a write operation. For example, with a RAID 5 LUN, a write hit eliminates the need to read, modify, and write the data. High hit ratios are desirable because each hit indicates at least one disk access that was not needed.
	You may want to compare the read and write hit ratios for the LUN with the read and write hit ratio for the entire storage system. For a LUN to have the best performance, the hit ratios should be higher than those for the storage system. A very low read or write hit rate for a busy LUN may mean that caching is not helping the LUN's performance.
-wsza	Write cache size for SP A; it is always equal to that of SP B.
-wszb	Write cache size for SP B; it is always equal to that of SPA.

For example:

```
ssmcli -d sc2d0l0 getcache
```

Example output is as follows.

```
SP Write Cache State Disabled
Cache Page Size: 2
Write Cache Mirrored: YES
Low Watermark: 0
High Watermark: 0
SPA Cache Pages: 0
SPB cache pages: 0
Unassigned cache pages: 0
Read Hit Ratio: 0
Prct Dirty Cache Pages: 0
Prct Cache Pages Owned
SPA Read Cache State Disabled
SPB Read Cache State Disabled
SPS Test Day: Sunday
SPS Test Time: 01:00
Physical Memory Size of CPU Ram = 3
Physical Memory Size of Front-End = 16
Physical Memory Size of Back-End = 16
SPA Read Cache Size = 0
SPB Read Cache Size = 0
SPA Write Cache Size = 0
SPB Write Cache Size = 0
SPA Optimized RAID 3 Memory Size = 6
SPB Optimized RAID 3 Memory Size = 6
Free Memory Size of CPU RAM = 2
Free Memory Size of Front-End = 10
Free Memory Size of Back End = 9
```

An SP's write cache is automatically enabled at power-on if the cache size is a valid number and the storage system has certain components; see "Hardware Requirements for Caching" on page 33 in Chapter 2.

Note: Caching statistics are meaningful only if the SP's write cache is enabled; use *setcache*.

getcontrol

The *getcontrol* subcommand returns information about the SP configuration. This subcommand is used with *ssmcli* as follows:

ssmcli getcontrol <optional switches>

Optional switches are as follows:

-allib	Idle and busy ticks for both buses.
-arv	Number of times a user request arrived while at least one other request was being processed.
-avg	Average number of requests seen by the SP.
-busy	Percentage of time the SP is busy.
-cbt	Controller idle and busy ticks.
-date	SP's system date.
-day	SP's system day.
-flush	Number of times the high watermark, low watermark, and idle unit turned on flushing. This parameter also displays the number of requests to flush write cache, and the number of write cache blocks flushed.
-he	Number of hard errors the SP has seen.
-ib1	Displays internal bus 1 idle and busy ticks.
-ib2	Displays internal bus 2 idle and busy ticks.
-idle	Percentage of time the SP is idle.
-max	Maximum number of requests seen by the SP.
-read	Number of reads received by the SP.
-rw	Displays the following:

- host read requests: the number of read requests that host makes to LUN

- host write requests: the number of write requests from host to LUN

- host blocks read: the number of blocks the host read from the LUN

- host blocks written: the number of blocks the host wrote to the LUN

159

-sc   Indicates whether system cache is disabled or enabled. For SP A, the output gives the state of both read and write cache; for example:

```
SP Read Cache State Disabled
SP Write Cache State Disabled
```

For SP B, the output is read cache only, because write cache is always the same as that for SP A.

-sf   Indicates whether system fault LED is on or off.

-sl   Indicates whether statistics logging is on or off.

-time   SP's system time.

-write   Number of writes received by SP.

For example:

```
ssmcli -d sc2d0l0 getcontrol
```

Example output is as follows:

```
System Fault LED: OFF
Statistics Logging: ON
SP Write Cache State: Disabled
Max Requests: 3
Average Requests: 0
Hard errors: 0
Total Reads: 145
Total Writes: 34642049
Prct Busy: 27
Prct Idle: 72
System Date: 04/11/1997

Day of the week: Friday
System Time: 8:58:57
read_requests: 145
write_requests: 34642049
blocks_read: 1633
blocks_written: 363696414
sum_queue_lengths_seen_by_arrivals: 401
arrivals_to_non_zero_queue: 400
hw_flush_on: 0
idle_flush_on: 5314
lw_flush_off: 0
```

```
write_cache_flushes: 98
write_cache_blocks_flushed: 1150
Internal bus 1 busy ticks: 4294967295
Internal bus 1 idle ticks: 4294967295
Internal bus 2 busy ticks: 0
Internal bus 2 idle ticks: 0
Internal bus 3 busy ticks: 0
Internal bus 3 idle ticks: 0
Internal bus 4 busy ticks: 0
Internal bus 4 idle ticks: 0
Internal bus 5 busy ticks: 0
Internal bus 5 idle ticks: 0
Controller busy ticks: 1403792
Controller idle ticks: 3775334
```

Note: Fibre Channel RAID SPs have only one internal bus. For 5400 model series, the internal bus 2 through 5 idle and busy ticks values are 0.

getcrus

The *getcrus* command returns state information on all or selected CRUs (customer replaceable units) in the subsystem except for disks (see the *getdisk* command). The *getcrus* command displays the state of the following CRUs: fan packs, link controller cards (LCCs), storage processors (SPs), standby power supplies (SPSs), and power supplies (also called voltage semi-regulated converters, or VSCs). The external Fibre Channel Hub is transparent to *ssmcli*.

Note: Despite the command name, not every component for which the *getcrus* command returns state information is customer-replaceable. Only fan packs, LCCs, and power supplies are customer-replaceable; see the *Origin FibreVault and Fibre Channel RAID Owner's Guide* for instructions. For failed SPs and SPSs, call your service provider.

getcrus is used with *ssmcli* as follows:

ssmcli getcrus <optional switches>

Optional switches are as follows:

-fana   Gets the state information on fan module A, which is the fan for the drives (rear fan) in a DPE, or the only fan module in a DAE.

If any fan in a module fails, a fault is displayed for its module, and other fans speed up to compensate. They cool the storage system indefinitely; however, if failure occurs in the normally functioning module, the temperature rises and the storage system shuts down after two minutes.

If a fan bank is down, you can determine whether it has failed by whether its button is amber, the event log lists fan failures, or the replace light on the fan module is lit. If a fan module has failed, replace it as soon as practical.

-fanb   Gets the state information on fan module B, which is the SP fan pack in a DPE (front fan). See **-fana** for further information.

-lcca   State information on the link controller card (LCC) for SP A.

-lccb   State information on the LCC for SP B.

-spa   Current operational state of SP A. It can be in any of these states:

- Present: The SP is the communication channel you are using to communicate with the chassis.

- Empty: The agent cannot talk to the SP because a communication channel specifying the SP is not in the agent's configuration file for the selected host. For example, the SP is connected to a different host than the SP in the communications channel for the chassis.

- Not Present: The SP that is in the communication channel to the selected chassis has failed or been removed.

- Removed: The SP was not present when the agent was started.

-spb   Gets state information on SP B. See the description of **-spa**.

-spsa   Gets state information on SPS A.

-spsb   Gets state information on SPS B.

-vsca   Gets state information on power supply A.

                   If a chassis has the maximum number of VSCs, it can continue operating if one VSC fails, but shuts down immediately if a second VSC fails. If a VSC is Down, you can determine whether it has failed by seeing if its button is amber, looking at the event log for VSC failures, or seeing if the Replace light on the VSC is lit. If the VSC has failed, replace the VSC as soon as practical.

-vscb   Gets the state information on power supply B. You can use getcrus without any switches to get the state information for every CRU except disks.

Table 6-5 summarizes states.

Table 6-5   VSC States

VSC State	Meaning
Present	Operating normally.
Empty	Failed or removed before the agent started running.
Not Connected	For an SP: the agent cannot talk to the SP because a communication channel specifying the SP is not in the agent's configuration file for the selected host. For example, the SP is connected to a different host than the SP in the communications channel for the chassis. ailed or removed before the agent started running.
Removed	Not present when the agent was started.

For example:

```
ssmcli -d sc2d0l0 getcrus
```

Example output is as follows:

```
DPE Enclosure 0
Enclosure 0 FAN A State: Present
Enclosure 0 FAN B State: Present
Enclosure 0 Power A State: Present
Enclosure 0 Power B State: Empty
SPA State: Empty
Enclosure 0 SPS A State: Empty
Enclosure 0 SPS B State: Empty
Enclosure 0 LCC A State: Present
Enclosure 0 LCC B State: Present
DAE Enclosure 1
Enclosure 1 FAN A State: Removed
Enclosure 1 Power A State: Removed
Enclosure 1 Power B State: Removed
Enclosure 1 LCC A State: Removed
Enclosure 1 LCC B State: Removed
```

getdisk

The *getdisk* command returns detailed status information about all or selected disks in the system. *getdisk* is used with *ssmcli* as follows:

ssmcli getdisk <optional switches>

Optional switches are as follows:

disk_position   Use the format e_d, where *e* is the enclosure number (1-10) and *d* is the disk number (0-9).

   When the **disk-position** is not entered, *getdisk* returns an inclusive report on all disks, regardless of what switches are set. To obtain only the information specified by the optional switches, set the **disk-position** switch in the *getdisk* command.

-bind   Percent of the disk that is bound.

-capacity   Disk capacity in blocks.

-hr   Total number of hard read errors for all the disk modules in the LUN that persisted through all the retries. A growing number of hard errors may mean that one or more of the LUN's disk modules is nearing the end of its useful life.

-hs   Hot spare information (NO if not hot spare)

-hw   Total number of hard write errors for all the disk modules in the LUN that persisted through all the retries. A growing number of hard errors may mean that one or more of the LUN's disk modules is nearing the end of its useful life.

-lun   Possible values include: a hexadecimal number, Unbound for an enclosure disk not bound to a LUN, or Empty for an empty enclosure disk slot.

-private   Private space of disk in blocks.

-product   Product ID of the disk.

-rb   What percent of the disk is rebuilt.

-read   Total number of read requests made to the disk module. You might find the LUN read information in *getlun* more useful because it is for the entire LUN, and not just for one of the disk modules in the LUN.

-rr	Number of read retries. Total number of times read and write requests to the disk module were retried.
-rs	Number of remapped sectors. Total number of disk sectors that were faulty when written to, and thus were remapped to a different part of the disk module.
-state	State of disk, as summarized in Table 6-6.

Table 6-6   Disk States (getdisk -state Output)

Disk State	Meaning
Binding	Being bound into a LUN.
Empty	Failed or removed before the agent started running.
Enabled	Either a hot spare on standby or part of a bound LUN that is assigned to (owned by) the SP you are using as the communication channel to the chassis. If the storage system has another SP, this module's status is Ready when you use the other SP as the communication channel to the chassis.
Equalizing	Data from a hot spare is being copied onto a replacement disk module.
Failed	Powered off or inaccessible.
Empty	Powered off by the SP, which can happen if a wrong size module is inserted.
Powering Up	Power is being applied to the disk module.
Ready	Module is part of a broken LUN or of a LUN that is bound and unassigned. This can mean that the disk module is part of a LUN that is not owned by the SP that you are using as the communication channel to the chassis. If the disk module is part of a LUN assigned to an SP other than the one you are using as the communication chassis, the module's status is either Enabled or Ready. It is Enabled when you use the other SP as the communication channel to the chassis.
Rebuilding	Module is either a hot spare or replacement disk module that replaced a failed module in a LUN. The data is being rebuilt on a hot spare or a replacement disk module.
Removed	Removed from the chassis; applies only to a disk module that is part of a LUN.

Table 6-6   Disk States (getdisk -state Output)

Disk State	Meaning
Hot Spare Ready	Module is either a hot spare or replacement disk module that replaced a failed module in a LUN. The data is being rebuilt on a hot spare or replacement disk module.
Unbound	Ready to be bound into a LUN.
Unformatted	Disk drive is not formatted and is thus unusable.

-serial   Serial number of the disk.

-sig   Bind signature (automatically assigned during bind).

-sr   Total number of soft read errors for all the disk modules in the LUN that disappeared before all the retries. A growing number of soft errors may indicate that one of the LUN's disk modules is nearing the end of its useful life.

-sw   Total number of soft write errors for all the disk modules in the LUN that disappeared before all the retries. A growing number of soft errors may indicate that one of the LUN's disk modules is nearing the end of its useful life.

-vendor   Vendor ID of the disk.

-wr   Number of write retries. Total number of times write requests to the disk module were retried.

-write   Total number of write requests made to the disk module. You might find the LUN write information in *getlun* more useful because it is for the entire LUN, and not just for one of the disk modules in the LUN.

The following example returns information for disk 1 in disk array enclosure (FibreVault) 1 (DAE 1):

```
ssmcli -d sc2d0l0 getdisk 1_1
```

Example output is as follows:

```
Enclosure 0 Disk 0
Vendor Id: <disk vendor name>
Product Id: ST19171FC
Product Revision: CC46
Lun: 3
State: Ready
Hot Spare: NO
Prct Rebuilt: 0
Prct Bound: 0
Serial Number: LA142784
Capacity: 8429
Private: 184320
Bind Signature: 9337
Hard Read Errors: 0
Hard Write Errors 0
Soft Read Errors 0
Soft Write Errors 0
Read Retries 0
Write Retries 0
Remapped Sectors 0
Number of Reads 0
Number of Writes 0
```

getlog

The *getlog* command returns the entire SP log, the newest entries to the log, or the oldest entries to the log. *getlog* is used with *ssmcli* as follows:

ssmcli getlog <optional switches>

Depending on the size of the unsolicited log, the *getlog* command may take several minutes to execute.

Optional switches are as follows:

+N   Displays the oldest *N* entries in the log, with the oldest entry first. If there are fewer than *N* entries, the entire log is displayed. *N* must be greater than 0.

-N   Displays the newest *N* entries in the log, with the oldest entry first. If there are fewer than *N* entries, the entire log is displayed. *N* must be greater than 0.

If a range of entries is not specified, the entire log is displayed, with the oldest entry first.

For example:

```
ssmcli -d sc2d0l0 getlog -5
```

The following example output shows the most recent entries in the log, which show a CRU that has failed, been replaced, rebuilt, and become enabled.

```
02/10/98 12:43:01 SP B                    (6e5) (FE Fibre loop operational) [0x00]    0    0
02/10/98 12:43:19 SP B                    (6e4) (FE Fibre loop down) [0x03]           0    0
02/10/98 12:43:19 SP B                    (6e5) (FE Fibre loop operational) [0x00]    0    0
02/10/98 12:43:20 Enclosure 0 Disk 2      (938) (Non R3 LUN-Can't Assign BW) [0x01]   0    0
02/10/98 12:43:20 Enclosure 0 Disk 0      (938) (Non R3 LUN-Can't Assign BW) [0x03]   0    0
```

getloop

The *getloop* command returns information about the fiber loop. *getloop* is used with *ssmcli* as follows:

ssmcli getloop

For example:

```
ssmcli -d sc2d0l0 getloop
```

Example output is as follows:

```
Needs Discovery: NO
Data Loop Failover: NO
Command Loop Failover: NO
Can Peer SP Failover: YES
Manual Discovery Enabled: NO
Can This SP Failover: NO
Illegal Cross Loop: NO
Connection #0: Enclosure 0
```

getlun

The *getlun* command returns information about a LUN and the CRUs that make up that LUN. If statistics logging is disabled, some fields are not printed. Other switches must be used in conjunction with the **-disk** switch. *getlun* is used with *ssmcli* as follows:

ssmcli getlun lun-number <optional switches>

where *lun-number* is the number of the LUN about which to get information. This switch is mandatory.

All other switches are optional:

-aa   Auto assign is enabled or disabled.

-at   Auto-trespass is enabled or disabled.

-bind   Percent of the LUN that is bound.

-bp   Number of blocks prefetched by read cache. This field is not printed if statistics logging is disabled.

-bread   Number of blocks read for each CRU. This switch must be used in conjunction with the **-disk** switch.

-brw   Number of host block read and write requests. This field is not printed if statistics logging is disabled.

-busy   Percentage of time that the disk is busy. This switch must be used in conjunction with the **-disk** switch.

-bwrite   Number of blocks written for each CRU. This switch must be used in conjunction with the **-disk** switch.

-capacity   LUN capacity in MB.

-crus   Names and states of disks in this LUN.

-default   SP is or is not the default owner of the LUN.

-disk   Disk statistics for the LUN. Disk statistics switches are only valid when the **-disk** switch is included.

-element   Stripe size (in blocks). Number of sectors that the storage system can read or write to a single disk module in the LUN. This size was specified when the LUN was bound. Stripe element size does not apply to a RAID 1 LUN, individual unit, or hot spare.

-ena	Disk is enabled or disabled. This switch must be used in conjunction with the **-disk** switch.
-ff	Number of times that a write had to flush a page to make room in the cache. This field is not printed if statistics logging is disabled.
-idle	Percentage of time that the LUN is idle. This switch must be used in conjunction with the **-disk** switch.
-idt	Idle delay time for caching (in 100-ms intervals).
-it	Idle threshold value for caching.
-mp	For variable length prefetching, gets the absolute maximum number of disk blocks to prefetch in response to a read request.
-nminus	Gets whether minimal latency reads is or is not supported. This field is not printed if statistics logging is disabled.
-owner	Which SP currently owns the LUN, SPA or SPB.
-pd min_size_disable	Gets the prefetch disable, which is the size in blocks of the smallest read request for which prefetching is disabled. Prefetching is disabled for all read requests whose size is equal or larger than the Prefetch Disable size.
-pic	Maximum number of I/Os that can be outstanding to the unit and have the unit still perform prefetching.
-prb	Percentage of the LUN that is rebuilt.
-prefetch	Type of prefetching that the LUN is using for read caching.
-prf	Percentage of read requests that caused a flush.
-ps	Prefetch size.
-psm	Prefetch size/multiplier. The amount of data prefetched for one host read request. For constant length prefetching, gets the read request in blocks. For variable length prefetching, gets the multiplier used to calculate the amount of data to prefetch. For example, if the prefetch size is 8, the amount of data to prefetch is 8 times the amount of data requested.
-qavg	Average queue depth for each CRU. This switch must be used in conjunction with the **-disk** switch.

-qmax	Maximum queue depth for each CRU. This switch must be used in conjunction with the **-disk** switch.
-rb	Maximum number of hours to rebuild failed disk.
-rc	LUN's read cache state (enabled or disabled).
-rchm	Read and write cache hits and misses. This field is not printed if statistics logging is disabled.
-reads	Number of reads received for each CRU. This switch must be used in conjunction with the **-disk** switch.
-remap	Number of remapped sectors. This switch must be used in conjunction with the **-disk** switch.
-ret	Retention priority for retaining prefetched data when the read cache becomes full. If the priority is Equal Priority, prefetched data has priority over host-requested data. If it is Equal, both types of data have the same priority.
-rh	Gets read-hit information if statistics logging is enabled. This field is not printed if statistics logging is disabled.
-rhist	Read histogram, which is an array of 10 locations that contain the number of reads. Element n of the array contains the number of reads that were larger than or equal to $2n$-1 and less than $2n$ blocks in size. For example, the first bucket contains 1 block reads, the second contains 2-3, the third 4-7 and so on. **-rhist** also displays read-histogram overflows, which are the number of I/O operations that were larger than 512 blocks. This field is not printed if statistics logging is disabled.
-rr	Total number of times read requests to all the disk modules in the LUN were retried. This switch must be used in conjunction with the **-disk** switch.
-rwf	Percent of write requests that caused a flush.
-rwr	Host read and write requests. This field is not printed if statistics logging is disabled.
-service	Average service time for each CRU. This switch must be used in conjunction with the **-disk** switch.
-srcp	Read cache configuration.
-ss	Segment size.

171

-ssm	Prefetch segment size/multiplier, the size of the segments that make up a prefetch operation. For constant length prefetching, **-ssm** gets the segment size (in blocks) of data prefetched in one read operation from the LUN. For variable length prefetching, **-ssm** gets the multiplier which determines the amount of data, relative to the amount of requested data, prefetched in one read operation. For example, if the segment multiplier is 4, the segment size is 4 times the amount of data requested.
-stripe	How many times an I/O crossed a stripe boundary on a RAID 5, RAID 0, or RAID 1/0 LUN.
-type	RAID type of the LUN as follows:

- RAID5: individual access array
- RAID3: parallel access array
- RAID1: mirrored pair
- RAID1/0: mirrored RAID 0 group
- RAID0: nonredundant individual access array
- Disk: individual unit
- Hot Spare: hot spare

-upb	Number of prefetched blocks not used by the read cache. This field is not printed if statistics logging is disabled.
-verify	Maximum verify time in hours. This field is not printed if statistics logging is disabled.
-was	Minimum block size request that bypasses cache.
-wc	LUN's write cache state (enabled or disabled).
-wch	Number of times that a write was completely satisfied by the cache, avoiding a disk operation. This field is not printed if statistics logging is disabled.
-wh	Write hit information, if statistics logging is enabled. This field is not printed if statistics logging is disabled.

-whist   Write histogram, which is an array of 10 locations that contain the number of writes. Element *n* of the array contains the number of writes that were larger than or equal to 2*n*-1 and less than 2*n* blocks in size. For example, the first bucket contains 1 block writes, the second contains 2 to 3 block writes, the third 4 to 7 block writes, and so on. **-whist** also displays the write histogram overflows, which are the number of I/O that were larger than 512 blocks. This field is not printed if statistics logging is disabled.

-wr   Total number of times write requests to all the disk modules in the LUN were retried. This switch must be used in conjunction with the **-disk** switch.

-writes   Number of writes received for each CRU. This switch must be used in conjunction with the **-disk** switch.

The following example retrieves information about LUN 0:

```
ssmcli -d sc2d0l0 getlun 0
```

Example output is as follows:

```
Prefetch size (blocks) = 4
Prefetch multiplier = 4
Segment size (blocks) = 4
Segment multiplier = 4
Maximum prefetch (blocks) = 512
Prefetch Disable Size (blocks) = 129
Prefetch count = 40

Variable length prefetching YES

Prefetched data retained YES

Max Verify Time  Hours
Read Histogram[0] 0
Read Histogram[1] 0
Read Histogram[2] 0
Read Histogram[3] 0
Read Histogram[4] 0
Read Histogram[5] 0
Read Histogram[6] 0
Read Histogram[7] 0
Read Histogram[8] 0
Read Histogram[9] 0
```

```
Read Histogram overflows 0

Write Histogram[0] 0
Write Histogram[1] 0
Write Histogram[2] 0
Write Histogram[3] 0
Write Histogram[4] 0
Write Histogram[5] 0
Write Histogram[6] 0
Write Histogram[7] 0
Write Histogram[8] 0
Write Histogram[9] 0
Write Histogram overflows 0

Read Requests 0
Write Requests 0
Blocks read 0
Blocks written 0
Read cache hits 0
Prefetched blocks 0
Unused prefetched blocks 0
Write cache hits 0
Forced flushes 0
Read Hit Ratio 0
Write Hit Ratio 0

RAID Type: Disk
Stripe Crossing: 0
Current owner:
Auto-trespass: DISABLED
Auto-assign: DISABLED
Write cache: ENABLED
Read cache: ENABLED
Idle Threshold: 0
Idle Delay Time: 20
Write Aside Size: 2048
Default Owner: SPB
Rebuild Time: 0
Read Hit Ratio: 0
Write Hit Ratio: 0
Prct Reads Forced Flushed: 0
Prct Writes Forced Flushed: 0
Prct Rebuilt: 0
Prct Bound: 0
Lun Capacity: 8429
```

```
Enclosure 0 Disk 3
Ready
Reads: 0
Writes: 0
Blocks Read: 0
Blocks Written: 0
Queue Max: 0
Queue Ave:0
Avg Service Time: 0
Prct Idle Not Available
Prct Busy Not Available
Remapped Sectors: 0
Read Retries: 0
Write Retries: 0
```

getsniffer

The *getsniffer* command fetches and displays background verify and sniff reports. For an explanation of this software, see "ssmgui Architecture" on page 2 in Chapter 1.

getsniffer is used with *ssmcli* as follows:

ssmcli getsniffer lun-number <optional switches>

The following example displays background verify and sniff information for LUN 1.

```
ssmcli -d sc2d0l0 getsniffer 1
```

Optional switches are as follows:

-nonv	Displays historical totals of all nonvolatile recovery verifies.
-curr	Displays the currently running full-unit verify.
-rec	Displays the most recently completed full-unit verify.
-alltot	Displays historical totals of all full-unit verifies.

Example output is as follows:

```
VERIFY RESULTS FOR UNIT 1

                Sniffing state:                ENABLED
                Sniffing rate(100 ms/IO):      4
                Background verify time(hours): 0

                Historical Total of all Non-Volatile Recovery
Verifies(0 passes)

--------------------------------------------------------------
                                 Corrected  Uncorrectable
                Checksum errors        0        0
                Write Stamp errors     0        0
                Time Stamp errors      0        0
                Shed Stamp errors      0        0
                Coherency errors       0        0

                Currently Running Full Unit Verify

--------------------------------------------------------------
                Verify State:         Sniff Running
                Percent Complete:      0
                                 Corrected  Uncorrectable
```

Page 35   (printed 4/9/98)

ssmcli(7)   UNIX System V   ssmcli(7)

```
                Checksum errors        0        0
                Write Stamp errors     0        0
                Time Stamp errors      0        0
                Shed Stamp errors      0        0
                Coherency errors       0        0

                Most Recently Completed Full Unit Verify

--------------------------------------------------------------
                                 Corrected  Uncorrectable
                Checksum errors        0        0
                Write Stamp errors     0        0
                Time Stamp errors      0        0
                Shed Stamp errors      0        0
                Coherency errors       0        0

                Historical Total of All Full Unit Verifies(0 passes)
```

```
----------------------------------------------------------------
                                     Corrected  Uncorrectable
                    Checksum errors        0        0
                    Write Stamp errors     0        0
                    Time Stamp errors      0        0
                    Shed Stamp errors      0        0
                    Coherency errors       0        0
```

r3wrbuff

The *r3wrbuff* command sets the state of RAID 3 write buffering. If there are no parameters, the command returns the current status of RAID 3 write buffering. *r3wrbuff* is used with *ssmcli* as follows:

ssmcli r3wrbuff [**-on**] [**-off**]

where **-on** enables RAID 3 write buffering and **-off** disables RAID 3 write buffering.

For example:

```
ssmcli -d sc2d0l0 r3wrbuff -on
```

Example output is as follows:

```
RAID 3 write buffering ENABLED.
```

When RAID 3 write buffering is enabled, the I/O has not been written to the media. Therefore, if a power cycle, SP failure, or other fatal error occurs, there is no guarantee that the last I/O is written to disk.

rebootSP

The *rebootSP* command reboots one or both of the storage processors in the system. If no switches are specified, the current SP is rebooted. *rebootSP* is used with *ssmcli* as follows:

ssmcli rebootSP <optional switches>

Optional switches are as follows:

-both　　　　Specifies both SPs.

-o　　　　　Suppresses the prompt that asks the user for permission.

For example:

```
ssmcli -d sc2d0l0 rebootSP -both
```

This subcommand has no output.

setcache

The *setcache* command changes system caching on the RAID SP. Using *setcache* with its various switches, you can enable, disable, and configure the cache. Read cache and write cache size switches do not work if that cache is enabled. *setcache* is used with *ssmcli* as follows:

ssmcli setcache <optional switches>

Optional switches are as follows:

-h *hi-watermark*

>   Write cache high watermark. The percentage of dirty pages, which, when reached, causes the SP to begin flushing the cache. The default is 100%. A lower value causes the SP to start flushing the cache sooner. The high watermark cannot be less than the low watermark. Watermark processing is turned off when both the low and high watermark values are 100.

-l *low-watermark*

>   Write cache low watermark. The percentage of cache dirty pages that determines when cache flushing stops. When the write cache low watermark is reached during a flush operation, the SP stops flushing the cache. The default value is 50%. The low watermark cannot be greater than the high watermark. Watermark processing is turned off when both the low and high watermark values are 100.

-m *mirror*   Enable or disable mirroring. A setting of 1 enables mirroring (the default). For the 5400 model series, a setting of 0, which disables mirroring, is not supported and an error message is reported.

-p *page-size*   Size (in KB) of pages into which to partition the cache. Valid sizes are 2 (the default), 4, 8, and 16. Page size is set to 2 regardless of enable/disable state. Generally, cache page size should be 8 KB for general IRIX file server applications, and 2 KB or 4 KB for database applications. This switch does not work if read or write cache is enabled.

-r3a *raid3-size-spa*

> Optimized RAID 3 size for SP A. For RAID 3 LUNs to be accessible, the minimum value is 2 MB.

-r3b *raid3-size-spb*

> Sets the optimized RAID 3 size for SP B. For RAID 3 LUNs to be accessible, the minimum value is 2 MB.

-rsza *read-cache-size-spa*

> Sets read cache size for SP A.

-rszb *read-cache-size-spb*

> Sets read cache size for SP B.

-rca *read-cache-spa*

> Sets the read cache state for SP A. A setting of 1 enables read cache; a setting of 0 disables.

-rcb *read-cache-spb*

> Sets read cache state for SP B. A setting of 1 enables; a setting of 0 disables.

-wc *write-cache*

> Sets write cache state. A setting of 1 enables write cache; a setting of 0 disables.

-wsza *write-cache-size-spa*

> Sets write cache size for SP A. Write cache on each SP must be equal. If you set the size for one SP, the software automatically sets the size in the other SP to match. Write caching requires two SPs with equal amounts of memory.

-wszb *write -cache-size-spb*

> Sets write cache size for SP B; see the explanation of **-wsza** *write-cache-size-spa*.

Note the following:

- Write caching requires two SPs with equal amounts of memory.

- The sum of an SP's write cache size and optimized RAID 3 memory size must be less than or equal to the SP's front-end size.

- The sum of an SP's read cache size and optimized RAID 3 memory size must be less than or equal to the SP's back-end size.

- RAID 3 memory must be the same size in the front and back ends.

Note: See also "Hardware Requirements for Caching" on page 33 and "Hardware Requirements for RAID 3 LUNs" on page 35 in Chapter 2.

This example enables the system write cache, partitioned into 8-KB pages, with a 50% low-watermark value and a 75% high-watermark value:

```
ssmcli -d sc2d0l0 setcache -wc 1 -p 8 -l 50 -h 75
```

This subcommand has no output.

setraid5

The *setraid5* command sets the state of mixed mode. You must enable or disable mixed mode for the LUNs in an array, depending on their type; see Table 6-7.

Table 6-7   Mixed Mode

Mixed Mode State	RAID 5, 1, 1/0, 0	RAID 3 and Hot Spares
Enabled	These LUNs can be bound and accessed.	These LUNs can be bound and accessed.
Disabled (default)	These LUNs can be bound, but not accessed until mixed mode is enabled.	These LUNs can be bound and accessed; RAID 3 operation is optimized.

Note the following:

- Mixed mode is disabled by default, which optimizes performance for the array's RAID 3 transfers. With mixed mode disabled, you can bind other RAID types, but you cannot access them.

- If you are binding or accessing RAID 5, 1, 1/0, or 0 LUNs, enable mixed mode.

- If RAID 5, 1, 1/0, 0 LUNs are bound when mixed mode is disabled, they are unowned; if you then enable mixed mode, they become owned.

- If mixed mode is enabled, disabling it reboots the array's SPs and restarts the LIC. Before beginning this process, make sure no users are conducting I/O with any filesystems or partitions on the array.

- RAID 3 and hot spares work whether mixed mode is enabled or disabled, but RAID 3 is optimized if it is disabled.

If there are no parameters, the command returns the current status of mixed mode. This command is used with *ssmcli* as follows:

ssmcli setraid5 [**-on**] [**-off**]

where **-on** enables mixed mode and **-off** disables mixed mode.

Caution: Disabling or enabling mixed mode (*ssmcli setraid5 -off)* reboots the array's SPs and restarts the LIC. Before beginning this process, make sure no users are conducting I/O with any filesystems or partitions on the array.

For example:

```
ssmcli -d sc2d0l0 setraid5 -on
```

Example output is as follows:

```
This operation will cause a storage subsystem reboot.

DO YOU WISH TO CONTINUE?   [n]   yes

Mixed mode RAID 5 is ENABLED.
```

setsniffer

The *setsniffer* command resets sniffer and background verify parameters. For an explanation of this software, see "ssmgui Architecture" on page 2 in Chapter 1. *setsniffer* is used with *ssmcli* as follows:

ssmcli getsniffer lun-number **0** | **1** *<optional switches>*

where **0** disables sniffing and **1** enables sniffing.

Optional switches are as follows:

-bv   STarts a background verify.

-cr   Clears the background verify reports.

-snrate *rate*   Specifies a sniff rate in units of 100 ms. This setting controls the rate at which sniffs are executed. Valid values are 1 through 254; the default is 30.

The following example sets sniffing on LUN 0 and starts a background verify.

```
ssmcli -d sc2d0l0 setsniffer 0 1 -bv
```

setstats

The *setstats* command sets statistics logging. If no optional switches are present, the command returns the current state of statistics logging.

The SP maintains a log of statistics for the LUNs, disk modules, and storage system caching that you can turn on and off. When the log is on it affects storage system performance, so you may want to leave it off unless you have a reason to monitor performance.

The log uses a 32-bit counter to maintain the statistics numbers, so when the counter is full, the statistics numbers start over at zero. As a result, you will see a sudden decrease in a statistics number if you view it shortly before the counter is full and shortly after the counter restarts at zero. If you want to keep the log turned on for more than two weeks, we recommend that you reset the log about every two weeks, so you know when the numbers start at zero.

setstats is used with *ssmcli* as follows:

ssmcli setstats <optional switches>

where **-on** sets statistics logging on and **-off** sets statistics logging off.

Note: If statistics logging is off, certain options in other subcommands are disabled (such as for *getlun*). The switch descriptions for the other subcommands specify whether statistics logging must be on for their switches to be valid.

For example:

```
ssmcli -d sc2d0l0 setstats -on
```

Example output is as follows:

```
Statistics logging is enabled.
```

trespass

This command sets the LUN's trespass value. *trespass* is used with *ssmcli* as follows:

ssmcli trespass [**mine**|**lun**|**all**] *lun-number*

where

mine   Performs whatever trespassing is necessary to take all of the LUNs that have this SP as a default owner.

lun   Specifies trespass on one LUN (specified by the *lun-number*).

all   Specifies trespass on all LUNs in the system.

The *lun-number* specifies the single LUN on which to trespass.

For example:

```
ssmcli -d sc2d0l0 trespass lun 2
```

This subcommand has no output.

unbind

The *unbind* command deconfigures LUNs from their current logical configuration. *unbind* destroys all data on the LUN. Therefore, when you execute an *unbind* command, you are prompted to verify that you really want to unbind the LUN. If you do not want the prompt to appear, you can override prompting with the **-o** switch.

Caution: Unbinding a LUN destroys any data on it. Observe these precautions:

- Back up any data you want to retain from the LUN before beginning the unbind procedure.

- Unbind only one LUN at a time.

- Never unbind the last bound LUN in a storage system. Unbinding the last LUN may prevent you from accessing the storage system.

- Never reboot a server unless at least one LUN is bound. If you unbind all of the LUNs and lose access to the storage system, refer to the release notice for the *ssmagent* for recovery procedures.

unbind is used with *ssmcli* as follows:

*ssmcli unbind lun-number <**-o**>*

where *lun-number* is the logical unit to deconfigure. The optional switch **-o** turns off the prompt that asks the user for permission.

This example destroys (deconfigures) LUN 3 without prompting for permission, and frees its physical disks to be reconfigured:

```
ssmcli -d sc2d0l0 unbind 3 -o
```

This subcommand has no output.

ssmcli Error Values

The *ssmcli* command returns the error values summarized in Table 6-8.

Table 6-8   ssmcli Error Values

Value	Description
0	No error
1	Invalid command-line parameters (switches)
2	Too few parameters
3	Too many parameters
4	Invalid bind type
5	Invalid LUN number
6	Invalid rebuild time
7	Disk names for RAID 1/0 must have an even number of arguments
8	Invalid number of disks in *bind* command
9	Valid disk names are of format a0, b1, ... e3, and so on
10	An enclosure containing one or more of the disks is not present
11	Disk already bound or binding

Table 6-8 (continued)   ssmcli Error Values

Value	Description
12	Invalid stripe size
13	Invalid disk name
14	Invalid cache flags
15	Invalid default owner
16	Incorrect number of *chglun* parameters
17	Unable to determine name of target host machine
18	Enable-disable flag invalid
19	Invalid usable cache size
20	Invalid page size
21	Invalid watermark value
22	Hi-watermark less than lo-watermark
23	No device name listed on command line
24	Invalid idle threshold
25	Invalid idle delay
26	Invalid write aside size
27	Disks must be on separate buses for bind
28	The agent was unable to configure any devices
29	LUN does not exist
30	LUN already exists
31	Cannot get current working directory for firmware command
32	Illegal position for hot spare
33	Invalid read cache value
34	Invalid read cache size
35	Invalid write cache value

Table 6-8 (continued)   ssmcli Error Values

Value	Description
36	Invalid write cache size
37	Invalid RAID3 cache size
38	Cannot access device
39	Switch not supported
40	Command is not supported
41	Unknown error
50	Agent: Memory allocation error
51	Agent: Invalid device name
52	Agent: Host connection not found
53	Agent: SP not found
54	Agent: Invalid SP suffix
55	Agent: Invalid LUN number
56	Agent: Invalid page size
57	Agent: Invalid command-line switch
58	Agent: Error *bind* command
59	Agent: Error *chglun* command
60	Agent: Error *unbind* command
61	Agent - Error *firmware* command
62	Agent: Error *trespass* command
63	Agent: Error *rebootSP* command
64	Agent: Generic error
65	Agent: Unknown error

The Fibre Channel Non-RAID Command-Line Interface

The fibre channel non-RAID (JBOD) command-line interface, *fccli*, is an interface to the SP (storage processor) in a FibreVault enclosure that contains non-RAID disks. The *fccli* program communicates directly with the JBOD agent *fcagent*. This software is the only way to communicate with JBOD storage; a graphical user interface is provided only for RAID storage. This CLI is accessible using a serial-port terminal, or remote login.

Note: For communication with RAID storage, use the graphical user interface explained in other chapters in this guide or the command-line interface explained in Chapter 6, "The Fibre Channel RAID Command-Line Interface."

This chapter is organized as follows:

- "The Agent fcagent" on page 188
- "fccli Subcommands and Syntax" on page 188
- "Determining the fcagent Version Number" on page 190
- "Getting Channel and Enclosure Status" on page 190
- "Taking JBOD Drives Offline" on page 193
- "Bringing Up a Replacement Drive" on page 194
- "Turning Disk Check LEDs On or Off" on page 194
- "Bypassing a JBOD Drive" on page 196
- "Bringing JBOD Drives Back Online (Unbypassing)" on page 197
- "Using IRIX Utilities to Get JBOD Information" on page 199

The Agent fcagent

The *fccli* works directly with the fibre channel agent *fcagent*. The *fcagent* must be running before *fccli* commands are accepted and executed by the fibre channel host system. To start the *fcagent*, follow these steps:

1. Make sure that the host system on which the agent is to run has the correct date and time.

2. Enter

   chkconfig fcagent on

   to turn on the *fcagent* flag.

3. Launch the agent by entering

   /etc/init.d/fcagent start

Stop the agent by entering

/etc/init.d/fcagent stop

fccli Subcommands and Syntax

Table 7-1 summarizes *fccli* subcommands.

Table 7-1   fccli Subcommands

Subcommand	Use
GETVERSION	Gets version number of *fcagent* on a particular host system
GETSTATUS	Gets channel and enclosure status
REMOVE	Takes a non-RAID drive offline
INSERT	Brings a drive online
LEDON	Turns disk module check LED on
LEDOFF	Turns disk module check LED off
BYPASS	Causes the host to bypass JBOD disk(s) in a specific enclosure on a specific channel
UNBYPASS	Brings bypassed disks back online

Command syntax is as follows:

fccli SUBCOMMAND [**-h** *hostname*] [*parameters*]

where

SUBCOMMAND

          Subcommand as listed in Table 7-1 and explained in this chapter.

-h *hostname*   Specifies hostname of target host, for when user is logged in remotely.

parameters   Possible parameters are as follows:

- **-c** *channelid*: specifies ID of a specific channel

  All subcommands except GETVERSION can take this parameter.

- **-e** *enclosureid*: specifies ID of a specific enclosure

  Only the subcommand GETSTATUS can take this parameter.

- **-t** *tid1* [*tid2* ... [*tidn*]: specifies ID(s) of target drives

  All subcommands except GETVERSION and GETSTATUS can take this parameter.

For more information on *fccli*, see its man page fccli(1M).

Determining the fcagent Version Number

You can obtain the version number of the *fcagent* that is running on a particular host system in two ways:

- Get the version information remotely.
- Get the version information directly from the host system's terminal interface.

If you are remotely logged into the host, enter

fccli GETVERSION -h *hostname*

To get the revision information directly from the host system running *fcagent*, enter

fccli GETVERSION

A sample output follows:

```
FC JBOD Agent version 1.1, Wed Mar 12 08:04:26 PST 1997.
```

In this example, the agent is identified as a JBOD, or non-RAID, agent version.

Getting Channel and Enclosure Status

To get a general channel and enclosure status for all the fibre channel devices on a host system, enter

fccli GETSTATUS

If you are remotely logged into the host, enter

fccli GETSTATUS -h *hostname*

Output looks like the following:

```
CHANNEL| DRIVES   POWER     FANS      PEER LCCS
-------+----------------------------------------
01     | OK       FAIL      OK        OK
02     | OK       OK        OK        OK
03     | OK       OK        OK        OK
04     | OK       OK        OK        OK
05     | OK       OK        OK        OK
06     | OK       OK        OK        OK
```

In channel 01 in the sample output, a power error is shown. Because the drives, fans, and link control cards (LCCs) show OK in the *GETSTATUS* output example, you can conclude that there is probably trouble with one of the two power supplies in the enclosure.

Note: In this example, channel 01 controls three enclosures. Each fibre channel site can have different configurations installed, and a single channel can control up to 11 (non-RAID) enclosures within the same rack. (The Fibre Channel Hub is transparent to *fccli*.)

To determine more about the power failure in the enclosure controlled by fibre channel 01, specify the channel:

`fccli GETSTATUS -c 01`

This command causes the *fcagent* to output a more detailed screen display showing the status of the enclosure(s) controlled by channel 01. The following example output shows that a power supply has failed:

```
ENCLOSURE | DRIVES     POWER     FANS      PEER LCC
----------+-----------------------------------------
        4 |   OK        FAIL      OK         OK
        5 |   OK        FAIL      OK         OK
       11 |   OK        FAIL      OK         OK
```

Always physically inspect the power supply before ordering a new part. The *GETSTATUS* command reports an unplugged power supply or a power supply with its breaker set at 0 (off), as a "failed" power supply.

If you try to obtain status on a channel that does not exist on the host, the *fcagent* reports an error; for example:

```
# fccli GETSTATUS -c 4
Channel 4 does not exist on 'localhost'.
```

For an enhanced output of information or to specify one particular enclosure supported by a particular channel, specify the enclosure as well as the channel with the **-c** and **-e** arguments to *GETSTATUS*. The following example requests information for channel 01 and enclosure 01:

```
fccli GETSTATUS -c 01 -e 01
```

Figure 7-1 shows example output.

```
Enclosure 1, status FAIL
  Vendor ID:     CLARiiON
  Product ID:    DISK ENC
  LCC SN#:
  LCC Mfg. Date:
  LCC code rev:  003
  FRUs:          10 disk slot(s)
                 1 fan slot(s)
                 2 power supply slot(s)
                 1 peer LCC slot(s)
        +---------------------------------------+------+
        | [1] FAIL                              |      |
        +---------------------------------------+      |
        |   |   |   |   |   |   |   |   |   |   |   | P    |
        | O | O | O | O | O | O | O | O | O | O | E   L |
        | K | K | K | K | K | K | K | K | K | K | E   L |
        | 10| 11| 12| 13| 14| 15| 16| 17| 18| 19| R   C |
        +---------------------------------------+      |
        |              FANS   OK                |  OK  |
        |        PS0 FAIL             PS1   OK  |      |
        +---------------------------------------+------+
```

Figure 7-1   Output for GETSTATUS With -c and -e Parameters

If you select a nonexistent or offline enclosure with a *GETSTATUS* inquiry, the *fcagent* reports an error similar to the following:

```
fccli GETSTATUS -c 1 -e 0
Enclosure 0 does not exist on channel 1 on host 'localhost'.
```

Taking JBOD Drives Offline

To remove a non-RAID fibre channel disk module from an operating enclosure use the *fccli REMOVE* command:

fccli REMOVE **-c** *channelid* **-t** *tid1 [tid2 ... [tidn]*

Note: You must use the **-c** (channel designator argument) and **-t** (target drive argument) with the command in order to properly spin down the target disk module.

In the following example, the tenth disk module on channel 01 is spun down and taken out of the fibre channel operational loop.

```
fccli REMOVE -c 1 -t 10
```

The *fccli REMOVE* command does not confirm that the disk module has been taken offline. Use a *GETSTATUS* inquiry such as `fccli GETSTATUS -c 1` or physically inspect the number 10 drive (and see the amber LED) to confirm that it is offline or bypassed (BP). The *GETSTATUS* shows something similar to:

```
+-------------------------------------+------+
| [1] FAIL                            |      |
+-------------------------------------+      |
|   |   |   |   |   |   |   |   |   |   | P  |
| O | O | O | O | O | O | O | O | O | B | E  L |
| K | K | K | K | K | K | K | K | K | P | E  L |
| 10| 11| 12| 13| 14| 15| 16| 17| 18| 19| R  C |
+-------------------------------------+      |
|              FANS   OK              | OK   |
|      PS0  FAIL            PS1   OK  |      |
+-------------------------------------+------+
```

The targeted disk module shows BP instead of OK and can now be physically removed from the enclosure.

If you use *fccli REMOVE* to remove a disk module on a channel that does not exist, you receive an error message similar to the following:

```
fccli REMOVE -c 1 -t 11
```

```
Drive 11 does not exist on channel 1 on host 'localhost'.
```

In this case, channel 01 controls only one enclosure with a total of ten disk modules. The targeted disk module (drive 11) does not exist and cannot be taken offline.

193

Bringing Up a Replacement Drive

After you install a new disk module, you must alert the host system to spin up the disk module and bring it online. This spin-up process uses the *INSERT* subcommand and requires the proper channel (**-c**) and targeted disk (**-t**):

fccli INSERT **-c** *channelid* **-t** *tid*

Note: During the replacement process, quiesce the loop's activity. Advise all users to unmount filesystems and stop running applications until the process is complete.

The following example spins up a replacement disk module at slot 10 on channel 1 and brings it back online.

```
fccli INSERT -c 1 -t 10
```

If you enter an invalid channel, enclosure, or drive number when bringing a new or replacement drive online, *fcagent* returns an error message.

Turning Disk Check LEDs On or Off

Each disk module slot in a FibreVault has two LEDs directly above it. The LED closest to the disk drive is a green activity LED, which lights during normal drive operations. The amber LED, farther from the drive, lights when there is a problem with the drive or when the drive is offline or missing. This amber LED can also be turned on using the *LEDON* subcommand with *fccli* with the proper arguments.

You may wish to turn on the LED of a particular drive to confirm the identity of a channel, enclosure, or a particular drive. Doing so can be especially helpful in systems with many cables, enclosures, and drives. Proper cable and enclosure labeling should greatly reduce this type of identification problem.

Use *LEDON* as follows:

fccli LEDON **-c** *channelid* **-t** *tid1 [tid2 ... [tidn]*

The following example turns on the check LEDs for disk 3 and 5 in enclosure 1 on channel 12.

```
fccli LEDON -c 12 -t 13 -t 15
```

194

Figure 7-2 shows output for this example.

```
                   +-------------------------------------+------+
                   | [1] FAIL                            |      |
                   +-------------------------------------+      |
Amber LED on ----> |   |   |   | * |   | * |   |   |   |   | P    |
                   | O | O | O | O | O | O | O | O | O | O | E  L |
                   | K | K | K | K | K | K | K | K | K | K | E  L |
                   | 10| 11| 12| 13| 14| 15| 16| 17| 18| 19| R  C |
                   +-------------------------------------+      |
                   |              FANS   OK              |   OK |
                   |       PS0 FAIL            PS1  OK   |      |
                   +-------------------------------------+------+
```

Figure 7-2   LEDON Example Output

The amber LEDs above disk drives 3 and 5 in enclosure 1 light up and stay lit until you turn them off with the *LEDOFF* subcommand.

Note in the sample output for *LEDON* that there is an asterisk above positions 13 and 15. The asterisk indicates that the amber LED has been turned on for enclosure 1, drive 3 and enclosure 1, drive 5.

To turn the LED off again, use *LEDOFF*:

fccli LEDOFF **-c** *channelid* **-t** *tid1 [tid2 ... [tidn]*

The following example turns off the amber check LED on the disk modules from the previous example.

```
fccli LEDOFF -c 12 -t 13 -t 15
```

Note: The *LEDOFF* subcommand is not intended for turning off a check LED on a drive that is experiencing an actual failure.

195

Bypassing a JBOD Drive

You can cause the host to bypass one or more disks in a specific enclosure on a specific channel by using the *BYPASS* subcommand with the **-c** and **-e** arguments:

fccli BYPASS **-c** *channelid* **-t** *tid1 [tid2 ... [tidn]*

Note: When you use *BYPASS* to take a drive offline, you cut off all I/O access. Before implementing the command, make sure that this action does not affect local or remote users.

In the example in Figure 7-3, five of the ten drives in enclosure 4 on channel 1 are taken offline.

```
# fccli BYPASS -c 01 -t 45 46 47 48 49
# fccli GETSTATUS -c 01 -e 4
 Enclosure 4, status FAIL
  Vendor ID:      CLARiiON
  Product ID:     DISK ENC
  LCC SN#:
  LCC Mfg. Date:
  LCC code rev:   003
  FRUs:           10 disk slot(s)
                  1 fan slot(s)
                  2 power supply slot(s)
                  1 peer LCC slot(s)


    +------------------------------------+------+
    | [ 4] FAIL                          |      |
    +------------------------------------+      |
    | | | | | | | | | | | | | | | | | | | P    |
    | O | O | O | O | O | B | B | B | B | B | E   L |
    | K | K | K | K | K | P | P | P | P | P | E   L |
    | 40| 41| 42| 43| 44| 45| 46| 47| 48| 49| R   C |
    +------------------------------------+      |
    |               FANS   OK            | OK   |
    |       PS0 FAIL            PS1   OK | |    |
    +------------------------------------+------+
```

Figure 7-3   BYPASS Example

The *fcagent* always reports the bypassing of a drive(s) to */var/adm/SYSLOG*. If you checked the *SYSLOG* immediately after the last command was issued, you would see something similar to the following:

```
system1 15# view /var/adm/SYSLOG
May  7 17:49:10 5D:system1 fcagent[427]: FRU reconfiguration: DISK #5
in enclosure 4 on channel 01: OK --> BYPASSED
May  7 17:49:10 5D:system1 fcagent[427]: FRU reconfiguration: DISK #6
in enclosure 4 on channel 01: OK --> BYPASSED
May  7 17:49:11 5D:system1 fcagent[427]: FRU reconfiguration: DISK #7
in enclosure 4 on channel 01: OK --> BYPASSED
May  7 17:49:11 5D:system1 fcagent[427]: FRU reconfiguration: DISK #8
in enclosure 4 on channel 01: OK --> BYPASSED
May  7 17:49:11 5D:system1 fcagent[427]: FRU reconfiguration: DISK #9
in enclosure 4 on channel 01: OK --> BYPASSED
```

Bringing JBOD Drives Back Online (Unbypassing)

To bring bypassed drives back online, have the host "unbypass" the disks in an enclosure that have been bypassed:

fccli UNBYPASS **-c** *channelid* **-t** *tid1 [tid2 ... [tidn]*

Note: When you use *UNBYPASS* to bring drives back online, the host has almost immediate access. Be sure this does not affect local or remote users before implementing the order.

For example, if five of the ten drives in enclosure 4 on channel 1 were offline, you would use the following commands to unbypass the drives and bring them back online and show the status after that operation. Figure 7-4 shows the commands and output.

```
# fccli UNBYPASS -c 01 -t 45 46 47 48 49
# fccli GETSTATUS -c 01 -e 4
 Enclosure 4, status FAIL
  Vendor ID:      CLARiiON
  Product ID:     DISK ENC
  LCC SN#:
  LCC Mfg. Date:
  LCC code rev:   003
  FRUs:           10 disk slot(s)
                  1 fan slot(s)
                  2 power supply slot(s)
                  1 peer LCC slot(s)


     +----------------------------------------+------+
     | [ 4] FAIL                              |      |
     +----------------------------------------+      |
     |   |   |   |   |   |   |   |   |   |   | | P    |
     | O | O | O | O | O | O | O | O | O | O | | E  L |
     | K | K | K | K | K | K | K | K | K | K | | E  L |
     | 40| 41| 42| 43| 44| 45| 46| 47| 48| 49| R  C |
     +----------------------------------------+      |
     |                 FANS   OK               | OK   |
     |        PS0 FAIL              PS1  OK    |      |
     +----------------------------------------+------+
```

Figure 7-4   UNBYPASS Example

As with the *BYPASS* command, the *fcagent* reports the unbypassing of a drive(s) to
/var/adm/SYSLOG. If you checked the *SYSLOG* immediately after the last command was
issued, you would see something similar to the following:

```
system1 15# view /var/adm/SYSLOG
May  7 17:59:10 5D:system1 fcagent[427]: FRU reconfiguration: DISK #5
in enclosure 4 on channel 01: OK --> UNBYPASSED
May  7 17:59:10 5D:system1 fcagent[427]: FRU reconfiguration: DISK #6
in enclosure 4 on channel 01: OK --> UNBYPASSED
May  7 17:59:11 5D:system1 fcagent[427]: FRU reconfiguration: DISK #7
in enclosure 4 on channel 01: OK --> UNBYPASSED
May  7 17:59:11 5D:system1 fcagent[427]: FRU reconfiguration: DISK #8
in enclosure 4 on channel 01: OK --> UNBYPASSED
May  7 17:59:11 5D:system1 fcagent[427]: FRU reconfiguration: DISK #9
in enclosure 4 on channel 01: OK --> UNBYPASSED
```

198

Using IRIX Utilities to Get JBOD Information

This section explains how to use IRIX utilities to get more information about JBOD storage:

- "Getting a Fibre Channel Hardware Graph Output" on page 199
- "Listing Fibre Channel Controller and Drive Numbers" on page 199

Getting a Fibre Channel Hardware Graph Output

To obtain hardware graph information for a local host system that is controlling a fibre channel disk array, enter a *grep* inquiry such as the following:

grep ^*controller_number* **/etc/ioconfig.conf**

The following example displays information on channel 12.

```
grep ^12 /etc/ioconfig.conf
```

The output might look like the following:

```
12 /hw/module/1/slot/io4/fibre_channel/pci/0/scsi_ctlr/0
```

This output tells you that module 1, slot io4, has a fibre channel enclosure connected to pci 0 (the first channel on the Fibre Channel XIO board).

Listing Fibre Channel Controller and Drive Numbers

You can use the *hinv* command at the operating system level to determine numbers and types of fibre devices on a particular host system, as follows:

hinv | grep Fibre

Such *hinv* output might look like the following:

```
Integral SCSI controller 12: Version Emerald Fibrechannel
Integral SCSI controller 13: Version Emerald Fibrechannel
Integral SCSI controller 10: Version Emerald Fibrechannel
Integral SCSI controller 11: Version Emerald Fibrechannel
Integral SCSI controller 22: Version Emerald Fibrechannel
Integral SCSI controller 23: Version Emerald Fibrechannel
```

To use *hinv* to get the listing of devices connected to the fibre channel controller(s), enter

hinv -c disk

The output might look like the following:

```
Integral SCSI controller 12: Version Emerald Fibrechannel

Disk drive: unit 19 on SCSI controller 12
Disk drive: unit 17 on SCSI controller 12
Disk drive: unit 15 on SCSI controller 12
Disk drive: unit 13 on SCSI controller 12
Disk drive: unit 11 on SCSI controller 12
Disk drive: unit 18 on SCSI controller 12
Disk drive: unit 16 on SCSI controller 12
Disk drive: unit 14 on SCSI controller 12
Disk drive: unit 12 on SCSI controller 12
Disk drive: unit 10 on SCSI controller 12
```

Error Recovery

This appendix contains these sections;

Restarting ssmagent

You might have to restart the agent *ssmagent* after you take any of these actions:

- changing an array so that communication between the front end and back end is affected; for example, attaching an array after the agent has been started

- restarting licensed internal code (LIC)

- partitioning RAID 3 memory

- modifying the */etc/config/ssmagent.config* file

- rebooting, unless you have configured the system to start the agent automatically

To determine whether restarting the agent is necessary, enter

scsicontrol -i *sc<controllerid>d<driveid>l<lunid>*

For example:

```
scsicontrol -i sc6d6l0
```

You do not need to restart the agent if you see an output such as

```
sc2d0l0:  Disk          SGI     RAID 5          0130
ANSI vers 2, ISO ver: 0, ECMA ver: 0; supports:  synch linkedcmds
cmdqueing
```

You need to restart the agent if you see a message such as `No such device` or `Cannot open`.

To start *ssmagent*, enter

/etc/init.d/ssm stop
/etc/init.d/ssm start

Depending on the number of RAID arrays you administer, it takes the agent from 30 seconds to several minutes to finish launching and be ready to administer them.

Re-Establishing Communication With an Array

Communication with the array is lost if you have rebooted a server for which there are no LUNs bound, or if all three database disk modules are inadvertently removed at the same time. To re-establish communication with the array, follow these steps:

1. Obtain a serial cable; see Appendix B, "Server Serial Connector Pinouts," for the pinout for the server you are using.

2. Connect the serial cable to a tty port on the server, such as tty2 (see the owner's guide for the server if necessary) and the serial port on the array.

3. In */etc/config/ssmagent.config*, comment out the line that specifies the device name and specify the tty port instead. For example:

```
# # device specifications
# *close* at the end is needed for low level (devscsi) support.

#device sc2d2l0  Unit-spA "Unit 8"
ttydevice tty2
```

4. Stop the agent:

   /etc/init.d/ssm stop

5. Restart the agent:

   /etc/init.d/ssm start

6. After you establish communication with the array, set up at least one LUN for the default SP.

7. In */etc/config/ssmagent.config*, remove the comment symbol from the line that specifies the device name and remove the line that specifies the tty port.

If the driver loses communication with the agent (for example, because of a broken cable connection between the host and the storage system, or if a host bus adapter is turned off), the driver on the host times out and logs off. To reinitialize the loop, enter

scsiha -lp *controllernumber*

For example:

scsiha -lp 2

Recovering From an Incorrect Bind

To stop a bind that is in progress, use the RAID CLI *rebootsp* subcommand; see "rebootSP" on page 177 in Chapter 6. Alternatively, you can use the GUI to remove all drives in the bind.

If a DPE has two LCCs (A and B), each DAE cabled to it must also have two LCCs. If a DAE has no LCC B, binding LUNs on SP B is restricted in some circumstances: If SP A uses disk modules 00 through 09 (that is, all the disk modules in the DPE) in LUNs, you cannot bind any LUNs on SP B, because the connection to the disk modules in the DAE is missing: there is no path through LCC B (absent) and no path through the disk modules on the DPE (all are bound on SP A).

Server Serial Connector Pinouts

This appendix contains these sections:

- "Origin2000 and Onyx2 Serial Pinout" on page 205
- "Origin200 Serial Pinouts" on page 207
- "OCTANE Serial Pinouts" on page 208

Origin2000 and Onyx2 Serial Pinout

Each Origin2000 or Onyx 2 system comes with two standard 9-pin serial ports. These ports can support either RS-232 or RS-422 interface devices. Figure B-1 shows the location and pinouts for a serial port. Optional additional serial ports are also available.
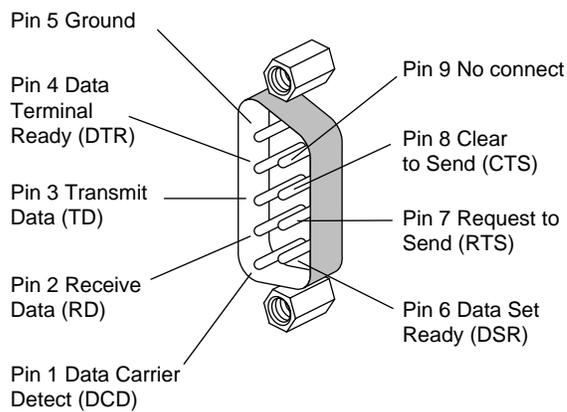
Figure B-1   Serial Port Location and Pinouts

This connector departs from traditional Silicon Graphics usage:

- Pin 2 (not pin 3) is Receive Data (RD)

- Pin 3 (not pin 2) is Transmit Data (TD)

- Pin 5 (not pin 7) is signal ground

- gender is female (not male)

Table B-1 shows the correspondence between this connector and the SP connector.

Table B-1   Origin2000/Onyx2 and SP Serial Connectors

Origin2000/Onyx2 Serial Connector Pin	SP Connector Pin
2	3
3	2
5	7

The RS-232 standard recommends the use of cables no longer than 50 feet (15.2 meters). (This standard should also be applied for an RS-422 connector.) Longer runs increases the possibility of line noise which can affect data transmission and cause errors. For cable runs longer than 50 feet (15.2 meters), use an appropriate extender device.

Origin200 Serial Pinouts

Figure B-2 shows pin numbering for the Origin200 serial port.
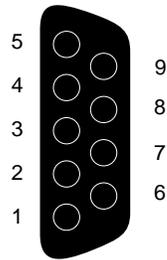
Figure B-2   Origin200 Serial Port Pinout Assignments

Table B-2 shows the connector pin assignments for the serial ports in RS-232 mode.

Table B-2   Origin200 Serial Port Pin Assignments, RS-232 Mode

Pin	Assignment
1	Data Carrier Detect (DCD)
2	Receive Data (RXD)
3	Transmit Data (TXD)
4	Data Terminal Ready (DTR)
5	Ground (GND)
6	Data Set Ready (DSR)
7	Request To Send (RTS)
8	Clear To Send (CTS)
9	(Not Connected)

OCTANE Serial Pinouts

The OCTANE workstation serial ports can operate in RS-232 electrical levels with a standard PC pinout; the range of supported baud rates is 300 baud to 460K baud. Figure B-3 shows pin numbering.
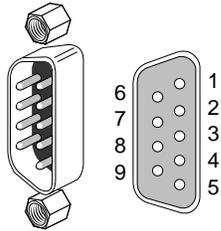
Figure B-3   OCTANE Serial Port

Table B-3 shows cable pinout assignments for the serial ports.

Table B-3   OCTANE PC-Compatible Serial Port Pinout

Pin	Assignment	Description
1	DCD	Data Carrier Detect
2	RD	Receive Data
3	TD	Transmit Data
4	DTR	Data Terminal Ready
5	SG	Signal Ground
6	DSR	Data Set Ready
7	RTS	Request to Send
8	CTS	Clear to Send
9	RI	Ring Indicator

For information on operating the serial ports with Macintosh style electrical levels and a Macintosh-compatible pinout or with ANSI/SMPTE 207M electrical levels and pinout, or for information on connecting a dumb terminal, see the *OCTANE Workstation Owner's Guide*.

Index

Tell Us About This Manual

As a user of Silicon Graphics products, you can help us to better understand your needs and to improve the quality of our documentation.

Any information that you provide will be useful. Here is a list of suggested topics:

- General impression of the document
- Omission of material that you expected to find
- Technical errors
- Relevance of the material to the job you had to do
- Quality of the printing and binding

Please send the title and part number of the document with your comments. The part number for this document is 007-3715-001.

Thank you!

Three Ways to Reach Us

- To send your comments by **electronic mail**, use either of these addresses:
  - On the Internet: techpubs@sgi.com
  - For UUCP mail (through any backbone site): *[your_site]*!sgi!techpubs
- To **fax** your comments (or annotated copies of manual pages), use this fax number: 650-932-0801
- To send your comments by **traditional mail**, use this address:

Technical Publications
Silicon Graphics, Inc.
2011 North Shoreline Boulevard, M/S 535
Mountain View, California 94043-1389